

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 <sup>2</sup> C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	24
Program Memory Size	8KB (2.75K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1.5K × 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	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f08ka102-e-ml

Email: info@E-XFL.COM

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

The internal oscillator block also provides a stable reference source for the Fail-Safe Clock Monitor (FSCM). This option constantly monitors the main clock source against a reference signal provided by the internal oscillator and enables the controller to switch to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.

### 1.1.4 EASY MIGRATION

Regardless of the memory size, all the devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve.

The consistent pinout scheme used throughout the entire family also helps in migrating to the next larger device. This is true when moving between devices with the same pin count, or even jumping from 20-pin to 28-pin devices.

The PIC24F family is pin compatible with devices in the dsPIC33 family, and shares some compatibility with the pinout schema for PIC18 and dsPIC30. This extends the ability of applications to grow from the relatively simple, to the powerful and complex.

### 1.2 Other Special Features

- Communications: The PIC24F16KA102 family incorporates a range of serial communication peripherals to handle a range of application requirements. There is an I<sup>2</sup>C<sup>™</sup> module that supports both the Master and Slave modes of operation. It also comprises UARTs with built-in IrDA<sup>®</sup> encoders/decoders and an SPI module.
- Real-Time Clock/Calendar: This module implements a full-featured clock and calendar with alarm functions in hardware, freeing up timer resources and program memory space for use of the core application.
- **10-Bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, and faster sampling speed. The 16-deep result buffer can be used either in Sleep to reduce power, or in Active mode to improve throughput.
- Charge Time Measurement Unit (CTMU) Interface: The PIC24F16KA102 family includes the new CTMU interface module, which can be used for capacitive touch sensing, proximity sensing and also for precision time measurement and pulse generation.

### 1.3 Details on Individual Family Members

Devices in the PIC24F16KA102 family are available in 20-pin and 28-pin packages. The general block diagram for all devices is displayed in Figure 1-1.

The devices are different from each other in two ways:

- 1. Flash program memory (8 Kbytes for PIC24F08KA devices, 16 Kbytes for PIC24F16KA devices).
- 2. Available I/O pins and ports (18 pins on two ports for 20-pin devices and 24 pins on two ports for 28-pin devices).
- 3. Alternate SCLx and SDAx pins are available only in 28-pin devices and not in 20-pin devices.

All other features for devices in this family are identical; these are summarized in Table 1-1.

A list of the pin features available on the PIC24F16KA102 family devices, sorted by function, is provided in Table 1-2.

Note: Table 1-1 provides the pin location of individual peripheral features and not how they are multiplexed on the same pin. This information is provided in the pinout diagrams on pages 4, 5 and 6 of the data sheet. Multiplexed features are sorted by the priority given to a feature, with the highest priority peripheral being listed first.

		Pin I	Number				
Function	20-Pin PDIP/SSOP/ SOIC	20-Pin QFN	28-Pin SPDIP/ SSOP/SOIC	28-Pin QFN	I/O	Input Buffer	Description
AN0	2	19	2	27	I	ANA	A/D Analog Inputs
AN1	3	20	3	28	I	ANA	
AN2	4	1	4	1	I	ANA	
AN3	5	2	5	2	I	ANA	
AN4	7	4	6	3	I	ANA	
AN5	8	5	7	4	I	ANA	
AN10	17	14	25	22	I	ANA	
AN11	16	13	24	21	I	ANA	
AN12	15	12	23	20	I	ANA	
U1BCLK	13	10	18	15	0	_	UART1 IrDA <sup>®</sup> Baud Clock
U2BCLK	9	6	11	8	0	_	UART2 IrDA Baud Clock
C1INA	8	5	7	4	I	ANA	Comparator 1 Input A (Positive Input)
C1INB	7	4	6	3	I	ANA	Comparator 1 Input B (Negative Input Option 1)
C1INC	5	2	5	2	I	ANA	Comparator Input C (Negative Input Option 2)
C1IND	4	1	4	1	I	ANA	Comparator Input D (Negative Input Option 3)
C10UT	17	14	25	22	0	—	Comparator 1 Output
C2INA	5	2	5	2	I	ANA	Comparator 2 Input A (Positive Input)
C2INB	4	1	4	1	I	ANA	Comparator 2 Input B (Negative Input Option 1)
C2INC	8	5	7	4	I	ANA	Comparator 2 Input C (Negative Input Option 2)
C2IND	7	4	6	3	I	ANA	Comparator 2 Input D (Negative Input Option 3)
C2OUT	14	11	20	17	0	—	Comparator 2 Output
CLKI	7	4	9	6	I	ANA	Main Clock Input Connection
CLKO	8	5	10	7	0	—	System Clock Output

