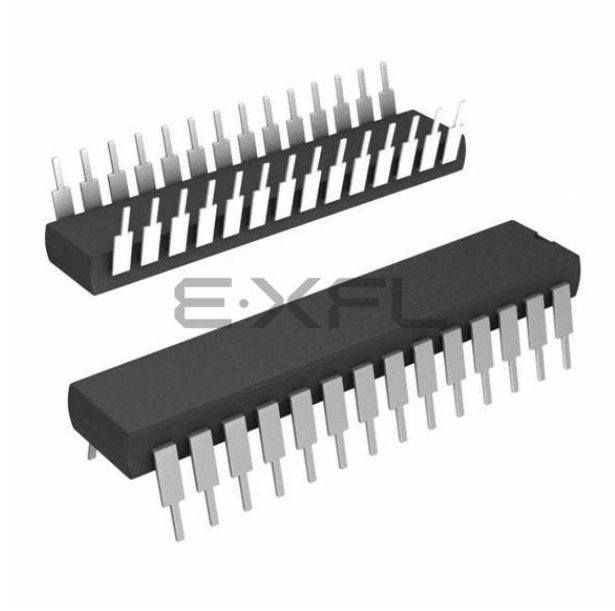


Welcome to E-XFL.COM

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	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf27k42-i-sp

TABLE 3: 48-PIN ALLOCATION TABLE FOR PIC18(L)F5XK42 (CONTINUED)

I/O	48-Pin TQFP	48-Pin UQFN	ADC	Voltage Reference	DAC	Comparators	Zero Cross Detect	I^2C	SPI	UART	DSM	Timers/SMT	CCP and PWM	CWG	CLC	NCO	Clock Reference (CLKR)	Interrupt-on-Change	Basic
RC2	40	40	ANC2	—	-	—	—	—	—	—	—	T5CKI(1)	CCP1(1)	—	—	—	—	IOCC2	—
RC3	41	41	ANC3	—	-	—	—	SCL1(3,4)	SCK1(1)	—	—	T2IN(1)	-	—	—	—	—	IOCC3	—
RC4	46	46	ANC4	—	—	—	—	SDA1(3,4)	SDI1(1)	—	—	—	—	—	—	—	—	IOCC4	—
RC5	47	47	ANC5	—	—	—	—	—	—	—	—	T4IN(1)	—	—	—	—	—	IOCC5	—
RC6	48	48	ANC6	—	—	—	—	—	—	CTS1(1)	—	—	—	—	—	—	—	IOCC6	—
RC7	1	1	ANC7	—	—	—	—	—	—	RX1(1)	—	—	—	—	—	—	—	IOCC7	—
RD0	42	42	AND0	—	—	—	—	—(4)	—	—	—	—	—	—	—	—	—	—	—
RD1	43	43	AND1	—	—	—	—	—(4)	—	—	—	—	—	—	—	—	—	—	—
RD2	44	44	AND2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD3	45	45	AND3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD4	2	2	AND4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD5	3	3	AND5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD6	4	4	AND6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD7	5	5	AND7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE0	27	27	ANE0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE1	28	28	ANE1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE2	29	29	ANE2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE3	20	20	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCE3	$\overline{MCLR}$ V_{PP}
RF0	36	36	ANF0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF1	37	37	ANF1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF2	38	38	ANF2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF3	39	39	ANF3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF4	12	12	ANF4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF5	13	13	ANF5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF6	14	14	ANF6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RF7	15	15	ANF7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.

2: All output signals shown in this row are PPS remappable.

3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

4: These pins can be configured for I^2C and SMB™ 3.0/2.0 logic levels; The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4/RD0/RD1 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I^2C specific or SMBus input buffer thresholds.

