

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·XFI

Product Status	Obsolete
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	1.5К х 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-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f08ka102-e-so

Email: info@E-XFL.COM

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

NOTES:

#### 3.3.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

- 1. 32-bit signed/16-bit signed divide
- 2. 32-bit unsigned/16-bit unsigned divide
- 3. 16-bit signed/16-bit signed divide
- 4. 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. Sixteen-bit signed and unsigned DIV instructions can specify any W register for both the 16-bit divisor (Wn), and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

### 3.3.3 MULTI-BIT SHIFT SUPPORT

The PIC24F ALU supports both single bit and single-cycle, multi-bit arithmetic and logic shifts. Multi-bit shifts are implemented using a shifter block, capable of performing up to a 15-bit arithmetic right shift, or up to a 15-bit left shift, in a single cycle. All multi-bit shift instructions only support Register Direct Addressing for both the operand source and result destination.

A full summary of instructions that use the shift operation is provided below in Table 3-2.

#### TABLE 3-2: INSTRUCTIONS THAT USE THE SINGLE AND MULTI-BIT SHIFT OPERATION

Instruction	Description
ASR	Arithmetic shift right source register by one or more bits.
SL	Shift left source register by one or more bits.
LSR	Logical shift right source register by one or more bits.

### 5.0 FLASH PROGRAM MEMORY

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 Flash Programming, refer to the "PIC24F Family Reference Manual", Section 4. "Program Memory" (DS39715).

The PIC24FJ64GA family of devices contains internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable when operating with VDD over 1.8V.

Flash memory can be programmed in three ways:

- In-Circuit Serial Programming<sup>™</sup> (ICSP<sup>™</sup>)
- Run-Time Self-Programming (RTSP)
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)

ICSP allows a PIC24F16KA102 device to be serially programmed while in the end application circuit. This is simply done with two lines for the programming clock and programming data (which are named PGCx and PGDx, respectively), and three other lines for power (VDD), ground (VSS) and Master Clear/Program Mode Entry Voltage (MCLR/VPP). This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or custom firmware to be programmed. Real-Time Self-Programming (RTSP) is accomplished using TBLRD (table read) and TBLWT (table write) instructions. With RTSP, the user may write program memory data in blocks of 32 instructions (96 bytes) at a time, and erase program memory in blocks of 32, 64 and 128 instructions (96,192 and 384 bytes) at a time.

The NVMOP<1:0> (NVMCON<1:0>) bits decide the erase block size.

### 5.1 Table Instructions and Flash Programming

Regardless of the method used, Flash memory programming is done with the table read and write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using the TBLPAG<7:0> bits and the Effective Address (EA) from a W register, specified in the table instruction, as depicted in Figure 5-1.

The TBLRDL and the TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in both Word and Byte modes.

The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can also access program memory in Word or Byte mode.



#### FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS

### 5.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 32 instructions or 96 bytes. RTSP allows the user to erase blocks of 1 row, 2 rows and 4 rows (32, 64 and 128 instructions) at a time and to program one row at a time. It is also possible to program single words.

The 1-row (96 bytes), 2-row (192 bytes) and 4-row (384 bytes) erase blocks, and single row write block (96 bytes) are edge-aligned, from the beginning of program memory.

When data is written to program memory using TBLWT instructions, the data is not written directly to memory. Instead, data written using table writes is stored in holding latches until the programming sequence is executed.

Any number of TBLWT instructions can be executed and a write will be successfully performed. However, 32 TBLWT instructions are required to write the full row of memory.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of TBLWT instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register.

Data can be loaded in any order and the holding registers can be written to multiple times before performing a write operation. Subsequent writes, however, will wipe out any previous writes.

Note:	Writing to a location multiple times without
	erasing it is not recommended.

All of the table write operations are single-word writes (two instruction cycles), because only the buffers are written. A programming cycle is required for programming each row.

#### 5.3 Enhanced In-Circuit Serial Programming

Enhanced ICSP uses an on-board bootloader, known as the program executive, to manage the programming process. Using an SPI data frame format, the program executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

#### 5.4 Control Registers

There are two SFRs used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls the blocks that need to be erased, which memory type is to be programmed and when the programming cycle starts.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 55h and AAh to the NVMKEY register. Refer to **Section 5.5 "Programming Operations"** for further details.

### 5.5 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. During a programming or erase operation, the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation and the WR bit is automatically cleared when the operation is finished.

#### 6.4.1 ERASE DATA EEPROM

The data EEPROM can be fully erased, or can be partially erased, at three different sizes: one word, four words or eight words. The bits, NVMOP<1:0> (NVMCON<1:0>), decide the number of words to be erased. To erase partially from the data EEPROM, the following sequence must be followed:

- 1. Configure NVMCON to erase the required number of words: one, four or eight.
- 2. Load TBLPAG and WREG with the EEPROM address to be erased.
- 3. Clear NVMIF status bit and enable NVM interrupt (optional).
- 4. Write the key sequence to NVMKEY.
- 5. Set the WR bit to begin erase cycle.
- 6. Either poll the WR bit or wait for the NVM interrupt (NVMIF set).

#### EXAMPLE 6-2: SINGLE-WORD ERASE

A typical erase sequence is provided in Example 6-2. This example shows how to do a one-word erase. Similarly, a four-word erase and an eight-word erase can be done. This example uses 'C' library procedures to manage the Table Pointer (builtin\_tblpage and builtin\_tbloffset) and the Erase Page Pointer (builtin\_tblwt1). The memory unlock sequence (builtin\_write\_NVM) also sets the WR bit to initiate the operation and returns control when complete.

```
int __attribute__ ((space(eedata))) eeData = 0x1234; // Variable located in EEPROM
    unsigned int offset;

    // Set up NVMCON to erase one word of data EEPROM
    NVMCON = 0x4058;

    // Set up a pointer to the EEPROM location to be erased
    TBLPAG = __builtin_tblpage(&eeData); // Initialize EE Data page pointer
    offset = __builtin_tbloffset(&eeData); // Initialize lower word of address
    __builtin_tblwtl(offset, 0); // Write EEPROM data to write latch
    asm volatile ("disi #5"); // Disable Interrupts For 5 Instructions
    __builtin_write_NVM(); // Issue Unlock Sequence & Start Write Cycle
```

