

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XF

Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	16KB (5.5K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VQFN Exposed Pad
Supplier Device Package	20-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f16ka101-e-mq

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Table of Contents

1.0	Device Overview	9
2.0	Guidelines for Getting Started with 16-Bit Microcontrollers	17
3.0	CPU	
4.0	Memory Organization	29
5.0	Flash Program Memory	45
6.0	Data EEPROM Memory	51
7.0	Resets	57
8.0	Interrupt Controller	
9.0	Oscillator Configuration	
10.0	Power-Saving Features	101
11.0	I/O Ports	113
12.0	Timer1	115
13.0	Timer2/3	117
14.0	Input Capture	123
15.0	Output Compare	125
16.0	Serial Peripheral Interface (SPI)	131
17.0	Inter-Integrated Circuit (I ² C [™])	139
18.0	Universal Asynchronous Receiver Transmitter (UART)	147
19.0	Real-Time Clock and Calendar (RTCC)	155
20.0	Programmable Cyclic Redundancy Check (CRC) Generator	167
21.0	High/Low-Voltage Detect (HLVD)	171
22.0	10-Bit High-Speed A/D Converter	173
23.0	Comparator Module	183
24.0	Comparator Voltage Reference	187
25.0	Charge Time Measurement Unit (CTMU)	189
26.0	Special Features	193
27.0	Development Support	203
28.0	Instruction Set Summary	207
29.0	Electrical Characteristics	215
30.0	Packaging Information	
Appe	endix A: Revision History	
Index	<	
The I	Microchip Web Site	
Custo	omer Change Notification Service	
Cust	omer Support	
Read	ler Kesponse	
Prod	uct Identification System	

NOTES:

REGISTER 3-2: CORCON: CPU CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0, HSC	R/W-0	U-0	U-0
—	—	—	—	IPL3 ⁽¹⁾	PSV	—	—
bit 7							bit 0

Legend:	HSC = Hardware Settable/C	learable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	l as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-4	Unimplemented: Read as '0'
bit 3	IPL3: CPU Interrupt Priority Level Status bit ⁽¹⁾
	1 = CPU interrupt priority level is greater than 70 = CPU interrupt priority level is 7 or less
bit 2	PSV: Program Space Visibility in Data Space Enable bit
	1 = Program space is visible in data space
	0 = Program space is not visible in data space
bit 1-0	Unimplemented: Read as '0'

Note 1: User interrupts are disabled when IPL3 = 1.

3.3 Arithmetic Logic Unit (ALU)

The PIC24F ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are 2's complement in nature. Depending on the operation, the ALU may affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array, or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

The PIC24F CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware division for 16-bit divisor.

3.3.1 MULTIPLIER

The ALU contains a high-speed, 17-bit x 17-bit multiplier. It supports unsigned, signed or mixed sign operation in several multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

TABLE 4-12: PORTA REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5 ⁽¹⁾	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISA	02C0	_	_	_	_	_	_	_		TRISA7 ⁽⁴⁾	TRISA6	_	TRISA4	TRISA3 ^(5,6)	TRISA2 ⁽⁵⁾	TRISA1	TRISA0	00DF
PORTA	02C2	_	-	-	_	—	_	_	_	RA7 ⁽⁴⁾	RA6	RA5	RA4 ⁽³⁾	RA3 ^(5,6)	RA2 ⁽⁵⁾	RA1 ⁽²⁾	RA0 ⁽²⁾	xxxx
LATA	02C4	_	—	—	—	—	_	_	_	LATA7 ⁽⁴⁾	LATA6	_	LATA4	LATA3 ^(5,6)	LATA2 ⁽⁵⁾	LATA1	LATA0	xxxx
ODCA	02C6	_	_	—	_	—	_			ODA7 ⁽⁴⁾	ODA6	_	ODA4	ODA3 ^(5,6)	ODA2 ⁽⁵⁾	ODA1	ODA0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: This bit is available only when MCLRE = 0.

2: A read of RA1 and RA0 results in '0' when debug is active on the PGC2/PGD2 pin.

3: A read of RA4 results in '0' when debug is active on the PGC3/PGD3 pin.

4: These bits are not implemented in 20-pin devices.

5: These bits are available only when the primary oscillator is disabled (POSCMD<1:0> = 00); otherwise read as '0'.

6: These bits are available only when the primary oscillator is disabled or EC mode is selected (POSCMD<1:0> = 00 or 11) and CLKO is disabled (OSCIOFNC = 0); otherwise read as '0'.

TABLE 4-13:PORTB REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11 ⁽³⁾	TRISB10 ⁽³⁾	TRISB9	TRISB8	TRISB7	TRISB6 ⁽³⁾	TRISB5 ⁽³⁾	TRISB4	TRISB3(3)	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11 ⁽³⁾	RB10 ⁽³⁾	RB9	RB8	RB7	RB6 ⁽³⁾	RB5 ⁽³⁾	RB4 ⁽²⁾	RB3 ⁽³⁾	RB2	RB1 ⁽¹⁾	RB0 ⁽¹⁾	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11 ⁽³⁾	LATB10 ⁽³⁾	LATB9	LATB8	LATB7	LATB6 ⁽³⁾	LATB5 ⁽³⁾	LATB4	LATB3 ⁽³⁾	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: A read of RB1 and RB0 results in '0' when debug is active on the PGEC1/PGED1 pins.

2: A read of RB4 results in '0' when debug is active on the PGEC3/PGED3 pins.

3: PORTB bits, 11, 10, 6, 5 and 3, are not implemented in 20-pin devices.

TABLE 4-14: PAD CONFIGURATION REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PADCFG1	02FC	_	—	—	_	—	—	—	—	—	—	—	SMBUSDEL	OC1TRIS	RTSECSEL1	RTSECSEL0	_	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-17: REAL-TIME CLOCK AND CALENDAR REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ALRMVAL	0620	Alarm Value Register Window Based on ALRMPTR<15:0>															xxxx	
ALCFGRPT	0622	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0624						RTCC	Value Registe	r Window Bas	ed on RTC	CPTR<15:0	>						xxxx
RCFGCAL	0626	RTCEN	-	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	0000
La mana di						un lus la num el e	alian al											

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-18: DUAL COMPARATOR REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CMSTAT	0630	CMSIDL	_	_	-	_	_	C2EVT	C1EVT	—	—	_	_	_	—	C2OUT	C10UT	0000
CVRCON	0632	_	_	_	_		_	_	_	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0634	CON	COE	CPOL	CLPWR		_	CEVT	COUT	EVPOL1	EVPOL0	_	CREF	_	_	CCH1	CCH0	0000
CM2CON	0636	CON	COE	CPOL	CLPWR		_	CEVT	COUT	EVPOL1	EVPOL0	_	CREF	_	_	CCH1	CCH0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-19: CRC REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CRCCON	0640	—	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0	CRCFUL	CRCMPT	-	CRCGO	PLEN3	PLEN2	PLEN1	PLEN0	0040
CRCXOR	0642		X<15:1>														_	0000
CRCDAT	0644							(CRC Data li	nput Registe	er							0000
CRCWDAT	0646	CRC Result Register														0000		

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

8.3 Interrupt Control and Status Registers

The PIC24F16KA102 family of devices implements a total of 22 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0, IFS1, IFS3 and IFS4
- · IEC0, IEC1, IEC3 and IEC4
- IPC0 through IPC5, IPC7 and IPC15 through IPC19
- INTTREG

Global interrupt control functions are controlled from INTCON1 and INTCON2. INTCON1 contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources. The INTCON2 register controls the external interrupt request signal behavior and the use of the AIV table.

The IFSx registers maintain all of the interrupt request flags. Each source of interrupt has a status bit, which is set by the respective peripherals, or external signal, and is cleared via software.

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals.