TABLE 1-1: DEVICE FEATURES

Features	PIC18(L)F26K42	PIC18(L)F27K42	PIC18(L)F45K42	PIC18(L)F46K42	PIC18(L)F47K42	PIC18(L)F55K42	PIC18(L)F56K42	PIC18(L)F57K42
Program Memory (Bytes)	65536	131072	32768	65536	131072	32768	65536	131072
Program Memory (Instructions)	32768	65536	16384	32768	65536	16384	32768	65536
Data Memory (Bytes)	4096	8192	2048	4096	8192	2048	4096	8192
Data EEPROM Memory (Bytes)	1024	1024	256	1024	1024	256	1024	1024
Packages	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN	40-pin PDIP 40-pin UQFN 44-pin TQFP 44-pin QFN	40-pin PDIP 40-pin UQFN 44-pin TQFP 44-pin QFN	40-pin PDIP 40-pin UQFN 44-pin TQFP 44-pin QFN	48-pin TQFP 48-pin UQFN	48-pin TQFP 48-pin UQFN	48-pin TQFP 48-pin UQFN
I/O Ports	A,B,C,E(1)	A,B,C,E(1)	A,B,C,D, E(1)	A,B,C,D, E(1)	A,B,C,D, E(1)	A,B,C,D, E(1), F	A,B,C,D, E(1), F	A,B,C,D, E(1), F
12-Bit Analog-to-Digital Conversion Module (ADC^2) with Computation Accelerator	5 internal 24 external	5 internal 24 external	5 internal 35 external	5 internal 35 external	5 internal 35 external	5 internal 43 external	5 internal 43 external	5 internal 43 external
Capture/Compare/ PWM Modules (CCP)	4							
10-Bit Pulse-Width Modulator (PWM)	4							
Timers (16-/8-bit)	4/3							
Serial Communications	1 UART, 1 UART with DMX/DALI/LIN, 2 I^2C, 1 SPI							
Complementary Waveform Generator (CWG)	3							
Zero-Cross Detect (ZCD)	1							
Data Signal Modulator (DSM)	1							
Signal Measurement Timer (SMT)	1							
5-bit Digital to Analog Converter (DAC)	1							
Numerically Controlled Oscillator (NCO)	1							

3.2.3 ISR PRIORITY > PERIPHERAL PRIORITY > MAIN PRIORITY

In this case, interrupt routines and peripheral operation (DMAx, Scanner) will stall the CPU. Interrupt will preempt peripheral operation. This results in lowest interrupt latency and highest throughput for the peripheral to access the memory.

3.2.4 PERIPHERAL 1 PRIORITY > ISR PRIORITY > MAIN PRIORITY > PERIPHERAL 2 PRIORITY

In this case, the Peripheral 1 will stall the execution of the CPU. However, Peripheral 2 can access the memory in cycles unused by Peripheral 1.

The operation of the System Arbiter is controlled through the following registers:

REGISTER 3-1: ISRPR: INTERRUPT SERVICE ROUTINE PRIORITY REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	ISRPR<2:0>		
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	HS = Hardware set

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **ISRPR<2:0>**: Interrupt Service Routine Priority Selection bits

REGISTER 3-2: MAINPR: MAIN ROUTINE PRIORITY REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-1/1
—	—	—	—	—	MAINPR<2:0>		
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	HS = Hardware set

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **MAINPR<2:0>**: Main Routine Priority Selection bits

REGISTER 3-3: DMA1PR: DMA1 PRIORITY REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0
—	—	—	—	—	DMA1PR<2:0>		
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	HS = Hardware set

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **DMA1PR<2:0>**: DMA1 Priority Selection bits

REGISTER 5-6: CONFIGURATION WORD 3H (30 0005h)

U-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	WDTCCS<2:0>			WDTCWS<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '1'	
-n = Value for blank device	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '1'

bit 5-3 **WDTCCS<2:0>:** WDT Input Clock Selector bits

If WDTE<1:0> Fuses = 2'b00:

These bits are ignored.

