9.1 <u>Instruc</u>	tion Descriptions		
ADDLW	Add Literal and W	ANDLW	AND Literal with W
Syntax:	[labe] ADDLW k	Syntax:	[labe] ANDLW k
Operands:	0 ≤k ≤255	Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)	Operation:	(W) .AND. (k) → (W)
Status Affected:	C, DC, Z	Status Affected:	Z
Encoding:	11 111x kkkk kkkk	Encoding:	11 1001 kkkk kkkk
Description:	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.	Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1	Words:	1
Cycles:	1	Cycles:	1
Q Cycle Activity:	Q1 Q2 Q3 Q4	Q Cycle Activity:	Q1 Q2 Q3 Q4
	Decode Read Process Write to data W		Decode Read Iteral "k" Process Write to data W
Example:	ADDLW 0x15	Example	ANDLW 0x5F
	Before Instruction W = 0x10 After Instruction W = 0x25		Before Instruction W = 0xA3 After Instruction W = 0x03
ADDWF	Add W and f	ANDWF	AND W with f
Syntax:	[labe] ADDWF f,d	Syntax:	[labe] ANDWF f,d
Operands:	0 ≤f ≤127	Operands:	0 ≤f ≤127

ADDWF	Add W and f			ANDWF	AND W	with f		
Syntax:	[labe] ADDWF	f,d		Syntax:	[labe] A	NDWF	f,d	
Operands:	$0 \le f \le 127$ $d \in [0,1]$			Operands:	$0 \le f \le 12$ $d \in [0,1]$	27		
Operation:	(W) + (f) → (de:	stination)		Operation:	(W) .ANI	O. (f) → (destinatio	on)
Status Affected:	C, DC, Z			Status Affected:	Z			
Encoding:	00 0111	dfff	ffff	Encoding:	0.0	0101	dfff	ffff
Description:	Add the contents contents of regist stored in the W re stored back in reg	er 'f'. If 'd' is egister. If 'd'	0 the result is	Description:	AND the V If 'd' is 0 t ter. If 'd' is register 'f'.	he result i 1 the res	is stored in	the W req
Words:	1			Words:	1			
Cycles:	1			Cycles:	1			
Q Cycle Activity:	Q1 Q2	Q3	Q4	Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode Read registe		Write to destination		Decode	Read register 'f'	Process data	Write to destination
Example	ADDWF FSR,	0		Example	ANDWF	FSR,	1	
	Before Instructi				Before In			
	W = ESR =	0x17 0xC2				W = ESR =	0x17 0xC2	
	After Instruction				After Inst		UNUZ	
	w =	0xD9				w =	0x17	
	FSR =	0xC2				FSR =	0x02	

1998 Microchip Technology Inc.

DS30430C-page 57

PIC16F8X

BTFSS	Bit Test	f, Skip if	Set		CALL	Call Sul	oroutine		
Syntax:	[labe] B	TFSS f,b			Syntax:	[label]	CALL I	(
Operands:	0 ≤f ≤12				Operands:	0 ≤ k ≤ 2	047		
	0 ≤b < 7				Operation:	(PC)+ 1-			
Operation:	skip if (f<	(b>) = 1				k → PC			
Status Affected:	None					•	H<4:3>) -	• PC<12	:11>
Encoding:	01	11bb	bfff	ffff	Status Affected:	None			
Description:		register 'f' i		he next	Encoding:	10	0kkk	kkkk	kkkk
	If bit 'b' is discarded instead, m	'1', then the and a NOF aking this a	next instr	ed	Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH.CALL			
Words:	1						ycle instru		INCALL
Cycles:	1(2)				Words:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4	Cycles:	2			
	Decode	Read register 'f'	Process data	No-Operat ion	Q Cycle Activity:	Q1	Q2	Q3	Q4
If Skip:	(2nd Cycle)				1st Cycle	Decode	Read literal 'k', Push PC	Process data	Write to PC
	Q1	Q2	Q3	Q4			to Stack		
	No-Operat ion	No-Operati on	No-Opera tion	No-Operat ion	2nd Cycle	No-Opera tion	No-Opera tion	No-Opera tion	No-Operat ion
Example	HERE FALSE	BTFSC	FLAG,1 PROCESS	CODE	Example	HERE	CALL	THERE	Į.
	TRUE	•				Before I	nstruction	1	
		:				After Ins		dress HE	RE
	Refere Ir	struction				Alter IIIs		dress TH	ERE
			address E	HERE			TOS = A	ddress HE	RE+1
	After Inst								
		if FLAG<12 PC =	> = 0, address F.	ATOD					
		if FLAG<1		ALGE					
		PC =	address T	RUE					