The IPCx registers are used to set the interrupt priority level for each source of interrupt. Each user interrupt source can be assigned to one of eight priority levels. The INTTREG register contains the associated interrupt vector number and the new CPU interrupt priority level, which are latched into the Vector Number (VECNUM<6:0>) and the Interrupt Level (ILR<3:0>) bit fields in the INTTREG register. The new interrupt priority level is the priority of the pending interrupt.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the same sequence listed in Table 8-2. For example, the INT0 (External Interrupt 0) is depicted as having a vector number and a natural order priority of 0. Thus, the INT0IF status bit is found in IFS0<0>, the INT0IE enable bit in IEC0<0> and the INT0IP<2:0> priority bits in the first position of IPC0 (IPC0<2:0>).

Although they are not specifically part of the interrupt control hardware, two of the CPU control registers contain bits that control interrupt functionality. The ALU STATUS register (SR) contains the IPL<2:0> bits (SR<7:5>). These indicate the current CPU interrupt priority level. The user may change the current CPU priority level by writing to the IPL bits.

The CORCON register contains the IPL3 bit, which together with IPL<2:0>, also indicates the current CPU priority level. IPL3 is a read-only bit so that the trap events cannot be masked by the user's software.

All interrupt registers are described in Register 8-1 through Register 8-21, in the following sections.

REGISTER 8-7: IFS3: INTERRUPT FLAG STATUS REGISTER 3

- RTCIF	U-0	R/W-0, HS	U-0	U-0	U-0	U-0	U-0	U-0
bit 15 bit	—	RTCIF	—	—	—	—	—	—
	bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—		—	—	—
bit 7							bit 0

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

bit 15 Unimplemented: Read as '0'

bit 14 **RTCIF:** Real-Time Clock and Calendar Interrupt Flag Status bit 1 = Interrupt request has occurred 0 = Interrupt request has not occurred

bit 13-0 Unimplemented: Read as '0'

REGISTER	R 8-12: IEC4	: INTERRUPT	ENABLE C	ONTROL REC	GISTER 4						
U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0				
_	—	CTMUIE	_	—		—	HLVDIE				
bit 15		· · ·			•		bit 8				
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0				
	_	—	—	CRCIE	U2ERIE	U1ERIE	_				
bit 7							bit 0				
Legend:											
R = Readab	ole bit	W = Writable b	bit	U = Unimplem	nented bit, read	d as '0'					
-n = Value a	at POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkn	iown				
bit 15-14	Unimplemer	Unimplemented: Read as '0'									
bit 13	CTMUIE: CTMU Interrupt Enable bit										
	1 = Interrupt request is enabled										
h:+ 40.0		0 = Interrupt request is not enabled									
DIC 12-9	Unimplemen										
DIT 8	HLVDIE: Hig	n/Low-voltage D	etect interrup	t Enable bit							
	0 = Interrupt	request is not er	eu nabled								
bit 7-4	Unimplemer	nted: Read as '0	,								
bit 3	CRCIE: CRC	Generator Inter	rupt Enable b	it							
	1 = Interrupt	request is enable	ed								
	0 = Interrupt	request is not er	abled								
bit 2	U2ERIE: UART2 Error Interrupt Enable bit										
	1 = Interrupt	1 = Interrupt request is enabled									
	0 = Interrupt	request is not er	abled								
bit 1	U1ERIE: UA	RT1 Error Interru	ipt Enable bit								
	1 = Interrupt	request is enable	ed Vabled								
bit Ω		ted. Bead as 'n	,								
	Jumplemen	ited. Itedu do U									

REGISTER	8-13: IPC	0: INTERRUPT	PRIORITY	CONTROL R	EGISTER 0		
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
	T1IP2	T1IP1	T1IP0		OC1IP2	OC1IP1	OC1IP0
bit 15							bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0
bit 7				•		•	bit 0
Legend:							
R = Readab	le bit	W = Writable	bit	U = Unimple	mented bit, rea	d as '0'	
-n = Value a	It POR	'1' = Bit is set		'0' = Bit is cle	eared	x = Bit is unkr	nown
bit 15	Unimpleme	ented: Read as ')'				
bit 14-12	T1IP<2:0>:	Timer1 Interrupt	Priority bits				
	111 = Interr	rupt is Priority 7 (highest priorit	y interrupt)			
	•						
	•						
	001 = Interr 000 = Interr	rupt is Priority 1 rupt source is dis	abled				
bit 11	Unimpleme	ented: Read as 'o)'				
bit 10-8	OC1IP<2:0:	>: Output Compa	re Channel 1	Interrupt Priori	ty bits		
	111 = Interr	rupt is Priority 7 (highest priorit	y interrupt)			
	•						
	•						
	001 = Interr 000 = Interr	rupt is Priority 1 rupt source is dis	abled				
bit 7	Unimpleme	ented: Read as 'o)'				
bit 6-4	IC1IP<2:0>	: Input Capture C	hannel 1 Inte	rrupt Priority bi	ts		
	111 = Interr	rupt is Priority 7 (highest priorit	y interrupt)			
	•						
	•						
	001 = Interr 000 = Interr	rupt is Priority 1 rupt source is dis	abled				
bit 3	Unimpleme	ented: Read as ')'				
bit 2-0 INTOIP<2:0>: External Interrupt 0 Priority bits							
	111 = Interr	rupt is Priority 7 (highest priorit	y interrupt)			
	•						
	•						
	• 001 = Interr	rupt is Priority 1					
	000 = Interr	rupt source is dis	abled				

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0	
_	—	_	_	—	RTCIP2	RTCIP1	RTCIP0	
bit 15							bit 8	
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
_	—	—	—	—		—	—	
bit 7							bit 0	
Legend:								
R = Readable bit W = V		W = Writable	bit	U = Unimplemented bit, read as '0'				
-n = Value a	t POR	'1' = Bit is set	t '0' = Bit is		ared	x = Bit is unkr	nown	
bit 15-11	Unimplemen	ted: Read as '	0'					
bit 10-8	RTCIP<2:0>:	Real-Time Clo	ck and Calend	lar Interrupt Prie	ority bits			
	111 = Interru	pt is Priority 7 (highest priority	interrupt)				
	•							
	•							
	•							
	001 = Interru	pt is Priority 1						
	000 = Interru	pt source is dis	abled					
bit 7-0	Unimplemen	ted: Read as '	0'					

REGISTER 8-20: IPC15: INTERRUPT PRIORITY CONTROL REGISTER 15

REGISTER 10-4: PMD2: PERIPHERAL MODULE DISABLE REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	I2C1MD
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0

—	—	_				—	OC1MD
bit 7							bit 0

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

bit 15-9 Unimplemented: Read as '0'

bit 8 I2C1MD: Input Capture 1 Module Disable bit 1 = Input Capture 1 module is disabled. All Input Capture registers are held in Reset and are not writable. 0 = Input Capture 1 module is writable

bit 7-1 Unimplemented: Read as '0'

bit 0 OC1MD: Input Compare 1 Module Disable bit

- 1 = Output Compare 1 module is disabled. All Output Compare registers are held in Reset and are not writable.
- 0 = Output Compare 1 module is writable

15.3 Pulse-Width Modulation (PWM) Mode

The following steps should be taken when configuring the output compare module for PWM operation:

- 1. Set the PWM period by writing to the selected Timer Period register (PRy).
- 2. Set the PWM duty cycle by writing to the OC1RS register.
- 3. Write the OC1R register with the initial duty cycle.
- 4. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
- Configure the output compare module for one of two PWM Operation modes by writing to the Output Compare Mode bits, OCM<2:0> (OC1CON<2:0>).
- 6. Set the TMRy prescale value and enable the time base by setting TON (TxCON<15>) = 1.
- Note: The OC1R register should be initialized before the output compare module is first enabled. The OC1R register becomes a read-only Duty Cycle register when the module is operated in the PWM modes. The value held in OC1R will become the PWM duty cycle for the first PWM period. The contents of the Output Compare 1 Secondary register, OC1RS, will not be transferred into OC1R until a time base period match occurs.

15.3.1 PWM PERIOD

The PWM period is specified by writing to PRy, the Timer Period register. The PWM period can be calculated using Equation 15-1.

