9.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 9-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 9-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 9-1 OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with $x = 0$. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer/Counter
TO	Time-out bit
PD	Power-down bit
dest	Destination either the W register or the specified register file location
[]	Options
()	Contents
\rightarrow	Assigned to
<>	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- · Byte-oriented operations
- Bit-oriented operations
- Literal and control operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μ s. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μ s.

Table 9-2 lists the instructions recognized by the MPASM assembler.

Figure 9-1 shows the general formats that the instructions can have.

Note: To maintain upward compatibility with future PIC16CXX products, <u>do not use</u> the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 9-1: GENERAL FORMAT FOR

Byte-oriented file register operations 13 8 7 6 0 OPCODE d f (FILE #) d d f (FILE #) d = 0 for destination W d = 1 for destination f f f f f d = 1 for destination f f = 7-bit file register operations 13 10 9 7 6 0 OPCODE b (BIT #) f (FILE #) f b = 3-bit bit address f = 7-bit file register address Literal and control operations General 13 8 7 0 OPCODE k (literal) k = 8-bit immediate value k 2ALL and GOTO instructions only 13 11 10 0 OPCODE k (literal) k = 11-bit immediate value k 14-bit immediate value						
13 8 7 6 0 OPCODE d f (FILE #) d d f (FILE #) d = 0 for destination W d = 1 for destination f f f = 7-bit file register address Bit-oriented file register operations 13 10 9 7 6 0 OPCODE b (BIT #) f (FILE #) f f f f b = 3-bit bit address f = 7-bit file register address f	Byte-oriented fil	ie regis	ster op	perat	tions	
$\begin{tabular}{ c c c c } \hline OPCODE & d & f(FILE \#) \\ \hline d = 0 \mbox{ for destination W} \\ \hline d = 1 \mbox{ for destination f} \\ \hline f = 7 \mbox{-bit file register address} \\ \hline \end{tabular}$	13	8	7	6		0
	OPCODE	Ξ	d		f (FILE #)	
Bit-oriented file register operations 13 10 9 7 6 0 OPCODE b (BIT #) f (FILE #) f f f b = 3-bit bit address f 7 6 0 f	d = 0 for de d = 1 for de f = 7 -bit file	estinati estinati e regis	on W on f ter ad	dres	s	
13 10 9 7 6 0 OPCODE b (BIT #) f (FILE #) f b = 3-bit bit address f 7-bit file register address Literal and control operations General 13 8 7 0 OPCODE k (literal) k (literal) k = 8-bit immediate value CALL and GOTO instructions only 13 11 10 0 OPCODE k (literal) k = 11-bit immediate value	Bit-oriented file	registe	er ope	ratio	ns	
OPCODE b (BIT #) f (FILE #) b = 3-bit bit address f = 7-bit file register address Literal and control operations General 13 8 7 0 OPCODE k (literal) k = 8-bit immediate value CALL and GOTO instructions only 13 11 10 0 OPCODE k (literal) k = 11-bit immediate value	13	10	9	7	6	0
$b = 3-bit bit address f = 7-bit file register address Literal and control operations General 13 8 7 0 OPCODE k (literal) k = 8-bit immediate value CALL and GOTO instructions only 13 11 10 0 OPCODE k (literal) k = 11-bit immediate value }$	OPCODE	Ξ	b (Bl	T #)	f (FILE #)	
13 8 7 0 OPCODE k (literal) k k k = 8-bit immediate value k k k CALL and GOTO instructions only 13 11 10 0 OPCODE k (literal) k k = 11-bit immediate value	General					
OPCODE k (literal) k = 8-bit immediate value CALL and GOTO instructions only 13 11 10 0 OPCODE k (literal) k = 11-bit immediate value	13		8	7		0
k = 8-bit immediate value CALL and GOTO instructions only 13 11 10 0 OPCODE k (literal) k = 11-bit immediate value	OPCODE	Ξ			k (literal)	
13 11 10 0 OPCODE k (literal) k = 11-bit immediate value	k = 8-bit in	imedia	ate val	lue		
13 11 10 0 OPCODE k (literal) k = 11-bit immediate value	CALL and GOTO	instruc	lions	only		
OPCODE k (literal) k = 11-bit immediate value						
k = 11-bit immediate value	13 1	1 10				0
	13 1 OPCODE	1 10		k (literal)	0