### 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



R/W-0, H	S R/W-0, HS	R/W-0	U-0	U-0	R/C-0, HS	R/W-0, HS	R/W-0
TRAPR	IOPUWR	SBOREN	_	—	DPSLP	СМ	PMSLP
bit 15	•				-	•	bit 8
R/W-0, H	S R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-1, HS	R/W-1, HS
EXTR	SWR	SWDTEN <sup>(2)</sup>	WDTO	SLEEP	IDLE	BOR	POR
bit 7							bit 0
Legend:		C = Clearable	bit	HS = Hardwa	are Settable bit		
R = Reada	ble bit	W = Writable I	oit	U = Unimpler	mented bit, reac	l as '0'	
-n = Value	at POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkn	own
bit 15	TRAPR: Trap	Reset Flag bit					
	1 = A Trap Co	onflict Reset ha	s occurred				
bit 11			s not occurred	J M Alagana Daga	at Elag bit		
DIL 14	1 = An illega	l oncode deter	oninitialized v	al address mes	de or uninitial	ized W reaiste	r used as an
	Address	Pointer caused	a Reset				
	0 = An illegal	l opcode or unir	nitialized W Re	eset has not o	ccurred		
bit 13	SBOREN: So	oftware Enable/I	Disable of BO	R bit			
	1 = BOR is tu	rned on in softw	vare				
bit 12 11	0 = BOR IS lu	ted: Deed es '	vare				
bit 10		n Sleen Mode F	laa hit				
bit TO	1 = Deep Slee	en has occurred	129 51				
	0 = Deep Slee	ep has not occu	irred				
bit 9	CM: Configura	ation Word Mis	match Reset I	Flag bit			
	1 = A Configu	ration Word Mi	smatch has o	ccurred			
	0 = A Configu	ration Word Mi	smatch has n	ot occurred			
bit 8	PMSLP: Prog	gram Memory P	ower During S	Sleep bit			
	0 = Program i	memory bias vo	ltage is powe	red down duri	ng Sleep ng Sleep		
bit 7	EXTR: Extern	al Reset (MCL	R) Pin bit				
	1 = A Master	Clear (pin) Res	et has occurr	ed			
	0 = A Master	Clear (pin) Res	et has not oc	curred			
bit 6	SWR: Softwa	re Reset (Instru	iction) Flag bi	t			
	1 = A  RESET	instruction has	been execute	d			
bit E	0 = A RESET	finstruction has	Dischle of W/F	cuted			
DIL S		nabled					
	0 = WDT is di	isabled					
bit 4	WDTO: Watcl	hdog Timer Tim	e-out Flag bit				
	1 = WDT time	e-out has occuri	ed				
	0 = WDT time	e-out has not oc	curred				
Note 1:	All of the Reset sta	itus bits may be	set or cleared	d in software. S	Setting one of th	ese bits in softv	vare does not
	cause a device Re	set.					
2:	If the FWDTEN Co	onfiguration bit i	s '1' (unprogra	ammed), the V	VDT is always e	enabled, regard	less of the

## REGISTER 7-1: RCON: RESET CONTROL REGISTER<sup>(1)</sup>

SWDTEN bit setting.

### 7.5.2 DETECTING BOR

When BOR is enabled, the BOR bit (RCON<1>) is always reset to '1' on any BOR or POR event. This makes it difficult to determine if a BOR event has occurred just by reading the state of BOR alone. A more reliable method is to simultaneously check the state of both POR and BOR. This assumes that the POR and BOR bits are reset to '0' in the software immediately after any POR event. If the BOR bit is '1' while the POR bit is '0', it can be reliably assumed that a BOR event has occurred.

Note: Even when the device exits from Deep Sleep mode, both the POR and BOR are set.

#### 7.5.3 DISABLING BOR IN SLEEP MODE

When BOREN<1:0> = 10, BOR remains under hardware control and operates as previously described. However, whenever the device enters Sleep mode, BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code, when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

REGISTER	8-19: IPC7:	: INTERRUPT		ONTROL RE	GISTER 7		
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0
bit 15							bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
	INT2IP2	INT2IP1	INT2IP0	<u> </u>		—	
bit 7							bit 0
Legend:							
R = Readab	le bit	W = Writable	bit	U = Unimplem	nented bit, read	l as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown
			_				
bit 15	Unimplemen	ted: Read as '	)'				
bit 14-12	U2TXIP<2:0>	>: UART2 Trans	smitter Interrup	t Priority bits			
	111 = Interru	pt is Priority 7 (	highest priority	interrupt)			
	•						
	•						
	001 = Interru	pt is Priority 1					
	000 = Interru	pt source is dis	abled				
bit 11	Unimplemen	ted: Read as '	כ'				
bit 10-8	U2RXIP<2:0	>: UART2 Rece	eiver Interrupt F	Priority 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 '	כי				
bit 6-4	INT2IP<2:0>	: External Interr	upt 2 Priority b	its			
	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 '	כי				

#### **REGISTER 8-24: INTTREG: INTERRUPT CONTROL AND STATUS REGISTER**

R-0	U-0	R/W-0	U-0	R-0	R-0	R-0	R-0
CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

U-0	R-0						
—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7							bit 0

Legend:									
R = Readable I	oit V	V = Writable bit	U = Unimplemented bit, read	as '0'					
-n = Value at P	OR "	1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown					
bit 15	CPUIRQ: Interr	upt Request from Interrupt	Controller CPU bit						
	1 = An interrup happen wh 0 = No interrup	<ul> <li>An interrupt request has occurred but has not yet been Acknowledged by the CPU (this with happen when the CPU priority is higher than the interrupt priority)</li> <li>No interrupt request is left unacknowledged</li> </ul>							
bit 14	Unimplemente	implemented: Read as '0'							
bit 13	VHOLD: Allows	Vector Number Capture a	nd Changes what Interrupt is	Stored in VECNUM bit					
	1 = VECNUM v	will contain the value of th	e highest priority pending inte	errupt, instead of the current					
	<ul> <li>0 = VECNUM will contain the value of the last Acknowledged interrupt (last interrupt that ha with higher priority than the CPU, even if other interrupts are pending)</li> </ul>								
bit 12	Unimplemente	d: Read as '0'							
bit 11-8	ILR<3:0>: New	CPU Interrupt Priority Leve	el bits						
	1111 = CPU Int	terrupt Priority Level is 15							
	•								
	•								
	0001 = CPU Int	terrupt Priority Level is 1							
	0000 = CPU Int	terrupt Priority Level is 0							
bit 7	Unimplemente	d: Read as '0'							
bit 6-0	VECNUM<6:0>	: Vector Number of Pendin	g Interrupt bits						
	0111111 = Inte	rrupt Vector pending is Nu	mber 135						
	•								
	•								
	• 0000001 = Inte	errunt Vector pending is Nu	mber 9						
	0000000 = Inte	errupt Vector pending is Nu	mber 8						

### 10.0 POWER-SAVING FEATURES

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, refer to the							
	"PIC24F Family Reference Manual",							
	"Section 39. Power-Saving Features							
	with Deep Sleep" (DS39727).							

The PIC24F16KA102 family of devices provides the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. All PIC24F devices manage power consumption in four different ways:

- Clock frequency
- Instruction-based Sleep, Idle and Deep Sleep modes
- Software controlled Doze mode
- Selective peripheral control in software

Combinations of these methods can be used to selectively tailor an application's power consumption, while still maintaining critical application features, such as timing-sensitive communications.

#### 10.1 Clock Frequency and Clock Switching

PIC24F devices allow for a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSC bits. The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 9.0** "Oscillator Configuration".

#### 10.2 Instruction-Based Power-Saving Modes

PIC24F devices have two special power-saving modes that are entered through the execution of a special PWRSAV instruction. Sleep mode stops clock operation and halts all code execution; Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. Deep Sleep mode stops clock operation, code execution and all peripherals except RTCC and DSWDT. It also freezes I/O states and removes power to SRAM and Flash memory. The assembly syntax of the PWRSAV instruction is shown in Example 10-1.

**Note:** SLEEP\_MODE and IDLE\_MODE are constants, defined in the assembler include file, for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

#### 10.2.1 SLEEP MODE

Sleep mode includes these features:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be reduced to a minimum provided that no I/O pin is sourcing current.
- The I/O pin directions and states are frozen.
- The Fail-Safe Clock Monitor does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode if the WDT or RTCC, with LPRC as the clock source, is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals may continue to operate in Sleep mode. This includes items, such as the input change notification on the I/O ports, or peripherals that use an external clock input. Any peripheral that requires the system clock source for its operation will be disabled in Sleep mode.