EQUATION 15-1: CALCULATING THE PWM PERIOD⁽¹⁾

PWM Period = $[(PRy) + 1] \bullet TCY \bullet (Timer Prescale Value)$ where:

PWM Frequency = 1/[PWM Period]

- Note 1: Based on Tcy = 2 * Tosc; Doze mode and PLL are disabled.
- Note: A PRy value of N will produce a PWM period of N + 1 time base count cycles. For example, a value of 7, written into the PRy register, will yield a period consisting of 8 time base cycles.

15.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the OC1RS register. The OC1RS register can be written to at any time, but the duty cycle value is not latched into OC1R until a match between PRy and TMRy occurs (i.e., the period is complete). This provides a double buffer for the PWM duty cycle and is essential for glitchless PWM operation. In PWM mode, OC1R is a read-only register.

Some important boundary parameters of the PWM duty cycle include:

- If the Output Compare 1 register, OC1R, is loaded with 0000h, the OC1 pin will remain low (0% duty cycle).
- If OC1R is greater than PRy (Timer Period register), the pin will remain high (100% duty cycle).
- If OC1R is equal to PRy, the OC1 pin will be low for one time base count value and high for all other count values.

See Example 15-1 for PWM mode timing details. Table 15-1 provides an example of PWM frequencies and resolutions for a device operating at 10 MIPS.

EQUATION 15-2: CALCULATION FOR MAXIMUM PWM RESOLUTION⁽¹⁾



Note 1: Based on Fcy = Fosc/2; Doze mode and PLL are disabled.

REGISTER 17-3: I2C1MSK: I2C1 SLAVE MODE ADDRESS MASK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—		—	—	—	AMSK9	AMSK8
bit 15		-				-	bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| AMSK7 | AMSK6 | AMSK5 | AMSK4 | AMSK3 | AMSK2 | AMSK1 | AMSK0 |
| bit 7 | | | | | | | bit 0 |

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

bit 15-10 Unimplemented: Read as '0'

bit 9-0

AMSK<9:0>: Mask for Address Bit x Select bits

1 = Enable masking for bit x of incoming message address; bit match is not required in this position
 0 = Disable masking for bit x; bit match is required in this position

REGISTER 17-4: PADCFG1: PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	SMBUSDEL	OC1TRIS ^(2,3)	RTSECSEL1 ^(1,3)	RTSECSEL0 ^(1,3)	—
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 15-5 Unimplemented: Read as '0'

bit 4 SMBUSDEL: SMBus SDA Input Delay Select bit

1 = The I^2C module is configured for a longer SMBus input delay (nominal 300 ns delay)

0 = The 1²C module is configured for a legacy input delay (nominal 150 ns delay)

bit 0 Unimplemented: Read as '0'

Note 1: To enable the actual RTCC output, the RTCOE (RCFGCAL<10>) bit needs to be set.

2: To enable the actual OC1 output, the OCPWM1 module has to be enabled.

3: Bits<3:1> are described in related chapters.

REGISTER 23-1: CMxCON: COMPARATOR x CONTROL REGISTERS (CONTINUED)

- bit 5 Unimplemented: Read as '0'
- bit 4 **CREF:** Comparator Reference Select bit (non-inverting input) 1 = Non-inverting input connects to the internal CVREF voltage
 - 0 = Non-inverting input connects to the CxINA pin
- bit 3-2 Unimplemented: Read as '0'
- bit 1-0 CCH<1:0>: Comparator Channel Select bits
 - 11 = Inverting input of comparator connects to VBG/2
 - 10 = Inverting input of comparator connects to CxIND pin
 - 01 = Inverting input of comparator connects to CxINC pin
 - 00 = Inverting input of comparator connects to CxINB pin

REGISTER 23-2: CMSTAT: COMPARATOR MODULE STATUS REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC
CMIDL	—	—	—	—	—	C2EVT	C1EVT
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC

bit	7

Legend:	HSC = Hardware Settable/Clearable bit					
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 15	CMIDL: Comparator Stop in Idle Mode bit
	 1 = Disable comparator interrupts when the device enters Idle mode; the module is still enabled 0 = Continue operation of all enabled comparators in Idle mode
bit 14-10	Unimplemented: Read as '0'
bit 9	C2EVT: Comparator 2 Event Status bit (read-only)
	Shows the current event status of Comparator 2 (CM2CON<9>).
bit 8	C1EVT: Comparator 1 Event Status bit (read-only)
	Shows the current event status of Comparator 1 (CM1CON<9>).
bit 7-2	Unimplemented: Read as '0'
bit 1	C2OUT: Comparator 2 Output Status bit (read-only)
	Shows the current output of Comparator 2 (CM2CON<8>).
bit 0	C1OUT: Comparator 1 Output Status bit (read-only)
	Shows the current output of Comparator 1 (CM1CON<8>).

C2OUT

C10UT

bit 0

25.2 Measuring Time

Time measurements on the pulse width can be similarly performed using the A/D module's internal capacitor (CAD) and a precision resistor for current calibration. Figure 25-2 displays the external connections used for time measurements, and how the CTMU and A/D modules are related in this application. This example also shows both edge events coming from the external CTED pins, but other configurations using internal edge sources are possible.

25.3 Pulse Generation and Delay

The CTMU module can also generate an output pulse with edges that are not synchronous with the device's system clock. More specifically, it can generate a pulse with a programmable delay from an edge event input to the module. When the module is configured for pulse generation delay by setting the TGEN bit (CTMUCON<12>), the internal current source is connected to the B input of Comparator 2. A capacitor (CDELAY) is connected to the Comparator 2 pin, C2INB, and the comparator voltage reference, CVREF, is connected to C2INA. CVREF is then configured for a specific trip point. The module begins to charge CDELAY when an edge event is detected. When CDELAY charges above the CVREF trip point, a pulse is output on CTPLS. The length of the pulse delay is determined by the value of CDELAY and the CVREF trip point.

Figure 25-3 shows the external connections for pulse generation, as well as the relationship of the different analog modules required. While CTED1 is shown as the input pulse source, other options are available. A detailed discussion on pulse generation with the CTMU module is provided in the "*PIC24F Family Reference Manual*".

FIGURE 25-2: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT



FIGURE 25-3: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION



R/P-1	U-0	U-0	U-0	U-0	U-0	R/P-1	R/P-1
DEBUG	—	_	—			FICD1	FICD0
bit 7							bit 0
Legend:							
R = Readable bit P = Programmable bit U = Ur				U = Unimplem	nented bit, read	l as '0'	
-n = Value at POR		'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	
bit 7	DEBUG: Back	ground Debugg	jer Enable bit				
	1 = Backgroun	d debugger is o	disabled				
0 = Background debugger functions are enabled							
bit 6-2	6-2 Unimplemented: Read as '0'						
bit 1-0	bit 1-0 FICD<1:0:> ICD Pin Select bits						
11 = PGC1/PGD1 are used for programming and debugging the device							

10 = PGC2/PGD2 are used for programming and debugging the device

REGISTER 26-7: FICD: IN-CIRCUIT DEBUGGER CONFIGURATION REGISTER

01 = PGC3/PGD3 are used for programming and debugging the device

00 = Reserved; do not use

TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)

