20.0 INSTRUCTION SET SUMMARY

ments to the previous PICmicro instruction sets, while maintaining an easy migration from these PICmicro The PIC18FXXX instruction set adds many enhanceMost instructions are a single program memory word (16-bits), but there are three instructions that require two program memory locations.

Each single word instruction is a 16-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 instruction set is highly orthogonal and is grouped into four basic categories:

- Byte-oriented operations
- **Bit-oriented** operations
- Literal operations
- The PIC18FXXX instruction set summary in Table 20-2 Control operations

operations. Table 20-1 shows the opcode field lists byte-oriented, bit-oriented, literal and control descriptions.

Most byte-oriented instructions have three operands:

- The file register (specified by 'f')
 - The destination of the result ď
 - The accessed memory (p, kq peilieds)

file register designator II specifies which file register is to be used by the instruction. (specified by 'a')

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

All bit-oriented instructions have three operands:

- The file register (specified by f')
 - The bit in the file register
- The accessed memory က်

(specified by 'b')

The bit field designator 'b' selects the number of the bit (specified by 'a')

nator " represents the number of the file in which the affected by the operation, while the file register desigbit is located.

The literal instructions may use some of the following

- A literal value to be loaded into a file register (sbecilied by 'k')
- The desired FSR register to load the literal value into (specified by f')
 - No operand required

(specified by '—',

The control instructions may use some of the following

- A program memory address (specified by 'n') operands:
- The mode of the Call or Return instructions (s, kq peilioeds)
- The mode of the Table Read and Table Write instructions (specified by 'm')
- No operand required

(specified by '—')

ble-word instructions. These three instructions were made double-word instructions so that all the required cuted as an instruction (by itself), it will execute as a information is available in these 32 bits. In the second All instructions are a single word, except for three douword, the 4-MSbs are 1's. If this second word is exe-NOP

program counter is changed as a result of the instruc-tion. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed instruction cycle, unless a conditional test is true or the All single word instructions are executed in a single as a NOP

The double-word instructions execute in two instruction

instruction execution time is 1 µs. If a conditional test is true or the program counter is changed as a result of an Thus, for an oscillator frequency of 4 MHz, the normal instruction, the instruction execution time is 2 μs. Iwo-word branch instructions (if true) would take 3 μs. One instruction cycle consists of four oscillator periods.

All examples use the format 'nnh' to represent a Figure 20-1 shows the general formats that the hexadecimal number, where h' signifies instructions can have.

hexadecimal digit.

lists the instructions recognized by the Microchip The Instruction Set Summary, shown in Table 20-2, Assembler (MPASMTM).

Section 20.1 provides a description of each instruction.

DS39564C-page 211 © 2006 Microchip Technology Inc.

PIC18FXX2

TABLE 20-1: OPCODE FIELD DESCRIPTIONS

Field	Description
_o	RAM access bit a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
qqq	Bit address within an 8-bit file register (0 to 7)
BSR	Bank Select Register. Used to select the current RAM bank.
ס	Destination select bit; d = 0: store result in WREG, d = 1: store result in file register f.
dest	Destination either the WREG register or the specified register file location
Ŧ	8-bit Register file address (0x00 to 0xFF)
ffs	12-bit Register file address (0x000 to 0xFFF). This is the source address.
fd	12-bit Register file address (0x000 to 0xFFF). This is the destination address.
_ل ې	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value)
label	Label name
шш	The mode of the TBLPTR register for the Table Read and Table Write instructions. Only used with Table Read and Table Write instructions:
*	No Change to register (such as TBLPTR with Table reads and writes)
+ *	Post-Increment register (such as TBLPTR with Table reads and writes)
· *	Post-Decrement register (such as TBLPTR with Table reads and writes)
* +	Pre-Increment register (such as TBLPTR with Table reads and writes)
п	The relative address (2's complement number) for relative branch instructions, or the direct address for Call/Branch and Return instructions
PRODH	Product of Multiply high byte
PRODL	Product of Multiply low byte
Ø	Fast Call/Retum mode select bit. s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
n	Unused or Unchanged
WREG	Working register (accumulator)
×	Don't care (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.
TBLPTR	21-bit Table Pointer (points to a Program Memory location)
TABLAT	8-bit Table Latch
TOS	Top-of-Stack
PC	Program Counter
PCL	Program Counter Low Byte
PCH	Program Counter High Byte
PCLATH	Program Counter High Byte Latch
PCLATU	Program Counter Upper Byte Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer
<u>10</u>	Time-out bit
<u>DD</u>	Power-down bit
C, DC, Z, OV, N	ALU status bits Carry, Digit Carry, Zero, Overflow, Negative
[]	Optional
^ 	Contents
1	Assigned to
^ v	Register bit field
Ψ	In the set of
italics	User defined term (font is counier)