The device will wake-up from Sleep mode on any of these events:

- On any interrupt source that is individually enabled
- · On any form of device Reset
- · On a WDT time-out

On wake-up from Sleep, the processor will restart with the same clock source that was active when Sleep mode was entered.

#### EXAMPLE 10-1: PWRSAV INSTRUCTION SYNTAX

PWRSAV	#SLEEP_MODE	; Put the device into SLEEP mode
PWRSAV	#IDLE_MODE	; Put the device into IDLE mode
BSET	DSCON, #DSEN	; Enable Deep Sleep
PWRSAV	#SLEEP_MODE	; Put the device into Deep SLEEP mode





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.

REGISTE	R 26-6:	FPOR	RESET CO	NFIGURATIO	ON REGISTE	R		
R/P-1	R/P	P-1	R/P-1	R/P-1	R/P-1	U-0	R/P-1	R/P-1
MCLRE <sup>(2</sup>	BOR	/1 <sup>(3)</sup>	BORV0 <sup>(3)</sup>	I2C1SEL <sup>(1)</sup>	PWRTEN		BOREN1	BOREN0
bit 7			•	•				bit 0
Legend:								
R = Reada	able bit		P = Programr	nable bit	U = Unimplen	nented bit, rea	ad as '0'	
-n = Value	at POR		'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown
bit 7	<b>MCLRE</b> 1 = MC 0 = RAS	E: MCLF	$\overline{R}$ Pin Enable b is enabled; RA pin is enabled;	it <sup>(2)</sup> 5 input pin is d MCLR is disab	isabled bled			
bit 6-5	BORV<	1:0>: B	Brown-out Rese	et Enable bits <sup>(3</sup>	)			
bit 4	11 = Br 10 = Br 01 = Br 00 = Lo	<ul> <li>11 = Brown-out Reset is set to the lowest voltage</li> <li>10 = Brown-out Reset</li> <li>01 = Brown-out Reset is set to the highest voltage</li> <li>00 = Low-Power Brown-out Reset occurs around 2.0V</li> </ul>						
	0 = Alte 1 = Defa	rnate lo ault loca	ocation for SCL ation for SCL1/	1/SDA1 pins SDA1 pins				
bit 3	<b>PWRTE</b> 0 = PW 1 = PW	E <b>N:</b> Pow RT is di RT is ei	ver-up Timer E isabled nabled	nable bit				
bit 2	Unimpl	emente	ed: Read as '0	,				
bit 1-0	BOREN	l<1:0>:	Brown-out Re	set Enable bits				
	11 = Bro 10 = Bro 01 = Bro 00 = Bro	own-ou own-ouf own-ou own-ou	t Reset is enat t Reset is enabl t Reset is cont t Reset is disa	oled in hardwar ed only while d rolled with the bled in hardwa	re; SBOREN bit evice is active a SBOREN bit se re; SBOREN bi	t is disabled nd disabled in etting it is disabled	Sleep; SBOREN	V bit is disabled
Note 1:	Applies on	ly to 28	-pin devices.					
2:	The MCLR user from a	RE fuse acciden	can only be ch ntally locking ou	anged when u ut the device fro	sing the VPP-Battern the low-voltation the low-voltation the low-voltation of the low-voltati	ased ICSP™ age test entry	mode entry. Thi	s prevents a
-								

3: Refer to Section 29.0, Electrical Characteristics for the BOR voltages.

Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
	BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
BTST	BTST	f,#bit4	Bit Test f	1	1	Z
	BTST.C	Ws,#bit4	Bit Test Ws to C	1	1	С
	BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
	BTST.C	Ws,Wb	Bit Test Ws <wb> to C</wb>	1	1	С
	BTST.Z	Ws,Wb	Bit Test Ws <wb> to Z</wb>	1	1	Z
BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
	BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	С
	BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
CALL	CALL	lit23	Call Subroutine	2	2	None
	CALL	Wn	Call Indirect Subroutine	1	2	None
CLR	CLR	f	f = 0x0000	1	1	None
	CLR	WREG	WREG = 0x0000	1	1	None
	CLR	Ws	Ws = 0x0000	1	1	None
CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO, Sleep
COM	COM	f	f = f	1	1	N, Z
	СОМ	f,WREG	WREG = f	1	1	N, Z
	COM	Ws.Wd	$Wd = \overline{Ws}$	1	1	N.Z
CP	CP	f	Compare f with WREG	1	1	C DC N OV Z
01	CP	- Wb.#lit5	Compare Wb with lit5	1	1	C DC N OV Z
	CP	Wb Wg	Compare Wb with Ws (Wb – Ws)	1	1	C, DC, N, OV, Z
CPO	CPO	f	Compare f with 0x0000	1	1	C, DC, N, OV, Z
61.0	CPO	± We	Compare Ws with 0x0000	1	1	C, DC, N, OV, Z
CPB	CPB	f	Compare f with WREG with Borrow	1	1	C, DC, N, OV, Z
61.5	CPB		Compare Wb with lit5 with Borrow	1	1	C, DC, N, OV, Z
	CDB	Wb We	Compare Wb with Ws with Borrow	1	1	C DC N OV Z
	Crb		$(Wb - Ws - \overline{C})$			0, 00, 11, 01, 2
CPSEQ	CPSEQ	Wb,Wn	Compare Wb with Wn, Skip if =	1	1 (2 or 3)	None
CPSGT	CPSGT	Wb,Wn	Compare Wb with Wn, Skip if >	1	1 (2 or 3)	None
CPSLT	CPSLT	Wb,Wn	Compare Wb with Wn, Skip if <	1	1 (2 or 3)	None
CPSNE	CPSNE	Wb,Wn	Compare Wb with Wn, Skip if ≠	1	1 (2 or 3)	None
DAW	DAW	Wn	Wn = Decimal Adjust Wn	1	1	С
DEC	DEC	f	f = f - 1	1	1	C, DC, N, OV, Z
	DEC	f,WREG	WREG = f –1	1	1	C, DC, N, OV, Z
	DEC	Ws,Wd	Wd = Ws - 1	1	1	C, DC, N, OV, Z
DEC2	DEC2	f	f = f - 2	1	1	C, DC, N, OV, Z
	DEC2	f,WREG	WREG = f – 2	1	1	C, DC, N, OV, Z
	DEC2	Ws,Wd	Wd = Ws - 2	1	1	C, DC, N, OV, Z
DISI	DISI	#lit14	Disable Interrupts for k Instruction Cycles	1	1	None
DIV	DIV.SW	Wm,Wn	Signed 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.SD	Wm,Wn	Signed 32/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UW	Wm,Wn	Unsigned 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide	1	18	N, Z, C, OV
EXCH	EXCH	Wns,Wnd	Swap Wns with Wnd	1	1	None
FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	С
FF1R	FF1R	Ws.Wnd	Find First One from Right (LSb) Side	1	1	с