yms.xvyms.x	Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
NAME NUMBERNompNetwork Number Number Number Number Number Number Number Number Number Number Number 	PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO, Sleep
NumberNumberOperation (Non-structure)12NoneNEFKENEPSERState Net Instructure (Non-structure)11NoneNERSITRESITSchware Device Reset13NoneNERSITRESITReturn from Inforupi13NoneNETURRETURReturn from Inforupi13NoneNETURRETURReturn from Subrounice13NoneNETURRETURReturn from Subrounice13NoneNETURRETURReturn from Subrounice13NoneNETURRETURReturn from Subrounice11C. N.ZNETURNEGEFoldet Left Mocgarry (I11C. N.ZNEGEFoldet Left MocGarry (I11N.ZNEGEFoldet Left No Carry (I11N.ZNEGEFoldet Right Mocgary (I11N.ZNEGEFoldet Right MocGarry (I1N.ZN.ZNEGEFoldet Right Moc	RCALL	RCALL	Expr	Relative Call	1	2	None
BREPAR BLILLIABepace Nach Instruction (Un) + 1 times11002007200		RCALL	Wn	Computed Call	1	2	None
RETEXTRen.Repeat Next Instruction (Wn) + 1 times11NoneRETEYTReturn With Literal IN from Interrupt13.(2)NoneRETURNUMEReturn With Literal IN from Interrupt13.(2)NoneRETURRETURReturn With Literal IN from Interrupt13.(2)NoneRETURPLOSC. N. ZNoneNoneNoneRETURReturn Mon Interrupt110.N. ZRELC. N. RELReturn Monoshourium110.N. ZRELS. N. RELReturn Monoshourium11N. ZRELS. N. RELReturn Monoshourium11N. ZRELR. N. RECFootale Left Moogh Carry Monoshourium11N. ZRELS. N. RELS. S. S	REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
BABETServerSolvane Device Reset11NoneBETTIERELIW * 11c10, W:Retum from Interrupt113(2)NoneRETUWSETUWRetum from Subrouine113(2)NoneRETUWIf = Note Let Ithrough Carry f11110, N. Z.RECf. NOEDCWREC = Rotate Let Ithrough Carry f11110, N. Z.RECf. NOEDCWREC = Rotate Let Ithrough Carry f11110, N. Z.RECf. NOEDCf. Fortate Let Ithrough Carry f1111N. Z.RECf. NOEDCf. Fortate Let Ithrough Carry f1111N. Z.RECf. NOEDCf. Fortate Let Ithrough Carry f1111N. Z.RECf. NOEDCf. Fortate Right Ithrough Carry f1111N. Z.RECf. NERGf. Fortate Right Ithrough Carry f1111N. Z. <td></td> <td>REPEAT</td> <td>Wn</td> <td>Repeat Next Instruction (Wn) + 1 times</td> <td>1</td> <td>1</td> <td>None</td>		REPEAT	Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
BETTEReturn of miderup11110Non-BETLHRETTEReturn with Useral IN/N13/20NoneRETTEReturn for Subrourine13/20NoneRECFFeltatie Left Invogin Carry f1110.N.Z.RECFFeltatie Left Invogin Carry f1110.N.Z.REM2.C.No.R.GWREG = Robate Left Invogin Carry MS111N.Z.REM6.R.GCFeltatie Left Invogin Carry MS11N.Z.N.Z.REM6.R.GCFeltatie Right Invogin Carry MS11N.Z.N.Z.REM6.R.GFeltatie Right Invogin Carry MS11N.Z.N.Z.REM6.R.W.GFeltatie Right Invogin Carry MS11N.Z.N.Z.RENC6.R.W.GFeltatie Right Invogin Carry MS11N.R.N.Z.RENC6.R.W.GMREG = Robate Right Invogin Carry MS11N.R.N.R.RENC6.R.W.GMREG = Robate Right Invogin Carry MS11N.R.N.R.RENC6.R.W.RGMREG = Robate Right Invogin Carry MS11 <td< td=""><td>RESET</td><td>RESET</td><td></td><td>Software Device Reset</td><td>1</td><td>1</td><td>None</td></td<>	RESET	RESET		Software Device Reset	1	1	None
звтли втлив110 лиReturn with Lien Wit	RETFIE	RETFIE		Return from Interrupt	1	3 (2)	None
вктовявстовяестоваесто	RETLW	RETLW	#lit10,Wn	Return with Literal in Wn	1	3 (2)	None
NBCffrr<	RETURN	RETURN		Return from Subroutine	1	3 (2)	None
NRC 10e, NRDNRREGNRREGI, II, IC, N, ZRINCE, NR, MGGN, ZN, ZN, ZRINCE, NR, MMREGRelate Left (No Carry) fIIN, ZRINCS, NR, MMREGRelate Left (No Carry) fIIN, ZRINCS, NR, MMREGFactore Left (No Carry) fIIN, ZRINCS, NR, MMREGFactore Relate Relation (No Carry) fIIN, ZRINCS, NR, MMREGRelate Relation (No Carry) fIIN, ZRINCS, NRMREGRelate Relation (No Carry) fIIN, ZRINCS, NRMREGRelate Relation (No Carry) fIIN, ZRINCS, NRMREGRelate Relation (No Carry) fIIN, Z	RLC	RLC	f	f = Rotate Left through Carry f	1	1	C, N, Z
IntermInter		RLC	f,WREG	WREG = Rotate Left through Carry f	1	1	C, N, Z
RLNC £ Class f = Rotate Left (No Carry) f 1 1 N, Z RLNC 6, MRR0 WREG = Rotate Left (No Carry) f 1 1 N, Z RRC f, RREG WREG = Rotate Left (No Carry) f 1 1 N, Z RRC f, RREG f = Rotate Right through Carry f 1 1 C, N, Z RRC f, MREG WREG = Rotate Right (No Carry) f 1 1 N, Z RRNC f, MREG f = Rotate Right (No Carry) f 1 N, Z N, Z RRNC f, MREG f = Rotate Right (No Carry) f 1 N, Z N, Z RRNC f, MREG MREG = Rotate Right (No Carry) f 1 N, Z N, Z SE f, MREG WREG = FORER N N N, Z N, Z SE f, MREG WREG = FORER 1 N N, Z N, Z SE f, MREG MREG = FORER 1 1 N, Z N, Z SE SE M, MA WREG		RLC	Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C, N, Z
FLNC F, WERG WREG = Rotate Left (No Carry) fr 1 1 N, Z RRC RLNC Wa, Wd Wd = Rotate Left (No Carry) fr 1 1 N, Z RRC f, WERG MRE Colle Right through Carry f 11 0 N, Z RRC f, WERG WREG = Rotate Right through Carry f 11 1 N, Z RRNC f, WERG MREG e Rotate Right (No Carry) f 11 N, Z RRNC f, WERG WREG = Rotate Right (No Carry) f 11 N, Z RRNC f, WERG WREG = Rotate Right (No Carry) f 11 N, Z RRNC W.R.WG WREG = Rotate Right (No Carry) f 11 N, Z SETM WR.WG WREG = FEFFh 11 N, R SETM WREG FEFFh 11 N, Z SET f, WERG WREG = Left Shift f 11 N, Z SE W.R.WA Wad Left Shift f 11 N, Z SE W.R.WA Wad Left Shift f 11 N, Z	RLNC	RLNC	f	f = Rotate Left (No Carry) f	1	1	N, Z
KLNC Ms, Md Wd = Rotate Left (No Carry) Ws 1 1 N, Z RRC f, NRRG WRRGE - Rotate Right through Carry f 1 1 C, N, Z RRC Ms, Md Wd = Rotate Right through Carry f 1 1 C, N, Z RRC f, NRRG WRRGE - Rotate Right (No Carry) f 1 1 N, Z RRNC f, NRRG WRRGE - Rotate Right (No Carry) f 1 1 N, Z RRNC f, NRRG WM = Rotate Right (No Carry) f 1 1 N, Z SE Ns, Md Wd = Rotate Right (No Carry) f 1 1 N, Z SE Ns, Md Wd = Rotate Right (No Carry) fW 1 1 N, Z SE Ns, Md Wd = Rotate Right (No Carry) fW 1 1 N, Z SE Ns, Md WRG = FERFh 1 1 None SETM MS FFFFh 1 1 N, Z SL f, NRRG WREG = For NREG 1 1 N, Z		RLNC	f,WREG	WREG = Rotate Left (No Carry) f	1	1	N, Z