© 2006 Microchip Technology Inc. DS39564C-page 212

GENERAL FORMAT FOR INSTRUCTIONS FIGURE 20-1:

Example Instruction	ADDWF MYREG, W, B	n n		MOVFF MYREG1, MYREG2						BSF MYREG, bit, B				MOVLW 0x7F					GOTO Label					CALL MYFUNC				BRA MYFUNC		BC MYFUNC
oriented file register operations	15 10 9 8 7 0 OPCODE d a f(FILE #)	d = 0 for result destination to be WREG register d = 1 for result destination to be file register (f) a = 0 to force Access Bank a = 1 for BSR to select bank f = 8-bit file register address	Byte to Byte move operations (2-word)	15 12 11 0 OPCODE f(Source FILE #)	2 11	1111 f (Destination FILE #)	f = 12-bit file register address	Bit-oriented file register operations	8 7	OPCODE b (BIT #) a f (FILE #)	b = 3-bit position of bit in file register (f) a = 0 to force Access Bank a = 1 for BSR to select bank f = 8-bit file register address	Literal operations	15 8 7 0	OPCODE k (literal)	k=8-bit immediate value	Control operations	CALL, GOTO and Branch operations	15 8 7 0	OPCODE n<7:0> (literal)	15 12 11 0	1111 n<19:8> (literal)	n = 20-bit immediate value	15 8 7 0	OPCODE S n<7:0> (literal)	15 12 11 0	S = Fast bit	15 11 10 0	OPCODE n<10:0> (literal)	15 87 0	OPCODE n<7:0> (literal)

DS39564C-page 213 © 2006 Microchip Technology Inc.