PIC16F8X

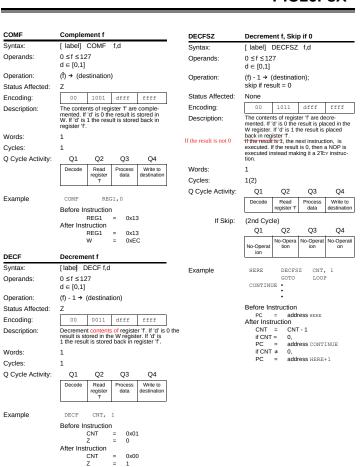
BCF	Bit Clear f	BTFSC	Bit Test, Skip if Clear
Syntax:	[labe] BCF f,b	Syntax:	[labe] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7	Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f)	Operation:	skip if (f) = 0
Status Affected:	None	Status Affected:	None
Encoding:	01 00bb bfff ffff	Encoding:	01 10bb bfff ffff
Description:	Bit 'b' in register 'f' is cleared.	Description:	If bit 'b' in register 'f' is '1' then the next
Words:	1		instruction is executed. If bit 'b', in register 'f', is '0' then the next
Cycles:	1		instruction is discarded, and a NOP is
Q Cycle Activity:	Q1 Q2 Q3 Q4		executed instead, making this a 2Tcy instruction.
	Decode Read Process Write register T	Words:	1
	Tr data register i	Cycles:	1(2)
E		Q Cycle Activity:	Q1 Q2 Q3 Q4
Example	BCF FLAG_REG, 7		Decode Read Process No-Opera
	Before Instruction FLAG REG = 0xC7		register T data ion
	After Instruction	If Skip:	(2nd Cycle)
	FLAG_REG = 0x47		Q1 Q2 Q3 Q4
			No-Operati No-Opera No-Operati No-Operati No-Operati No-Operati No-Operation
			ion
		Example	HERE BTFSC FLAG, 1
			FALSE GOTO PROCESS_CODE
			TRUE •
			•
			Before Instruction PC = address HERE
			After Instruction
BSF	Bit Set f		if FLAG<1> = 0, PC = address TRUE
Syntax:	[labe] BSF f,b		if FLAG<1>=1,
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7		PC = address FALSE
Operation:	1 → (f)		
Status Affected:	None		
Encoding:	01 01bb bfff ffff		
Description:	Bit 'b' in register 'f' is set.		
Words:	1		
Cycles:	1		
Q Cycle Activity:	Q1 Q2 Q3 Q4		
,	Decode Read Process Write		
	register data register f		
Example	BSF FLAG_REG, 7		
	Before Instruction		
	FLAG_REG = 0x0A After Instruction		
	FLAG_REG = 0x8A		