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

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

yms.xvyms.xvyms.xvyms.xvyms.vvRelativeQuito Silego yr dia model1yUWDD. SilegoRCN1.ivorRelative Call12NoneREPSTRiscl.winComputed Call111NoneREPSTRepeat Next Instruction (W1+11 times111NoneRESTRepeat Next Instruction (W1+11 times132NoneRESTRepeat Next Instruction (W1+11 times133NoneRESTRepeat Next Instruction (W1+11 times132NoneRETTReturn from Instruction (W1+11 times)132NoneRETTReturn from Subroutine132None111NoneRETTReturn from Subroutine11111None11NoneRETTReturn from Subroutine1111None11None11NoneRETTReturn from Subroutine1111None11None11None11None11None11None11None11None11None1None11None11None11None11None11None11None11None11<	Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
NAME NAME NAMENop-ReprReprNop-Non-REPLATALLI ACOCONDUCTION12Non-REPLATILLIARepat Nat Instruction (M) + 1 Imos11Non-REPLATINTASolvant Devole Reset13Non-RETTEISTATUSolvant Devole Reset13Non-RETTEISTATUSolvant Devole Reset13Non-RETTEISolvant Devole Reset13Non-Solvant Devole Reset13Non-RETTEISTATUReturn Internut Non-13Non-Solvant Devole ResetNon-Non-RETTEISTATUReturn Internut Non-13Non-Non-Non-Non-RETTEISixuaReturn Internut Non-110Non-Non-Non-Non-RETTEISixuaNon-Internut Non-110Non-<	PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO, Sleep
No.LNu.Compute Call12NoneNEFERREPSERsite14Repeat Next Instruction (Mr) + 1 times11NoneNESSTRESSTSchware Device Reset133NoneNESSTREVITReturn from Inforupi133NoneNETTRETURReturn from Inforupi133NoneNETTRETURReturn from Inforupi Carny f131NoneNETTRETURReturn from Subrounine131NoneNETTNECFactale Left through Carny f111N.ZRETURNEGFactale Left through Carny f111N.ZRECFactale Left Noc Carny f111N.ZRECFactale Explore Repatite Left Noc Carny f11N.ZRECFactale Repatite Repatite Noc Carny f11N.ZRECFactale Repatite Repatite Repatite Noc Carny f11N.ZRECFactale Repatite	RCALL	RCALL	Expr	Relative Call	1	2	None
BREPAR BLILLIABepace Nach Instruction (Un) + 1 times11No20073237VinSoftware Device Resolution1131No20172017Software Device Resolution1131No20172017Return Vin Interrupt1131No20172017Return Vin Interrupt1131No20172017Construction1131No20172018ConstructionSoftwareNoNo20172018ConstructionSoftware1131No20172018ConstructionSoftware1111NoNo20182018ConstructionSoftwareSoftwareNoNoNo20182018ConstructionSoftwareSoftwareSoftwareNoNoNo20192018ConstructionSoftwareSoftwareSoftwareNoNoNo20192018ConstructionSoftwareSoftwareSoftwareNoNoNo20192018ConstructionSoftwareSoftwareSoftwareNoNoNo20192018ConstructionSoftwareSoftwareSoftwareNoNoNo20192018ConstructionSoftwareSoftwareSoftwareNoNoNo20192018ConstructionSoftwareSoftwareSoftwareNoNoNo<		RCALL	Wn	Computed Call	1	2	None
REEXTNnRepeat Next Instruction (VM) + 1 times11NoRETYReturn VM Conscrete1.13.0NoneRETYNTT NReturn VM Conscrete1.13.0NoneRETURNTT NNoReturn VM Conscrete1.13.0NoneRETURNTT NNaReturn VM Conscrete1.13.0NoneRETURPLAC1.11.11.10.1	REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
BASETServerSolvane Device Reset11NoneBETTIRRETLWRELW from Interrupt1.13.02NoneRETUWRETUW WIL Interain WWIL Interain		REPEAT	Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
BETTEReturn form interruptf.19.109.000BETLHRETTEReturn with Useral InVinof.13.02NoneBETLReturn for Subrourinef.13.02NoneRLCf.f.F.Return for Subrourinef.13.02NoneRLCf.f.F.F.Return for Subrourinef.1f.1C. N. Z.RLMCf.f.S.S.S.S.S.S.S.S.S.RLMCf.f.S. <td>RESET</td> <td>RESET</td> <td></td> <td>Software Device Reset</td> <td>1</td> <td>1</td> <td>None</td>	RESET	RESET		Software Device Reset	1	1	None
звтли втлив110 лиReturn with Lien Myn111	RETFIE	RETFIE		Return from Interrupt	1	3 (2)	None
<table-container>вктовявстовяестоваесто</table-container>	RETLW	RETLW	#lit10,Wn	Return with Literal in Wn	1	3 (2)	None
NUCPickPi	RETURN	RETURN		Return from Subroutine	1	3 (2)	None
NRC (NRC)NRC (NRC)NRC (NRC)NRC (NRC)NRC (NRC)NRC (NRC)NRC (NRC)NRC (NRC)NRC (NRC)NRC)NRC (NRC)NRC)NRC (NRC)NRC)NRC)NRCNRC)	RLC	RLC	f	f = Rotate Left through Carry f	1	1	C, N, Z
InterpretationInterp		RLC	f,WREG	WREG = Rotate Left through Carry f	1	1	C, N, Z
RLNC         £         C         f = Rotate Left (No Carry) f         1         1         N, Z           RLNC         (F, MRR0         WREG = Rotate Left (No Carry) f         1         1         N, Z           RRC         f         F. Rotate Left (No Carry) f         1         1         N, Z           RRC         f         F. Rotate Right through Carry f         1         1         C, N, Z           RRC         F, MREG         WREG = Rotate Right through Carry f         1         1         C, N, Z           RRNC         f, MREG         f = Rotate Right (No Carry) f         1         1         N, Z           RRNC         f, MREG         f = Rotate Right (No Carry) f         1         1         N, Z           RRNC         f, MREG         MREG = Rotate Right (No Carry) f         1         N         N.Z           SE         ws. Mad         Wate Rotate Right (No Carry) f         1         N         N.Z           SE         f # Rotate Right (No Carry) f         1         N         N.Z           SE         f ws. Mad         Wate Rotate Right (No Carry) f         1         N.Z           SE         f ws. Mad         Wate Rotate Right (No Carry) f         1         N.Z           S		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           SETM         Wa, Md         Wd = Rotate Right (No Carry) f         11         N, Z           SETM         WR.MG         WREG = FEFFh         11         N, R           SETM         WREG         FFFFh         11         N, Z           SL         r, WERG         WREG = Left Shift f         11         N, Z           SL         W.N, MA         Wd = Left Shift f         11         N, Z           SL         W.N, MA         Wd = 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, NREG         MREG = Rotate Right through Carry f         1         1         C, N, Z           RRC         Ms, Md         Wd = Rotate Right through Carry f         1         1         C, N, Z           RENC         F, NREG         WREG = Rotate Right (No Carry) f         1         1         N, Z           RENC         F, NREG         WREG = Rotate Right (No Carry) f         1         1         N, Z           RENC         F, NREG         WREG = 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) f         1         1         N, Z           SE         Ns, Md         Wd = Rotate Right (No Carry) f         1         1         N, Z           SE         Ns, Md         Wd = Rotate Right Nos         11         1         N, Z           SE         Ns, Md         WREG = For Strift         1         1         N, O           SE         Ns, Nd         WREG = I colt Shift Ws by Ms         1         1         N		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         EaxC         f         F         Rotate Right through Carry Ms         11         1         C, N, Z           RRNC         EaxC         f         F         Rotate Right (No Carry) M         11         1         N, Z           RRNC         Ws, Md         Wd = Rotate Right (No Carry) MS         11         1         N, Z           SS         Ws, Md         Wd = Rotate Right (No Carry) MS         11         1         N, Z           SS         Ws, Md         Wd = Sign-Extended WS         11         1         N, Z           SS         Ws, Md         Wd = Sign-Extended WS         11         11         N, Z           SSTM         Ws         Ma         Ws = FFFFh         11         11         None           SL         f.NRS         WREG = Left Shift f         11         11         N, Z           SL         Ms, Ma         Wd = Left Shift Wbb Wins         11         11         N, Z           SL         Ms, Md         Wd = Left Shift Wbb Wins		RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N, Z
RRC         f, NRSG         WREG = Rotate Right through Carry ff         1         1         C, N, Z           RRC         Ws, vid         Wd = Rotate Right through Carry Vis         1         1         C, N, Z           RRNC         f. Rotate Right (No Carry) f         11         1         N, Z           RRNC         f. NREG         WREG = Rotate Right (No Carry) Vis         11         1         N, Z           SE         Ks. Mad         Wd = Rotate Right (No Carry) Vis         11         1         N, Z           SET         Ks. Mad         Wid = Rotate Right (No Carry) Vis         11         1         N, Z           SET         Ws., Wad         Water Rotate Right (No Carry) Vis         11         1         N, Z           SET         Ws., Wad         Water Rotate Right (No Carry) Vis         11         1         N, Z           SET         Ws., Wad         Water Rotate Right (No Carry) Vis         11         1         N, Z           SET         Ws. Water         Ws. Set FFFFh         11         1         C, N, OV, Z           SL         F.         Ms., Mad         Wd = Left Shift Wb by Wiss         11         1         C, N, OV, Z           SL         Ms., Mad         Wid = Left Shift Wb by Wiss	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, RRC         F         Rotate Right (No Carry) f         1         1         N, Z           RENC         F, WEG         Rotate Right (No Carry) f         1         1         N, Z           SE         We, Wd         Wd = Rotate Right (No Carry) Ws         1         1         N, Z           SE         We, Wd         Wd = Rotate Right (No Carry) Ws         1         1         N, Z           SE         We, Wd         Wd = Rotate Right (No Carry) Ws         1         1         None           SETM         WeRG         FFFFh         1         1         None           SETM         WREG         FFFFh         1         1         None/X           SL         f, WREG         MeRG = ferf Shift Ms         1         1         C, N, OV, Z           SL         We, Wd         Wd = Left Shift Ms         1         1         N, OV, Z           SL         We, Md         Wd = Left Shift Ms by Ms         1         1         N, OV, Z           SL         We, Md         Wd = Left Shift Ms by Ms         1         1         N, OV, Z           SL		RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C, N, Z