PIC16F8X

TABLE 9-2 PIC16FXX INSTRUCTION SET

ADDWF ANDWF CLRF CLRW COMF DECF DECFSZ INCF INCFSZ IORWF MOVE	ds f, d f, d f, d f, d f, d f, d f, d f, d	BYTE-ORIENTED FILE REG Add W and f AND W with f Clear f Clear W Complement f Decrement f Decrement f Increment f	ISTER OPE	MSb RATIC 00 00 00 00 00	0111 0101 0001 0001 1001	dfff dfff lfff 0xxx	LSb ffff ffff xxxx	Affected C,DC,Z Z Z Z	1,2 1,2 2
ADDWF CLRF CLRW COMF DECF DECFSZ INCF INCFSZ IORWF MOVE	f, d f, d f - f, d f, d f, d f, d f, d f, d	BYTE-ORIENTED FILE REG Add W and f AND W with f Clear f Clear W Complement f Decrement f Decrement f, Skip if 0 Increment f	ISTER OPE	RATIC 00 00 00 00 00 00 00 00	0111 0101 0001 0001 1001	dfff dfff lfff 0xxx	ffff ffff ffff xxxx	C,DC,Z Z Z Z	1,2 1,2 2
ADDWF ANDWF CLRF CLRW COMF DECF DECFSZ INCF INCFSZ IORWF MOVE	f, d f, d f - f, d f, d f, d f, d f, d f, d	Add W and f AND W with f Clear f Clear W Complement f Decrement f Decrement f, Skip if 0 Increment f	1 1 1 1 1 1	00 00 00 00 00	0111 0101 0001 0001 1001	dfff dfff lfff 0xxx	ffff ffff ffff xxxx	C,DC,Z Z Z Z	1,2 1,2 2
ANDWF CLRF CLRW COMF DECF DECFSZ INCF INCFSZ IORWF MOVE	f, d f - f, d f, d f, d f, d f, d f, d	AND W with f Clear f Clear W Complement f Decrement f Decrement f, Skip if 0 Increment f	1 1 1 1 1	00 00 00	0101 0001 0001 1001	dfff lfff 0xxx	ffff ffff xxxx	Z Z Z	1,2 2
CLRF CLRW COMF DECF DECFSZ INCF INCFSZ IORWF	f - f, d f, d f, d f, d f, d	Clear f Clear W Complement f Decrement f Decrement f, Skip if 0 Increment f	1 1 1 1(2)	00 00 00	0001 0001 1001	lfff Oxxx	ffff xxxx	Z Z	2
CLRW COMF DECF DECFSZ INCF INCFSZ IORWF	- f, d f, d f, d f, d f, d f, d	Clear W Complement f Decrement f Decrement f, Skip if 0 Increment f	1 1 1 1(2)	00	0001 1001	0xxx	xxxx	Z	
COMF DECF DECFSZ INCF INCFSZ IORWF	f, d f, d f, d f, d f, d f, d	Complement f Decrement f Decrement f, Skip if 0 Increment f	1 1 1(2)	00	1001	2666			
DECF DECFSZ INCF INCFSZ IORWF MOVE	f, d f, d f, d f, d f, d	Decrement f Decrement f, Skip if 0 Increment f	1	0.0		arri	ffff	Z	1,2
DECFSZ INCF INCFSZ IORWF MOVE	f, d f, d f, d f, d	Decrement f, Skip if 0 Increment f	1(2)	00	0011	dfff	ffff	Z	1,2
INCF INCFSZ IORWF MOVE	f,d f,d f.d	Increment f	1(2)	00	1011	dfff	ffff		1,2,3
INCFSZ IORWF MOVE	f,d f.d		1	00	1010	dfff	ffff	Z	1,2
IORWF	f.d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
MOVE	.,	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	С	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	С	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
		BIT-ORIENTED FILE REGIS	TER OPER	RATIO	١S				
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f. b	Bit Test f. Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
		LITERAL AND CONTRO	L OPERAT	IONS					
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
LRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	TO,PD	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk	-	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	TO.PD	
SUBLW	k	Subtract W from literal	1	11	110×	kkkk	kkkk	C.DC.Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTE, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

9.1 Instruction Descriptions

ADDLW	Add Literal and W					
Syntax:	[<i>label</i>] A	DDLW	k			
Operands:	$0 \le k \le 2$	55				
Operation:	(W) + k -	→ (W)				
Status Affected:	C, DC, Z					
Encoding:	11	111x	kkkk	kkkk		
Description:	The conte added to t result is pl	nts of the he eight b aced in th	W register it literal 'k' e W regist	are and the ter.		
Words:	1					
Cycles:	1					
Q Cycle Activity:	Q1	Q2	Q3	Q4		
	Decode	Read literal 'k'	Process data	Write to W		
Example:	ADDLW	0x15				

Before Instruction							
W	=	0x10					
After Instruction							
W	=	0x25					
	_	UNEO					

ADDWF	Add W a	nd f				
Syntax:	[<i>label</i>] A	DDWF	f,d			
Operands:	$\begin{array}{l} 0 \leq f \leq 12 \\ d \in [0,1] \end{array}$	7				
Operation:	(W) + (f)	\rightarrow (desti	nation)			
Status Affected:	C, DC, Z					
Encoding:	00	0111	dfff	ffff		
Description:	Add the co register 'f'. in the W re stored bac	ontents of If 'd' is 0 egister. If k in regis	the W reg the result i 'd' is 1 the ter 'f'.	ister with is stored result is		
Words:	1					
Cycles:	1					
Q Cycle Activity:	Q1	Q2	Q3	Q4		
	Decode	Read register 'f'	Process data	Write to destination		
Example	ADDWF	FSR,	0			
	Before Instruction					
		W =	0x17			
	After Inst	FSK =	UXC2			
	7 4101 11131	W =	0xD9			
		FSR =	0xC2			