DS30430C-page 58

1998 Microchip Technology Inc.

PIC16F8X

CLRF	Clear f	CLRW	Clear W
Syntax:	[labe] CLRF f	Syntax:	[label] CLRW
Operands:	0 ≤ f ≤ 127	Operands:	None
Operation:	00h → (f) 1 → Z	Operation:	00h → (W) 1 → Z
Status Affected:	Z	Status Affected:	Z
Encoding:	00 0001 1fff ffff	Encoding:	00 0001 0xxx xxxx
Description:	The contents of register 'f' are cleared and the Z bit is set.	Description:	W register is cleared. Zero bit (Z) is set.
Words:	1	Words:	1
Cycles:	1	Cycles:	1
Q Cycle Activity:	Q1 Q2 Q3 Q4	Q Cycle Activity:	Q1 Q2 Q3 Q4
	Decode Read Process Write register 'f'		Decode No-Opera tion Process Write to W
Example	CLRF FLAG REG	Example	CLRW
	Before Instruction		Before Instruction
	FLAG_REG = 0x5A		W = 0x5A After Instruction
	After Instruction FLAG_REG = 0x00 Z = 1		W = 0x00 Z = 1
		CLRWDT	Clear Watchdog Timer
		Syntax:	[label] CLRWDT
		Operands:	None
		Operation:	00h → WDT 0 → WDT prescaler, 1 → \overline{TO} 1 → \overline{PD}
		Status Affected:	TO, PD
		Encoding:	00 0000 0110 0100
		Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits TO and PD are set.
		Words:	1
		Cycles:	1
		Q Cycle Activity:	Q1 Q2 Q3 Q4
			Decode No-Opera tion Process data WDT Counter
		Example	CLRWDT
			Before Instruction
			WDT counter = ?
			After Instruction
			After Instruction WDT counter = 0x00
			After Instruction



□ 1998 Microchip Technology Inc.

PIC16F8X

DS30430C-page 61

INCFSZ	Increment f, Skip if 0	IORLW	Inclusive OR Literal with W
Syntax:	[label] INCFSZ f,d	Syntax:	[label] IORLW k
Operands:	0 ≤f ≤127	Operands:	0 ≤ k ≤ 255
	d ∈ [0,1]	Operation:	(W) .OR. k → (W)
Operation:	(f) + 1 → (destination), skip if result = 0	Status Affected:	Z
Status Affected:	None	Encoding:	11 1000 kkkk kkkk
Encoding:	00 1111 dfff ffff	Description:	The contents of the W register is
Description:	The contents of register 'f' are incre-		OR'ed with the eight bit literal 'k'. The result is placed in the W register.
Description.	mented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is	Words:	1
If the result is not 0	placed back in register 'f'. If the result is 1, the next instruction is	Cycles:	1
if the result is not 0	ratio result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead making it a 2T cy instruc-	O Cycle Activity:	01 02 03 04
	tion.	, ,	Decode Read Process Write to
Words:	1		literal 'k' data W
Cycles:	1(2)	Example	IORLW 0x35
Q Cycle Activity:	Q1 Q2 Q3 Q4	Zampio	Before Instruction
	Decode Read Process Write to		W = 0x9A
If Skip:			After Instruction W = 0xBF
п экір.	(2nd Cycle) Q1 Q2 Q3 Q4		Z = 1
	No-Operat ion No-Operat on No-Operation		Z = 0 !!!!
Example	HERE INCFSZ CNT, 1 GOTO LOOP CONTINUE • •		
	Before Instruction PC = address HERE After Instruction CNT = CNT + 1 if CNT = 0, PC = address CONTINUE if CNT# 0, PC = address HERE + 1		

PIC16F8X

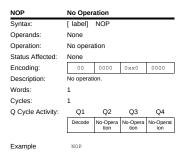
GOTO	Unconditi	ional Br	anch		INCF	Increme	nt f		
Syntax:	[label]	GOTO	k		Syntax:	[label]	INCF 1	f,d	
Operands:	0 ≤ k ≤ 20	47			Operands:	0 ≤ f ≤ 12			
	k → PC<1					d ∈ [0,1]			
1	PCLATH<	:4:3>→ F	PC<12:11	L>	Operation:	(f) + 1 →	(destina	ition)	
Status Affected:	None				Status Affected:	Z			
Encoding:	10	1kkk	kkkk	kkkk	Encoding:	0.0	1010	dfff	ffff
	GOTO is an eleven bit ir into PC bits PC are load GOTO is a tv	nmediate <10:0>. Ted from F	value is lo The upper PCLATH<4	aded bits of 1:3>.	Description:	The conte mented. If the W reg placed ba	'd' is 0 th ister. If 'd'	is 1 the re	placed in
Words:	1	•			Words:	1			
Cycles:	2				Cycles:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4	Q Cycle Activity:	Q1	Q2	Q3	Q4
1st Cycle	Decode	Read literal 'k'	Process data	Write to PC		Decode	Read register 'f'	Process data	Write to destination
2nd Cycle	No-Operat ion	No-Operat ion	No-Opera tion	No-Operat ion	Example	INCF	CNT,	1	
						Before Ir			
Example	GOTO TH	ERE					CNT 7	= 0xFI = 0	F
	After Instr					After Ins	_	- 0	
	Р	PC = /	Address	THERE			CNT	= 0x00)
							Z	= 1	

DS30430C-page 62 🗆 1998 Microchip Technology Inc.

