B.5 12-Bit Core Instruction Set

Microchip's base-line 8-bit microcontroller family uses a 12-bit wide instruction set. All instructions execute in a single instruction cycle unless otherwise noted. Any unused opcode is executed as a NOP. The instruction set is grouped into the following catagories:

Table B.5: 12-Bit Core Literal and Control Operations

Hex	Mnemonic		Description	Function
Ekk	ANDLW	k	AND literal and W	k .AND. W → W
9kk	CALL	k	Call subroutine	PC + 1 → TOS, k → PC
004	CLRWDT		Clear watchdog timer	0 → WDT (and Prescaler if assigned)
Akk	GOTO	k	Goto address (k is nine bits)	k → PC(9 bits)
Dkk	IORLW	k	Incl. OR literal and W	k .OR. W → W
Ckk	MOVLW	k	Move Literal to W	k → W
002	OPTION		Load OPTION Register	W → OPTION Register
8kk	RETLW	k	Return with literal in W	k → W, TOS → PC
003	SLEEP		Go into Standby Mode	0 → WDT, stop oscillator
00f	TRIS	f	Tristate port f	W → I/O control reg f
Fkk	XORLW	k	Exclusive OR literal and W	k .XOR. W → W

Table B.6: 12-Bit Core Byte Oriented File Register Operations

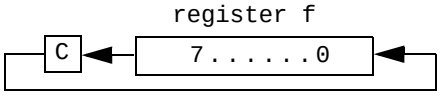
Hex	Mnemonic		Description	Function
1Cf	ADDWF	f,d	Add W and f	W + f → d
14f	ANDWF	f,d	AND W and f	W .AND. f → d
06f	CLRF	f	Clear f	0 → f
040	CLRW		Clear W	0 → W
24f	COMF	f,d	Complement f	.NOT. f → d
0Cf	DECF	f,d	Decrement f	f - 1 → d
2Cf	DECFSZ	f,d	Decrement f, skip if zero	f - 1 → d, skip if zero
28f	INCF	f,d	Increment f	f + 1 → d
3Cf	INCFSZ	f,d	Increment f, skip if zero	f + 1 → d,skip if zero
10f	IORWF	f,d	Inclusive OR W and f	W .OR. f → d
20f	MOVF	f,d	Move f	f → d
02f	MOVWF	f	Move W to f	W → f
000	NOP		No operation	
34f	RLF	f,d	Rotate left f	

Table B.6: 12-Bit Core Byte Oriented File Register Operations (Continued)

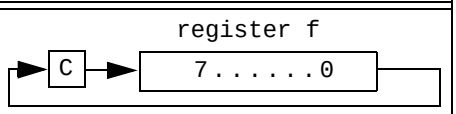
Hex	Mnemonic		Description	Function
30f	RRF	f,d	Rotate right f	
08f	SUBWF	f,d	Subtract W from f	f - W → d
38f	SWAPF	f,d	Swap halves f	f(0:3) ↔ f(4:7) → d
18f	XORWF	f,d	Exclusive OR W and f	W .XOR. f → d

Table B.7: 12-Bit Core Bit Oriented File Register Operations

Hex	Mnemonic		Description	Function
4bf	BCF	f,b	Bit clear f	0 → f(b)
5bf	BSF	f,b	Bit set f	1 → f(b)
6bf	BTFSC	f,b	Bit test, skip if clear	skip if f(b) = 0
7bf	BTFSS	f,b	Bit test, skip if set	skip if f(b) = 1

B.6 14-Bit Core Instruction Set

Microchip's mid-range 8-bit microcontroller family uses a 14-bit wide instruction set. This instruction set consists of 36 instructions, each a single 14-bit wide word. Most instructions operate on a file register, f, and the working register, W (accumulator). The result can be directed either to the file register or the W register or to both in the case of some instructions. A few instructions operate solely on a file register (BSF for example). The instruction set is grouped into the following catagories:

Table B.8: 14-Bit Core Literal and Control Operations

Hex	Mnemonic		Description	Function
3Ekk	ADDLW	k	Add literal to W	k + W → W
39kk	ANDLW	k	AND literal and W	k .AND. W → W
2kkk	CALL	k	Call subroutine	PC + 1 → TOS, k → PC
0064	CLRWDT	T	Clear watchdog timer	0 → WDT (and Prescaler if assigned)
2kkk	GOTO	k	Goto address (k is nine bits)	k → PC(9 bits)
38kk	IORLW	k	Incl. OR literal and W	k .OR. W → W
30kk	MOVLW	k	Move Literal to W	k → W
0062	OPTION		Load OPTION register	W → OPTION Register
0009	RETFIE		Return from Interrupt	TOS → PC, 1 → GIE
34kk	RETLW	k	Return with literal in W	k → W, TOS → PC
0008	RETURN		Return from subroutine	TOS → PC
0063	SLEEP		Go into Standby Mode	0 → WDT, stop oscillator
3Ckk	SUBLW	k	Subtract W from literal	k - W → W
006f	TRIS	f	Tristate port f	W → I/O control reg f
3Akk	XORLW	k	Exclusive OR literal and W	k .XOR. W → W

Table B.9: 14-Bit Core Byte Oriented File Register Operations

Hex	Mnemonic		Description	Function
07ff	ADDWF	f,d	Add W and f	W + f → d
05ff	ANDWF	f,d	AND W and f	W .AND. f → d
018f	CLRF	f	Clear f	0 → f
0100	CLRW		Clear W	0 → W
09ff	COMF	f,d	Complement f	.NOT. f → d
03ff	DECF	f,d	Decrement f	f - 1 → d
0Bff	DECFSZ	f,d	Decrement f, skip if zero	f - 1 → d, skip if 0
0Aff	INCF	f,d	Increment f	f + 1 → d
0Fff	INCFSZ	f,d	Increment f, skip if zero	f + 1 → d, skip if 0
04ff	IORWF	f,d	Inclusive OR W and f	W .OR. f → d
08ff	MOVF	f,d	Move f	f → d

Table B.9: 14-Bit Core Byte Oriented File Register Operations (Continued)

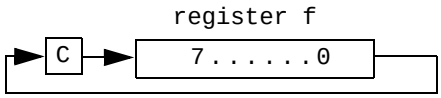
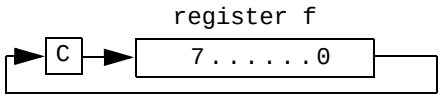
Hex	Mnemonic		Description	Function
008f	MOVWF	f	Move W to f	W → f
0000	NOP		No operation	
0Dff	RLF	f,d	Rotate left f	
0Cff	RRF	f,d	Rotate right f	
02ff	SUBWF	f,d	Subtract W from f	f - W → d
0Eff	SWAPF	f,d	Swap halves f	f(0:3) ↔ f(4:7) → d
06ff	XORWF	f,d	Exclusive OR W and f	W .XOR. f → d

Table B.10: 14-Bit Core Bit Oriented File Register Operations

Hex	Mnemonic		Description	Function
1bff	BCF	f,b	Bit clear f	0 → f(b)
1bff	BSF	f,b	Bit set f	1 → f(b)
1bff	BTFSC	f,b	Bit test, skip if clear	skip if f(b) = 0
1bff	BTFSS	f,b	Bit test, skip if set	skip if f(b) = 1

Table B.11: 12-Bit/14-Bit Core Special Instruction Mnemonics

Mnemonic		Description	Equivalent Operation(s)		Status
ADDCF	f,d	Add Carry to File	BTFSC INCF	3,0 f,d	Z
ADDDCF	f,d	Add Digit Carry to File	BTFSC INCF	3,1 f,d	Z
B	k	Branch	GOTO	k	-
BC	k	Branch on Carry	BTFSC GOTO	3,0 k	-
BDC	k	Branch on Digit Carry	BTFSC GOTO	3,1 k	-
BNC	k	Branch on No Carry	BTFSS GOTO	3,0 k	-
BNDC	k	Branch on No Digit Carry	BTFSS GOTO	3,1 k	-
BNZ	k	Branch on No Zero	BTFSS GOTO	3,2 k	-
BZ	k	Branch on Zero	BTFSC GOTO	3,2 k	-