Otherwise:

000 = WDT reference clock is the 31.0 kHz LFINTOSC
001 = WDT reference clock is the 31.25 kHz MFINTOSC
010 = WDT reference clock is SOSC
011 = Reserved (default to LFINTOSC)
•
•
110 = Reserved (default to LFINTOSC)
111 = Software control

bit 2-0 **WDTCWS<2:0>:** WDT Window Select bits

WDTCWS<2:0>	Window at POR			Software Control of Window	Keyed Access Required?
	Value	Window Delay Percent of Time	Window Opening Percent of Time		
000	000	87.5	12.5	No	Yes
001	001	75	25		
010	010	62.5	37.5		
011	011	50	50		
100	100	37.5	62.5		
101	101	25	75		
110	111	n/a	100		
111	111	n/a	100	Yes	No

6.3 Register Definitions: BOR Control

REGISTER 6-1: BORCON: BROWN-OUT RESET CONTROL REGISTER

R/W-1/u	U-0	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	—	—	—	—	—	—	BORRDY
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	q = Value depends on condition

bit 7 **SBOREN:** Software Brown-out Reset Enable bit

If BOREN ≠ 01:
SBOREN is read/write, but has no effect on the BOR.
If BOREN = 01:
1 = BOR Enabled
0 = BOR Disabled

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **BORRDY:** Brown-out Reset Circuit Ready Status bit

1 = The Brown-out Reset Circuit is active and armed
0 = The Brown-out Reset Circuit is disabled or is warming up

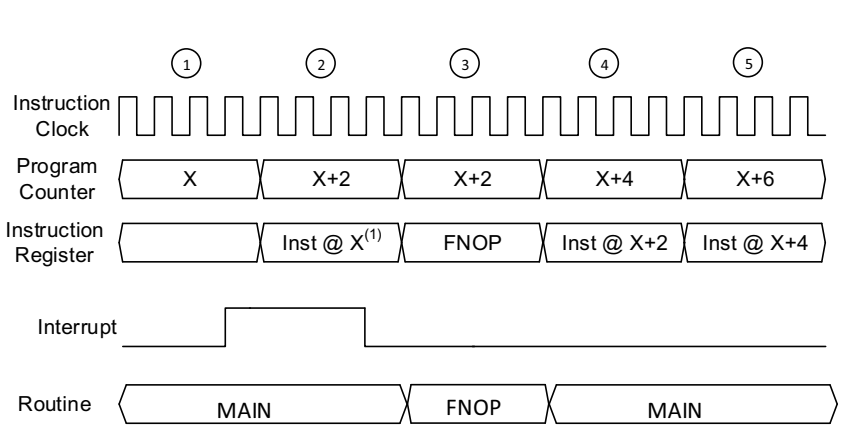
9.7.1 ABORTING INTERRUPTS

If the last instruction before the interrupt controller vectors to the ISR from main routine clears the GIE, PIE or PIR bit associated with the interrupt, the controller executes one force NOP cycle before it returns to the main routine.

Figure 9-10 illustrates the sequence of events when a peripheral interrupt is asserted and then cleared on the last executed instruction cycle.

If the GIE, PIE or PIR bit associated with the interrupt is cleared prior to vectoring to the ISR, then the controller continues executing the main routine.

FIGURE 9-10: INTERRUPT TIMING DIAGRAM - ABORTING INTERRUPTS



Note 1: Inst @ X clears the interrupt flag, Example BCF INTCON0, GIE.

10.4 Register Definitions: Voltage Regulator Control

REGISTER 10-1: VREGCON: VOLTAGE REGULATOR CONTROL REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1
—	—	—	—	—	—	VREGPM	Reserved
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 7-2 **Unimplemented:** Read as '0'

bit 1 **VREGPM:** Voltage Regulator Power Mode Selection bit
- 1 = Low-Power Sleep mode enabled in Sleep(2)
  Draws lowest current in Sleep, slower wake-up
- 0 = Normal Power mode enabled in Sleep(2)
  Draws higher current in Sleep, faster wake-up

bit 0 **Reserved:** Read as '1'. Maintain this bit set.

Note 1: Not present in LF parts.

2: See **Section 44.0 "Electrical Specifications"**.