PIC16F8X

IORWF	Inclusive OR W with f	MOVLW	Move Literal to W
Syntax:	[label] IORWF f,d	Syntax:	[label] MOVLW k
Operands:	0 ≤f ≤127	Operands:	0 ≤ k ≤ 255
	$d \in [0,1]$	Operation:	k → (W)
Operation:	(W) .OR. (f) → (destination)	Status Affected:	None
/	₹ Z!!!	Encodina:	11 00xx kkkk kkkk
Encoding:	00 0100 dfff ffff	Description:	The eight bit literal 'k' is loaded into W
Description:	Inclusive OR the W register with contents of register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.	Words:	register. The don't cares will assemble as 0's.
Words:	1		1
Cycles:	1	Cycles: O Cycle Activity:	
Q Cycle Activity:	01 02 03 04	Q Cycle Activity:	Q1 Q2 Q3 Q4 Decode Read Process Write to
Q Cycle Activity.	Decode Read Process Write to		literal 'k' data W
	register data destination	Example	MOVLW 0x5A
			After Instruction
Example	IORWF RESULT, 0 Before Instruction		W = 0x5A
	RESULT = 0x13 W = 0x91 After Instruction RESULT = 0x13 W = 0x93 Z = 1		
	Z = 0 !!!!		
MOVF	Z=0 !!!	MOVWF	Move W to f
MOVF Syntax:		Syntax:	[label] MOVWF f
	Move f	Syntax: Operands:	[label] MOVWF f 0 ≤f ≤127
Syntax:	Move f [label] MOVF f,d	Syntax: Operands: Operation:	[label] MOVWF f 0 ≤f ≤127 (W) → (f)
Syntax: Operands: Operation:	Move f [label] MOVF f,d 0 ≤ f ≤ 127 d ∈ [0,1] (f) → (destination)	Syntax: Operands: Operation: Status Affected:	[label] MOVWF f $0 \le f \le 127$ (W) \rightarrow (f) None
Syntax: Operands: Operation: Status Affected:	Move f [label] MOVF f,d 0 ≤f ≤ 127 d ∈ [0,1] (f) → (destination) Z	Syntax: Operands: Operation: Status Affected: Encoding:	[label] MOVWF f $0 \le f \le 127$ (W) \rightarrow (f) None $0 0 0000 1 fff ffff$
Syntax: Operands: Operation: Status Affected: Encoding:	Move f [label] MOVF f,d $0 \le f \le 127$ d ∈ [0,1] (f) → (destination) Z $0 \ge 1000$ dfff ffff	Syntax: Operands: Operation: Status Affected:	[label] MOVWF f $0 \le f \le 127$ (W) \Rightarrow (f) None $0 = 0.000 = 0.000 = 0.000$ Move data from W register to register
Syntax: Operands: Operation: Status Affected:	Move f [label] MOVF f,d 0 ≤f ≤127 d ∈ [0,1] (f) → (destination) Z 00 1000 dfff ffff The contents of register f is moved to a	Syntax: Operands: Operation: Status Affected: Encoding:	[label] MOVWF f $0 \le f \le 127$ (W) \rightarrow (f) None $0 0 0000 1 fff ffff$
Syntax: Operands: Operation: Status Affected: Encoding:	Move f [label] MOVF f,d $0 \le f \le 127$ d ∈ [0,1] (f) \Rightarrow (destination) Z $0 \ge f \le 127$ d ∈ [0,1] The contents of register f is moved to a destination dependant upon the status of d. Ifd $= 0$, destination is W register. If	Syntax: Operands: Operation: Status Affected: Encoding: Description:	[label] MOVWF f $0 \le f \le 127$ (W) \rightarrow (f) None 0 0 0000 1 f f f f f f f Move data from W register to register f.
Syntax: Operands: Operation: Status Affected: Encoding:	Move f [label] MOVF f,d 0 ≤f ≤ 127 d ∈ [0,1] (f) → (destination) Z 00 1000 dfff ffff The contents of register f is moved to a destination dependant upon the status	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words:	[label] MOVWF f $0 \le f \le 127$ (W) \rightarrow (f) None $0 0 0000 1 fff fff$ Move data from W register to register f.
Syntax: Operands: Operation: Status Affected: Encoding:	Move f [label] MOVF f,d 0 ≤f ≤ 127 d ∈ [0,1] (f) → (destination) Z 00 1000 dfff ffff The contents of register f is moved to a destination dependant upon the status of d. if d = 0, destination is W register. If d = 1, the destination is F register. If d = 1, the destination is N register. If d = 1, the	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles:	[label] MOVWF f 0 ≤ f ≤ 127 (W) → (f) None 0 0 0000 1 f f f f f f f f Move data from W register to register T. 1 Q1 Q2 Q3 Q4 Decode Read Process Write
Syntax: Operands: Operation: Status Affected: Encoding:	Move f [label] MOVF f,d $0 \le f \le 127$ d ∈ [0,1] (f) \Rightarrow (destination) Z $0 \le f \le 127$ d ∈ [0,1] (f) \Rightarrow (destination) Z $0 \le f \le 127$ The contents of register f is moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is life register f itself. d = 1 is useful to test a file register since status flag Z is affected. 1	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles:	[label] MOVWF f $0 \le f \le 127$ (W) \Rightarrow (f) None $0 = 0000 1 \text{ fff} \text{ffff}$ Move data from W register to register r. 1 1 $Q1 Q2 Q3 Q4$