Table B.11: 12-Bit/14-Bit Core Special Instruction Mnemonics (Continued)

Mnemonic	Description	Equivalent Operation(s)	Status
CLRC	Clear Carry	BCF 3,0	-
CLRDC	Clear Digit Carry	BCF 3,1	-
CLRZ	Clear Zero	BCF 3,2	-
LCALL k			
LGOTO k			
MOVFW f	Move File to W	MOVF f,0	Z
NEGF f,d	Negate File	COMF f,1 INCF f,d	Z
SETC	Set Carry	BSF 3,0	-
SETDC	Set Digit Carry	BSF 3,1	-
SETZ	Set Zero	BSF 3,2	-
SKPC	Skip on Carry	BTFSS 3,0	-
SKPDC	Skip on Digit Carry	BTFSS 3,1	-
SKPNC	Skip on No Carry	BTFSC 3,0	-
SKPNDC	Skip on No Digit Carry	BTFSC 3,1	-
SKPNZ	Skip on Non Zero	BTFSC 3,2	-
SKPZ	Skip on Zero	BTFSS 3,2	-
SUBCF f,d	Subtract Carry from File	BTFSC 3,0 DECF f,d	Z
SUBDCF f,d	Subtract Digit Carry from File	BTFSC 3,1 DECF f,d	Z
TSTF f	Test File	MOVF f,1	Z

B.7 16-Bit Core Instruction Set

Microchip's high-performance 8-bit microcontroller family uses a 16-bit wide instruction set. This instruction set consists of 55 instructions, each a single 16-bit wide word. Most instructions operate on a file register, f, and the working register, W (accumulator). The result can be directed either to the file register or the W register or to both in the case of some instructions. Some devices in this family also include hardware multiply instructions. A few instructions operate solely on a file register (BSF for example).

Table B.12: 16-Bit Core Data Movement Instructions

Hex	Mnemonic	Description	Function
6pff	MOVFP f,p	Move f to p	f → p
b8kk	MOVLB k	Move literal to BSR	k → BSR (3:0)
bakx	MOVLP k	Move literal to RAM page select	k → BSR (7:4)
4pff	MOVPF p,f	Move p to f	p → W
01ff	MOVWF f	Move W to F	W → f
a8ff	TABLRD t,i,f	Read data from table latch into file f, then update table latch with 16-bit contents of memory location addressed by table pointer	TBLATH → f if t=1, TBLATL → f if t=0; ProgMem(TBLPTR) → TBLAT; TBLPTR + 1 → TBLPTR if i=1
acff	TABLWT t,i,f	Write data from file f to table latch and then write 16-bit table latch to program memory location addressed by table pointer	f → TBLATH if t = 1, f → TBLATL if t = 0; TBLAT → ProgMem(TBLPTR); TBLPTR + 1 → TBLPTR if i=1
a0ff	TLRD t,f	Read data from table latch into file f (table latch unchanged)	TBLATH → f if t = 1 TBLATL → f if t = 0
a4ff	TLWT t,f	Write data from file f into table latch	f → TBLATH if t = 1 f → TBLATL if t = 0

Table B.13: 16-Bit Core Arithmetic and Logical Instruction

Hex	Mnemonic	Description	Function
b1kk	ADDLW k	Add literal to W	(W + k) → W
0eff	ADDWF f,d	Add W to F	(W + f) → d
10ff	ADDWFC f,d	Add W and Carry to f	(W + f + C) → d
b5kk	ANDLW k	AND Literal and W	(W .AND. k) → W
0aff	ANDWF f,d	AND W with f	(W .AND. f) → d
28ff	CLRF f,d	Clear f and Clear d	0x00 → f,0x00 → d
12ff	COMF f,d	Complement f	.NOT. f → d
2eff	DAW f,d	Dec. adjust W, store in f,d	W adjusted → f and d
06ff	DECF f,d	Decrement f	(f - 1) → f and d
14ff	INCF f,d	Increment f	(f + 1) → f and d
b3kk	IORLW k	Inclusive OR literal with W	(W .OR. k) → W

Table B.13: 16-Bit Core Arithmetic and Logical Instruction (Continued)