REGISTER 15-21: DMAxDCNTH: DMAx DESTINATION COUNT HIGH REGISTER

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	DCNT<11:8>			
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **DCNT<11:8>:** Current Destination Byte Count

REGISTER 15-22: DMAxSIRQ: DMAx START INTERRUPT REQUEST SOURCE SELECTION 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
—	SIRQ<6:0>						
bit 7							bit 0

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

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

bit 6-0 **SIRQ<6:0>:** DMAx Start Interrupt Request Source Selection bits
Please refer to Table 15-2 for more information.

REGISTER 15-23: DMAxAIRQ: DMAx ABORT INTERRUPT REQUEST SOURCE SELECTION 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
—	AIRQ<6:0>						
bit 7							bit 0

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

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

bit 6-0 **AIRQ<6:0>:** DMAx Interrupt Request Source Selection bits
Please refer to Table 15-2 for more information.

18.0 INTERRUPT-ON-CHANGE

PORTA, PORTB, PORTC and pin RE3 of PORTE can be configured to operate as Interrupt-on-Change (IOC) pins on PIC18(L)F26/27/45/46/47/55/56/57K42 family devices. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual port pin, or combination of port pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 18-1 is a block diagram of the IOC module.

18.1 Enabling the Module

To allow individual port pins to generate an interrupt, the IOCIE bit of the PIEx register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

18.2 Individual Pin Configuration

For each port pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both associated bits of the IOCxP and IOCxN registers, respectively.

18.3 Interrupt Flags

The IOCAFx, IOCBFx, IOCCFx and IOCEF3 bits located in the IOCAF, IOCBF, IOCCF and IOCEF registers respectively, are status flags that correspond to the interrupt-on-change pins of the associated port. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the PIR0 register reflects the status of all IOCAFx, IOCBFx, IOCCFx and IOCEF3 bits.

18.4 Clearing Interrupt Flags

The individual status flags, (IOCAFx, IOCBFx, IOCCFx and IOCEF3 bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

EXAMPLE 18-1: CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)

```
MOVLW   0xff
XORWF   IOCAF, W
ANDWF   IOCAF, F
```

18.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCxF register will be updated prior to the first instruction executed out of Sleep.

REGISTER 23-5: CCPRxH: CCPx REGISTER HIGH BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
RH<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0

MODE = Capture Mode:
RH<7:0>: MSB of captured TMR1 value
MODE = Compare Mode:
RH<7:0>: MSB compared to TMR1 value
MODE = PWM Mode && FMT = 0:
RH<7:2>: Not used
RH<1:0>: CCPW<9:8> – Pulse-Width MS 2 bits
MODE = PWM Mode && FMT = 1:
RH<7:0>: CCPW<9:2> – Pulse-Width MS 8 bits

TABLE 23-4: SUMMARY OF REGISTERS ASSOCIATED WITH CCPx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPxCON	EN	—	OUT	FMT	MODE<3:0>				350
CCPxCAP	—	—	—	—	—	—	CTS<1:0>		352
CCPRxL	CCPRx<7:0>								352
CCPRxH	CCPRx<15:8>								353
CCPTMRS0	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		351

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the CCP module.

REGISTER 24-2: CCPTMRS1: CCP TIMERS CONTROL REGISTER 1

R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1
P8TSEL<1:0>		P7TSEL<1:0>		P6TSEL<1:0>		P5TSEL<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **P8TSEL<1:0>:** PWM8 Timer Selection bits
11 = PWM8 based on TMR6
10 = PWM8 based on TMR4
01 = PWM8 based on TMR2
00 = Reserved

bit 5-4 **P7TSEL<1:0>:** PWM7 Timer Selection bits
11 = PWM7 based on TMR6
10 = PWM7 based on TMR4
01 = PWM7 based on TMR2
00 = Reserved

bit 3-2 **P6TSEL<1:0>:** PWM6 Timer Selection bits
11 = PWM6 based on TMR6
10 = PWM6 based on TMR4
01 = PWM6 based on TMR2
00 = Reserved

bit 1-0 **P5TSEL<1:0>:** PWM5 Timer Selection bits
11 = PWM5 based on TMR6
10 = PWM5 based on TMR4
01 = PWM5 based on TMR2
00 = Reserved

29.10 Register Definitions: ZCD Control

REGISTER 29-1: ZCDCON: ZERO-CROSS DETECT CONTROL REGISTER

R/W-0/0	U-0	R-x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
SEN	—	OUT	POL	—	—	INTP	INTN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **SEN:** Zero-Cross Detect Software Enable bit
This bit is ignored when ZCDSEN configuration bit is set.
1= Zero-cross detect is enabled.
0= Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls.