ANDLW	AND Lite	eral with	w				
Syntax:	[<i>label</i>] A	NDLW	k				
Operands:	$0 \le k \le 25$	55					
Operation:	(W) .AND	0. (k) \rightarrow ((W)				
Status Affected:	Z						
Encoding:	11	1001	kkkk	kkkk			
Description:	The conter AND'ed wird result is pl	nts of W r ith the eig aced in th	egister are ht bit literal ne W regist	e I 'k'. The ler.			
Words:	1						
Cycles:	1						
Q Cycle Activity:	Q1	Q2	Q3	Q4			
	Decode	Read literal "k"	Process data	Write to W			
Example	ANDLW	0x5F					
	Before In	struction	1				
	W = 0xA3 After Instruction W = 0x03						
ANDWF	AND W v	vith f					
ANDWF Syntax:	AND W v	vith f NDWF	f,d				
ANDWF Syntax: Operands:	AND W $([abel] A = 0 \le f \le 12]$ $d \in [0,1]$	vith f NDWF 27	f,d				
ANDWF Syntax: Operands: Operation:	AND W w [<i>label</i>] Al $0 \le f \le 12$ $d \in [0,1]$ (W) .AND	vith f NDWF 27 0. (f) → (r	f,d destinatio	n)			
ANDWF Syntax: Operands: Operation: Status Affected:	AND W w [<i>label</i>] Al 0 ≤ f ≤ 12 d ∈ [0,1] (W) .AND Z	vith f NDWF 27 0. (f) → (r	f,d destinatio	n)			
ANDWF Syntax: Operands: Operation: Status Affected: Encoding:	AND W v [<i>label</i>] Al $0 \le f \le 12$ $d \in [0,1]$ (W) .ANE Z 00	vith f NDWF 27 0. (f) → (r 0101	f,d destinatio	n) ffff			
ANDWF Syntax: Operands: Operation: Status Affected: Encoding: Description:	AND W w [label] Al $0 \le f \le 12$ $d \in [0,1]$ (W) .ANE Z 00 AND the W is 0 the resister 'f'.	vith f NDWF 7 0. (f) \rightarrow (r 0101 V register sult is sto 1 the res	f,d destination dfff with regist red in the V ult is store	n) ffff er 'f'. If 'd' W regis- d back in			
ANDWF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words:	AND W v [label] Al $0 \le f \le 12$ $d \in [0,1]$ (W) ANE Z 00 AND the W is 0 the re- ter. If d'is register 'f'. 1	vith f NDWF 7 0. (f) \rightarrow (f) 0101 V register sult is sto 1 the res	f,d destinatio dfff with regist red in the V ult is store	n) ffff er 'f. If 'd' W regis- d back in			
ANDWF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles:	AND W v [label] Al $0 \le f \le 12$ $d \in [0,1]$ (W) ANE Z 00 AND the W is 0 the re: ter. If d'is register 'f'. 1	vith f NDWF 7 0. (f) \rightarrow (f) $\sqrt{100}$ v register sult is sto 1 the res	f,d destinatio afff with regist red in the V ult is store	n) ffff er 'f. If 'd' N regis- d back in			
ANDWF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	AND W v [label] Al $0 \le f \le 12$ $d \in [0,1]$ (W) ANE Z 00 AND the W is 0 the re- ter. If d'is register 'f'. 1 1 Q1	vith f NDWF 7 0. (f) \rightarrow (c 0101 V register sult is sto 1 the res Q2	f,d destinatio dfff with regist red in the b ult is store	n) ffff rf.lf'd' N regis- d back in Q4			
ANDWF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	AND W v [labef] Al $0 \le f \le 12$ $d \in [0,1]$ (W) .ANE Z AND the V is 0 the rev ter. If 'd' is register 'f'. 1 2 Q1 Decode	vith f NDWF 7 0. (f) \rightarrow (f) $\sqrt{register}$ 1 the res Q2 Read register T	f,d destination dfff with regist red in the N ult is store Q3 Process data	n) ffff er 'f. If 'd' W regis- d back in Q4 Write to destination			
ANDWF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	AND W v $[label] Al 0 \le f \le 12$ $d \in [0,1]$ (W) AND Z AND the V is 0 the rev ter. If d' is register 'f'. 1 Q1 Decode ANDWF	vith f NDWF 7 0. (f) \rightarrow (f) 0101 V register sult is stor 1 the res Q2 Read register T FSR.	f,d destinatio dfff with regist red in the \ult ult is store Q3 Process data	n) ffff er 'f. If 'd' W regis- d back in Q4 Write to destination			
ANDWF Syntax: Operands: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	AND W v [label] Al $0 \le f \le 12$ $d \in [0,1]$ (W) AND Z AND the V is 0 the rev ter. If d' is register 'f'. 1 Q1 Decode ANDWF Before In	vith f NDWF 7 0. (f) \rightarrow (f) 0101 $\sqrt{register}$ sult is sto 1 the res Q2 Read register $\frac{1}{T}$ FSR,	f,d destinatio dfff with regist red in the V ult is store Q3 Process data	n) ffff er 'f. If 'd' W regis- d back in Q4 Write to destination			

PIC16F8X

BCF	Bit Clear	f			BTFSC
Syntax:	[<i>label</i>] BC	CF f,b			Syntax:
Operands:	$\begin{array}{l} 0 \leq f \leq 12 \\ 0 \leq b \leq 7 \end{array}$	27			Operands
Operation:	$0 \rightarrow (f < b;$	>)			Operation
Status Affected:	None				Status Aff
Encoding:	01	00bb	bfff	ffff	Encoding:
Description:	Bit 'b' in re	gister 'f' is	s cleared.		Descriptio
Words:	1				
Cycles:	1				
Q Cycle Activity:	Q1	Q2	Q3	Q4	
	Decode	Read register 'f'	Process data	Write register 'f'	Words: Cycles:
Example	BCF	FLAG	REG, 7		Q Cycle A
	Before In After Inst	struction FLAG_RE ruction	EG = 0xC7	•	ľ
BSF	Bit Set f				
Syntax:	[<i>label</i>] BS	SF f,b			
Operands:	$0 \le f \le 12$ $0 \le b \le 7$	7			
Operation:	$1 \rightarrow (f < b;$	>)			
Status Affected:	None				
Encoding:	01	01bb	bfff	ffff	
Description:	Bit 'b' in re	gister 'f' is	s set.		
Words:	1				
Cycles:	1				
Q Cycle Activity:	Q1	Q2	Q3	Q4	
	Decode	Read register 'f'	Process data	Write register 'f'	
Example	BSF	FLAG_F	REG, 7		
	Before In	struction) =G = 0×0	۵	
	After Inst	ruction	_0 = 0.00	7	
		FLAG RE	EG = 0x8/	4	