TABLE 1-2. FIG24FIORATUZ FAMILT FINOUT DESCRIPTIONS	TABLE 1-2:	PIC24F16KA102 FAMILY PINOUT DESCRIPTIONS
---	------------	--

**Legend:** ST = Schmitt Trigger input buffer, ANA = Analog level input/output,  $I^2C^{TM} = I^2C/SMBus$  input buffer

**Note 1:** Alternative multiplexing when the I2C1SEL Configuration bit is cleared.

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
WREG0	0000								Working	Register 0							
WREG1	0002								Working	Register 1							
WREG2	0004								Working	Register 2							
WREG3	0006								Working	Register 3							
WREG4	0008								Working	Register 4							
WREG5	000A								Working	Register 5							
WREG6	000C								Working	Register 6							
WREG7	000E								Working	Register 7							
WREG8	0010				Working Register 8												
WREG9	0012			Working Register 9													
WREG10	0014			Working Register 10													
WREG11	0016								Working F	Register 11							
WREG12	0018								Working F	Register 12							
WREG13	001A								Working F	Register 13							
WREG14	001C								Working F	Register 14							
WREG15	001E								Working F	Register 15							
SPLIM	0020							Stack	Pointer Lin	nit Value Re	egister						
PCL	002E							Progra	m Counter	Low Byte R	legister						
PCH	0030	_	— — — — — — Program Counter Register High Byte														
TBLPAG	0032	—	—	_	—	_	—	_	—			Table N	lemory Pag	e Address I	Register		
PSVPAG	0034	_	_	_	-	_	_	_	_		F	Program Spa	ace Visibility	Page Add	ress Regist	er	
RCOUNT	0036							REP	EAT Loop (	Counter Reg	gister						
SR	0042		—	—		—	—		DC	IPL2	IPL1	IPL0	RA	N	OV	Z	С
CORCON	0044	—	—	—	—	—	—	_	—	—	—	_		IPL3	PSV	—	_
DISICNT	0052	_	_						Disab	le Interrupts	Counter R	egister					

#### TABLE 4-3: **CPU CORE REGISTERS MAP**

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

All

xxxx

IADLE 4-0. IIIVIER REGISTER IVIAF	<b>FABLE 4-6</b> :	TIMER 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
TMR1	0100								Timer1	Register								0000
PR1	0102								Timer1 Per	iod Registe	r							FFFF
T1CON	0104	TON	—	TSIDL	—		_		—	_	TGATE	TCKPS1	TCKPS0		TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register													0000			
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)												0000				
TMR3	010A								Timer3	Register								0000
PR2	010C								Timer2 Per	iod Registe	r							FFFF
PR3	010E	Timer3 Period Register											FFFF					
T2CON	0110	TON	_	TSIDL	_	_	_	_	_	_	TGATE	TCKPS1	TCKPS0	T32	_	TCS	_	0000
T3CON	0112	TON	_	TSIDL	_	_	_	_	_	_	TGATE	TCKPS1	TCKPS0	_	_	TCS	_	0000

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

### TABLE 4-7: INPUT CAPTURE 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
IC1BUF	0140	Input Capture 1 Register F												FFFF				
IC1CON	0142	—	-	ICSIDL	—					ICTMR	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000

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

### TABLE 4-8: OUTPUT COMPARE 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
OC1RS	0180		Output Compare 1 Secondary Register												FFFF			
OC1R	0182		Output Compare 1 Register FI											FFFF				
OC1CON	0184	_	_	OCSIDL	_	_	—	_	_	_	_	_	OCFLT	OCTSEL	OCM2	OCM1	OCM0	0000