RRC f f = Rotate Right through Carry f 1 1 C, N, Z RC f, NEBO WREG = Rotate Right through Carry f 1 1 C, N, Z RRNC Easc f = Cotate Right through Carry Ms 1 1 C, N, Z RRNC Easc f = Cotate Right through Carry Ms 1 1 N, Z RRNC F, NREG WREG = Rotate Right (No Carry) MS 1 1 N, Z SET F, NREG WREG = Rotate Right (No Carry) MS 1 1 N, Z SET Ws, Md Wd = Sign-Extended WS 1 1 N, Z SET Ws, Md Wd = Sign-Extended WS 1 1 N, Z SET WREG f = Rotate Right Mrough Carry MS 1 1 N, Z SET Ws, Md Wd = Sign-Extended WS 1 1 N, Z SET Ws, Md WREG = I celt Shift 1 1 1 N, C SL f , NREG MM = Uelf Shift Wb by MIS 1 1 <th< td=""><td></td><td>RLNC</td><td>Ws,Wd</td><td>Wd = Rotate Left (No Carry) Ws</td><td>1</td><td>1</td><td>N, Z</td></th<>		RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N, Z
RRC f, WREG WREG = Rotate Right through Carry ff 1 1 C, N, Z RRC We, Nd We = Rotate Right through Carry MS 1 1 C, N, Z RRNC f, MREG F. Rotate Right (No Carry) f 1 1 N, Z RRNC f, MREG WREG = Rotate Right (No Carry) f 1 1 N, Z SE MS, Mad Wel = Rotate Right (No Carry) MS 1 1 N, Z SET MS, Mad Wal = Rotate Right (No Carry) MS 1 1 N, Z SET MS, Mad Wal = Rotate Right (No Carry) MS 1 1 N, Z SET MS, Mad WREG = FFFFh 1 1 None SET MREG MREG = Left Shift f 1 1 N, NO, Z SL MS, Man, Xnd Wal = Left Shift Wob y Wns 1 1 N, Z SL MS, Man, Xnd Wal = Left Shift Wob y Wns 1 1 N, Z SL MS, Man, Xnd Wal = Left Shift Wob y Mis 1 1	RRC	RRC	f	f = Rotate Right through Carry f	1	1	C, N, Z
RRC We, Md Wd = Rotate Right through Carry Ws 1 1 C, N, Z RENC F. F. Rotate Right (No Carry) f 1 1 N, Z RENC F., WEG Rotate Right (No Carry) f 1 1 N, Z SE F. Wa, Wd Wd = Rotate Right (No Carry) Ws 1 1 N, Z SE Wa, Wd Wd = Rotate Right (No Carry) Ws 1 1 N, Z SE Wa, Wd Wd = Rotate Right (No Carry) Ws 1 1 None SETM WRSD Mref SertFFFh 1 1 None SE F. HREG FLERTShift 1 1 C, N, OV, Z SL F. WREG Inf IMW 1 1 C, N, OV, Z SL W., Wd Wd = Left Shift Wb Dy Wns 1 1 C, N, OV, Z SL W., Wa Wo = Left Shift Wb Dy Wns 1 1 C, OC, N, OV, Z SL W., Wa Wom = Left Shift Wb Dy Wns 1		RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C, N, Z
RRNC f Rotate Right (No Carry) f 1 1 N, Z RRNC f, WREG WREG = Rotate Right (No Carry) f 1 1 N, Z RRNC Wa, Wd Wd = Rotate Right (No Carry) Ws 1 1 N, Z SE Ws, Wd Wd = Ston-Exclended Ws 1 1 C, N, Z SETM SETM f f FFFFh 1 1 None SETM WREG WREG = FFFFh 1 1 C, N, OV, Z None SET f, WREG WREG = Left Shift f 1 1 C, N, OV, Z SL f, Wna, Wd Wd = Left Shift Wb by Wins 1 1 N, Z SL Wb, Wa, Wd Wnd = Left Shift Wb by Wins 1 1 N, Z SUB f, WREG WREG = Left Shift Wb by Wins 1 1 N, Z SUB f, WREG Wref = f-WREG 1 1 N, Z SUB f, WREG Wref = f-WREG 1 1 C, DC, N, OZ </td <td></td> <td>RRC</td> <td>Ws,Wd</td> <td>Wd = Rotate Right through Carry Ws</td> <td>1</td> <td>1</td> <td>C, N, Z</td>		RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C, N, Z
RRNC f, WREG WREG = Rotate Right (No Carry) f 1 1 N, Z SE NRNC Wa, Wd Wd = Rotate Right (No Carry) Ws 1 1 N, Z SE SE Wa, Wnd Wnd = Sign-Extended Ws 1 1 C, N, Z SETM f f FFFFh 1 1 None SETM WREG WREG = FFFFh 1 1 None SL f f Left Shift 1 1 C, N, OV, Z SL f, MRRG WREG = Left Shift f 1 1 N, Z SL ws, Wd Wd = Left Shift Wb by Wns 1 1 N, Z SL Wb, Mna, Wnd Wnd = Left Shift Wb by Bifs 1 1 N, Z SUB f f FFFEN 1 1 N, Z SUB f MRRG Wd = Left Shift Wb by Mns 1 1 N, Z SUB SUB f, MRRG Wd = Left Shift Wb by Mns 1 1 </td <td>RRNC</td> <td>RRNC</td> <td>f</td> <td>f = Rotate Right (No Carry) f</td> <td>1</td> <td>1</td> <td>N, Z</td>	RRNC	RRNC	f	f = Rotate Right (No Carry) f	1	1	N, Z
RRNC Ws, Wd Wd = Rotate Right (No Carry) Ws 1 1 N, Z SE SE Wa, Wnd Wnd = Sign-Extended Ws 1 1 C, N, Z SETM £ F FFFFh 1 1 None SETM WREG WREG = FFFFh 1 1 None SL f f f 1 1 1 None SL f f f f 1 1 1 None SL f f f f f 1 1 1 0 NOV,Z SL Ms, Mad Wd = Left Shift f 1 1 1 N,Z NOV,Z SL Ws, Mad Wd = Left Shift Wb by Wns 1 1 1 N,Z NOV,Z SL Ws, Mad Wd = Left Shift Wb by Wns 1 1 1 C, N,OV,Z SUB f f f f F N,V,Z NOV,Z </td <td></td> <td>RRNC</td> <td>f,WREG</td> <td>WREG = Rotate Right (No Carry) f</td> <td>1</td> <td>1</td> <td>N, Z</td>		RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N, Z
SE SE NS, Mnd Wnd = Sign-Extended Ws 1 1 C, N, Z SETM f FFFFh 11 1 None SETM KEM WREG WREG = FFFFh 11 1 None SL f f=Left Shift f 11 1 None SL f, WREG WREG = Left Shift f 11 1 C, N, OV, Z SL f, WREG WREG = Left Shift Wb by Wns 11 1 C, N, OV, Z SL Ns, Wad Wde Left Shift Wb by Wns 11 1 N, Z SUB f, WREG Wnd = Left Shift Wb by Wns 1 1 N, Z SUB f f=-WREG Wnd = Left Shift Wb by It5 1 1 N, Z SUB f.WREG WREG = FWREG 11 1 C, DC, N, OV, Z SUB f.WREG WREG = FWREG 1 1 C, DC, N, OV, Z SUB f.WREG WREG = WREG = C, C, O 1 1 C, DC, N, OV, Z		RRNC	Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N, Z
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	SE	SE	Ws,Wnd	Wnd = Sign-Extended Ws	1	1	C, N, Z
SETM WREG WREG = FFFFh 1 1 1 None SL f f Left Shift f 1 1 0	SETM	SETM	£	f = FFFFh	1	1	None
SETM Ws Ws = FFFFh 1 1 None SL f f=Left Shift f 1 1 C, N, OV, Z SL f, WREG WREG = Left Shift f 1 1 C, N, OV, Z SL Ws, Wd Wd = Left Shift Wb by Wns 1 1 C, N, OV, Z SL Ws, Wa, Wnd Wd = Left Shift Wb by Wns 1 1 N, Z SL Wb, Wns, Wnd Wnd = Left Shift Wb by Wns 1 1 N, Z SUB f f=f-WREG 1 1 C, DC, N, OV, Z SUB # Jit10, Wn Wn = Wn - IN10 1 1 C, DC, N, OV, Z SUB # Jit10, Wn Wd = Wb - Ws 1 1 C, DC, N, OV, Z SUB # Jit10, Wn Wd = Wb - Ws 11 1 C, DC, N, OV, Z SUBB # Jit10, Wn Wd = Wb - Ws 11 1 C, DC, N, OV, Z SUBB # Jit10, Wn Wn = Wn - IN10 1 1 C, DC, N, OV, Z SUBB		SETM	WREG	WREG = FFFFh	1	1	None
$ SL & SL & f & f = Left Shift f & 1 & 1 & 1 & 1 & C, N, OV, Z \\ \hline SL & f, WREG & WREG = Left Shift f & 1 & 1 & C, N, OV, Z \\ \hline SL & Wb, Wa & Wd = Left Shift Wb & 1 & 1 & C, N, OV, Z \\ \hline SL & Wb, Wa, Wnd & Wnd = Left Shift Wb & Wns & 1 & 1 & N, Z \\ \hline SL & Wb, #115, Wnd & Wnd = Left Shift Wb by Wns & 1 & 1 & N, Z \\ \hline SL & Wb, #115, Wnd & Wnd = Left Shift Wb by Wns & 1 & 1 & N, Z \\ \hline SUB & f & f = f - WREG & 1 & 1 & C, DC, N, OV, Z \\ \hline SUB & #11c10, Wn & Wn = Wn - Wn - Wn - Wn - Wn & 1 & 1 & C, DC, N, OV, Z \\ \hline SUB & #11c10, Wn & Wn = Wn - Wn - Wn - Wn & 1 & 1 & C, DC, N, OV, Z \\ \hline SUB & #11c10, Wn & Wn = Wn - Wn - Wn & 1 & 1 & C, DC, N, OV, Z \\ \hline SUB & Wb, #11t5, Wd & Wd = Wb - Ws & 1 & 1 & C, DC, N, OV, Z \\ \hline SUB & Wb, #11t5, Wd & Wd = Wb - Ws & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & K & f = f - WREG - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & f, WREG & WREG = f - WREG - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & Wb, #11t5, Wd & Wd = Wb - Ws - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & Wb, #11t5, Wd & Wd = Wb - Ws - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & Wb, #11t5, Md & Wd = Wb - Ws - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & Wb, #11t5, Md & Wd = Wb - Ws - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & Wb, #11t5, Md & Wd = Wb - Wb - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & Wb, #11t5, Md & Wd = WB - Wb - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBB & Wb, #11t5, Md & Wd = WB - Wb - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBR & Wb, Ws, Md & Wd = Ws - Wb & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBR & Wb, Ws, Md & Wd = WREG - f - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBR & Wb, Ws, Wd & Wd = WREG - f - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBBR & Wb, Ws, Wd & Wd = WREG - f - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBBR & Wb, Ws, Wd & Wd = WREG - f - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBBR & Wb, Ws, Wd & Wd = WREG - F - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBBR & Wb, Ws, Wd & Wd = WREG - WREG - f - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBBR & Wb, Ws, Wd & Wd = WREG - WREG - f - (C) & 1 & 1 & C, DC, N, OV, Z \\ \hline SUBBR & Wb, Ws, Wd & Wd = WREG - WREG - f - (C) & 1 & 1 $		SETM	Ws	Ws = FFFFh	1	1	None
SL f, WREG WREG = Left Shift f 1 1 1 0, N, O, Z SL Ws, Wd Wd = Left Shift Ws 1 1 0, N, O, Z SL Wb, Wns, Wnd Wnd = Left Shift Wb by Wns 1 1 N, Z SL Wb, #lit5, Wnd Wnd = Left Shift Wb by Uhs 1 1 N, Z SUB f MREG f=nWREG 1 1 N, Z SUB f, WREG Wnd = Left Shift Wb by Uhs 1 1 C, DC, N, OV, Z SUB f, WREG Wh Hits 1 1 C, DC, N, OV, Z SUB f, WREG WN HEG = f-WREG 1 1 C, DC, N, OV, Z SUB Wb, Ws, Wd Wd = Wb - Ws 1 1 C, DC, N, OV, Z SUB Wb, #lit5, Md Wd = Wb - Ws 1 1 C, DC, N, OV, Z SUBB f. WREG MREG = f-WREG - (C) 1 1 C, DC, N, OV, Z SUBB f. WREG WREG = WREG - (C) 1 1 C, DC, N, OV	SL	SL	£	f = Left Shift f	1	1	C, N, OV, Z
SL Ws, Wd Wd = Left Shift Ws 1 1 1 C, N, OV, Z SL Wb, Wns, Wnd Wnd = Left Shift Wb by Wns 1 1 N, Z SL Wb, #lits, Wnd Wnd = Left Shift Wb by Nits 1 1 N, Z SUB SUB f f=f-WREG 1 1 N, Z SUB f, WREG WREG = f-WREG 1 1 C, DC, N, OV, Z SUB f, WREG WM = Wn - III10 1 1 C, DC, N, OV, Z SUB Wb, Ms, Wd Wd = Wn - Ws 1 1 C, DC, N, OV, Z SUB Wb, Ms, Wd Wd = Wn - Ws 1 1 C, DC, N, OV, Z SUB Wb, Ms, Wd Wd = Wn - Ws 1 1 C, DC, N, OV, Z SUBB f.WREG MREG = f-WREG - (C) 1 1 C, DC, N, OV, Z SUBB f.WREG MREG = MREG - (C) 1 1 C, DC, N, OV, Z SUBB #lit10, Wn Wn = Wn - IIt10 - (C) 1 1 C, DC, N		SL	f,WREG	WREG = Left Shift f	1	1	C, N, OV, Z
SL Wb, Wns, Wnd Wnd = Left Shift Wb by Wns 1 1 N, Z SUB f f=f-WREG 1 1 N, Z SUB f f=f-WREG 1 1 N, Z SUB f, WREG WREG=f-WREG 1 1 C, DC, N, OV, Z SUB flit10, Wn Wn = Wn - H10 1 1 C, DC, N, OV, Z SUB Wb, Ws, Wd Wd = Wb - Ws 1 1 C, DC, N, OV, Z SUB Wb, Ws, Wd Wd = Wb - Ws 1 1 C, DC, N, OV, Z SUB Wb, Ws, Wd Wd = Wb - Ws 1 1 C, DC, N, OV, Z SUB Wb, Ws, Wd Wd = Wb - Ws 1 1 C, DC, N, OV, Z SUB Wb, Ws, Wd Wd = Wb - Ws MEG = I, WREG - ICO 1 1 C, DC, N, OV, Z SUBB f, WREG M = Wn - INTO - ICO 1 1 C, DC, N, OV, Z SUBB f, Ws, Wd Wd = Wo - Ws - ICO 1 1 C, DC, N, OV, Z		SL	Ws,Wd	Wd = Left Shift Ws	1	1	C, N, OV, Z
SL Wb,#lits,Wnd Wnd = Left Shift Wb by lit5 1 1 N,Z SUB f f=f-WREG 1 1 C,DC,N,OV,Z SUB f,WREG WREG = f-WREG 1 1 C,DC,N,OV,Z SUB #1110,Wn Wn = Wn - lit10 1 1 C,DC,N,OV,Z SUB Wb,#s,Wd Wd = Wb - Ws 1 1 C,DC,N,OV,Z SUB Wb,#1it5,Wd Wd = Wb - Ws 1 1 C,DC,N,OV,Z SUB Wb,#1it5,Wd Wd = Wb - Ws 1 1 C,DC,N,OV,Z SUBB f f=f-WREG N 1 1 C,DC,N,OV,Z SUBB f.WREG Wd = Wb - Ws 1 1 C,DC,N,OV,Z SUBB f.WREG WREG = F-WREG - (C) 1 1 C,DC,N,OV,Z SUBB f.WREG WREG = F-WREG - (C) 1 1 C,DC,N,OV,Z SUBB #1110,Wn Wn = Wn = Untlift- (C) 1 1 C,DC,N,OV,Z SUBB Wb,Ws,		SL	Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N, Z
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$		SL	Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N, Z
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	SUB	SUB	f	f = f – WREG	1	1	C, DC, N, OV, Z
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		SUB	f,WREG	WREG = f – WREG	1	1	C, DC, N, OV, Z
SUB Wb, Ws, Wd Wd = Wb - Ws 1 1 C, DC, N, OV, Z SUB Wb, #11t5, Wd Wd = Wb - Ht5 1 1 1 C, DC, N, OV, Z SUBB SUBB f f=f-WREG-(\overline{C}) 1 1 C, DC, N, OV, Z SUBB f, WREG WREG = f-WREG-(\overline{C}) 1 1 C, DC, N, OV, Z SUBB f, WREG WREG = f-WREG-(\overline{C}) 1 1 C, DC, N, OV, Z SUBB #1110, Wn Wn = Wn - Ht10-(\overline{C}) 1 1 C, DC, N, OV, Z SUBB Wb, Ws, Wd Wd = Wb - Ws-(\overline{C}) 1 1 C, DC, N, OV, Z SUBB Wb, Ws, Wd Wd = Wb - Ht5-(\overline{C}) 1 1 C, DC, N, OV, Z SUBR f f=WREG f 1 1 C, DC, N, OV, Z SUBR f, WREG Mvb, Ws, Wd Wd = Ws - Wb 1 1 1 C, DC, N, OV, Z SUBR f, WREG Mvb, Ws, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z		SUB	#lit10,Wn	Wn = Wn – lit10	1	1	C, DC, N, OV, Z
SUB Wb, #lit5, Wd Wd = Wb - lit5 1 1 C, D, N, OV, Z SUBB f f=f-WREG-(C) 1 1 C, D, N, OV, Z SUBB f, WREG WREG = f-WREG-(C) 1 1 C, D, N, OV, Z SUBB f, WREG WREG = f-WREG-(C) 1 1 C, D, N, OV, Z SUBB #lit10, Wn Wn = Wn - lit10-(C) 1 1 C, D, N, OV, Z SUBB #lit10, Wn Wn = Wn - lit10-(C) 1 1 C, D, N, OV, Z SUBB wb, Ws, Wd Wd = Wb - Ws - (C) 1 1 C, D, N, OV, Z SUBB Wb, #lit5, Wd Wd = Wb - lit5 - (C) 1 1 C, D, N, OV, Z SUBR f f WREG = WREG - f 1 1 C, D, N, OV, Z SUBR f, WREG Wd = Ws - Wb Wd = Ws - Wb 1 1 C, D, C, N, OV, Z SUBR f, WREG f WREG = WREG - f 1 1 C, D, C, N, OV, Z SUBR Wb, #lit5, Wd Wd = Ws - Wb <td></td> <td>SUB</td> <td>Wb,Ws,Wd</td> <td>Wd = Wb – Ws</td> <td>1</td> <td>1</td> <td>C, DC, N, OV, Z</td>		SUB	Wb,Ws,Wd	Wd = Wb – Ws	1	1	C, DC, N, OV, Z
SUBB f f=f-WREG-(C) 1 1 C, DC, N, OV, Z SUBB f, WREG WREG = f-WREG - (C) 1 1 C, DC, N, OV, Z SUBB flit10, Wn Wn = Wn - lit10 - (C) 1 1 C, DC, N, OV, Z SUBB #lit10, Wn Wn = Wn - lit10 - (C) 1 1 C, DC, N, OV, Z SUBB Wb, Ws, Wd Wd = Wb - Ws - (C) 1 1 C, DC, N, OV, Z SUBB Wb, #lit5, Wd Wd = Wb - Ws - (C) 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = Wb - lit5 - (C) 1 1 C, DC, N, OV, Z SUBR f WREG MREG - f 1 1 C, DC, N, OV, Z SUBR f, WREG WREG = WREG - f 1 1 1 C, DC, N, OV, Z SUBR Wb, Ws, Wd Wd = Ws - Wb 1 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = lit5 - Wb 1 1 1 C, DC, N, OV, Z SUBBR f, WREG		SUB	Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C. DC. N. OV. Z