Syntax: Operation: Status Affected: Encoding: Description: Words: Cycles:	Move f [label] MOVF f,d $0 \le f \le 127$ d ∈ [0,1] (f) \Rightarrow (destination) Z $0 \le f \le 127$ d ∈ [0,1] (f) \Rightarrow (destination) Z $0 \le f \le 127$ The contents of register f is moved to a destination dependant upon the status of d. if d = 0, destination is W register. If d = 1, the destination is life register f itself. d = 1 is useful to test a file register risince status flag Z is affected. 1	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	[label] MOVWF f $0 \le f \le 127$ (W) \Rightarrow (f) None $0 = 0.000 = 1 \text{ frf} = 1 \text{ frff}$ Move data from W register to register fr. 1 Q1 Q2 Q3 Q4 Decode Read Process Write register data register
Syntax: Operation: Operation: Status Affected: Encoding: Description: Words:	Move f [label] MOVF f,d $0 \le f \le 127$ $d \in [0,1]$ (f) → (destination) Z $0 \ge 1000$ The contents of register f is moved to a destination dependant upon the status of d. If $d = 0$, destination is W register. If $d = 1$, the destination is like register f itself. $d = 1$ is the destination is like register f itself. $d = 1$ is subsult to test a file register since status flag Z is affected. 1 Q1 Q2 Q3 Q4	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles:	
Syntax: Operation: Status Affected: Encoding: Description: Words: Cycles:	Move f [label] MOVF f,d 0 ≤f ≤127 d ∈ [0,1] (f) → (destination) Z The contents of register f is moved to a destination dependant upon the status of d. If d = 0, destination is Wegister. If d = 1, the destination is Wegister. If d = 1, the destination is Wegister. If d = 1, the destination is Wegister. If the status of d = 1 is useful to test a file register f itself. d = 1 is useful to test a file register f itself. d = 1 is useful to test a file register. 1 1 Q1 Q2 Q3 Q4 Decode Read Process Write to	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	[label] MOVWF f 0 ≤ f ≤ 127 (W) → (f) None 00 0000 1 f f f f f f f f f f f f f f f
Syntax: Operation: Status Affected: Encoding: Description: Words: Cycles:	Move f [label] MOVF f,d 0 ≤f ≤127 d ∈ [0,1] (f) → (destination) Z The contents of register f is moved to a destination dependant upon the status of d. If d = 0, destination is Wegister. If d = 1, the destination is Wegister. If d = 1, the destination is Wegister. If d = 1, the destination is Wegister. If the status of d = 1 is useful to test a file register f itself. d = 1 is useful to test a file register f itself. d = 1 is useful to test a file register. 1 1 Q1 Q2 Q3 Q4 Decode Read Process Write to	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	[label] MOVWF f 0 ≤ f ≤ 127 (W) → (f) None 0 0 0000 1 fff fff Move data from W register to register T. 1 Q1 Q2 Q3 Q4 Decode Read Process register T MOVWF OPTION_REG Before Instruction OPTION = 0xFF W = 0x4F
Syntax: Operation: Status Affected: Encoding: Description: Words: Cycles:	Move f [label] MOVF f,d 0 ≤f ≤ 127 d ∈ [0,1] (f) → (destination) Z 00 1000 dfff ffff The contents of register f is moved to a destination dependant upon the status of d. Ifd = 0, destination is W register. If d = 1, the destination is life register f isself. d = 1 is useful to test a file register since status flag Z is affected. 1 Q1 Q2 Q3 Q4 Decode Read Process Write to destination destination	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	
Syntax: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	Move f [label] MOVF f,d $0 \le f \le 127$ d ∈ [0,1] (f) \Rightarrow (destination) Z $0 \le f \le 100$ 1000 dfff ff f	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	
Syntax: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	Move f [label] MOVF f,d 0 ≤ f ≤ 127 d ∈ [0,1] (f) → (destination) Z 00 1000 dfff ffff The contents of register f is moved to a destination dependant upon the status of d. Ifd = 0, destination is W register. If d = 1, the destination is like register f itself. d = 1 is useful to test a file register since status flag Z is affected. 1 Q1 Q2 Q3 Q4 Decode Read Process Write to register f The destination is which is the register of data destination MOVF FSR, 0	Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	[label] MOVWF f 0 ≤ f ≤ 127 (W) → (f) None 0 0 0000 1 fff fff Move data from W register to register T. 1 Q1 Q2 Q3 Q4 Decode Read process register T MOVWF OPTION_REG Before Instruction OPTION = 0xFF W = 0x4F After Instruction OPTION = 0x4F