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

bit 5 **OUT:** Zero-Cross Detect Data Output bit
ZCDPOL bit = 0:
1 = ZCD pin is sinking current
0 = ZCD pin is sourcing current
ZCDPOL bit = 1:
1 = ZCD pin is sourcing current
0 = ZCD pin is sinking current

bit 4 **POL:** Zero-Cross Detect Polarity bit
1 = ZCD logic output is inverted
0 = ZCD logic output is not inverted

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **INTP:** Zero-Cross Detect Positive-Going Edge Interrupt Enable bit
1 = ZCDIF bit is set on low-to-high ZCD_output transition
0 = ZCDIF bit is unaffected by low-to-high ZCD_output transition

bit 0 **INTN:** Zero-Cross Detect Negative-Going Edge Interrupt Enable bit
1 = ZCDIF bit is set on high-to-low ZCD_output transition
0 = ZCDIF bit is unaffected by high-to-low ZCD_output transition

TABLE 29-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
ZCDCON	SEN	—	OUT	POL	—	—	INTP	INTN	462

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the ZCD module.

31.16 Clock Accuracy with Asynchronous Operation

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

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value of the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 7.2.2.3 "Internal Oscillator Frequency Adjustment"** for more information.

The other method adjusts the value of the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 31.17.1 "Auto-Baud Detect"**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change of the peripheral clock frequency.

31.17 UART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is a 16-bit timer that is dedicated to the support of the UART operation.

The UxBRGH, UxBRGL register pair determines the period of the free running baud rate timer. The multiplier of the baud rate period is determined by the BRGS bit in the UxCON0 register.

Table 31-1 contains the formulas for determining the baud rate. Example 31-1 provides a sample calculation for determining the baud rate and baud rate error.

The high baud rate range (BRGS = 1) is intended to extend the baud rate range up to a faster rate when the desired baud rate is not possible otherwise. Using the normal baud rate range (BRGS = 0) is recommended when the desired baud rate is achievable with either range.

Writing a new value to the UxBRGH, UxBRGL 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 RXIDL bit to make sure that the receive operation is idle before changing the system clock.

EXAMPLE 31-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, BRGS = 0:

$$Desired\ Baud\ Rate = \frac{FOSC}{16([UxBRG] + 1)}$$

$$X = \frac{\frac{FOSC}{Desired\ Baud\ Rate}}{16} - 1$$

$$= \frac{\frac{16000000}{9600}}{16} - 1$$

$$= [103.17] = 103$$

$$Calculated\ Baud\ Rate = \frac{16000000}{16(103 + 1)}$$

$$= 9615$$

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

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

TABLE 31-1: BAUD RATE FORMULAS

BRGS	BRG/UART Mode	Baud Rate Formula
1	High Rate	FOSC/[4 (n+1)]
0	Normal Rate	FOSC/[16(n+1)]

Legend: n = value of UxBRGH, UxBRGL register pair.