RRNC         f         Relate 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-Excleded Ws         1         1         C, N, Z           SETM         SETM         f         f         FFFFh         1         1         None           SETM         WREG         WREG         WREG = Left Shift f         1         1         C, N, OV, Z           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, WREG         WREG = Left Shift Wb by Wist         1         1         N, Z           SUB         f, WD, Wns, Wnd         Wnd = Left Shift Wb by Wist         1         1         N, Z           SUB         f, WREG         WREG = Left Shift Wb by Wist         1         1         N, Z           SUB         f, WD, Wist, Wnd         Wnd = Left Shift Wb by Wist         1		RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C, N, Z
RRNC         f, WREG         WREG = Rotate Right (No Carry) M         1         1         N, Z           SE         NA, Wd         Wd = Rotate Right (No Carry) MS         1         1         N, Z           SE         SE M, Wnd         Wnd = Sign-Extended Ws         1         1         N, Z           SETM         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, WREG         WREG = Left Shift M         1         1         N, OV, Z           SL         Ns, Wd         Wd = Left Shift MS         1         1         N, Z           SL         Ns, Wnd         Wd = Left Shift MS         1         1         N, Z           SL         Ns, Md         Wd = Left Shift MS         1         1         N, Z           SL         Ns, Mind         Wnd = Left Shift MS         1         1         N, Z           SUB         SUB         f.         MS         MG         1         1         C, D, C, N, Z           SUB         SUB         f.	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         WREG         WREG         1         1         1         None           SL         f         MREG         WREG         WREG         1         1         1         0, NO, Z           SL         f         WREG         WREG         1         1         1         N, Z           SL         Wb, Wns, Wnd         Wd <left by="" shift="" td="" wb="" wns<="">         1         1         1         N, Z           SUB         f         MREG         F         MREG         1         1         N, Z           SUB         f         MREG         WREG         1         1         1         D, D, O, V, Z           SUB</left>		RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N, Z
SE         SE         Ws.Wnd         Wnd = Sign-Extended Ws         1         1         C. N. Z           SETM         f         FFFFh         11         1         None           SETM         K         f         FFFFh         11         1         None           SETM         WREG         WREG         FFFFh         11         1         None           SL         f         f=Left Shift f         11         1         C. N. OV, Z           SL         f.ws.Wa         WREG = Left Shift MS         11         1         C. N. OV, Z           SL         ws.Wa         WMd = Left Shift Wb by Wns         11         1         N, Z           SUB         MD.Wns.Wnd         Wnd = Left Shift Wb by Wns         1         1         N, Z           SUB         f.ws.Wad         Wnd = Left Shift Wb by Wns         1         1         N, Z           SUB         f.ws.Wad         WREG = f-WREG         11         1         N, Z           SUB         f.ws.Wad         Wd = Wb - Ws         1         1         C, DC, N, OV, Z           SUB         f.ws.Wd         Wd = Wb - Ws         1         1         1         C, DC, N, OV, Z           SUB		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
SETMWREGWREG = FFFFh11NoneSETMWBKSFFFFh11NoneSLSLffff11NoneSLSLffFf11C, N, OV, ZSLKNREGWREGWREGI10C, N, OV, ZSLWb, WRS, WadWdUd110N, ZSLWb, Whit, WadWdI11N, ZSLWb, Whit, WadWndI110C, DC, N, OV, ZSUBffFWREG110C, DC, N, OV, ZSUBf, WREGWREGWREGI110C, DC, N, OV, ZSUBMb, Whit, WadWn = UntSin110C, DC, N, OV, ZSUBWhit, S, WdWdWdWbS110C, DC, N, OV, ZSUBWb, Wit, WdWdWdWdWdM110C, DC, N, OV, ZSUBWb, Wit, WdWdWdWGI110C, DC, N, OV, ZSUBSUBffWREGFWREGI110C, DC, N, OV, ZSUBSUBfMEGWBWGI110C, DC, N, OV, ZSUBBSUBfMEGMEGIIIC, DC, N, OV, ZSU	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, #lit5, Wnd         Wnd = Left Shift Wb by Wns         1         1         N, Z           SUB         f         f=f-WREG         1         1         C, DC, N, OV, Z           SUB         #lit10, Wn         Wn = Wn - INHO         1         1         C, DC, N, OV, Z           SUB         #lit10, Wn         Wn = Wn - INHO         1         1         C, DC, N, OV, Z           SUB         #lit10, Wn         Wd = Wb - Ws         1         1         C, DC, N, OV, Z           SUBB         f         f=f-WREG         1         1         C, DC, N, OV, Z           SUBB         f         f=f-WREG         1         1         C, DC, N, OV, Z           SUBB         f         f=f-WRE		SETM	WREG	WREG = FFFFh	1	1	None
$ SL \qquad SL \qquad f \qquad f = Left Shift f \qquad 1 \qquad 1 \qquad 1 \qquad C, N, OV, Z \\ \hline SL \qquad f, WREG \qquad WREG = Left Shift f \qquad 1 \qquad 1 \qquad C, N, OV, Z \\ \hline SL \qquad Wb, WREG \qquad WREG = Left Shift Ws \qquad 1 \qquad 1 \qquad C, N, OV, Z \\ \hline SL \qquad Wb, Wa, Wd \qquad Wd = Left Shift Ws & 1 \qquad 1 \qquad N, Z \\ \hline SL \qquad Wb, Wb, Hits, Wnd \qquad Wnd = Left Shift Wb by Wns \qquad 1 \qquad 1 \qquad N, Z \\ \hline SL \qquad Wb, Wb, Hits, Wnd \qquad Wnd = Left Shift Wb by Wls \qquad 1 \qquad 1 \qquad N, Z \\ \hline SL \qquad Wb, Wb, Hits, Wnd \qquad Wnd = Left Shift Wb by Wls \qquad 1 \qquad 1 \qquad N, Z \\ \hline SUB \qquad SUB \qquad f \qquad f = f - WREG \qquad 11 \qquad 1 \qquad 0, D, OV, Z \\ \hline SUB \qquad Hitle, WREG \qquad WREG = f - WREG \qquad 11 \qquad 11 \qquad C, DC, N, OV, Z \\ \hline SUB \qquad Hitle, Wb, Ws, Wd \qquad Wd = Wb - Ws \qquad 1 \qquad 1 \qquad 0, D, O, N, OV, Z \\ \hline SUB \qquad Hitle, Wb, Hits, Wd \qquad Wd = Wb - Ws \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUB \qquad Wb, Hitle, Wd \qquad Wd = Wb - Ws \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUB \qquad Wb, Hitle, Wd \qquad Wd = Wb - Hitle \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Kb, Hitle, Wd \qquad Wd = Wb - Hitle \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Kb, Hitle, Wd \qquad Wd = Wb - Hitle \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad f, WREG \qquad WREG = f - WREG - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Kb, Hitle, Wd \qquad Wd = Wb - Hitle \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Hitle, Wd \qquad Wd = Wb - Hitle \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Hb, Hitle, Wd \qquad Wd = Wb - Ws - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Hitle, Wd \qquad Wd = Wb - Hitle - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Hitle, Wd \qquad Wd = Wb - Hitle - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Hitle, Wd \qquad Wd = Wb - Wb - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Hitle, Wd \qquad Wd = Wb - Wb - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Hitle, Wd \qquad Wd = Wb - Wb - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Hitle, Wd \qquad Wd = WB - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Ws, Wd \qquad Wd = WB - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Ws, Wd \qquad Wd = WB - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Wb, Hitle, Wd \qquad Wd = WB - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Wb, Hitle, Wd \qquad Wd = WB - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Wb, Hitle, Wd \qquad Wd = WB - (C) \qquad 1 \qquad 1 \qquad C, DC, N, OV, Z \\ \hline SUBB \qquad Wb, Wb, Hitle, W$		SETM	Ws	Ws = FFFFh	1	1	None