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

	-15.																	
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
ADC1BUF0	0300								A/D Data	a Buffer 0								xxxx
ADC1BUF1	0302								A/D Data	a Buffer 1								xxxx
ADC1BUF2	0304								A/D Data	a Buffer 2								xxxx
ADC1BUF3	0306								A/D Data	a Buffer 3								xxxx
ADC1BUF4	0308								A/D Data	a Buffer 4								xxxx
ADC1BUF5	030A								A/D Data	a Buffer 5								xxxx
ADC1BUF6	030C								A/D Data	a Buffer 6								xxxx
ADC1BUF7	030E								A/D Data	a Buffer 7								xxxx
ADC1BUF8	0310								A/D Data	a Buffer 8								xxxx
ADC1BUF9	0312								A/D Data	a Buffer 9								xxxx
ADC1BUFA	0314								A/D Data	Buffer 10								xxxx
ADC1BUFB	0316								A/D Data	Buffer 11								xxxx
ADC1BUFC	0318								A/D Data	Buffer 12								xxxx
ADC1BUFD	031A								A/D Data	Buffer 13								xxxx
ADC1BUFE	031C								A/D Data	Buffer 14								xxxx
ADC1BUFF	031E								A/D Data	Buffer 15								xxxx
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG2	VCFG1	VCFG0	OFFCAL	—	CSCNA		—	BUFS	_	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0324	ADRC	_	_	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	_	_	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0328	CH0NB	—	—	—	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1PCFG	032C	_	_	_	PCFG12	PCFG11	PCFG10	_	_	—	_	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	_	_	_	CSSL12	CSSL11	CSSL10	_	_	_	_	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000

#### TABLE 4-15: A/D REGISTER MAP

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

### TABLE 4-16: CTMU 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
CTMUCON	033C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	033E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	_	_	_	_	_	_	_	_	0000

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

### 7.0 RESETS

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on Resets, refer to the *"PIC24F Family Reference Manual"*, Section 40. "Reset with Programmable Brown-out Reset" (DS39728).

The Reset module combines all Reset sources and controls the device Master Reset Signal, SYSRST. The following is a list of device Reset sources:

- POR: Power-on Reset
- MCLR: Pin Reset
- SWR: RESET Instruction
- WDTR: Watchdog Timer Reset
- · BOR: Brown-out Reset
- Low-Power BOR/Deep Sleep BOR
- TRAPR: Trap Conflict Reset
- IOPUWR: Illegal Opcode Reset
- UWR: Uninitialized W Register Reset

Figure 7-1 displays a simplified block diagram of the Reset module.

Any active source of Reset will make the SYSRST signal active. Many registers associated with the CPU and peripherals are forced to a known Reset state. Most registers are unaffected by a Reset; their status is unknown on a Power-on Reset (POR) and unchanged by all other Resets.

Note: Refer to the specific peripheral or CPU section of this manual for register Reset states.

All types of device Reset will set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 7-1). A POR will clear all bits except for the BOR and POR bits (RCON<1:0>) which are set. The user may set or clear any bit at any time during code execution. The RCON bits only serve as status bits. Setting a particular Reset status bit in software will not cause a device Reset to occur.

The RCON register also has other bits associated with the Watchdog Timer (WDT) and device power-saving states. The function of these bits is discussed in other sections of this manual.

**Note:** The status bits in the RCON register should be cleared after they are read so that the next RCON register value, after a device Reset, will be meaningful.

### FIGURE 7-1: RESET SYSTEM BLOCK DIAGRAM



### REGISTER 8-21: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0
bit 15							bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—
bit 7							bit 0
Legend:	. L . L		L :4		<b></b>	l = = (0)	
R = Readable		vv = vvritable	DIT	U = Unimplen	nented bit, read		
-n = value at	PUR	T = Bit is set		U = Bit is clea	ared	x = Bit is unkr	lown
bit 15	Unimplemen	ted: Read as 'i	ר <b>י</b>				
bit 14-12	CRCIP<2.0>	CRC Generat	or Error Interru	int Priority hits			
Sit III IZ	111 = Interru	pt is Priority 7 (	highest priority	/ interrupt)			
	•						
	•						
	•	nt in Driarity 1					
	001 – Interru	pt is Fliolity 1 pt source is dis	abled				
bit 11	Unimplemen	ted: Read as '	)'				
bit 10-8	U2ERIP<2:0>	-: UART2 Error	Interrupt Prior	ritv 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	Unimplemen	ted: Read as '	D'				
bit 6-4	U1ERIP<2:0>	-: UART1 Error	Interrupt Prior	rity bits			
	111 = Interru	pt is Priority 7 (	highest priority	/ interrupt)			
	•						
	•						
	001 = Interru	pt is Priority 1					
	000 = Interru	pt source is dis	abled				
bit 3-0	Unimplemen	ted: Read as '	o'				

### 8.4 Interrupt Setup Procedures

#### 8.4.1 INITIALIZATION

#### To configure an interrupt source:

- 1. Set the NSTDIS Control bit (INTCON1<15>) if nested interrupts are not desired.
- 2. Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits, for all enabled interrupt sources, may be programmed to the same non-zero value.