	Bit Test,	Skip if Cl	ear				
	[<i>label</i>] BT	FSC f,b					
ds:	$\begin{array}{l} 0 \leq f \leq 12 \\ 0 \leq b \leq 7 \end{array}$	7					
on:	skip if (f<	b>) = 0					
ffected:	None						
g:	01	10bb	bfff	ffff			
tion:	If bit 'b' in register 'f' is '1' then the next instruction is executed. If bit 'b', in register 'f', is '0' then the next instruction is discarded, and a NOP is executed instead, making this a 2TcY instruction.						
	1						
	1(2)						
Activity:	Q1	Q2	Q3	Q4			
	Decode	Read register 'f'	Process data	No-Operat ion			
If Skip:	(2nd Cyc	le)					
	Q1	Q2	Q3	Q4			
	No-Operat ion	No-Operati on	No-Opera tion	No-Operat ion			
e	HERE FALSE TRUE	BTFSC GOTO •	FLAG,1 PROCESS_	_CODE			
	Before In	struction PC = a	ddress H	ERE			
	After Inst	ruction					
	i	if FLAG<1>	= 0,	שווס			
		if FLAG<1>	=1.	NUL			

W = 0x17

FSR = 0x02

After Instruction

PC = address FALSE

BTFSS	Bit Test	Bit Test f, Skip if Set CALL			CALL				
Syntax:	[<i>label</i>] BT	FSS f,b			Syntax:				
Operands:	$0 \le f \le 12$	$0 \le f \le 127$							
	0 ≤ b < 7				Operation:				
Operation:	skip if (f<	b>) = 1							
Status Affected:	None								
Encoding:	01	11bb	bfff	ffff	Status Affect				
Description:	If bit 'b' in instruction If bit 'b' is ' discarded instead, m	to ti 'b' in register 't' is '0' then the next nstruction is executed. I bit 'b' is '1', then the next instruction is discarded and a NOP is executed nstead, making this a ZTCY instruction.							
Words:	1								
Cycles:	1(2)				Words:				
Q Cycle Activity:	Q1	Q2	Q3	Q4	Cycles:				
	Decode	Read register 'f'	Process data	No-Operat ion	Q Cycle Act				
If Skip:	(2nd Cyc	le)	1st						
	Q1	Q2	Q3	Q4					
	No-Operat ion	No-Operati on	No-Opera tion	No-Operat ion	2nd				
Example	HERE FALSE TRUE	BTFSC GOTO •	FLAG,1 PROCESS	_CODE	Example				
	Before In	struction							
	After Inct	PC = a	ddress H	IERE					
	Anerinsi	if FLAG<1>	= 0.						
		PC =	address Fi	ALSE					

if FLAG<1> = 1, PC = address TRUE

	[label]	CALL k	[
:	$0 \le k \le 2047$								
:	$\begin{array}{l} (PC)+1 \rightarrow TOS, \\ k \rightarrow PC<10:0>, \\ (PCLATH<4:3>) \rightarrow PC<12:11> \end{array}$								
ected:	None								
:	10	0kkk	kkkk	kkkk					
on:	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 is a two cycle instruction.								
Activity:	Q1	Q2	Q3	Q4					
st Cycle	Decode	Read literal 'k', Push PC to Stack	Process data	Write to PC					
nd Cycle	No-Opera tion	No-Opera tion	No-Opera tion	No-Operat ion					
	HERE	CALL	THERE						

Call Subroutine

Before Instruction PC = Address HERE

After Instruction

PC = Address THERE TOS = Address HERE+1

PIC16F8X

CLRF	Clear f				CLRW	Clear W	
Syntax:	[<i>label</i>] C	LRF f			Syntax:	[label]	CLRW
Operands:	$0 \le f \le 12$	27			Operands:	None	
Operation:	$00h \rightarrow (f + Z)$ 1 $\rightarrow Z$)			Operation:	$00h \rightarrow (1)$ 1 $\rightarrow Z$	W)
Status Affected:	Z				Status Affected:	z	
Encoding:	00	0001	lfff	ffff	Encodina:	0.0	0001
Description:	The conte and the Z	nts of regi bit is set.	ister 'f' are	cleared	Description:	W registe set.	r is cleare
Words:	1				Words:	1	
Cycles:	1				Cvcles:	1	
Q Cycle Activity:	Q1	Q2	Q3	Q4	Q Cycle Activity:	Q1	Q2
	Decode	Read register 'f'	Process data	Write register 'f'		Decode	No-Opera tion
Example	CLRF	FLAC	G_REG	11	Example	CLRW	
	Before In	struction	1			Before In	nstruction
	After Inst	FLAG_RE ruction FLAG_RE	EG =	0x5A 0x00		After Ins	truction W =
		Ζ	=	1			Z =

	Clear W			
	[label]	CLRW		
ds:	None			
on:	$\begin{array}{c} 00h \rightarrow (V \\ 1 \rightarrow Z \end{array}$	V)		
Affected:	Z			
g:	00	0001	0xxx	xxxx
tion:	W register set.	is cleared	. Zero bit (Z) is
	1			
	1			
Activity:	Q1	Q2	Q3	Q4
	Decode	No-Opera tion	Process data	Write to W
e	CLRW			
	Before In	struction W =	0x5A	

W = 0x00Z = 1

	<u>.</u>					
CLRWDI	Clear Watchdog Timer					
Syntax:	[label] CLRWDT					
Operands:	None					
Operation:	$\begin{array}{l} 00h \rightarrow WDT \\ 0 \rightarrow WDT \ prescaler, \\ 1 \rightarrow \overline{TO} \\ 1 \rightarrow \overline{PD} \end{array}$					
Status Affected:	TO, PD					
Encoding:	0 0	0000	0110	0100		
Description:	CLRWDT in dog Timer of the WD set.	struction r It also res T. Status b	esets the set <u>s th</u> e pr its TO and	Watch- escaler I PD are		
Words:	1					
Cycles:	1					
Q Cycle Activity:	Q1	Q2	Q3	Q4		
	Decode	No-Opera tion	Process data	Clear WDT Counter		
Example	CLRWDT					
	Before In	struction				
	After loot	WDT cour	iter =	?		
	Aner Inst	WDT cour	ter =	0x00		
		WDT pres	caler=	0		
		=				