PIC18FXX2

TABLE 20-2: PIC18FXXX INSTRUCTION SET

STATE-ORIENTED FILE REGISTER OPERATIONS NSD	Mnemonic,	nic,			16-	Bit Instru	16-Bit Instruction Word	ord	Status	
1, ct a Add WHEG and Carry bit of 1 0010 01dao fffeff fffeff C.DC, Z.OV, N. 1.2 1, ct a Add WHEG and Carry bit of 1 0010 01dao fffeff fffeff C.DC, Z.OV, N. 1.2 1, ct a Compare f with WHEG skip 1 0010 11da fffeff fffeff Z.N 1.2 1, ct a Compare f with WHEG skip 1 2 or 3) 0110 010a fffeff fffeff None 1.2 1, ct a Compare f with WHEG skip 1 2 or 3) 0110 010a fffeff fffeff None 1.2 1, ct a Compare f with WHEG skip 1 2 or 3) 0110 010a fffeff fffeff None 1.2 1, ct a Compare f with WHEG skip 1 2 or 3) 0110 010a fffeff fffeff None 1.2 1, ct a Compare f with WHEG skip 1 2 or 3) 0110 010a fffeff fffeff None 1.2 1, ct a Compare f with WHEG skip 1 2 or 3) 0110 010a fffeff fffeff None 1.2 2, dt a Compare f with WHEG skip 1 2 or 3) 0110 010a fffeff fffeff None 1.2 3, dt a Compare f with WHEG skip 1 2 or 3) 0110 010a fffeff fffeff None 1.2 4, dt a Compare f with WHEG swith 1 1 0110 010a fffeff fffeff None 1.2 4, dt a None (source) 1 2 or 3) 0110 010a fffeff fffeff None 1.2 5, dt dt switch 0 1 2 or 3) 0110 010a fffeff fffeff None 1.2 6, dt A Switch 0 1 2 or 3) 0110 010a fffeff fffeff None 1.2 6, dt Rotate Left fithrough Carry 1 0 0 0 0 0 0 6, dt Rotate Right (No Carry) 1 0 0 0 0 0 0 6, dt Switch 0 0 0 0 0 0 0 6, dt Switch 0 0 0 0 0 0 0 0 6, dt Switch 0 0 0 0 0 0 0 0 6, dt Switch 0 0 0 0 0 0 0 0 6, dt Switch 0 0 0 0 0 0 0 0 6, dt Switch 0 0 0 0 0 0 0 0 6, dt Switch 0 0 0 0 0 0 0 0 7, dt Rotate 0 0 0 0 0 0 0 0 0 8, dt Switch 0 0 0 0 0 0 0 0 0 9, dt Switch 0 0 0 0 0 0 0 0 0 1, dt	Opera	spu	Description	cycles	MSb			LSb	Affected	Notes
Compared with WREG skip = 1	BYTE-ORI	ENTED F	ILE REGISTER OPERATIONS							
Compare furth WREG swip 1 0010 oda fffff ffff C DC Z OV, N 1.2 1.2	ADDWF	f, d, a	Add WREG and f	+	0010	01da0	ffff	ffff	8	1,2
1, d. a AND WHEG with f	ADDWFC	f, d, a	Add WREG and Carry bit to f	_	0010	0da	ffff	ffff	20	1,2
1, d. a Clear f 1, d. a Compare f with WREG, skip = 1 2 or 3 0.00	ANDWF	f, d, a	AND WREG with f	_	0001	01da	ffff	ffff	z	1,2
1, d, a Complement Comparer with WREG, skip = 1 1 1 1 1 1 1 1 1 1	CLRF	f, a	Clear f	_	0110	101a	ffff	ffff	Z	2
Compare fwith WREG, skip = 1 (2 or 3) 0110 010a ffff ffff None 4 (2 or 3) 0110 010a ffff ffff None 1.2 (3 or 3) 0110 010a ffff ffff None 1.2 (3 or 3) 0110 010a ffff ffff None 1.2 (3 or 3) 0110 010a ffff ffff None 1.2 (3 or 3) 0110 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (3 or 3) 010 010a ffff ffff None 1.2 (4 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a fffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (5 or 3) 010 010a ffff ffff None 1.2 (COMF	f, d, a	Complement f	_	0001	11da	ffff	ffff	Z, N	1,2
1, a Compare f with WREG, skip > 1 (2 or 3) 0.10 0.10 a ffff ffff None 1.2 3.	CPFSEQ	f, a	Compare f with WREG, skip =	'n	0110	001a	ffff	ffff	None	4
1, d, a Decrement, Skip if Not	CPFSGT	f,	Compare f with WREG, skip >	1 (2 or 3)	0110	010a	££££	ffff	None	4
1 (d. a) Decrement (Skip) frot 0 1 (2 or 3) 0.010 1.24 C. Co. C. OV. N. I. 2. 3. 1.2. 3. 2 (d. a) Decrement (Skip) frot 0 1 (2 or 3) 0.100 1.04a fffff ffff ffff from 1.2. 3. 1 (d. a) Increment (Skip) frot 0 1 (2 or 3) 0.101 1.04a fffff ffff ffff ffff from 1.2. 3. 1 (d. a) Increment (Skip) frot 0 1 (2 or 3) 0.101 1.04a ffff ffff ffff from 1.2. 3. 1 (d. a) Increment (Skip) frot 0 1 (2 or 3) 0.101 1.04a ffff fff fff fff from 1.2. 3. 1 (d. a) Increment (Skip) frot 0 1 (2 or 3) 0.101 1.04a ffff fff fff fff fff fff fff from 1.2. 3. 1 (d. a) Increment (Skip) frot 0 1 (2 or 3) 0.101 1.04a ffff fff fff fff fff fff fff fff fff f	CPFSLT	ť,	Compare f with WREG, skip <	ŏ	0110	0000	ffff	ffff	None	1,2