PIC16F8X



RETFIE	Return f	rom Inte	rrupt	
Syntax:	[label]	RETFIE		
Operands:	None			
Operation:	TOS → F 1 → GIE	PC,		
Status Affected:	None			
Encoding:	0.0	0000	0000	1001
	and Top o PC. Interro Global Inte (INTCON« instruction	upts are er errupt Enal :7>). This i	nabled by s ble bit, GIE	etting
Words:	1			
Cycles:	2			
Q Cycle Activity:	Q1	Q2	Q3	Q4
1st Cycle	Decode	No-Opera tion	Set the GIE bit	Pop from the Stack
2nd Cycle	No-Operat ion	No-Opera tion	No-Opera tion	No-Operat ion
Example	RETFIE			

OPTION	Load Op	tion Re	gister	
Syntax:	[label]	OPTION	1	
Operands:	None			
Operation:	(W) → O	PTION		
Status Affected:	None			
Encoding:	00	0000	0110	0010
Description:	The conter loaded in t instruction patibility w Since OPT register, th it.	he OPTIC is suppor ith PIC16 TION is a	ON register ted for coo C5X produ readable/w	This de com- icts.
Words:	1			
Cycles:	1			
Example				
	with fu	ture PiC1	ard comp 6CXX pro is instruct	ducts,