TO PD = 1 = 1

COMF	Complement f	DECFSZ
Syntax:	[label] COMF f,d	Syntax:
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]	Operands:
Operation:	$(\bar{f}) \rightarrow (destination)$	Operation:
Status Affected:	Z	
Encoding:	00 1001 dfff ffff	Status Affec
Description:	The contents of register 'f' are comple- mented. If 'd' is 0 the result is stored in W. If 'd' is 1 the result is stored back in register 'f'.	Encoding: Description:
Words:	1	
Cycles:	1	
Q Cycle Activity:	Q1 Q2 Q3 Q4	
	Decode Read Process Write to	Words:
		Cycles:
Example	COMF REG1,0	Q Cycle Acti
	Before Instruction	
	REG1 = 0x13	If S
	After Instruction REG1 = 0x13	
	W = 0xEC	
DECF	Decrement f	
Syntax:	[label] DECF f,d	Example
Syntax: Operands:	[<i>label</i>] DECF f,d 0 ≤ f ≤ 127	Example
Syntax: Operands:	[<i>label</i>] DECF f,d $0 \le f \le 127$ $d \in [0,1]$	Example
Syntax: Operands: Operation:	[<i>label</i>] DECF f,d 0 ≤ f ≤ 127 d ∈ [0,1] (f) - 1 → (destination)	Example
Syntax: Operands: Operation: Status Affected:	[<i>label</i>] DECF f,d $0 \le f \le 127$ $d \in [0,1]$ (f) - 1 → (destination) Z	Example
Syntax: Operands: Operation: Status Affected: Encoding:	[<i>label</i>] DECF f,d $0 \le f \le 127$ $d \in [0,1]$ (f) - 1 \rightarrow (destination) Z 00 0011 dfff ffff	Example
Syntax: Operands: Operation: Status Affected: Encoding: Description:		Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words:	[<i>label</i>] DECF f,d $0 \le f \le 127$ $d \in [0,1]$ (f) - 1 \rightarrow (destination) Z Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. 1	Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles:	[<i>label</i>] DECF f,d $0 \le f \le 127$ $d \in [0,1]$ (f) - 1 \rightarrow (destination) Z Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. 1	Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:	[<i>label</i>] DECF f,d $0 \le f \le 127$ $d \in [0,1]$ (f) - 1 \rightarrow (destination) Z Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. 1 1 Q1 Q2 Q3 Q4	Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity:		Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity: Example	[<i>label</i>] DECF f,d 0 ≤ f ≤ 127 d ∈ [0,1] (f) - 1 → (destination) Z 00 0011 dfff ffff Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. 1 1 2 Q1 Q2 Q3 Q4 Decode Read Process Write to data destination T 1 DECF CNT, 1	Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity: Example	[<i>label</i>] DECF f,d 0 ≤ f ≤ 127 d ∈ [0,1] (f) - 1 → (destination) Z 00 0011 dfff ffff Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. 1 1 2 Q1 Q2 Q3 Q4 Decode Read Process Write to data destination DECF CNT, 1 Before Instruction	Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity: Example	[<i>label</i>] DECF f,d 0 ≤ f ≤ 127 d ∈ [0,1] (f) - 1 → (destination) Z 00 0011 dfff ffff Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. 1 1 2 Q1 Q2 Q3 Q4 Decode Read Process Write to data destination DECF CNT, 1 Before Instruction CNT = 0x01	Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity: Example	[<i>label</i>] DECF f,d 0 ≤ f ≤ 127 d ∈ [0,1] (f) - 1 → (destination) Z 00 0011 dfff ffff Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. 1 1 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Decode Read Process Write to data destination DECF CNT, 1 Before Instruction CNT = 0x01 Z = 0 After Instruction	Example
Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Q Cycle Activity: Example		Example

FSZ	Decrement f, Skip if 0							
ax:	[label]	DECFSZ	f,d					
ands:	$0 \le f \le 127$ $d \in [0,1]$							
ation:	(f) - 1 \rightarrow (destination); skip if result = 0							
s Affected:	None							
ding:	00	1011	dfff	ffff				
ription:	The conten mented. If ' W register. back in regi If the result executed. If executed in tion.	ts of registe d' is 0 the r If 'd' is 1 th ister 'f'. is 1, the ne the result stead make	er 'f' are de esult is plac e result is p ext instructi is 0, then a ing it a 2Tc	cre- ced in the blaced on, is NOP is Y instruc-				
s:	1							
S:	1(2)							
cle Activity:	Q1	Q2	Q3	Q4				
	Decode	Read register 'f'	Process data	Write to destination				
If Skip:	(2nd Cyc	le)						
	Q1	Q2	Q3	Q4				
	No-Operat ion	No-Opera tion	No-Operat ion	No-Operati on				
ple	HERE	DECFS GOTO	Z CNT, LOOF	1				
	CONTINU	JE •						

Before Instruction

PC	=	address HERE
After Inst	ructio	on
CNT	=	CNT - 1
if CNT	=	0,
PC	=	address CONTINUE
if CNT	≠	0,
PC	=	address HERE+1