2 f, d. a Decrement, Skip if 0 1 (2 or 3) 0.00 11da fffff ffff fff ff g g g g g g g g g g g g g g g g g g g <	DECF			_	0000	01da	ffff	ffff	Ŋ	က်
1, d. a Decrement, Skip if Not 0 1 (2 or 3) 0100 11da fffff Iffff None 1, 2, 3 1, d. a Increment, Skip if Not 0 1 (2 or 3) 010 10da ffff ffff None 1, 2, 3 1, d. a Increment, Skip if Not 0 1 (2 or 3) 010 10da ffff ffff None 1, 2 1, d. a Increment, Skip if Not 0 1 (2 or 3) 010 10da ffff ffff ffff None 1, 2 1, d. a Move fs, (source) to 1st word 2 1100 ffff ffff ffff ffff ffff ffff None 1, 2 <	DECFSZ		Decrement f, Skip if 0	ö	0010	11da	ffff	ffff	None	က်
f, d, a Increment f, Skip if O 1 0.010 10da ffff ffff ffff ffff ffff fff fff None 1, 2, 3. f, d, a Increment f, Skip if O 1 (2 or 3) 0.011 11da ffff ffff ffff ffff ffff ffff fff fff	DCFSNZ		Decrement f, Skip if Not 0	č	0100	11da	ffff	ffff	None	1,2
f, d, a Incementf, Skip if 0 1 (2 or 3) 0011 11da ffff ffff ffff ffff None f, d, a Inclusive ON WEG with f 1 (2 or 3) 0100 10da ffff ffff ffff ffff Inclusive ON WEG with f 1 (2 or 3) 0100 10da ffff ffff ffff ffff Inclusive ON WEG with f 1 (3 or 3) 0100 1111 ffff ffff ffff ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff ffff ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff fff Inclusive ON WEG with f 1 000 0110 ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff Inclusive ON WEG with f 1 000 0110 ffff ffff Inclusive ON WEG with f 1 000 0110 fffff Inclusive ON WEG fffff Inclusive O	NCF	f, d, a	Increment f	_	0010	10da	££££	ffff		က်
f, d, all increment f, Skip if Not 0 1 (2 or 3) 0.000 and fifte	INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011	11da	ffff	ffff	None	4
f, d, a Move Is, fource) to Test Move Is, did a Move Is, fource) to Test	INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100	10da	fff	ffff	None	1,2
f.d. a Move f 1 1000 Effe Effe Effe Z. N	IORWF	f, d, a	Inclusive OR WREG with f	_	1000	00da	££££	ffff	Z, N	1,2
f. for Move fs, (source) to 1st word 1 110 ffff ffff ffff 1 110 1 1 1 1 1 1 1	MOVF	f, d, a	Move f	-	0101	00da	ffff	ffff	Z, N	_
f, a Move WREG tof 1 1111 ffff ff fff fff fff fff fff fff fff ff	MOVFF	fs, fd	Move f _s (source) to 1st word	2	1100	££££	ffff	ffff	None	
f, a Multiply WREG with f			f _d (destination) 2nd word		1111	ffff	ffff	ffff		
f, a Negate if the Mone f, a Negate if the Mone f, d a Negate if through Carry i, d, a Rotate Left flow Carry) 1 010 110a ffff ffff ffff C, Z, N 010 01da ffff ffff fff C, Z, N 010 01da ffff ffff fff Z, N 010 01da ffff ffff None ffff fff Z, N 010 01da ffff ffff None ffff fff Z, N 010 01da ffff ffff None ff Z, D, Z, D, N 010 01da ffff ffff None ff Z, D, Z, D, N 010 01da ffff ffff None ff Z, D, Z, D, N 010 01da ffff ffff None ff Z, D, Z, D, N 010 01da ffff ffff None ff Z, D,	MOVWF	ť,	Move WREG to f	_	0110	111a	ffff	ffff	None	
f, a. Nogate f, d. a. 1 0110 110a fffff ffff ffff C. DC, Z. OV, N. I. d. a. f, d. a. Rotate Left (No Carry) 1 0110 01da ffff ffff ffff C. Z. N. I. O. I. O. I. O. I. O. I. O. I. A. S. A.	MULWF	f, a	Multiply WREG with f	_	0000	001a	ffff	ffff		
f, d, a Rotate Leff through Carry) 1 0.011 0.1da £ffff £fff £fff £fff £fff £fff £fff Z.N f, d, a Rotate Right frough Carry) 1 0.010 0.0da £ffff £fff Z.N f, d, a Subtract from WREG with t, d, a Subtract MREG from f with toporrow 1 0.10 1.0da £ffff £fff £fff C, DC, Z, OV, N B f, d, a Subtract WREG from f with top orrow 1 0.10 1.0da £fff £fff C, DC, Z, OV, N b f, d, a Subtract WREG from f with top orrow f, d, a Subtract WREG from f with top orrow 1 0.01 1.0da £fff £fff C, DC, Z, OV, N f, d, a Subtract WREG from f with top orrow f, d, a Subtract WREG with f 1 0.01 1.0da £fff £fff C, DC, Z, OV, N Dorrow f, d, a Subtract WREG with f 1 0.011 1.0da £fff £fff None Efff ffff None L, O, C, Z, OV, N N None L, O, C, Z, OV, N N L, O, C, Z, OV, N N <	NEGF	ť,	Negate f	_	0110	110a	ffff	ffff	C, DC, Z, OV, N	1,2