Note:	At a device	e Rese	t, the	IPC	Cx regi	isters are
	initialized,	such	that	all	user	interrupt
	sources are	e assig	gned t	o P	riority	Level 4.

- 3. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx register.
- 4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

#### 8.4.2 INTERRUPT SERVICE ROUTINE

The method that is used to declare an ISR and initialize the IVT with the correct vector address depends on the programming language (i.e., C or assembler) and the language development toolsuite that is used to develop the application. In general, the user must clear the interrupt flag in the appropriate IFSx register for the source of the interrupt that the ISR handles. Otherwise, the ISR will be re-entered immediately after exiting the routine. If the ISR is coded in assembly language, it must be terminated using a RETFIE instruction to unstack the saved PC value, SRL value and old CPU priority level.

### 8.4.3 TRAP SERVICE ROUTINE (TSR)

A Trap Service Routine (TSR) is coded like an ISR, except that the appropriate trap status flag in the INTCON1 register must be cleared to avoid re-entry into the TSR.

#### 8.4.4 INTERRUPT DISABLE

All user interrupts can be disabled using the following procedure:

- 1. Push the current SR value onto the software stack using the PUSH instruction.
- 2. Force the CPU to Priority Level 7 by inclusive ORing the value OEh with SRL.

To enable user interrupts, the POP instruction may be used to restore the previous SR value.

Only user interrupts with a priority level of 7 or less can be disabled. Trap sources (Levels 8-15) cannot be disabled.

The DISI instruction provides a convenient way to disable interrupts of Priority Levels 1-6 for a fixed period. Level 7 interrupt sources are not disabled by the DISI instruction.

NOTES:





Control princontrols our channel.
 Each output compare channel can use one of two selectable time bases. Refer to the device data sheet for the time bases associated with the module.

### 17.0 INTER-INTEGRATED CIRCUIT (I<sup>2</sup>C<sup>™</sup>)

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the Inter-Integrated Circuit, refer to the *"PIC24F Family Reference Manual"*, Section 24. "Inter-Integrated Circuit™ (I<sup>2</sup>C™)" (DS39702).

The Inter-Integrated Circuit (I<sup>2</sup>C) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial data EEPROMs, display drivers, A/D Converters, etc.

The  $I^2C$  module supports these features:

- Independent master and slave logic
- 7-bit and 10-bit device addresses
- General call address, as defined in the I<sup>2</sup>C protocol
- Automatic clock stretching to provide delays for the processor to respond to a slave data request
- Both 100 kHz and 400 kHz bus specifications
- Configurable address masking
- Multi-Master modes to prevent loss of messages in arbitration
- Bus Repeater mode, allowing the acceptance of all messages as a slave regardless of the address
- Automatic SCL

Figure 17-1 illustrates a block diagram of the module.

### 17.1 Pin Remapping Options

The I<sup>2</sup>C module is tied to a fixed pin. To allow flexibility with peripheral multiplexing, the I2C1 module in 28-pin devices can be reassigned to the alternate pins, designated as SCL1 and SDA1 during device configuration.

Pin assignment is controlled by the I2C1SEL Configuration bit. Programming this bit (= 0) multiplexes the module to the SCL1 and SDA1 pins.

## 17.2 Communicating as a Master in a Single Master Environment

The details of sending a message in Master mode depends on the communications protocol for the device being communicated with. Typically, the sequence of events is as follows:

- 1. Assert a Start condition on SDA1 and SCL1.
- 2. Send the I<sup>2</sup>C device address byte to the slave with a write indication.
- 3. Wait for and verify an Acknowledge from the slave.
- 4. Send the first data byte (sometimes known as the command) to the slave.
- 5. Wait for and verify an Acknowledge from the slave.
- 6. Send the serial memory address low byte to the slave.
- 7. Repeat Steps 4 and 5 until all data bytes are sent.
- 8. Assert a Repeated Start condition on SDA1 and SCL1.
- 9. Send the device address byte to the slave with a read indication.
- 10. Wait for and verify an Acknowledge from the slave.
- 11. Enable master reception to receive serial memory data.
- 12. Generate an ACK or NACK condition at the end of a received byte of data.
- 13. Generate a Stop condition on SDA1 and SCL1.