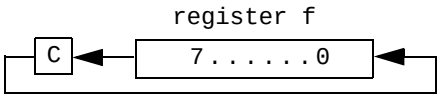
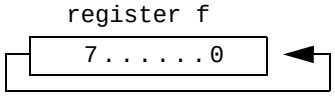
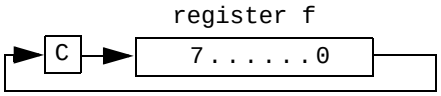
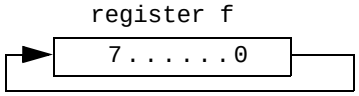
Hex	Mnemonic		Description	Function
08ff	IORWF	f,d	Inclusive or W with f	(W .OR. f) → d
b0kk	MOVLW	k	Move literal to W	k → W
bckk	MULLW	k	Multiply literal and W	(k x W) → PH:PL
34ff	MULWF	f	Multiply W and f	(W x f) → PH:PL
2cff	NEGW	f,d	Negate W, store in f and d	(W + 1) → f,(W + 1) → d
1aff	RLCF	f,d	Rotate left through carry	
22ff	RLNCF	f,d	Rotate left (no carry)	
18ff	RRCF	f,d	Rotate right through carry	
20ff	RRNCF	f,d	Rotate right (no carry)	
2aff	SETF	f,d	Set f and Set d	0xff → f,0xff → d
b2kk	SUBLW	k	Subtract W from literal	(k - W) → W
04ff	SUBWF	f,d	Subtract W from f	(f - W) → d
02ff	SUBWFB	f,d	Subtract from f with borrow	(f - W - c) → d
1cff	SWAPF	f,d	Swap f	f(0:3) → d(4:7), f(4:7) → d(0:3)
b4kk	XORLW	k	Exclusive OR literal with W	(W .XOR. k) → W
0cff	XORWF	f,d	Exclusive OR W with f	(W .XOR. f) → d

Table B.14: 16-Bit Core Bit Handling Instructions

Hex	Mnemonic		Description	Function
8bff	BCF	f,b	Bit clear f	0 → f(b)
8bff	BSF	f,b	Bit set f	1 → f(b)
9bff	BTFSC	f,b	Bit test, skip if clear	skip if f(b) = 0
9bff	BTFSS	f,b	Bit test, skip if set	skip if f(b) = 1
3bff	BTG	f,b	Bit toggle f	.NOT. f(b) → f(b)

Table B.15: 16-Bit Core Program Control Instructions

Hex	Mnemonic	Description	Function
ekkk	CALL k	Subroutine call (within 8k page)	PC+1 → TOS,k → PC(12:0), k(12:8) → PCLATH(4:0), PC(15:13) → PCLATH(7:5)
31ff	CPFSEQ f	Compare f/w, skip if f = w	f-W, skip if f = W
32ff	CPFSGT f	Compare f/w, skip if f > w	f-W, skip if f > W
30ff	CPFSLT f	Compare f/w, skip if f< w	f-W, skip if f < W
16ff	DECFSZ f,d	Decrement f, skip if 0	(f-1) → d, skip if 0
26ff	DCFSNZ f,d	Decrement f, skip if not 0	(f-1) → d, skip if not 0
ckkk	GOTO k	Unconditional branch (within 8k)	k → PC(12:0) k(12:8)→ PCLATH(4:0), PC(15:13) → PCLATH(7:5)
1eff	INCFSZ f,d	Increment f, skip if zero	(f+1)→ d, skip if 0
24ff	INFSNZ f,d	Increment f, skip if not zero	(f+1) → d, skip if not 0
b7kk	LCALL k	Long Call (within 64k)	(PC+1) → TOS; k → PCL, (PCLATH)→ PCH
0005	RETFIE	Return from interrupt, enable interrupt	(PCLATH) → PCH:k → PCL 0 → GLINTD
b6kk	RETLW k	Return with literal in W	k → W, TOS → PC, (PCLATH unchanged)
0002	RETURN	Return from subroutine	TOS → PC (PCLATH unchanged)
33ff	TSTFSZ f	Test f, skip if zero	skip if f = 0