PIC16F8X

GOTO	Uncondi	tional Br	anch		INCF		Increme	nt f		
Syntax:	[label]	GOTO	k		Synta	ax:	[label]	INCF	f,d	
Operands:	$0 \le k \le 20$	047			Opera	ands:	$0 \le f \le 12$	27		
Operation:	$k \rightarrow PC < PC < PCLATH$:10:0> <4:3> →	PC<12:11	1>	Opera	ation:	d ∈ [0,1] (f) + 1 →	(destina	ation)	
Status Affected:	None				Statu	s Affected:	Z			
Encoding:	10	lkkk	kkkk	kkkk	Enco	ding:	00	1010	dfff	ffff
Description:	GOTO is an eleven bit into PC bit PC are loa GOTO is a	n unconditi immediate ts <10:0>. aded from two cycle i	onal brand value is lo The upper PCLATH </td <td>ch. The baded bits of 4:3>.</td> <td>Desc</td> <td>ription:</td> <td>The conte mented. In the W reg placed ba</td> <td>ents of reg i 'd' is 0 th ister. If 'd' ck in regis</td> <td>jister 'f' ar le result is ' is 1 the r ster 'f'.</td> <td>e incre- s placed in esult is</td>	ch. The baded bits of 4:3>.	Desc	ription:	The conte mented. In the W reg placed ba	ents of reg i 'd' is 0 th ister. If 'd' ck in regis	jister 'f' ar le result is ' is 1 the r ster 'f'.	e incre- s placed in esult is
Words:	1				Word	s:	1			
Cycles:	2				Cycle	es:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4	Q Cy	cle 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	Exam	nple	INCF	CNT ,	1	
							Before Ir	structior	ı	
Example	GOTO T	HERE						CNT	= 0xF	F
	After Inst	ruction PC =	Address	THERE			After Ins	∠ truction CNT	= 0 = 0x0	00

Z = 1

INCFSZ	Increment f, Skip if 0				IORLW
Syntax:	[label]	INCFSZ	f,d		Syntax:
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]				Operands: Operation:
Operation:	(f) + 1 \rightarrow (destination), skip if result = 0				Status Affected
Status Affected:	None				Encoding:
Encoding:	00	1111	dfff	ffff	Description:
Description:	The conte mented. If the W regi placed bac If the resul executed. cuted inste tion.	nts of regis 'd' is 0 the ster. If 'd' i ck in regist t is 1, the If the resu ead making	ster 'f' are e result is p s 1 the res ter 'f'. next instru It is 0, a No g it a 2TCY	incre- blaced in sult is oction is OP is exe- r instruc-	Words: Cycles: Q Cycle Activity
Words:	1				
Cycles:	1(2)				Example
Q Cycle Activity:	Q1	Q2	Q3	Q4	Example
	Decode	Read register 'f'	Process data	Write to destination	
If Skip:	(2nd Cyc	le)			
	Q1	Q2	Q3	Q4	
	No-Operat ion	No-Opera tion	No-Opera tion	No-Operati on	
Example	HERE CONTIN	INCFS GOTO JE •	Z CI LO	NT, 1 OP	
	Before In PC After Inst CNT if CNT PC if CNT PC	struction = add ruction = CN = 0, = add ≠ 0, = add	ress HERE [+ 1 ress CONT ress HERE	INUE +1	

W	Inclusive	Inclusive OR Literal with W						
x:	[label]	IORLW	k					
ands:	$0 \le k \le 2$	$0 \le k \le 255$						
ation:	(W) .OR.	(W) .OR. $k \rightarrow$ (W)						
s Affected:	Z							
ding:	11	1000	kkkk	kkkk				
iption:	The conter OR'ed with result is pl	The contents of the W register is OR'ed with the eight bit literal 'k'. The result is placed in the W register.						
s:	1							
s:	1							
cle Activity:	Q1	Q2	Q3	Q4				
	Decode	Read literal 'k'	Process data	Write to W				
	TODIW	0x35						
ple	TOKTW	011000						
ple	Before In	struction	1					
ple	Before In	struction W =	0x9A					
ple	Before In After Inst	struction W = ruction W =	0x9A 0xBF					

PIC16F8X

IORWF	Inclusive OR W with	f MOVLW
Syntax:	[label] IORWF f,d	Svntax:
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]	Operand
Operation:	(W) OR (f) \rightarrow (destination)	Operation)
Status Affected	7	Status A
Encoding:	00100dff	f ffff Encodin
Description:	Inclusive OR the W regist ter 'f'. If 'd' is 0 the result is W register. If 'd' is 1 the re back in register 'f'.	ter with regis- s placed in the soult is placed Words:
Words:	1	Cycles:
Cycles:	1	Q Cycle
Q Cycle Activity:	Q1 Q2 Q	3 Q4
	Decode Read Proc	ess Write to
	register dat 'f'	ta destination Example
Example	TORWE RESU	τ.π. Ο
NOVE	After Instruction RESULT = W = Z =	0x13 0x93 1 MOVWF
MOVE		Syntax:
Syntax:	[label] MOVF f,d	Operano
Operands:	$0 \le t \le 127$ d = [0,1]	Operatio
Operation:	(f) \rightarrow (destination)	Status A
Status Affected:	Z	Encodin
Encoding:	00 1000 dff	f ffff Descript
Description:	The contents of register f destination dependant up of d. If $d = 0$, destination is fil itself, $d = 1$ is useful to test ter since status flag Z is a	is moved to a on the status Wregister. If le register f st a file regis- fiferted
Words:	1	
Cycles:	1	
Q Cycle Activity:	Q1 Q2 Q	3 Q4 Example
	Decode Read register dat	ta Write to destination
Example	MOVF FSR, 0	

After Instruction W = value in FSR register Z = 1

I	Move Literal to W						
	[label]	MOVLW	/ k				
ds:	$0 \le k \le 2\xi$	55					
on:	$k \to (W)$						
Affected:	None						
ig:	11	00xx	kkkk	kkkk			
tion:	The eight I register. Th as 0's.	bit literal 'l he don't c	k' is loaded ares will as	l into W semble			
	1						
	1						
Activity:	Q1	Q2	Q3	Q4			
	Decode	Read literal 'k'	Process data	Write to W			
е	MOVLW	0x5A					