1998 Microchip Technology Inc.

DS30430C-page 65

PIC16F8X

RLF	Rotate Left f the	ough Ca	rry	RRF	Rotate F	Right f th	rough C	arry
Syntax:	[label]	RLF f	,d	Syntax:	[label]	RRF f,	d	
Operands:	$0 \le f \le 127$ $d \in [0,1]$			Operands:	$0 \le f \le 12$ $d \in [0,1]$	27		
Operation:	See description	below		Operation:	See desc	cription b	elow	
Status Affected:	С			Status Affected:	С			
Encoding:	00 1101	dfff	ffff	Encoding:	0.0	1100	dfff	ffff
Description:	The contents of recone bit to the left the Flag. If 'd' is 0 the in Wiregister. If 'd' is back in register 'f'.	rough the esult is pla	Carry ced in the	Description:	The conte one bit to Flag. If 'd' W register back in re	the right the rest. If 'd' is 1 gister 'f'.	nrough the esult is pla	Carry ced in the
							regioter	
Words:	1			Words:	1			
Cycles:	1			Cycles:	1			
Q Cycle Activity:	Q1 Q2	Q3	Q4	Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode Read register	Process data	Write to destination		Decode	Read register 'f'	Process data	Write to destination
Example	RLF RE	G1,0		Example	RRF		REG1,0	
	Before Instructio	struction			Before Instruction			
	REG1 C After Instruction REG1 W C	= 0	0 0110 0 0110 0 1100		After Inst	REG1 C truction REG1 W C	= 0 = 111	0 0110 0 0110 1 0011

PIC16F8X

RETLW	Return wit	th Litor	ol in W		RETURN	Doturn f	rom Sub	routino	
Syntax:	[label] F				Syntax:	[label]			
Operands:	0 ≤ k ≤ 255		κ.		Operands:	None	KLIOK		
•		•			•				
Operation:	k → (W); TOS → PC				Operation:	TOS → I	SC.		
Status Affected:	None	•			Status Affected:	None			
		0.1			Encoding:	0.0	0000	0000	1000
Encoding: Description:	11 01xx kkkk kkkk The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the			ne eight ter is	Description:	POPed ar is loaded	nd the top	ine. The st of the stack ogram coul ction.	(TOS)
	return addre	return address). This is a two cycle			Words:	1			
	instruction.				Cycles:	2			
Words:	1				O Cycle Activity:	01	Q2	Q3	Q4
Cycles:	2				1st Cycle		No-Opera	No-Opera	Pop from
Q Cycle Activity:	Q1	Q2	Q3	Q4	13t Cycle	Decode	tion	tion	the Stack
1st Cycle	Decode	Read literal 'k'	No-Opera tion	Write to W, Pop from the Stack	2nd Cycle	No-Operat ion	No-Opera tion	No-Opera tion	No-Opera tion
2nd Cycle	No-Operat ion	No-Opera tion	No-Opera tion	No-Operat ion	Example	RETURN After Inte	errupt		
Example	CALL TABLE ADDWF PC RETLW k1 RETLW k2	;offse					PC =	TOS	
	RETLW kn Before Inst W After Instru	; End of truction / = uction	of table 0x07 value of k8	3					

DS30430C-page 66 1998 Microchip Technology Inc.

PIC16F8X

SLEEP Syntax: [label] SLEEP Operands: None $\begin{array}{l} 00h \rightarrow \ \text{WDT}, \\ 0 \rightarrow \ \text{WDT prescaler}, \\ 1 \rightarrow \ \overline{\text{TO}}, \\ 0 \rightarrow \ \overline{\text{PD}} \end{array}$ Operation: TO, PD Status Affected: TO, PD

00 0000 0110 0011

The power-down status bit, PD is cleared. Time-out status bit, TO is set. Watchdog Timer and its prescaler are cleared.

The processor is put into SLEEP mode with the scillator stopped. See Section 14.8 for more details. Encoding: Description: Words: Cycles: Q2 Q Cycle Activity: Q1 Q3 Q4 Decode No-Opera tion No-Opera tion Go to Sleep Example: SLEEP

Syntax:	[label]	SUBL	W k			
Operands:	0 ≤k ≤25					
Operation:	k - (W) →					
Status Affected:	C, DC, Z	(**)				
	11	110x	kkkk	kkkk		
Encoding: Description:	The conter ment meth	nts of W rod) from	egister is si the eight bi	ubtracted (2 t literal 'k'.		
		is placed	in the W re	gister.		
Words:	1					
Cycles:	1					
Q Cycle Activity:	Q1	Q2	Q3	Q4		
	Decode	Read literal 'k	Process data	Write to W		
Example 1:	SUBLW		0x02			
	Before Instruction					
		w =	1			
		C = 7 =	?			
	Z = ? After Instruction					
	Aitei iiisti	w =	1			
		C =		is positive		
		Z =	0			
Example 2:	Before In:	struction	1			
		w =	2			
		C = 7 =	?			
	After Instruction					
		w =	0			
		C =	1; result	is zero		
		Z =	1			
Example 3:	Before Instruction					
		W =	3			
		Z =	?			
	After Instruction					
		W =	0xFF			