Table B.16: 16-Bit Core Special Control Instructions

Hex	Mnemonic	Description	Function
0004	CLRWT	Clear watchdog timer	0 → WDT,0→ WDT prescaler, 1 → PD, 1 → TO
0003	SLEEP	Enter Sleep Mode	Stop oscillator,power down, 0 → WDT, 0 → WDT Prescaler 1 → PD, 1 → TO

B.8 Key to Enhanced 16-Bit Core Instruction Set

Field	Description
File Addresses	
f	8-bit file register address
fs	12-bit source file register address
fd	12-bit destination file register address
dest	W register if d = 0; file register if d = 1
r	0, 1, or 2 for FSR number
Literals	
kk	8-bit literal value
kb	4-bit literal value
kc	bits 8-11 of 12-bit literal value
kd	bits 0-7 of 12-bit literal value
Offsets, Addresses	
nn	8-bit relative offset (signed, 2's complement)
nd	11-bit relative offset (signed, 2's complement)
ml	bits 0-7 of 20-bit program memory address
mm	bits 8-19 of 20-bit program memory address
xx	any 12-bit value
Bits	
b	bits 9-11; bit address
d	bit 9; 0=W destination; 1=f destination
a	bit 8; 0=common block; 1=BSR selects bank
s	bit 0 (bit 8 for CALL); 0=no update; 1(fast)=update/save W, STATUS, BSR

B.9 Enhanced 16-Bit Core Instruction Set

Microchip's new high-performance 8-bit microcontroller family uses a 16-bit wide instruction set. This instruction set consists of 76 instructions, each a single 16-bit wide word (2 bytes). Most instructions operate on a file register, f, and the working register, W (accumulator). The result can be directed either to the file register or the W register or to both in the case of some instructions. A few instructions operate solely on a file register (BSF for example)

Table B.17: Enhanced 16-Bit Core Literal Operations

Hex	Mnemonic		Description	Function
0F*kk*	ADDLW	kk	ADD literal to WREG	W+kk → W
0B*kk*	ANDLW	kk	AND literal with WREG	W **.**AND**.** kk → W
0004	CLRWDT		Clear Watchdog Timer	0 → WDT, 0 → WDT postscaler, 1 → TO,1 → PD
0007	DAW		Decimal Adjust WREG	if W<3:0> >9 or DC=1, W<3:0>+6→W<3:0>, else W<3:0> → W<3:0>; if W<7:4> >9 or C=1, W<7:4>+6→W<7:4>, else W<7:4> → W<7:4>;
09*kk*	IORLW	kk	Inclusive OR literal with WREG	W **.**OR**.** kk → W
EF*kc* F0*kd*	LFSR	r,kd:kc	Load 12-bit Literal to FSR (second word)	kd:kc → FSRr
01*kb*	MOVLB	kb	Move literal to low nibble in BSR	kb → BSR
0E*kk*	MOVLW	kk	Move literal to WREG	kk → W
0D*kk*	MULLW	kk	Multiply literal with WREG	W * kk → PRODH:PRODL
08*kk*	SUBLW	kk	Subtract W from literal	kk–W → W
0A*kk*	XORLW	kk	Exclusive OR literal with WREG	W **.**XOR**.** kk → W

Table B.18: Enhanced 16-Bit Core Memory Operations

Hex	Mnemonic		Description	Function
0008	TBLRD *		Table Read (no change to TBLPTR)	Prog Mem (TBLPTR) → TABLAT
0009	TBLRD *+		Table Read (post-increment TBLPTR)	Prog Mem (TBLPTR) → TABLAT TBLPTR +1 → TBLPTR
000A	TBLRD *-		Table Read (post-decrement TBLPTR)	Prog Mem (TBLPTR) → TABLAT TBLPTR -1 → TBLPTR
000B	TBLRD +*		Table Read (pre-increment TBLPTR)	TBLPTR +1 → TBLPTR Prog Mem (TBLPTR) → TABLAT
000C	TBLWT *		Table Write (no change to TBLPTR)	TABLAT → Prog Mem(TBLPTR)
000D	TBLWT *+		Table Write (post-increment TBLPTR)	TABLAT → Prog Mem(TBLPTR) TBLPTR +1 → TBLPTR