After Instruction W = 0x5A

MOVWF	Move W to f
Syntax:	[label] MOVWF f
Operands:	$0 \le f \le 127$
Operation:	$(W) \rightarrow (f)$
Status Affected:	None
Encoding:	00 0000 lfff ffff
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Q Cycle Activity:	Q1 Q2 Q3 Q4
	Decode Read register 'f' Process Write register 'f
Example	MOVWF OPTION_REG
	Before Instruction OPTION = 0xFF W = 0x4F
	After Instruction OPTION = 0x4F W = 0x4F

NOP	No Oper	ation			RETFIE
Syntax:	[label]	NOP			Syntax:
Operands:	None				Operan
Operation:	No opera	ation			Operatio
Status Affected:	None				
Encoding:	00	0000	0xx0	0000	Status A
Description:	No operat	tion.			Encodin
Words:	1				Descrip
Cycles:	1				
Q Cycle Activity:	Q1	Q2	Q3	Q4	
	Decode	No-Opera tion	No-Opera tion	No-Operat ion	
	1	1			Words:
Example	NOP				Cycles:

RETFIE	Return fi	rom Inter	rupt		
Syntax:	[label]	RETFIE			
Operands:	None				
Operation:	$\begin{array}{l} TOS \rightarrow F \\ 1 \rightarrow GIE \end{array}$	PC,			
Status Affected:	None				
Encoding:	00	0000	0000	1001	
Description:	Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two cycle instruction.				
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				
	After Inte	rrupt			
		PC =	TOS 1		

PIC16F8X

RETLW	Return with Literal in W RETU					
Syntax:	[label] RETLW k Syntax					
Operands:	$0 \le k \le 25$	55			Opera	
Operation:	$\begin{array}{l} k \rightarrow (W); \\ TOS \rightarrow F \end{array}$	с			Opera Status	
Status Affected:	None				Encoc	
Encoding:	11	01xx	kkkk	kkkk	Descr	
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instructions works					
Words:	1				Cycles	
Cycles:	2				Q Cyc	
Q Cycle Activity:	Q1	Q2	Q3	Q4		
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-Operat ion	Examp	
Example	CALL TABLE ADDWF PC RETLW k1 RETLW k2 RETLW kn Before In	<pre>% cont ;offset ;W now ;W = off ;Begin t ; ; End of struction W -</pre>	ains tabl : value : has table Eset .able : table	e e value		
	After Inst	vv = ruction W =	ux07 value of kt	3		

JRN	Return fr	om Sub	routine	
x:	[label]	RETURI	N	
ands:	None			
ation:	$TOS\toP$	C		
s Affected:	None			
ding:	00	0000	0000	1000
ription:	Return from POPed an is loaded in is a two cy	m subrouti d the top o nto the pro cle instruc	ine. The st of the stack gram coun ction.	ack is (TOS) iter. This
s:	1			
s:	2			
cle Activity:	Q1	Q2	Q3	Q4
1st Cycle	Decode	No-Opera tion	No-Opera tion	Pop from the Stack
2nd Cycle	No-Operat ion	No-Opera tion	No-Opera tion	No-Opera tion
ple	RETURN			

After Interrupt

OPTION	Load Option Register				
Syntax:	[label]	OPTION	٧		
Operands:	None				
Operation:	$(W)\toO$	PTION			
Status Affected:	None				
Encoding:	00	0000	0110	0010	
Description:	The conter loaded in t instruction patibility w Since OPT register, th it.	nts of the he OPTIC is support ith PIC16 FION is a he user ca	W register DN register rted for coo C5X produ readable/v n directly a	are This de com- icts. vritable address	
Words:	1				
Cycles:	1				
Example					
	To maintain upward compatibility with future PIC16CXX products, do not use this instruction.				

© 1998 Microchip Technology Inc.

RLF	Rotate Left f through Carry				RRF
Syntax:	[label]		RLF f,	d	Syntax:
Operands:	$\begin{array}{l} 0 \leq f \leq 12 \\ d \in \ [0,1] \end{array}$	7			Operands:
Operation:	See desc	ription b	elow		Operation:
Status Affected:	С				Status Affected:
Encoding:	00	1101	dfff	ffff	Encoding:
Description:	The conter one bit to t Flag. If 'd' i W register. back in reg	nts of reg he left thi s 0 the re . If 'd' is 1 jister 'f'.	ister 'f' are rough the esult is plac the result Register f	Totated Carry ced in the is stored	Description:
Words:	1				Words:
Cycles:	1				Cycles:
Q Cycle Activity:	Q1	Q2	Q3	Q4	Q Cycle Activity:
	Decode	Read register 'f'	Process data	Write to destination	
Example	RLF	REG	31,0		Example
	Before Instruction				
	I	REG1	= 111	0 0110	
	(^ { + + + + + + - +	C	= 0		
	Alter Insti	REG1	- 111	0 0110	
	1	W	= 110	0 1100	
	(С	= 1		

	Rotate R	ight f th	rough C	arry
	[label]	RRF f,	d	
	$0 \le f \le 12$	7		
	$d\in [0,1]$			
	See desc	ription b	elow	
ed:	С			
	00	1100	dfff	ffff
	one bit to t Flag. If 'd' W register back in reg	the right t is 0 the re . If 'd' is 1 gister 'f'.	hrough th esult is pla the result Register t	e Carry iced in the t is placed
	1			
	1			
ity:	Q1	Q2	Q3	Q4
	Decode	Read register 'f'	Process data	Write to destination
	RRF		REG1,0	
	Before In	struction	1	
		REG1	= 111	.0 0110
	After Inst	ruction	= 0	
		REG1	= 111	.0 0110
		W	= 011	.1 0011
		U	= 0	