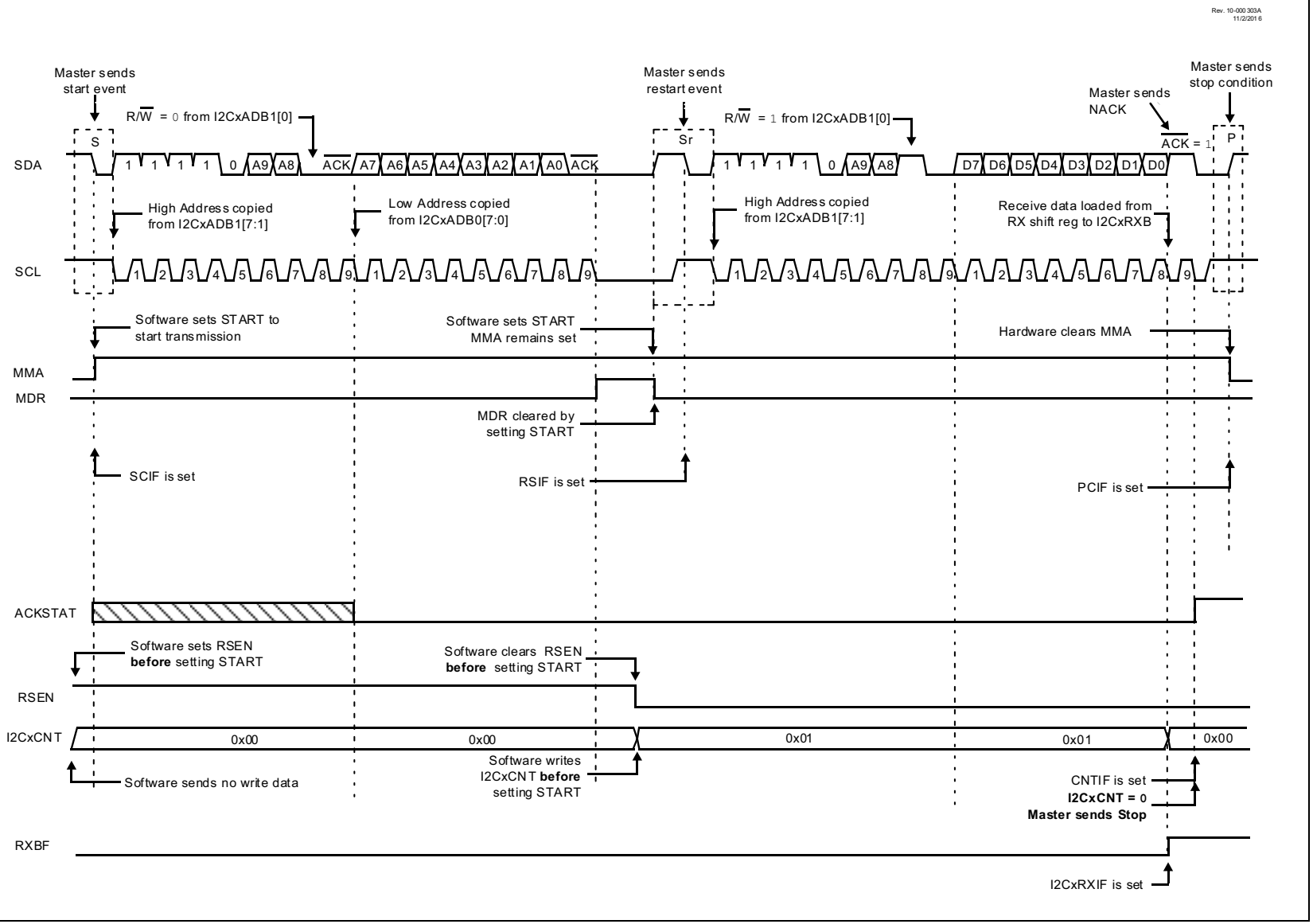
32.8.3.4 Receiver Overflow and Transmitter Underflow Interrupts

The receiver overflow interrupt triggers if data is received when the RXFIFO is already full and RXR = 1. In this case, the data will be discarded and the RXOIF bit will be set. The receiver overflow interrupt flag is the RXOIF bit of SPIxINTF. The receiver overflow interrupt enable bit is the RXOIE bit of SPIxINTE.

The Transmitter Underflow interrupt flag triggers if a data transfer begins when the TXFIFO is empty and TXR = 1. In this case, the most recently received data will be transmitted and the TXUIF bit will be set. The transmitter underflow interrupt flag is the TXUIF bit of SPIxINTF. The transmitter underflow interrupt enable bit is the TXUIE bit of SPIxINTE.