Table B.18: Enhanced 16-Bit Core Memory Operations (Continued)

Hex	Mnemonic		Description	Function
000E	TBLWT *-		Table Write (post-decrement TBLPTR)	TABLAT → Prog Mem(TBLPTR) TBLPTR -1 → TBLPTR
000F	TBLWT +*		Table Write (pre-increment TBLPTR)	TBLPTR +1 → TBLPTR TABLAT → Prog Mem(TBLPTR)

Table B.19: Enhanced 16-Bit Core Control Operations

Hex	Mnemonic		Description	Function
E2*nn*	BC	nn	Relative Branch if Carry	if C=1, PC+2+2*nn→ PC, else PC+2→PC
E6*nn*	BN	nn	Relative Branch if Negative	if N=1, PC+2+2*nn→PC,else PC+2→PC
E3*nn*	BNC	nn	Relative Branch if Not Carry	if C=0, PC+2+2*nn→PC, else PC+2→PC
E7*nn*	BNN	nn	Relative Branch if Not Negative	if N=0, PC+2+2*nn→PC, else PC+2→PC
E5*nn*	BNOV	nn	Relative Branch if Not Overflow	if OV=0, PC+2+2*nn→PC, else PC+2→PC
E1*nn*	BNZ	nn	Relative Branch if Not Zero	if Z=0, PC+2+2*nn→PC, else PC+2→PC
E4*nn*	BOV	nn	Relative Branch if Overflow	if OV=1, PC+2+2*nn→PC, else PC+2→PC
E0*nd*	BRA	nd	Unconditional relative branch	PC+2+2*nd→ PC
E0*nn*	BZ	nn	Relative Branch if Zero	if Z=1, PC+2+2*nn→PC, else PC+2→PC
EC*ml* F*mm*	CALL	mm:ml,s	Absolute Subroutine Call (second word)	PC+4 → TOS, mm:ml → PC<20:1>, if s=1, W → WS, STATUS → STATUSS, BSR → BSRS
EF*ml* F*mm*	GOTO	mm:ml	Absolute Branch (second word)	mm:ml → PC<20:1>
0000	NOP		No Operation	No operation
0006	POP		Pop Top of return stack	TOS-1 → TOS
0005	PUSH		Push Top of return stack	PC +2→ TOS
D8*nd*	RCALL	nd	Relative Subroutine Call	PC+2 → TOS, PC+2+2*nd→PC
00FF	RESET		Generate a Reset (same as MCR reset)	same as MCLR reset
0010	RETFIE	s	Return from interrupt (and enable interrupts)	TOS → PC, 1 → GIE/GIEH or PEIE/GIEL, if s=1, WS → W, STATUSS → STATUS, BSRS → BSR, PCLATU/PCLATH unchngd.
0C*kk*	RETLW	kk	Return from subroutine, literal in W	kk → W,
0012	RETURN	s	Return from subroutine	TOS → PC, if s=1, WS → W, STATUSS → STATUS, BSRS → BSR, PCLATU/PCLATH are unchanged
0003	SLEEP		Enter SLEEP Mode	0 → WDT, 0 → WDT postscaler, 1 → TO, 0 → PD

Table B.20: Enhanced 16-Bit Core Bit Operations

Hex	Mnemonic		Description	Function
9*bf*	BCF	f,b,a	Bit Clear f	0 → f<b>
8*bf*	BSF	f,b,a	Bit Set f	1 → f<b>
B*bf*	BTFSC	f,b,a	Bit test f, skip if clear	if f<b>=0, PC+4→PC, else PC+2→PC
A*bf*	BTFSS	f,b,a	Bit test f, skip if set	if f<b>=1, PC+4→PC, else PC+2→PC
7*bf*	BTG	f,b,a	Bit Toggle f	f<b> → f<b>