f, d, a Rotate Left (No Carry) 1 000 00da fffff ffff ffff fff ff ff ff ff ff ff	RLCF	f, d, a	Rotate Left f through Carry	_	0011	01da	ffff	Efff	C, Z, N	
f, d, a Robate Right f through Carry 1 0011 00da ffff ffff C.Z.N f, d, a Subtract Rfrom WREG with borrow 1 0101 0104 ffff ffff ffff ffff ffff None e f, d, a Subtract Fffrom WREG from f with borrow 1 0101 11da ffff ffff C, DC, Z, OV, N N e f, d, a Subtract WREG from f with borrow 1 0101 11da ffff ffff C, DC, Z, OV, N N e f, d, a Subtract WREG from f with borrow 1 0101 11da ffff ffff C, DC, Z, OV, N N f, d, a Subtract WREG from f with borrow 1 1 0101 10da ffff ffff None f, d, a Subtract WREG with f 1 1 001 10da ffff ffff None f, d, a Bit Clear f 1 1 001 10da ffff ffff None f	RLNCF	f, d, a	Rotate Left f (No Carry)	_	0010	01da	ffff	ffff	Z, N	1,2
1, d, a Poste Right (No Carry)	RRCF	f, d, a	Rotate Right f through Carry	_	0011	00da	ffff	ffff	C, Z, N	
1	RRNCF		Rotate Right f (No Carry)	_	0010	00da	ffff	ffff	Z, N	
WB	SETF	ť,	Set f	_	0110	100a	ffff	ffff		
Description	SUBFWB		Subtract f from WREG with	-	0101	01da	fff	ffff	z, ov,	1,2
Note 1, 0, a Subtract WHEG from furth 1 0.101 1.0da ffff ffff C, DC, Z, OV, N	į		borrow	,		:			1	
Provided Windows 1 1 1 1 1 1 1 1 1	SUBWE	ס ע	Subtract WHEG from I		TOTO	Lida	TITI	TITI	C, DC, 2, OV, N	c
PF 1, d, a Swap nibbles in f 1 12 or 3 10 10 10 2 Eff. fiff. None		ŝ	borrow	_	T 0 T 0	4000	7777	7777	, DC, L, CV, N	J.,
SSZ f, a	SWAPF	f.d.a	Swap nibbles in f		0011	10da	ffff	ffff	None	4
NHENTED FILE REGISTER OPERATIONS 1 1 1 1 1 1 1 1 1	TSTES7		Test f. skip if 0	č	0110	6110	FFFF	FFFF	None	2
Index Fig. 1 1001 bbba fiff fiff Index Fiff None f, b, a Bit Setf 1 1000 bbba fiff fiff None i, b, a Bit Testf, Skip if Clear 1(2 or 3) 1011 bbba fiff fiff None i, b, a Bit Testf, Skip if Set 1(2 or 3) 1010 bbba fiff fiff None f, d, a Bit Teggle i 1 0111 bbba fiff fiff None	XORWF	f, d, a	Exclusive OR WREG with f	;	0001	10da	ffff	ffff	Z, Z	ı Î
f, b, a Bit Clearf 1 1001 bbba ffff ffff None 1, b, a Bit Satf 1 1,000 bbba ffff ffff None 1, co. b, a Bit Tastf, Skip if Clear 1, co. 3) 1011 bbba ffff ffff None 1, co. 3) 1010 bbba ffff ffff None 1, co. 3) 1010 bbba ffff ffff None 1, co. 4, a Bit Tastf, Skip if Set 1 1, co. 3) 1010 bbba ffff ffff None 1, co. 3) 1011 bbba ffff fff None 1, co. 3) 1011 bbba ffff ffff None 1, co. 3) 1011 bbba ffff fff None 1, co. 3) 1011 bbba ffff ffff None 1, co. 3) 1011 bbba ffff None 1, co. 3) 1011 bb	BIT-ORIEN	TED FIL	E REGISTER OPERATIONS							
(i, b, a) Bit Setf 1000 bbba ffff ffff None i; 0, a Bit Testf, Skip if Glear 1(2 or 3) 1010 bbba ffff ffff None i; 0, a Bit Testf, Skip if Set 1(2 or 3) 1010 bbba ffff ffff None f, d, a Bit Testf, Skip if Set 1 11 bbba ffff ffff None	BCF	t b a	Bit Clear f	_	1001	hhha	FFF	ffff	None	1 2
C f, b, a Bit Test f, Skip if Clear 1 (2 or 3) 1011 bbba f.ff.f. f.ff. None 1 (2 or 3) 1010 bbba f.ff.f. f.ff. None f.d, a Bit Toggle f 1 (2 or 3) 1011 bbba f.ff.f. f.ff. None	BSF	f, b, a			1000	bbba	ffff	ffff	None	1,2
SS f, b, a Bit Test f, Skip if Set 1 (2 or 3) 1010 bbba f.f.f.f. f. f. none f, d, a Bit Toggle f 1 1 11 bbba f.f.f. f. f. none	BTFSC	f, b, a		1 (2 or 3)	1011	bbba	ffff	ffff	None	3,4
f, d, a Bit Togglef 11 0111 bbba ffff ffff None 1,	BTFSS	f, b, a		1 (2 or 3)	1010	bbba	ffff	ffff	None	3,4
	вта		Bit Toggle f	_	0111	bbba	ffff	ffff	None	