PIC16F8X

SLEEP		SUBLW	Subtract W from Literal
Syntax:	[label] SLEEP	Syntax:	[label] SUBLW k
Operands:	None	Operands:	$0 \le k \le 255$
Operation:	$00h \rightarrow WDT$,	Operation:	$k \text{ - } (W) \to (W)$
	$0 \rightarrow WDT$ prescaler,	Status Affected:	C, DC, Z
	$1 \rightarrow IO,$ $0 \rightarrow \overline{PD}$	Encoding:	11 110x kkkk kkkk
Status Affected: Encoding:	TO, PD 00 0000 0110 0011	Description:	The W register is subtracted (2's comple- ment method) from the eight bit literal 'k'. The result is placed in the W register.
Description:	The power-down status bit, PD is	Words:	1
·	cleared. Time-out status bit, TO is	Cycles:	1
	caler are cleared.	Q Cycle Activity:	Q1 Q2 Q3 Q4
The processor is put into SLEEP mode with the oscillator stopped. See Section 14.8 for more details.	The processor is put into SLEEP mode with the oscillator stopped. See Section 14.8 for more details.		Decode Read Process Write to W data
Words:	1	Example 1:	SUBLW 0x02
Cycles:	1		Before Instruction
Q Cycle Activity:	Q1 Q2 Q3 Q4 Decode No-Opera tion No-Opera Sleep Sleep		W = 1 C = ? Z = ?
Example:	CI FFD		After Instruction
Example.			W = 1 C = 1; result is positive Z = 0
		Example 2:	Before Instruction
			W = 2 C = ? Z = ?
			After Instruction
			W = 0 C = 1; result is zero Z = 1
		Example 3:	Before Instruction
			W = 3 C = ? Z = ?
			After Instruction
			W = 0xFF C = 0; result is negative Z = 0

Swap Nibbles in f

SUBWF	Subtract W f	rom f	f		SWA	PF
Syntax:	[label] SI	JBWF	f,d		Synta	ax:
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]				Oper	ands:
Operation:	(f) - (W) \rightarrow (d	estina	ation)		Oper	ation:
Status Affected:	C, DC, Z					
Encoding:	00 0	010	dfff	ffff	Statu	s Affect
Description:	Subtract (2's co ister from regis stored in the W result is stored	Subtract (2's complement method) W reg- ster from register 'f'. If 'd' is 0 the result is tored in the W register. If 'd' is 1 the esult is stored back in register 'f'.				
Words:	1					
Cycles:	1				Word	S:
Q Cycle Activity:	Q1	Q2	Q3	Q4	Cycle	S:
	Decode F reg	Read jister 'f'	Process data	Write to destination	Q Cy	cle Activ
Example 1:	SUBWF		REG1,1			
	Before Instru	ction			Exam	ıple
	REG1	=	3			
	W	=	2			
	z	=	?			
	After Instruct	ion				
	REG1	=	1			
	w	=	2 1. recult is	n o o i ti vo		
	z	=	0	positive		
Example 2:	Before Instru	ction			TRI	s
	REG1	=	2		Syr	ntax:
	w	=	2		Ope	erands:
	z	=	?		Op	eration:
	After Instruct	ion			Sta	tus Affe
	REG1	=	0		End	coding:
	W	=	2		Des	scriptior
	C Z	=	1; result is 1	zero		
Example 3:	- Before Instru	ction				
	REG1	=	1			
	W	=	2		Wo	rds:
	C	=	?		Сус	les:
	∠ After Instruct	= ion	t		Exa	ample
	RFG1	=	0xFF			
	W	=	2			
	C	=	0; result is	negative		
	Ζ	=	U			

	[label]	SWAPF 1	f,d				
ds:	$0 \le f \le 127$ $d \in [0,1]$						
on:	(f<3:0>) - (f<7:4>) -	ightarrow (destin $ ightarrow$ (destin	ation<7:4 ation<3:0	4>), 0>)			
Affected:	None						
g:	00	1110	dfff	ffff			
tion:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.						
	1						
	1						
Activity:	Q1	Q2	Q3	Q4			
	Decode	Read register 'f'	Process data	Write to destination			
e	SWAPF	REG,	0				
	Before In	struction					
		REG1	= 0x4	45			
	After Inst	ruction					
		REG1	= 0xA	45			
		W	= 0x5	5A			
	Load IR	IS Regis	ter				
K:	[Iabel]	IRIS I	ſ				
nas:	1212C						
tion:	$(VV) \rightarrow II$	RIS regis	ter f;				
Affected:	None						
ling:	00	0000	0110	Offf			
iption:	I he instruction is supported for code compatibility with the PIC16C5X prod- ucts. Since TRIS registers are read- able and writable, the user can directly address them.						
5:	1						
s:	1						
ple							
ple	To mair with fu do n	ntain upwa Iture PIC1 Iot use thi	ard comp 6CXX pro s instruct	atibility ducts, tion.			

PIC16F8X

XORLW	Exclusive OR Literal with W	XORWF	Exclusive OR W with f
Syntax:	[<i>label</i>] XORLW k	Syntax:	[label] XORWF f,d
Operands:	$0 \le k \le 255$	Operands:	0 ≤ f ≤ 127 d ∈ [0.1]
Operation: Status Affected: Encoding: Description:	(W) .XOR. $k \rightarrow$ (W) Z 11 1010 kkkk kkkk The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W regis-	Operation: Status Affected: Encoding: Description:	
Words: Cycles: Q Cycle Activity:	ter. 1 1 Q1 Q2 Q3 Q4	Words: Cycles:	result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. 1 01 02 03 04
	Decode Read Process Write to literal 'k' data W	Q Oyoic Activity.	Decode Read Process Write to destination
Example:	XORLW 0xAF Before Instruction W = 0xB5	Example	XORWF REG 1 Before Instruction
	After Instruction W = 0x1A		$\begin{array}{rcl} REG &=& 0xAF\\ W &=& 0xB5\\ \end{array}$ After Instruction

REG = 0x1A W = 0xB5