SL         f, WREG         WREG = Left Shift f         1         1         1         C, N, OY, Z           SL         Ws, Wd         Wd = Left Shift Ws         1         1         0, N, OY, Z           SL         Wb, Wns, Wnd         Wnd = Left Shift Wb by Wns         1         1         N, Z           SUB         f         Mb, #1it5, Wnd         Wnd = Left Shift Wb by Wns         1         1         N, Z           SUB         f         MREG         F = F-WREG         1         1         N, Z           SUB         f, WREG         HINEG         1         1         C, DC, N, OY, Z           SUB         f, WREG         WNd = Left Shift Wb by III5         1         1         C, DC, N, OY, Z           SUB         f, WREG         WN Haits, Nd         Wn = Wn-Elft O         1         1         C, DC, N, OY, Z           SUB         Wb, #1it5, Wd         Wd = Wb - Ws         1         1         C, DC, N, OY, Z           SUBB         f. WREG         MREG = f - WREG - (C)         1         1         C, DC, N, OY, Z           SUBB         f. WREG         WREG = H-WREG - (C)         1         1         C, DC, N, OY, Z           SUBB         f. WREG         WREG = WREG - (C)	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 Ilfs         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         #lit10, Wn         Wn = Wn - lit10         1         1         C, DC, N, OV, Z           SUB         #lit10, Wn         Wn = Wn - lit10         1         1         C, DC, N, OV, Z           SUB         Wb, Ms, 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         B         F, WREG         WREG = f - WREG - (C)         1         1         C, DC, N, OV, Z           SUB         B         Hitt0, Wn         Wn = Wn - lit10 - (C)         1         1         C, DC, N, OV, Z           SUB         # Hit10, Wn         Wn = Wn - lit10 - (C)		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           SL         Wb, #lits, Wnd         Wnd = Left Shift Wb by lit5         1         1         N, Z           SUB         f         f         f         f         f         N, Z           SUB         f         wREG         WREG = f         WREG         1         1         C, DC, N, OV, Z           SUB         f, WREG         WREG = f         WREG         1         1         C, DC, N, OV, Z           SUB         #lit10, Wn         Wn = Wn - lit10         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         f, WREG         WREG = f         WREG - (C)         1         1         C, DC, N, OV, Z           SUB         #lit10, Wn         Wn = Wn - lit10 - (C)         1         1         C, DC, N, OV, Z           SUB         #lit10, Wn         Wn = Wn - lit10 - (C)         1         1         C, DC, N, OV, Z           SUB         Wb, Ws, Wd         Wd = Wb - Ws - (C)		SL	Ws,Wd	Wd = Left Shift Ws	1	1	C, N, OV, Z
SL         Wb,#lit5,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,#lit5,Wd         Wd = Wb - Ws         1         1         C,DC,N,OV,Z           SUB         Wb,#lit5,Wd         Wd = Wb - Ws         1         1         C,DC,N,OV,Z           SUB         Wb,#lit5,Wd         Wd = Wb - Ws         1         1         C,DC,N,OV,Z           SUBB         f.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 - lit10-(C)         1         1         C,DC,N,OV,Z           SUBB         #1110,Wn         Wn = Wn - [it10-(C)         1         1         C,DC,N,OV,Z           SUBB         #1110,Wn         Wn = Wn - [it15-(C)         1         1         C,DC,N,OV,Z           SUBB         #1110,W		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
SUB         f, WREG         WREG = f-WREG         1         1         C, DC, N, OV, Z           SUB         #1it10, Wn         Wn = Wn - lit10         1         1         C, DC, N, OV, Z           SUB         Wb, Ws, 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         1         1         1         C, DC, N, OV, Z           SUBB         f, WREG         f=f-WREG         1         1         1         C, DC, N, OV, Z           SUBB         f, WREG         WREG = f-WREG - (C)         1         1         1         C, DC, N, OV, Z           SUBB         #1it10, Wn         Wn = Wn - lit10 - (C)         1         1         1         C, DC, N, OV, Z           SUBB         #1it10, Wn         Wn = Wn - lit10 - (C)         1         1         1         C, DC, N, OV, Z           SUBB         #1it10, Wn         Wn = Wn - lit10 - (C)         1         1         1         C, DC, N, OV, Z           SUBB         Wb, Ws, Wd         Wd = Wb - Ws - (C)         1         1         1         C, DC, N, OV, Z           SU	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-(C)         1         1         1         C, DC, N, OV, Z           SUBB         f, WREG         WREG = f-WREG-(C)         1         1         1         C, DC, N, OV, Z           SUBB         f, WREG         WREG = f-WREG-(C)         1         1         1         C, DC, N, OV, Z           SUBB         #1110, Wn         Wn = Wn - Ht10-(C)         1         1         C, DC, N, OV, Z           SUBB         #1110, Wn         Wn = Wn - Ht10-(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, H115, Wd         Wd = Wb - Ws - (C)         1         1         C, DC, N, OV, Z           SUBR         f         Wb, Ws, Wd         Wd = Ws - Wb         1         1         C, DC, N, OV, Z           SUBR         f.WREG         WREG = WREG - f - (C)         1         1         C, DC, N, OV, Z           SUBBR		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, Hit5, Wd         Wd = Wb - lit5 - (C)         1         1         C, D, N, OV, Z           SUBR         KBR         f         F         WREG = WREG - f         1         1         C, D, N, OV, Z           SUBR         JUBR         Mb, Ws, Wd         Wd = Ws - Wb         1         1         C, D, C, N, OV, Z           SUBR         JUBR         Wb, Ws, Wd         Wd = Ws - Wb         1         1         C, D, C, N, OV, Z           SUBR         SUBR         M, MEG		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,Hit5,Wd         Wd = Wb - Ws - (C)         1         1         C, DC, N, OV, Z           SUBR         Wb,Ws,Wd         Wd = Wb - Ws - (C)         1         1         C, DC, N, OV, Z           SUBR         Wb,Hit5,Wd         Wd = Wb - Ws - (C)         1         1         C, DC, N, OV, Z           SUBR         f.WREG         WREG = WREG - f         1         1         C, DC, N, OV, Z           SUBR         f.WREG         f = WREG - f - (C)         1         1         C, DC, N, OV, Z           SUBR         Wb,Ws,Wd         Wd = Ws - Wb         1         1         C, DC, N, OV, Z           SUBR         f.WREG         WREG = WREG - f - (C)         1         1         C, DC, N,		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 = IIL10 (C)         1         1         C, DC, N, OY, Z           SUBB         Wb, Ws, Wd         Wd = Wb - Ws - (C)         1         1         C, DC, N, OY, Z           SUBB         Wb, #lit5, Wd         Wd = Wb - III5 - (C)         1         1         C, DC, N, OY, Z           SUBR         SUBR         f         F         F         C, DC, N, OY, Z           SUBR         SUBR         f         G         C, DC, N, OY, Z           SUBR         f, WREG         WREG = WREG - f         1         1         C, DC, N, OY, Z           SUBR         Wb, Ws, Wd         Wd = Ws - Wb         1         1         C, DC, N, OY, Z           SUBR         Wb, Ws, Wd         Wd = Ws - Wb         1         1         C, DC, N, OY, Z           SUBR         Wb, Ws, Wd         Wd = Ws - Wb         1         1         C, DC, N, OY, Z           SUBR         SUBR         f         F         WREG = WREG - f - (C)         1         1         C, DC, N, OY, Z           SUBR         SUBR         f, WREG         WREG = WREG - f - (C)         1         1         C, DC, N, OY, Z           SUBR         Wb, Ws, Wd         Wd = Ws - Wb - (C)         1         1		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         WREG = 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 - Wb         1         1         C, DC, N, OY, Z           SUBR         f, WREG         WB, WS, Wd         Wd = WS - Wb         1         1         C, DC, N, OY, Z           SUBR         Mb, #1it5, Wd         Wd = WS - Wb         1         1         C, DC, N, OY, Z           SUBR         Wb, #1it5, Wd         Wd = WS - Wb         1         1         C, DC, N, OY, Z           SUBBR         f, WREG         MREG = OF - (C)         1         1         C, DC, N, OY, Z           SUBR         f, WREG         Wb, #1it5, Wd         Wd = WS - Wb - (C)         1         1         C, DC, N, OY, Z           SUBR         Wb, #1it5, Wd         Wd = It5 - Wb - (C)         1         1         C, DC, N, OY, Z           SUBR <td< td=""><td></td><td>SUBB</td><td>#11C10,WH</td><td>V(1 - V(1 - H(1) - (C))</td><td>1</td><td>1</td><td>C, DC, N, OV, Z</td></td<>		SUBB	#11C10,WH	V(1 - V(1 - H(1) - (C))	1	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         McG = 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         Wb, #lit5, Wd         Wd = It5 - Wb - (C)         1         1         C, DC, N, OV, Z <t< 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></t<>		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           SWAP         Wn         Wn         Nn B		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 = lit5 - Wb         1         1         C, DC, N, OV, Z           SUBBR         f         F         REG - 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           SUBBR         Wb, #lit5, Wd         Wd = Ws - Wb - (C)         1         1         C, DC, N, OV, Z           SWAP         Wb, #lit5, Wd         Wd = IIt5 - Wb - (C)         1         1         C, DC, N, OV, Z           SWAP         Wn         Wn         None         None         None           TBLRDH         Wn         Read Prog<23:16>	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, #lit5, 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, 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         Wb, #lit5, Wd         Wd = lit5 - Wb - (C)         1         1         C, DC, N, OV, Z           SWAP         Wn         Wn         No e         1         1         None           SWAP         Wn         Wn         Nn = 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         £         f         f         WREG = n(C)         1         1         C, DC, N, OV, Z           SUBBR         f         WREG         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         Wn         Wn         Nine 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}{c ccccccccccccccccccccccccccccccccccc$		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