Table B.21: Enhanced 16-Bit Core File Register Operations

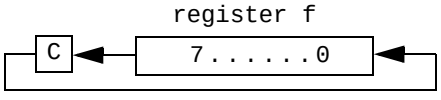
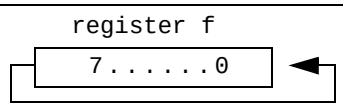
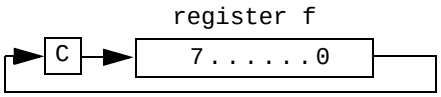
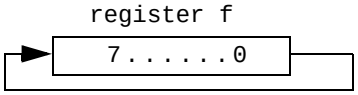
Hex	Mnemonic		Description	Function
24*f*	ADDWF	f,d,a	ADD WREG to f	W+f → dest
20*f*	ADDWFC	f,d,a	ADD WREG and Carry bit to f	W+f+C → dest
14*f*	ANDWF	f,d,a	AND WREG with f	W **.**AND**.** f → dest
6A*f*	CLRF	f,a	Clear f	0 → f
1C*f*	COMF	f,d,a	Complement f	f → dest
62*f*	CPFSEQ	f,a	Compare f with WREG, skip if f=WREG	f–W, if f=W, PC+4 → PC else PC+2 → PC
64*f*	CPFSGT	f,a	Compare f with WREG, skip if f > WREG	f–W, if f > W, PC+4 → PC else PC+2 → PC
60*f*	CPFSLT	f,a	Compare f with WREG, skip if f < WREG	f–W, if f < W, PC+4 → PC else PC+2 → PC
04*f*	DECF	f,d,a	Decrement f	f–1 → dest
2C*f*	DECFSZ	f,d,a	Decrement f, skip if 0	f–1 → dest, if dest=0, PC+4 → PC else PC+2 → PC
4C*f*	DCFSNZ	f,d,a	Decrement f, skip if not 0	f–1 → dest, if dest ≠ 0, PC+4 → PC else PC+2 → PC
28*f*	INCF	f,d,a	Increment f	f+1 → dest
3C*f*	INCFSZ	f,d,a	Increment f, skip if 0	f+1 → dest, if dest=0, PC+4 → PC else PC+2 → PC
48*f*	INFSNZ	f,d,a	Increment f, skip if not 0	f+1 → dest, if dest ≠ 0, PC+4 → PC else PC+2 → PC
10*f*	IORWF	f,d,a	Inclusive OR WREG with f	W **.**OR**.** f → dest
50*f*	MOVF	f,d,a	Move f	f → dest
C*fs* F*fd*	MOVFF	fs,fd	Move fs to fd (second word)	fs → fd
6E*f*	MOVWF	f,a	Move WREG to f	W → f
02*f*	MULWF	f,a	Multiply WREG with f	W * f → PRODH:PRODL
6C*f*	NEGF	f,a	Negate f	f +1 → f
34*f*	RLCF	f,d,a	Rotate left f through Carry	

Table B.21: Enhanced 16-Bit Core File Register Operations (Continued)

Hex	Mnemonic		Description	Function
44*f*	RLNCF	f,d,a	Rotate left f (no carry)	
30*f*	RRCF	f,d,a	Rotate right f through Carry	
40*f*	RRNCF	f,d,a	Rotate right f (no carry)	
48*f*	SETF	f,a	Set f	0xFF → f
54*f*	SUBFWB	f,d,a	Subtract f from WREG with Borrow	W–f–C → dest
5C*f*	SUBWF	f,d,a	Subtract WREG from f	f–W → dest
58*f*	SUBWFB	f,d,a	Subtract WREG from f with Borrow	f–W–C → dest
38*f*	SWAPF	f,d,a	Swap nibbbles of f	f<3:0> → dest<7:4>, f<7:4> → dest<3:0>
66*f*	TSTFSZ	f,a	Test f, skip if 0	PC+4 → PC, if f=0, else PC+2 → PC
18*f*	XORWF	f,d,a	Exclusive OR WREG with f	W **.**XOR**.** f → dest

B.10 Hexadecimal to Decimal Conversion

Byte				Byte			
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

Using This Table: For each Hex digit, find the associated decimal value. Add the numbers together. For example, Hex A38F converts to 41871 as follows:

Hex 1000's Digit		Hex 100's Digit		Hex 10's Digit		Hex 1's Digit		Result
40960	+	768	+	128	+	15	=	41871 Decimal