### REGISTER 17-2: I2C1STAT: I2C1 STATUS REGISTER

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT			_	BCL	GCSTAT	ADD10
bit 15							bit 8
R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/A	Р	S	R/W	RBF	TBF
bit 7							bit 0
Legend:		C = Clearab	le bit	HS = Hardwa	re Settable bit	HSC = Hardware S	ettable/Clearable bit
R = Readat	ole bit	W = Writable	e bit	U = Unimplen	nented bit, read	as '0'	
-n = Value a	at POR	'1' = Bit is se	et	'0' = Bit is clea	ared	x = Bit is unknown	
bit 15	ACKSTAT:	Acknowledg	ge Status bit				
	1 = NACK	was detected	d last				
	U = ACK w Hardware i	s set or clea	r at of Acknov	vledae.			
bit 14	TRSTAT: T	ransmit Stat	us bit				
	(When ope	erating as I <sup>2</sup> C	;™ master; ap	plicable to ma	aster transmit o	peration.)	
	1 = Master	transmit is i	n progress (8	bits + ACK)			
	0 = Master Hardware i	transmit is r	not in progress	8 er transmissio	n. hardware is	clear at end of slave	e Acknowledge
bit 13-11	Unimplem	ented: Rear	l as '∩'		, naraware ie		s / toki towiedge.
bit 10	BCI · Mast	er Rus Collis	ion Detect bit				
	1 = A bus o	collision has	been detecte	d during a ma:	ster operation		
	0 = No coll	ision		0	·		
	Hardware i	s set at dete	ction of bus c	ollision.			
bit 9	GCSTAT: (	General Call	Status bit				
	1 = General 0 = General	al call addres	s was receive	eived			
	Hardware i	s set when a	ddress match	nes general ca	all address; har	dware is clear at Sto	p detection.
bit 8	ADD10: 10	)-Bit Address	Status bit				
	1 = 10-bit a	address was	matched				
	0 = 10-bit a	address was	not matched	of motobod 1	) hit addraaa; h	ardwara ia alaar at	Stan datastian
hit 7		s set at mate	Detect bit	or matched it	o-bit address, fi		Stop detection.
	1 = An atte	mot to write	to the I2C1TF	RN register fai	led because the	e l <sup>2</sup> C module is bus	v
	0 = No coll	ision					y
	Hardware i	s set at occu	irrence of writ	e to I2C1TRN	while busy (cle	eared by software).	
bit 6	I2COV: Re	ceive Overflo	ow Flag bit				
	1 = A byte	was received	d while the I2	C1RCV registe	er is still holding	g the previous byte	
	Hardware i	s set at atter	npt to transfe	r to I2C1RCV	(cleared by sof	tware).	
bit 5	D/A: Data/	Address bit (	when operatii	ng as I <sup>2</sup> C slav	e)	,	
	1 = Indicate	es that the la	ist byte receiv	ed was data			
	0 = Indicates that the last byte received was the device address						
hit 4		s clear at devi	ce address ma	atch; nardware	is set by a write	to 12C1 I RN or by re	ception of slave byte.
DIT 4	P: Stop Dit 1 = Indicate	as that a Sta	n hit has hoor	h detected least	+		
	1 = finiteate 0 = Stop bi	t was not de	tected last	TUELECIEU IAS	ι		
	Hardware i	s set or clea	r when Start,	Repeated Star	rt or Stop is det	ected.	

### REGISTER 19-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER<sup>(1)</sup> (CONTINUED)

bit 7-0	CAL<7:0>: RTC Drift Calibration bits						
	01111111 = Maximum positive adjustment; adds 508 RTC clock pulses every one minute						
	•						
	01111111 = Minimum positive adjustment; adds 4 RTC clock pulses every one minute 00000000 = No adjustment						
	11111111 = Minimum negative adjustment; subtracts 4 RTC clock pulses every one minute						
	10000000 = Maximum negative adjustment; subtracts 512 RTC clock pulses every one minute						

- **Note 1:** The RCFGCAL register is only affected by a POR.
  - 2: A write to the RTCEN bit is only allowed when RTCWREN = 1.
  - 3: This bit is read-only; it is cleared to '0' on a write to the lower half of the MINSEC register.