SUBS Image (c) Image (c) <thimage (c)<="" th=""> <thimage (c)<="" th=""> <thimage< td=""><td>SUBB</td><td>SUBB</td><td>f</td><td>$f = f - WREG - (\overline{C})$</td><td>1</td><td>1</td><td>C DC N OV Z</td></thimage<></thimage></thimage>	SUBB	SUBB	f	$f = f - WREG - (\overline{C})$	1	1	C DC N OV Z
SUBB F, MRES		SIIBB	f WPFC	WREG = $f - WREG - (\overline{C})$	1	1	C DC N OV Z
SUBB #11L10, WI WI WI WI WI I		GUDD	#14510 Mm	$W_{n} = W_{n} \text{iii} 10 (\overline{C})$	1	1	C, DC, N, OV, Z
SUBB WD, WS, Wd Wd = WD - WS - (C) 1 1 C, DC, N, OY, Z SUBB WD, #1it5, Wd Wd = Wb - lit5 - (C) 1 1 C, DC, N, OY, Z SUBR SUBR f f MREG f 1 1 C, DC, N, OY, Z SUBR SUBR f MREG MREG - f 1 1 C, DC, N, OY, Z SUBR f, WREG WREG = WREG - f 1 1 C, DC, N, OY, Z SUBR f, WD, WS, Wd Wd = WS - WD 1 1 C, DC, N, OY, Z SUBR f, WB, WS, Wd Wd = WS - WD 1 1 C, DC, N, OY, Z SUBR WD, #1it5, Wd Wd = It5 - WD 1 1 C, DC, N, OY, Z SUBBR f, WREG f = WREG - f - (C) 1 1 C, DC, N, OY, Z SUBBR f, WREG WD, WS, Wd Wd = WS - WD - (C) 1 1 C, DC, N, OY, Z SUBR WD, WS, Wd Wd = It5 - WD - (C) 1 1 C, DC, N, OY, Z SWAP		SUBB	#11C10,WH	V(1 - V(1 - H(1) - (C))		1	C, DC, N, OV, Z
SUBB Wb, #lit5, Wd Wd = Wb - Nt5 - (C) 1 1 C, DC, N, OV, Z SUBR SUBR f f WREG = MREG - f 1 1 C, DC, N, OV, Z SUBR f, WREG WREG = WREG - f 1 1 C, DC, N, OV, Z SUBR f, Wb, Ws, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR SUBR f Mb, #lit5, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBBR f, WREG MREG = G, C, C 1 1 C, DC, N, OV, Z SUBBR f, WREG Wb, Ws, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SUBR Wb, Ws, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = It5 - Wb - (C) 1 1 C, DC, N, OV, Z SWAP <td></td> <td>SUBB</td> <td>WD,WS,Wd</td> <td>Wd = VVB - VVS - (C)</td> <td>1</td> <td>1</td> <td>C, DC, N, OV, Z</td>		SUBB	WD,WS,Wd	Wd = VVB - VVS - (C)	1	1	C, DC, N, OV, Z
SUBR f f = WREG - f 1 1 C, DC, N, OV, Z SUBR f, WREG WREG = WREG - f 1 1 C, DC, N, OV, Z SUBR Wb, Ws, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR Wb, #1it5, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR Wb, #1it5, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR SUBBR f G, DC, N, OV, Z 1 1 C, DC, N, OV, Z SUBR Wb, #1it5, Wd Wd = Ws - Mb 1 1 C, DC, N, OV, Z SUBBR f, WREG Wb, Ws, Wd WREG = WREG - f - (C) 1 1 C, DC, N, OV, Z SUBBR f, WREG Wb, Ws, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SUBR Wb, #1it5, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SWAP Wn Wn None 1 1 None TBLRDH TBLRDH Ws, Wd		SUBB	Wb,#lit5,Wd	Wd = Wb - lit5 - (C)	1	1	C, DC, N, OV, Z
SUBR f, WREG WREG = WREG - f 1 1 C, DC, N, OV, Z SUBR Wb, Ws, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = It5 - Wb 1 1 C, DC, N, OV, Z SUBBR f f WREG = WREG - f - (C) 1 1 C, DC, N, OV, Z SUBBR f, WREG WREG = WREG - f - (C) 1 1 C, DC, N, OV, Z SUBBR f, WREG Wb, Ws, Wd WREG = WREG - f - (C) 1 1 C, DC, N, OV, Z SUBBR f, WREG Wb, Ws, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SWAP Wh, #lit5, Wd Wd = It5 - Wb - (C) 1 1 C, DC, N, OV, Z SWAP Wn Wn None None None TBLRDH Wn Read Prog<23	SUBR	SUBR	f	f = WREG – f	1	1	C, DC, N, OV, Z
SUBR Wb, Ws, Wd Wd = Ws - Wb 1 1 C, DC, N, OV, Z SUBR Wb, #1it5, Wd Wd = lit5 - Wb 1 1 C, DC, N, OV, Z SUBBR SUBBR f f G, DC, N, OV, Z 1 1 C, DC, N, OV, Z SUBBR SUBBR f MREG f= WREG - f - (C) 1 1 C, DC, N, OV, Z SUBBR f, WREG WREG = WREG - f - (C) 1 1 C, DC, N, OV, Z SUBBR wb, #s, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SUBR Wb, #lit5, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SWAP Wb, #lit5, Wd Wd = lit5 - Wb - (C) 1 1 C, DC, N, OV, Z SWAP Wn Wn Nne None 1 1 None TBLRDH TBLRDH Ws, Wd Read Prog<23:16> to Wd<7:0> 1 2 None		SUBR	f,WREG	WREG = WREG – f	1	1	C, DC, N, OV, Z
SUBR Wb, #lit5, Wd Wd = lit5 - Wb 1 1 C, DC, N, OV, Z SUBBR SUBBR f f WREG = n(C) 1 1 C, DC, N, OV, Z SUBBR f, WREG f WREG = MREG - f - (C) 1 1 C, DC, N, OV, Z SUBBR f, WREG WREG = WREG - f - (C) 1 1 C, DC, N, OV, Z SUBBR wb, Ws, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SUBBR Wb, #lit5, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SWAP Wb, #lit5, Wd Wd = lit5 - Wb - (C) 1 1 C, DC, N, OV, Z SWAP SWAP.b Wn Wn = Nibble Swap Wn 1 1 None TBLRDH TBLRDH Ws, Wd Read Prog<23:16> to Wd<7:0> 1 2 None		SUBR	Wb,Ws,Wd	Wd = Ws – Wb	1	1	C, DC, N, OV, Z
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		SUBR	Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C, DC, N, OV, Z
SUBBR f, WREG WREG = WREG - f - (C̄) 1 1 C, DC, N, OV, Z SUBBR Wb, Ws, Wd Wd = Ws - Wb - (C̄) 1 1 C, DC, N, OV, Z SUBBR Wb, Ws, Wd Wd = Ws - Wb - (C̄) 1 1 C, DC, N, OV, Z SWAP Wb, #lit5, Wd Wd = lit5 - Wb - (C̄) 1 1 C, DC, N, OV, Z SWAP SWAP.b Wn Wn = Nibble Swap Wn 1 1 None SWAP Wn Wn = Nibble Swap Wn 1 1 None TBLRDH TBLRDH Ws, Wd Read Prog<23:16> to Wd<7:0> 1 2 None	SUBBR	SUBBR	f	$f = WREG - f - (\overline{C})$	1	1	C, DC, N, OV, Z
SUBBR Wb, Ws, Wd Wd = Ws - Wb - (C) 1 1 C, DC, N, OV, Z SUBBR Wb, #lit5, Wd Wd = lit5 - Wb - (C) 1 1 C, DC, N, OV, Z SWAP SWAP.b Wn Wn = Nibble Swap Wn 1 1 None SWAP Wn Wn = Byte Swap Wn 1 1 None TBLRDH TBLRDH Ws, Wd Read Prog<23:16> to Wd<7:0> 1 2 None		SUBBR	f,WREG	WREG = WREG – f – (\overline{C})	1	1	C, DC, N, OV, Z
SUBBR Wb,#lit5,Wd Wd = lit5 - Wb - (C̄) 1 1 C, DC, N, OV, Z SWAP SWAP.b Wn Wn = Nibble Swap Wn 1 1 None SWAP Wn Wn = Nibble Swap Wn 1 1 None SWAP Wn Wn = Byte Swap Wn 1 1 None TBLRDH TBLRDH Ws,Wd Read Prog<23:16> to Wd<7:0> 1 2 None		SUBBR	Wb,Ws,Wd	$Wd = Ws - Wb - (\overline{C})$	1	1	C, DC, N, OV, Z
SWAP SWAP.b Wn Wn = Nibble Swap Wn 1 1 None SWAP Wn Wn = Byte Swap Wn 1 1 None TBLRDH TBLRDH Ws.Wd Read Prog<23:16> to Wd<7:0> 1 2 None		SUBBR	Wb,#lit5,Wd	$Wd = lit5 - Wb - (\overline{C})$	1	1	C, DC, N, OV, Z
SWAP Wn Wn = Byte Swap Wn 1 1 None TBLRDH TBLRDH Ws, Wd Read Prog<23:16> to Wd<7:0> 1 2 None	SWAP	SWAP.b	Wn	Wn = Nibble Swap Wn	1	1	None
TBLRDH TBLRDH Ws, Wd Read Prog<23:16> to Wd<7:0> 1 2 None		SWAP	Wn	Wn = Byte Swap Wn	1	1	None
	TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None