#### TABLE 29-13: COMPARATOR DC SPECIFICATIONS

Operating Conditions: $2.0V < VDD < 3.6V$ 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	Мах	Units	Comments	
D300	VIOFF	Input Offset Voltage*	_	20	40	mV		
D301	VICM	Input Common Mode Voltage*	0	_	Vdd	V		
D302	CMRR	Common Mode Rejection Ratio*	55		_	dB		

\* Parameters are characterized but not tested.

#### TABLE 29-14: COMPARATOR VOLTAGE REFERENCE DC SPECIFICATIONS

$\begin{array}{llllllllllllllllllllllllllllllllllll$								
Param No.	Symbol	Characteristic	Min	Тур	Max	Units	Comments	
VRD310	CVRES	Resolution	VDD/24		Vdd/32	LSb		
VRD311	CVRAA	Absolute Accuracy	—		AVDD - 1.5	LSb		
VRD312	CVRur	Unit Resistor Value (R)	_	2k	—	Ω		

#### TABLE 29-15: INTERNAL VOLTAGE REFERENCES

Operating Conditions: $2.0V < VDD < 3.6V$ 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	Мах	Units	Comments	
	Vbg	Internal Band Gap Reference	1.14	1.2	1.26	V		
	TIRVST	Internal Reference Stabilization Time	—	200	250	μS		

#### TABLE 29-16: CTMU CURRENT SOURCE SPECIFICATIONS

DC CHARACTERISTICS			$\begin{array}{ll} \mbox{Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq TA \leq +85^{\circ}C \mbox{ for Industrial} \\ -40^{\circ}C \leq TA \leq +125^{\circ}C \mbox{ for Extended} \end{array}$						
Param No.	<sup>n</sup> Sym Characteristic		Min	Typ <sup>(1)</sup>	Мах	Units	Conditions		
	IOUT1	CTMU Current Source, Base Range		550		nA	CTMUICON<1:0> = 01		
	IOUT2	CTMU Current Source, 10x Range	—	5.5	—	μA	CTMUICON<1:0> = 10		
	IOUT3	CTMU Current Source, 100x Range	—	55	—	μA	CTMUICON<1:0> = 11		