PIC16F8X

SUBWF	Subtract V	V from	f		SWAPF	Swap Ni	bbles in	f	
Syntax:	[label]	SUBWI	F f,d	•	Syntax:	[label]	SWAPF	f,d	
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$				Operands:	$0 \le f \le 12$ $d \in [0,1]$			
Operation:	(f) - (W) →	(destina	ation)		Operation:		→ (destir		
Status Affected:	C, DC, Z					(f<7:4>)	→ (destir	ation<3:	0>)
Encoding:	0.0	0010	dfff	ffff	Status Affected:	None		1	,
Description:	Subtract (2's				Encoding:	0.0	1110	dfff	ffff
	result is store	red in the	W register.	If 'd' is 1 the	Description:	register 'f' is placed	r and lowe are excha in W regis in register	nged. If 'd ter. If 'd' is	d' is 0 the
Words:	1				Words:	1	iii registei		
Cycles:	1				Cycles:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4	Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode	Read register 'f'	Process write to		Q Cycle Activity.	Decode	Read	Process	Write to
			I				register 'f'	data	destination
Example 1:	SUBWF		REG1,1		Fuerente	0113.00	220		
	Before Instruction				Example	SWAPF REG, 0			
	REG1	=	3			Before II	nstruction		
	W C	=	2				REG1	= 0x	A5
	Z	=	?			After Ins			
	After Instru	iction					REG1 W	= 0x = 0x	A5 5A
	REG1	=	1				vv	= UX	3A
	W	=	2	manish on					
	C Z	=	1; result is 0	positive					
Example 2:	Before Inst	ruction			TRIS	Load TF	IS Regis	ter	
•	REG1	=	2		Syntax:	[label	TRIS		
	w	=	2		Operands:	5 ≤ f ≤ 7			
	C Z	=	?		Operation:	(W) → T	RIS regis	ter f:	
	After Instru	ıction			Status Affected:	٠,,			
	RFG1	=	0		Encoding:	0.0	0000	0110	Offf
	W	=	2		Description:	The instru	ction is su	pported fo	r code
	C 7	=	1; result is	zero			lity with the		
Example 3:	Before Inst	_	1				e TRIS reç writable, th nem.		
	REG1	=	1		Words:	1			
	W C	=	2		Cycles:	1			
	Z	=	?		Example				
	After Instruction				To mair	ntain upw	ard comp	atibility	
	REG1	=	0xFF			with fu	ture PIC1	6CXX pro	ducts,
		=	2			do n	ot use thi	s instruct	tion.
	W C	=	0; result is						

PIC16F8X

XORLW	Exclusive OR Literal with W	XORWF	Exclusive OR W with f			
Syntax:	[label XORLW k	Syntax: [labe] XORWF f,d				
Operands:	0 ≤k ≤255	Operands:	$0 \le f \le 127$ $d \in [0,1]$ (W) .XOR. (f) \rightarrow (destination)			
Operation:	(W) .XOR. k → (W)	Operation:				
Status Affected:	Z	Status Affected:				
Encoding:	11 1010 kkkk kkkk	Encoding:	00 0110 dfff ffff			
Description:	The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.	Description:	Exclusive OR the contents of the W register with contents of register 'f'. If 'd' 0 the result is stored in the W register. I			
Words:	1	Words:	1 the result is stored back in register 'f'. 1			
Cycles:	1	Cycles:	1			
Q Cycle Activity:	Q1 Q2 Q3 Q4	O Cycle Activity:	01 02 03 04			
	Decode Read Process Write to data W	Q 2,000 i 000 ii,	Decode Read Process Write to register data destination			
Example:	XORLW 0xAF		T Gestinate			
	Before Instruction	Example	XORWF REG 1			
	W = 0xB5		Before Instruction			
	After Instruction W = 0x1A		REG = 0xAF W = 0xB5			
	5.27		After Instruction			
			REG = 0x1A W = 0xB5			