### REGISTER 19-2: 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	RTSECSEL1 <sup>(1)</sup>	RTSECSEL0 <sup>(1)</sup>	_
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit			U = Unimplemented bit, read as '0'				

bit 15-5	Unimplemented: Rea	ad as '0'
----------	--------------------	-----------

#### bit 4-3 Described in Section 15.0 "Output Compare" and Section 17.0 "Inter-Integrated Circuit (I<sup>2</sup>C<sup>™</sup>)".

bit 2-1 RTSECSEL<1:0>: RTCC Seconds Clock Output Select bits<sup>(1)</sup>

'1' = Bit is set

- 11 = Reserved; do not use
  - 10 = RTCC source clock is selected for the RTCC pin (can be LPRC or SOSC, depending on the RTCOSC (FDS<5>) bit setting)

'0' = Bit is cleared

- 01 = RTCC seconds clock is selected for the RTCC pin
- 00 = RTCC alarm pulse is selected for the RTCC pin
- bit 0 Unimplemented: Read as '0'



-n = Value at POR

x = Bit is unknown



### FIGURE 22-1: 10-BIT HIGH-SPEED A/D CONVERTER BLOCK DIAGRAM

REGISTER 22-1: A	D1CON1: A/D CO	NTROL REGISTER 1
------------------	----------------	------------------

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON <sup>(1)</sup>	—	ADSIDL	—	—	_	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0, HSC	R/W-0, HSC
SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE
bit 7							bit 0

Legend:	HSC = Hardware Settal	ole/Clearable bit	
R = Readable	bit W = Writable bit	U = Unimplemented bit	, read as '0'
-n = Value at I	POR '1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
	(1)		
bit 15	ADON: A/D Operating Mode bit <sup>(1)</sup>		
	<ul><li>1 = A/D Converter module is operating</li><li>0 = A/D Converter is off</li></ul>	9	
bit 14	Unimplemented: Read as '0'		
bit 13	ADSIDL: Stop in Idle Mode bit		
	<ul><li>1 = Discontinue module operation who</li><li>0 = Continue module operation in Idle</li></ul>	en device enters Idle mode mode	
bit 12-10	Unimplemented: Read as '0'		
bit 9-8	FORM<1:0>: Data Output Format bits		
	11 = Signed fractional (sddd dddd d	ld00 0000)	
	10 = Fractional (dddd dddd dd00 0 01 = Signed integer (sass sasd ddo	000) 3d dddd)	
	$00 = \text{Integer} (0000 \ 00\text{dd} \ \text{ddd} \ \text{ddd}$	d)	
bit 7-5	SSRC<2:0>: Conversion Trigger Sour	ce Select bits	
	111 = Internal counter ends sampling	and starts conversion (auto-c	onvert)
	110 = CTMU event ends sampling and	d starts conversion	
	100 = Reserved		
	011 = Reserved		
	010 = Timer3 compare ends sampling	and starts conversion	reion
	000 = Clearing SAMP bit ends sampli	ng and starts conversion	301
bit 4-3	Unimplemented: Read as '0'	0	
bit 2	ASAM: A/D Sample Auto-Start bit		
	<ul><li>1 = Sampling begins immediately afte</li><li>0 = Sampling begins when SAMP bit is</li></ul>	r last conversion completes; \$ is set	SAMP bit is auto-set
bit 1	SAMP: A/D Sample Enable bit		
	1 = A/D sample/hold amplifier is samp 0 = A/D sample/hold amplifier is holdin	ling input Ig	
bit 0	DONE: A/D Conversion Status bit		
	1 = A/D conversion is done		
	0 = A/D conversion is not done		

**Note 1:** Values of ADC1BUFn registers will not retain their values once the ADON bit is cleared. Read out the conversion values from the buffer before disabling the module.

### TABLE 28-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
<n:m></n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
bit4	4-bit bit selection field (used in word addressed instructions) $\in \{015\}$
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0000h1FFFh}
lit1	1-bit unsigned literal $\in \{0,1\}$
lit4	4-bit unsigned literal ∈ {015}
lit5	5-bit unsigned literal $\in \{031\}$
lit8	8-bit unsigned literal $\in \{0255\}$
lit10	10-bit unsigned literal $\in$ {0255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal $\in \{016384\}$
lit16	16-bit unsigned literal $\in \{065535\}$
lit23	23-bit unsigned literal $\in$ {08388608}; LSB must be '0'
None	Field does not require an entry, may be blank
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512511}
Slit16	16-bit signed literal ∈ {-3276832767}
Slit6	6-bit signed literal ∈ {-1616}
Wb	Base W register ∈ {W0W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd], [++Wd], [Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd], [++Wnd], [Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor working register pair (direct addressing)
Wn	One of 16 working registers ∈ {W0W15}
Wnd	One of 16 destination working registers ∈ {W0W15}
Wns	One of 16 source working registers ∈ {W0W15}
WREG	W0 (working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws], [++Ws], [Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns], [++Wns], [Wns], [Wns+Wb] }

Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
GOTO	GOTO	Expr	Go to Address	2	2	None
	GOTO	Wn	Go to Indirect	1	2	None
INC	INC	f	f = f + 1	1	1	C, DC, N, OV, Z
	INC	f,WREG	WREG = f + 1	1	1	C, DC, N, OV, Z
	INC	Ws,Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
INC2	INC2	f	f = f + 2	1	1	C, DC, N, OV, Z
	INC2	f,WREG	WREG = f + 2	1	1	C, DC, N, OV, Z
	INC2	Ws,Wd	Wd = Ws + 2	1	1	C, DC, N, OV, Z
IOR	IOR	f	f = f .IOR. WREG	1	1	N, Z
	IOR	f,WREG	WREG = f .IOR. WREG	1	1	N, Z
	IOR	#lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N, Z
	IOR	Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N, Z
	IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N, Z
LNK	LNK	#lit14	Link Frame Pointer	1	1	None
LSR	LSR	f	f = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	f,WREG	WREG = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C, N, OV, Z
	LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N, Z
	LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N, Z
MOV	MOV	f,Wn	Move f to Wn	1	1	None
	MOV	[Wns+Slit10],Wnd	Move [Wns+Slit10] to Wnd	1	1	None
	MOV	f	Move f to f	1	1	N, Z
	MOV	f,WREG	Move f to WREG	1	1	N, Z
	MOV	#lit16,Wn	Move 16-bit Literal to Wn	1	1	None
	MOV.b	#lit8,Wn	Move 8-bit Literal to Wn	1	1	None
	MOV	Wn,f	Move Wn to f	1	1	None
	MOV	Wns,[Wns+Slit10]	Move Wns to [Wns+Slit10]	1	1	None
	MOV	Wso,Wdo	Move Ws to Wd	1	1	None
	MOV	WREG, f	Move WREG to f	1	1	N, Z
	MOV.D	Wns,Wd	Move Double from W(ns):W(ns+1) to Wd	1	2	None
	MOV.D	Ws,Wnd	Move Double from Ws to W(nd+1):W(nd)	1	2	None
MUL	MUL.SS	Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	None
	MUL.SU	Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	None
	MUL.US	Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	None
	MUL.UU	Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	None
	MUL.SU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	None
	MUL.UU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	None
	MUL	f	W3:W2 = f * WREG	1	1	None
NEG	NEG	f	$f = \overline{f} + 1$	1	1	C, DC, N, OV, Z
	NEG	f,WREG	WREG = $\overline{f}$ + 1	1	1	C, DC, N, OV, Z
	NEG	Ws,Wd	$Wd = \overline{Ws} + 1$	1	1	C. DC. N. OV. Z
NOP	NOP		No Operation	1	1	None
	NOPR		No Operation	1	1	None
POP	POP	f	Pop f from Top-of-Stack (TOS)	1	1	None
	POP	Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
	POP.D	Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd+1)	1	2	None
	POP.S		Pop Shadow Registers	1	1	All
PUSH	PUSH	f	Push f to Top-of-Stack (TOS)	1	1	None
	PUSH	Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
	PUSH.D	Wns	Push W(ns):W(ns+1) to Top-of-Stack (TOS)	1	2	None
	PUSH.S		Push Shadow Registers	1	1	None

### TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)

### TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)

yms.xvyms.x	Assembly Mnemonic		Assembly Syntax	Description		# of Cycles	Status Flags Affected
NAME NUMBERNompNetwork 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.REM2.C.S.N.BWREG = Robate Left Invogin Carry MS11N.Z.REM2.R.GF. Robate Left Invogin Carry MS11N.Z.N.Z.REM6.R.GF. Fotate Englith Invogin Carry MS11N.Z.N.Z.REM6.N.BWREG = Robate Right Invogin Carry MS11N.R.N.Z.RENC6.N.BWREG = Robate Right Invogin Carry MS11N.R.N.Z.RENC6.N.BWREG = Robate Right Invogin Carry MS11N.R.N.R.RENC6.N.BWREG = Robate Right Invogin Carry MS11N.R.N.R.<	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           SE         f, MREG         WREG = FOREN         1         N         N, Z           SE         f, MREG         MREG = FOREN         1         N         N           SE         f, MREG         MREG = FOREN         1         N         N		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         S, NrB         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         f , NREG         WREG = Left Shift         1         1         1         None           SL         f , NREG         M M = UeleftShift Wob y MS         1         1 </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>		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         C, DC, N, OX, Z           SUB         f, WREG         Wref = f-WREG         1         1         C, DC, N, OX,		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         FFFFh         1         1         N, Z           SUB         f         MRRG         Wd = Left Shift Wb by Mns         1         1         N, Z           SUB         f.gus b, Mna, Mnd         Wnd = Left Shift Wb by Mis         1         1	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           SL         Ws, Mad         Wd = Wb - MSG         1         1         1         C, DC, N,OV,Z		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, MREG         WREG = f-WREG         11         1         N, Z           SUB         f, MREG         WReG = f-WREG         11         1         C, DC, N, OV, Z           SUB         f, MREG         WREG = F-WREG         1         1         C, DC, N, OV, Z           SUB         f, MREG         WREG = F-WREG         1         1         C, DC, N, OV, Z           SUB         f,		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         1         1         C, DC, N, OV, Z           SUBB         # Jit10, Wn         Wd = Wb - Ws         1         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 - 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 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=f-WREG         1         1         N, Z           SUB         f, WREG         WREG = f-WREG         1         1         C, DC, N, OV, Z           SUB         f, WREG         WN H = Wn-UH10         1         1         C, DC, N, OV, Z           SUB         Wb, #lit5, Md         Wd = Wb - Ws         1         1         C, DC, N, OV, Z           SUB         Wb, #lit5, Md         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, O	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,		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         #lit10, 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><math>f = f - WREG - (\overline{C})</math></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 = OF - (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> <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         Wn         Wn         Wn = Nibble Swap Wn         1         1         None           SWAP         Wn         Wn         Wn = Byte Swap Wn         1         1         None           TBLRDH         TBLRD	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         Nn = Nibble Swap Wn         1         1         None           SWAP         Wn         Wn         Mn = Byte Swap Wn         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

NOTES:

#### TABLE 29-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)

DC CHARACTERISTICS			$\begin{array}{llllllllllllllllllllllllllllllllllll$			
Parameter No.	Typical <sup>(1)</sup>	Max	Units			Conditions
Power-Down C	Current (IPD): I	PMD Bits are	Set, PMSLP	Bit is '0' <sup>(2)</sup>		
DC60		0.200		-40°C		
DC60a		0.200		+25°C		
DC60b	0.025	0.870	μA	+60°C	1.8V	
DC60c		1.350		+85°C		
DC60d		10.00		+125°C		Base Power-Down Current
DC60e		0.540		-40°C		(Sleep) <sup>(3)</sup>
DC60f		0.540		+25°C		
DC60g	0.105	1.680	μΑ	+60°C	3.3V	
DC60h		2.450		+85°C		
DC60i		11.00		+125°C		
DC70		0.150		-40°C		
DC70a		0.150		+25°C		
DC70b	0.020	0.430	μA	+60°C	1.8V	
DC70c		0.630		+85°C		
DC70d		3.00		+125°C		Base Deep Sleep Current
DC70e		0.300		-40°C		Base Deep Sleep Guilent
DC70f		0.300		+25°C		
DC70g	0.035	0.700	μA	+60°C	3.3V	
DC70h		0.980		+85°C		
DC70i		5.00		+125°C		
DC61		0.65		-40°C		
DC61a		0.65		+25°C		
DC61b	0.55	0.65	μA	+60°C	1.8V	
DC61c		0.65		+85°C		
DC61d		1.20		+125°C		Watchdog Timer Current (WDT)(3,4)
DC61e		0.95		-40°C		
DC61f		0.95		+25°C		
DC61g	0.87	0.95	μA	+60°C	3.3V	
DC61h		0.95		+85°C		
DC61i		1.50		+125°C		

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

2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as outputs and set low. WDT, etc., are all switched off.

3: The ∆ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

**4:** Current applies to Sleep only.

5: Current applies to Sleep and Deep Sleep.

6: Current applies to Deep Sleep only.