Note 1: When a PORT register is modified as a function of itself (e.g., MOVF PORTE, 1, 0), 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.

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.

4: Some instructions are 2-word instructions. The second word of these instructions will be executed as a NOP, if some instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.

5: If the Table Write starts the write cycle to internal memory, the write will continue until terminated.

© 2006 Microchip Technology Inc. DS39564C-page 214

TABLE 20-2: PIC18FXXX INSTRUCTION SET (CONTINUED)

Mnemonic,	onic,	2000	200	16-	Bit Instr	16-Bit Instruction Word	ord	Status	4014
Operands	spu	Describing	cycles	MSb			rSb	Affected	NOTES
CONTROL OPERATIONS	OPERA	TIONS							
BC	ے	Branch if Carry	1 (2)	1110	0010	nnnn	nnnn	None	
BN	_	Branch if Negative	1 (2)	1110	0110	nnnn	nnnn	None	
BNC	_	Branch if Not Carry	1 (2)	1110	0011	nnnn	nnnn	None	
BNN	_	Branch if Not Negative	1 (2)	1110	0111	nnnn	nnnn	None	
BNOV	_	Branch if Not Overflow	1 (2)	1110	0101	nnnn	nnnn	None	
BNZ	c	Branch if Not Zero	2	1110	1000	nnnn	nnnn	None	
BOV	_	Branch if Overflow	1 (2)	1110	0100	nnnn	nnnn	None	
BRA	۵	Branch Unconditionally	1 (2)	1101	0nnn	nnnn	nnnn	None	
BZ	-	Branch if Zero	1 (2)	1110	0000	nnnn	nnnn	None	
CALL	n, s	Call subroutine1st word	2	1110	1108	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	KKKK		
CLRWDT	I	Clear Watchdog Timer	_	0000	0000	0000	0100	<u>TO, PD</u>	
DAW	1	Decimal Adjust WREG	_	0000	0000	0000	0111	ပ	
GОТО	۵	Go to address1st word	2	1110	1111	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
NOP	I	No Operation	_	0000	0000	0000	0000	None	
NOP	I	No Operation	-	1111	XXXX	XXXXX	XXXX	None	4
POP	I	Pop top of return stack (TOS)	_	0000	0000	0000	0110	None	
PUSH	I	Push top of return stack (TOS)	-	0000	0000	0000	0101	None	
RCALL	_	Relative Call	2	1101	lnnn	nnnn	nnnn	None	
RESET		Software device RESET	_	0000	0000	1111	1111	All	
RETFIE	s	Return from interrupt enable	2	0000	0000	1000	8000	GIE/GIEH,	
								PEIE/GIEL	
RETLW	¥	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
RETURN	s	Return from Subroutine	2	0000	0000	1000	0018	None	
SLEEP	l	Go into Standby mode	_	0000	0000	0000	0011	<u>TO, PD</u>	

Note 1: When a PORT register is modified as a function of itself (e.g., MOVF PORTB, 1, 0), 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.

 3. If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is
- 4: Some instructions are 2-word instructions. The second word of these instructions will be executed as a NOP, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.
 - 5: If the Table Write starts the write cycle to internal memory, the write will continue until terminated.

DS39564C-page 215 © 2006 Microchip Technology Inc.

PIC18FXX2

TABLE 20-2: PIC18FXXX INSTRUCTION SET (CONTINUED)

Mnemonic,		1	4	Bit Inst	16-Bit Instruction Word	Vord	Status	1
Operands	Describinon	cycles	MSb			rSb	Affected	salon
LITERAL OPERATIONS	SNOI							
ADDLW k	Add literal and WREG	_	0000	1111	kkk	kkkk	C, DC, Z, OV, N	
ANDLW k	AND literal with WREG	_	0000	1011	kkk	kkkk	Z, N	
IORLW k	Inclusive OR literal with WREG	_	0000	1001	kkk	kkkk	Z, N	
LFSR f, k	Move literal (12-bit) 2nd word	5	1110	1110	3300	kkkk	None	
	to FSRx 1st word		1111	0000	kkkk	kkkk		
MOVLB k	Move literal to BSR<3:0>	-	0000	1000	0000	kkkk	None	
MOVLW k	Move literal to WREG	_	0000	1110	kkk	kkkk	None	
MULLW k	Multiply literal with WREG	_	0000	1101	KKKK	kkkk	None	
RETLW k	Return with literal in WREG	2	0000	1100	kkk	kkkk	None	
SUBLW k	Subtract WREG from literal	_	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW k	Exclusive OR literal with WREG	_	0000	1010	kkkk	kkkk	Z, N	
DATA MEMORY ↔	DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS							
TBLRD*	Table Read	2	0000	0000	0000	1000	None	
TBLRD*+	Table Read with post-increment		0000	0000	0000	1001	None	
TBLRD*-	Table Read with post-decrement		0000	0000	0000	1010	None	
TBLRD+*	Table Read with pre-increment		0000	0000	0000	1011	None	
TBLWT*	Table Write	2 (5)	0000	0000	0000	1100	None	
TBLWT*+	Table Write with post-increment		0000	0000	0000	1101	None	
TBLWT*	Table Write with post-decrement		0000	0000	0000	1110	None	
TBLWT+*	Table Write with pre-increment		0000	0000	0000	1111	None	

Note 1: When a PORT register is modified as a function of itself (e.g., MOVF PORTB, 1, 0), 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 constant device, the data will be written back with a constant of the configured as input and is driven low by an external device, the data will be cleared if assigned.

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

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.

4: Some instructions are 2-word instructions. The second word of these instructions will be executed as a NOP, unless the first word of the instruction ratrieves the information embedded in these 16-bits. This ensures that all program memory

locations have a valid instruction.