**Note 1:** Nominal value at the center point of the current trim range (CTMUICON<7:2> = 000000).

#### TABLE 29-22: AC SPECIFICATIONS

Symbol	Characteristics	Min	Тур	Max	Units
TLW	BCLKx High Time	20	Tcy/2	_	ns
THW	BCLKx Low Time	20	(TCY * BRGx) + TCY/2	—	ns
TBLD	BCLKx Falling Edge Delay from UxTX	-50	—	50	ns
Твно	BCLKx Rising Edge Delay from UxTX	Тсү/2 – 50	—	Tcy/2 + 50	ns
TWAK	Min. Low on UxRX Line to Cause Wake-up	—	1	—	μS
TCTS	Min. Low on UxCTS Line to Start Transmission	Тсү	—	—	ns
TSETUP	Start bit Falling Edge to System Clock Rising Edge Setup Time	3	—	—	ns
TSTDELAY	Maximum Delay in the Detection of the Start bit Falling Edge	—	—	TCY + TSETUP	ns

#### TABLE 29-23: A/D CONVERSION TIMING REQUIREMENTS<sup>(1)</sup>

A/D CHARACTERISTICS			$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$					
Param No.	Symbol	Characteristic	Min.	Тур	Max.	Units	Conditions	
Clock Parameters								
AD50	Tad	A/D Clock Period	75	_	—	ns	Tcy = 75 ns, AD1CON3 is in the default state	
AD51	TRC	A/D Internal RC Oscillator Period	_	250	—	ns		
		Conver	sion Ra	te				
AD55	TCONV	Conversion Time	_	12	—	TAD		
AD56	FCNV	Throughput Rate	_	_	500	ksps	$AVDD \ge 2.7V$	
AD57	TSAMP	Sample Time	_	1	—	TAD		
AD58	TACQ	Acquisition Time	750	_	—	ns	(Note 2)	
AD59	Tswc	Switching Time from Convert to Sample			(Note 3)			
AD60	TDIS	Discharge Time	0.5	_	—	TAD		
		Clock P	aramete	ers				
AD61	TPSS	Sample Start Delay from Setting Sample bit (SAMP)	2		3	TAD		

**Note 1:** Because the sample caps will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

2: The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (VDD to Vss or Vss to VDD).

3: On the following cycle of the device clock.

## TABLE 29-28:RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER<br/>AND BROWN-OUT RESET TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Co (unless otherwise stat Operating temperature			onditions: 1.8V to 3.6V (red) $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended		
Param No.	Symbol	Characteristic	Min.	Typ <sup>(1)</sup>	Max.	Units	Conditions	
SY10	TmcL	MCLR Pulse Width (low)	2	_	_	μS		
SY11	TPWRT	Power-up Timer Period	50	64	90	ms		
SY12	TPOR	Power-on Reset Delay	1	5	10	μS		
SY13	Tioz	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	_		100	ns		
SY20	TWDT	Watchdog Timer Time-out Period	0.85	1.0	1.15	ms	1.32 prescaler	
			3.4	4.0	4.6	ms	1:128 prescaler	
SY25	TBOR	Brown-out Reset Pulse Width	1			μS		
SY35	TFSCM	Fail-Safe Clock Monitor Delay	_	2	2.3	μS		
SY45	TRST	Configuration Update Time		20		μS		
SY55	TLOCK	PLL Start-up Time	_	1		ms		
SY65	Tost	Oscillator Start-up Time	—	1024	—	Tosc		
SY75	TFRC	Fast RC Oscillator Start-up Time	—	1	1.5	μS		
SY85	TLPRC	Low-Power Oscillator Start-up Time		_	100	μS		

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

#### FIGURE 29-8: BAUD RATE GENERATOR OUTPUT TIMING



20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



### RECOMMENDED LAND PATTERN

	MILLIMETERS				
Dimension	MIN	NOM	MAX		
Contact Pitch	E	1.27 BSC			
Contact Pad Spacing	С		9.40		
Contact Pad Width (X20)	Х			0.60	
Contact Pad Length (X20)	Y			1.95	
Distance Between Pads	Gx	0.67			
Distance Between Pads	G	7.45			

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2094A

## **Worldwide Sales and Service**

#### AMERICAS

Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/ support Web Address: www.microchip.com

Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455

Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088

Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075

**Cleveland** Independence, OH Tel: 216-447-0464 Fax: 216-447-0643

**Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924

Detroit Farmington Hills, MI Tel: 248-538-2250 Fax: 248-538-2260

Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453

Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608

Santa Clara Santa Clara, CA Tel: 408-961-6444 Fax: 408-961-6445

Toronto Mississauga, Ontario, Canada Tel: 905-673-0699 Fax: 905-673-6509

#### ASIA/PACIFIC

Asia Pacific Office Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon Hong Kong Tel: 852-2401-1200 Fax: 852-2401-3431 Australia - Sygney

Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

**China - Beijing** Tel: 86-10-8569-7000 Fax: 86-10-8528-2104

**China - Chengdu** Tel: 86-28-8665-5511 Fax: 86-28-8665-7889

**China - Chongqing** Tel: 86-23-8980-9588 Fax: 86-23-8980-9500

**China - Hangzhou** Tel: 86-571-2819-3187 Fax: 86-571-2819-3189

**China - Hong Kong SAR** Tel: 852-2401-1200 Fax: 852-2401-3431

**China - Nanjing** Tel: 86-25-8473-2460 Fax: 86-25-8473-2470

**China - Qingdao** Tel: 86-532-8502-7355 Fax: 86-532-8502-7205

**China - Shanghai** Tel: 86-21-5407-5533 Fax: 86-21-5407-5066

China - Shenyang Tel: 86-24-2334-2829 Fax: 86-24-2334-2393

**China - Shenzhen** Tel: 86-755-8203-2660 Fax: 86-755-8203-1760

**China - Wuhan** Tel: 86-27-5980-5300 Fax: 86-27-5980-5118

**China - Xian** Tel: 86-29-8833-7252 Fax: 86-29-8833-7256

**China - Xiamen** Tel: 86-592-2388138 Fax: 86-592-2388130

**China - Zhuhai** Tel: 86-756-3210040 Fax: 86-756-3210049

#### ASIA/PACIFIC

India - Bangalore Tel: 91-80-3090-4444 Fax: 91-80-3090-4123

**India - New Delhi** Tel: 91-11-4160-8631 Fax: 91-11-4160-8632

India - Pune Tel: 91-20-2566-1512 Fax: 91-20-2566-1513

**Japan - Yokohama** Tel: 81-45-471- 6166 Fax: 81-45-471-6122

**Korea - Daegu** Tel: 82-53-744-4301 Fax: 82-53-744-4302

Korea - Seoul Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934

Malaysia - Kuala Lumpur Tel: 60-3-6201-9857 Fax: 60-3-6201-9859

**Malaysia - Penang** Tel: 60-4-227-8870 Fax: 60-4-227-4068

Philippines - Manila Tel: 63-2-634-9065 Fax: 63-2-634-9069

Singapore Tel: 65-6334-8870 Fax: 65-6334-8850

**Taiwan - Hsin Chu** Tel: 886-3-5778-366 Fax: 886-3-5770-955

**Taiwan - Kaohsiung** Tel: 886-7-536-4818 Fax: 886-7-330-9305

Taiwan - Taipei Tel: 886-2-2500-6610 Fax: 886-2-2508-0102

**Thailand - Bangkok** Tel: 66-2-694-1351 Fax: 66-2-694-1350

#### EUROPE

Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829

France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

**Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44

Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781

Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340

**Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91

**UK - Wokingham** Tel: 44-118-921-5869 Fax: 44-118-921-5820

08/02/11