Both of these interrupts will only occur in Slave mode, as Master mode will not allow the RXFIFO to overflow or the TXFIFO to underflow.

FIGURE 33-22: I²C MASTER, 10-BIT ADDRESS, RECEPTION (USING RSTEN BIT)



REGISTER 36-9: ADPREL: ADC PRECHARGE TIME CONTROL REGISTER (LOW BYTE)

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
PRE<7:0>							
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 7-0 **PRE<7:0>**: Precharge Time Select bits
See Table 36-4.

REGISTER 36-10: ADPREH: ADC PRECHARGE TIME CONTROL REGISTER (HIGH BYTE)

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
—	—	—	PRE<12:8>				
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 PRE<12:8>: Precharge Time Select bits
See Table 36-4.

Note: If PRE is not equal to '0', then ADACQ = b'00000000 means Acquisition time is 256 clocks of the selected ADC clock.

TABLE 36-4: PRECHARGE TIME

ADPRE	Precharge time
1 1111 1111 1111	8191 clocks of the selected ADC clock
1 1111 1111 1110	8190 clocks of the selected ADC clock
1 1111 1111 1101	8189 clocks of the selected ADC clock
...	...
0 0000 0000 0010	2 clocks of the selected ADC clock
0 0000 0000 0001	1 clock of the selected ADC clock
0 0000 0000 0000	Not included in the data conversion cycle

CALLW **Subroutine Call Using WREG**

Syntax: CALLW

Operands: None

Operation: (PC + 2) → TOS,
(W) → PCL,
(PCLATH) → PCH,
(PCLATU) → PCU

Status Affected: None

Encoding:

0000	0000	0001	0100

Description: First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a NOP instruction while the new next instruction is fetched. Unlike CALL, there is no option to update W, Status or BSR.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read WREG	PUSH PC to stack	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    CALLW
```
Before Instruction
PC = address (HERE)
PCLATH = 10h
PCLATU = 00h
W = 06h
After Instruction
PC = 001006h
TOS = address (HERE + 2)
PCLATH = 10h
PCLATU = 00h
W = 06h

CLRF **Clear f**

Syntax: CLRF f {,a}

Operands: 0 ≤ f ≤ 255
a ∈ [0,1]

Operation: 000h → f
1 → Z

Status Affected: Z

Encoding:

0110	101a	ffff	ffff

Description: Clears the contents of the specified register.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example:
```
CLRF    FLAG_REG, 1
```
Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

CPFSLT **Compare f with W, skip if f < W**

Syntax: CPFSLT f {,a}

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	000a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction. If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    CPFSLT REG, 1
NLESS   :
LESS    :
```

Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG < W;
PC = Address (LESS)
If REG ≥ W;
PC = Address (NLESS)

DAW **Decimal Adjust W Register**

Syntax: DAW

Operands: None

Operation: If [W<3:0> > 9] or [DC = 1] then
(W<3:0>) + 6 → W<3:0>;
else
(W<3:0>) → W<3:0>;

If [W<7:4> + DC > 9] or [C = 1] then
(W<7:4>) + 6 + DC → W<7:4> ;
else
(W<7:4>) + DC → W<7:4>

Status Affected: C

Encoding:

0000	0000	0000	0111

Description: DAW adjusts the 8-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register W	Process Data	Write W

Example1:
```
DAW
```
Before Instruction
W = A5h
C = 0
DC = 0
After Instruction
W = 05h
C = 1
DC = 0

Example 2:
Before Instruction
W = CEh
C = 0
DC = 0
After Instruction
W = 34h
C = 1
DC = 0

RLNCF — Rotate Left f (No Carry)

Syntax: RLNCF f {,d {,a}}

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

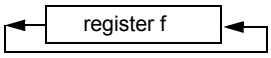
Operation: (f<n>) → dest<n + 1>,
(f<7>) → dest<0>