Param No.	Symbol	Characteristic	Min	Тур	Мах	Units	Comments
300	TRESP	Response Time ^{*(1)}	_	150	400	ns	
301	Тмс2оv	Comparator Mode Change to Output Valid [*]	—	—	10	μS	

TABLE 29-26: COMPARATOR TIMINGS

*

Parameters are characterized but not tested.

Note 1: Response time is measured with one comparator input at (VDD – 1.5)/2, while the other input transitions from Vss to VDD.

TABLE 29-27: COMPARATOR VOLTAGE REFERENCE SETTLING TIME SPECIFICATIONS

Param No.	Symbol	Characteristic	Min	Тур	Max	Units	Comments
VR310	TSET	Settling Time ⁽¹⁾			10	μS	

Note 1: Settling time is measured while CVRR = 1 and CVR<3:0> bits transition from '0000' to '1111'.



FIGURE 29-19: SPIX MODULE SLAVE MODE TIMING CHARACTERISTICS (CKE = 0)

TABLE 29-38: SPIX MODULE SLAVE MODE TIMING REQUIREMENTS (CKE = 0)

АС СН	AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended				
Param No.	Symbol	Characteristic	Min	Тур ⁽¹⁾	Мах	Units	Conditions	
SP70	TscL	SCKx Input Low Time	30	_		ns		
SP71	TscH	SCKx Input High Time	30	_	_	ns		
SP72	TscF	SCKx Input Fall Time ⁽²⁾	—	10	25	ns		
SP73	TscR	SCKx Input Rise Time ⁽²⁾	—	10	25	ns		
SP30	TdoF	SDOx Data Output Fall Time ⁽²⁾	—	10	25	ns		
SP31	TdoR	SDOx Data Output Rise Time ⁽²⁾	—	10	25	ns		
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—		30	ns		
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20		—	ns		
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20			ns		
SP50	TssL2scH, TssL2scL	\overline{SSx} to SCKx \uparrow or SCKx Input	120		_	ns		
SP51	TssH2doZ	$\overline{\text{SSx}}$ \uparrow to SDOx Output High-Impedance ⁽³⁾	10	_	50	ns		
SP52	TscH2ssH TscL2ssH	SSx after SCKx Edge	1.5 Tcy + 40	—	_	ns		

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

3: Assumes 50 pF load on all SPIx pins.

20-Lead Plastic Quad Flat, No Lead Package (MQ) - 5x5 mm Body [QFN] With 0.40mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			3.35
Optional Center Pad Length	T2			3.35
Contact Pad Spacing	C1		4.50	
Contact Pad Spacing	C2		4.50	
Contact Pad Width (X20)	X1			0.40
Contact Pad Length (X20)	Y1			0.55
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2139A