Status Affected: N, Z

Encoding:

0100	01da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: RLNCF REG, 1, 0

Before Instruction
REG = 1010 1011
After Instruction
REG = 0101 0111

RRCF — Rotate Right f through Carry

Syntax: RRCF f {,d {,a}}

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

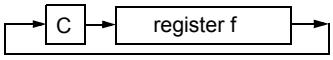
Operation: (f<n>) → dest<n – 1>,
(f<0>) → C,
(C) → dest<7>

Status Affected: C, N, Z

Encoding:

0011	00da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the CARRY flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



Words: 1

Cycles: 1

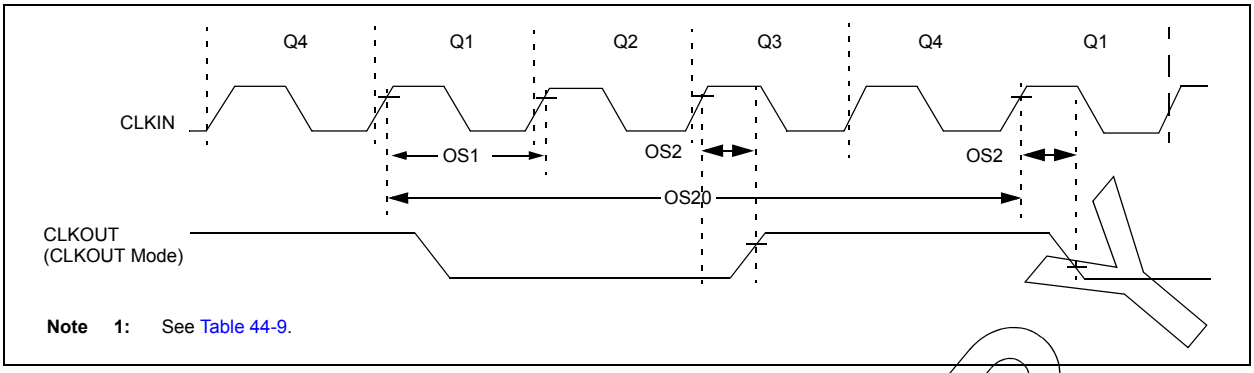
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: RRCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 0111 0011
C = 0

FIGURE 44-5: CLOCK TIMING



Note 1: See Table 44-9.

TABLE 44-9: EXTERNAL CLOCK/OSCILLATOR TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
ECL Oscillator							
OS1	F_{ECL}	Clock Frequency	—	—	500	kHz	
OS2	T_{ECL_DC}	Clock Duty Cycle	40	—	60	%	
ECM Oscillator							
OS3	F_{ECM}	Clock Frequency	—	—	8	MHz	
OS4	T_{ECM_DC}	Clock Duty Cycle	40	—	60	%	
ECH Oscillator							
OS5	F_{ECH}	Clock Frequency	—	—	64	MHz	
OS6	T_{ECH_DC}	Clock Duty Cycle	40	—	60	%	
LP Oscillator							
OS7	F_{LP}	Clock Frequency	—	—	100	kHz	**Note 4**
XT Oscillator							
OS8	F_{XT}	Clock Frequency	—	—	4	MHz	**Note 4**
HS Oscillator							
OS9	F_{HS}	Clock Frequency	—	—	20	MHz	**Note 4**
Secondary Oscillator							
OS10	F_{SEC}	Clock Frequency	32.4	32.768	33.1	kHz	
System Oscillator							
OS20	F_{OSC}	System Clock Frequency	—	—	64	MHz	**(Note 2, Note 3)**

* 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: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: The system clock frequency (FOSC) is selected by the "main clock switch controls" as described in **Section 10.0 "Power-Saving Operation Modes"**.

3: The system clock frequency (FOSC) must meet the voltage requirements defined in the **Section 44.2 "Standard Operating Conditions"**.

4: LP, XT and HS oscillator modes require an appropriate crystal or resonator to be connected to the device. For clocking the device with the external square wave, one of the EC mode selections must be used.