5. If the Table Write starts the write cycle to internal memory, the write will continue until terminated.

© 2006 Microchip Technology Inc. DS39564C-page 216

20.1 Instruction Set

ADDLW	ADD literal to W	al to W			ADDWF	ADD W to f		
Syntax:	[<i>label</i>] A	[label] ADDLW k	~		Syntax:	[label] ADDWF	f [,d [,a]	_
Operands:	$0 \le k \le 255$	τÕ			Operands:	$0 \le f \le 255$		
Operation:	$(W) + k \rightarrow W$	Α,				d ∈ [0,1]		
Status Affected:	N, OV, C, DC, Z	DC, Z			Č	a∈ [0,1]		
Encoding:	0000	1111	kkkk	kkkk	Operation:	$(VV) + (T) \rightarrow VV$		
Description:	The conte	The contents of W are added to the	re add	ad to the	Status Affected:	N, OV, C, DC, Z		
	8-bit litera	8-bit literal 'k' and the result is	ie resu	It is	Encoding:	0010 01da	ffff	FEEE
	placed in W	×			Description:	Add W to register f'. If d' is 0, the	f'. If d is	, 0, the
Words:						result is stored in W. If 'd' is 1, the	W. If d is	s 1, the
						result is stored back in register If	ck in regi	ster If
Cycles:	-					(default). If 'a' is 0, the Access	, the Acc	ess
Q Cycle Activity:						Bank will be selected. If 'a' is 1, the	ted. If 'a'	is 1, the
ğ	Q2	8		Q 4		BSR is used.		
Decode	Read	Process		Write to W	Words:	-		

	>	
89	Process	Data
Q2	Read	iteral 'k'
ğ	Decode	

	Decode	Read literal 'k'	Process Data	≥
Exar	Example:	ADDLW	0x15	
	Before Instruction	ıction		
	///	0×10		

uction	0x10	tion	0x25
nstri	П	truc	п
Before Instruction	≯	After Instruction	>

REG, 0, 0
ADDWF
Example:

Q4
Write to
destination

Q2 Read register f*

Q Cyde Activity: Q1 Decode

ဗ

REG,						
ADDWF	_	0x17	0xC2		0xD9	0xC2
AI	ructio	П	П	ction	11	П
Example:	Before Instruction	>	REG	After Instruction	8	REG

PIC18FXX2

ADDWFC	ADD W and Carry bit to f	bit to f	ANDLW	AND literal with W	
Syntax:	[label] ADDWFC f[,d[,a]	f [,d [,a]	Syntax:	[label] ANDLW k	
Operands:	$0 \le f \le 255$ d $\in [0,1]$		Operands: Operation:	$0 \le k \le 255$ (W) AND. $k \to W$	
Operation:	$a \in [0,1]$ (W) + (f) + (C) \rightarrow dest	est	Status Affected:		
Status Affected:	N,OV, C, DC, Z		Encoding:	0000 1011 KKK	kkkk
Encoding:	0010 00da	fiff ffff	Description:	the 8-bit literal 'K'. The result is	Jed with
Description:	Add W, the Carry Flag and data	lag and data		placed in W.	
	memory location 'f'. If 'd' is 0, the	If d is 0, the	Words:	-	
	result is placed in W. II of is 1, the result is placed in data memory loca-	v. II u is i, tile ata memory loca-	Cycles:	-	
	tion I'. If 'a' is 0, the Access Bank	Access Bank	Q Cycle Activity:		
	will be selected. If 'a' is 1, the BSR	a' is 1, the BSR	8	Q2 Q3	Ω
	will not be overridden.	en.	Decode	Read literal Process Writ	Write to W
Words:	-			1	
Cycles:	-				
Q Cycle Activity:			Example:	ANDLW 0x5F	
ğ	Q2 Q3	Q 4	Before Instruction	nction	
Decode	Read Process	s Write to	8	= 0xA3	
	register ff Data	destination	After Instruction	tion	
			Μ	= 0x03	

REG, 0, 1 ADDWFC Example:

© 2006 Microchip Technology Inc. DS39564C-page 218

DS39564C-page 217

Syntax: [label] ANDWF f [d [a]] Syntax: [label] BC n Operands: 0 ≤ f ≤ 255 4 ∈ [0,1] 0 128 ≤ n ≤ 127 128 ≤ n ≤ 127 Operands: 4 ∈ [0,1] 0 0 110 0 10 Operation: (W) -AND. (f) → dest 1 0<	ANDWF	AND W with f	ith f			BC		Branch if Carry	Carry		
des: $0 \le f \le 255$ de $[0,1]$ and $[0,1]$ on: (W) .AND. $(f) \rightarrow dest$ on: (W) .AND. $(f) \rightarrow dest$ fifferled: $N.Z$ Groundlist of W are AND'ed with register f' . If f' is f . the result is stored back in register f' (default). If f' is f . the BSR will not be overridden (default). be overridden (default). 1	Syntax:	[label] A	NDWF	f [,d [,a		Synt	ax:	[<i>label</i>] B	C C		
d ∈ [0,1] a ∈ [0,1] b	Operands:	0 ≤ f ≤ 255	10			Ope	rands:	-128 ≤ n ≤	127		
on: (W) .AND. (f) → dest N.Z Encoding: 0.001 0.1da ffff ffff 1		$d\in [0,1]$ $a\in [0,1]$				Ope	ration:	if carry bit (PC) + 2	is '1' :+2n → F	S S	
Wifected: N.Z	Operation:	(W) AND	. (f) → de	sst		Statu	us Affected:	None			
tion: The contents of Ware AND'ed with register "I" if "d" is "the result is stored back in register "(default). If "a" is "the result is stored back in register "(default). If "a" is 0, the Access Bank will be selected. If "a" is 1, the BSR will not be overridden (default). Words: 1 1 Activity: Q Q Q Q Q Ord Activity: Activity: <td>Status Affected:</td> <td>N,N</td> <td></td> <td></td> <td></td> <td>Enc</td> <td>odina:</td> <td>1110</td> <td></td> <td>nnnn</td> <td>uuuu</td>	Status Affected:	N,N				Enc	odina:	1110		nnnn	uuuu
tion: The contents of W are AND'ed with register "f" if "d" is ", the result is stored back in register "(default). If "a" is 0, the Access Bank will be selected. If "a" is 1, the BSR will not be overridden (default). 1	Encoding:	0001	01da	ffff	ffff	Desc	cription:	f the Carn	v bit is '1'	then	the the
Selection a is 1, title Both will find be overridden (default).	Description:	The contered register f'. stored in V stored bac 'a' is 0, the	nts of W If 'd' is 0 W. If 'd' is k in regis Access	are AND), the res 1, the res ster ff (de Bank wi	r'ed with sult is ssult is sfault). If			program w The 2's co added to th have incre instruction	vill branch mplemen he PC. Semented to mented to the new the new This inch	it numit ince the o fetch addre	oer '2n' is ne PC will I the next ss will be
1 Words: 1 1 Cycles: 1(2) 2 Cycles: 1(2) 3 Cycles: 1(3) 3 Cycles: 1(3		selected. I	n લાડા, lden (deશિ	ine bor ault).	WIII TIOL			a two-cycle	e instructi	ion.	n is men
1 Cycles: 1(2) 2 Cycle Activity: Q 1 Q2 Q3 Q4 If Jump:	Words:	-				Wor	:sp	-			
Q2 Q3 Q4 If Jump: Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q3 Q3 Q3 Q4 <	Cycles:	-				Cyc	es:	1(2)			
Q2 Q3 Q4 If Jump: Read Process Write to legister IP Q1 Q2 Q3 register IP Data destination Decode Read literal Process	Q Cycle Activity:					g	ycle Activity:				
Read Process Write to legister IP Q1 Q2 Q3 register IP Data destination Decode Read literal Process	ō	05	Ö		Q	<u>ا</u> کے	:dwr				
Data destination Decode Read literal Process	Decode	Read	Proces		rite to		5	Ø5	ဗ		Q
		register f	Data		tination		Decode	Read literal	Process		Write to PC

	Decode	<u> </u>	Read	Process	SSe		Writ
		reg	register If	Data	g		destin
Exa	Example:	AN	ANDWF	REG,	0,0	0	
	Before Instruction	ctior	_				
	8	П	0x17				
	REG	П	0xC2				
	After Instruction	lion					
	>	П	0x02				

No operation

operation

operation

If No Jump: Ö

S

ž

Process Data No operation

Q4 No operation

Q3 Process Data

Q2 Read literal 'n'

Decode

PEG, O,		0x17	0xC2		0x02	0xC2
ANDME	uction	ô =	6	tion	ô =	0
Adilliple.	Before Instruction	×	REG	After Instruction	×	REG

2		(HERE)			address (HERE+12)		address (HERE+2)
BC		address (HERE		÷	address	ö	address
ple: HERE	Before Instruction	PC =	After Instruction	If Carry =	PC	If Carry =	P _C
Example:	8		⋖				

= 1; = address (Jump) = 0; = address (HERE+2)

After Instruction
If Negative =
If Negative =
If Negative =

PIC18FXX2

BCF	Bit Clear f	r f			BN	Branch if	Branch if Negative	
Syntax:	[label] BCF		f,b[,a]		Syntax:	[<i>label</i>] BN	c N	
Operands:	$0 \le f \le 255$	55			Operands:	-128 ≤ n ≤ 127	127	
	0≤b≤7 a∈ [0,1]				Operation:	if negative bit is '1' (PC) + 2 + 2n → P	if negative bit is '1' (PC) + 2 + 2n → PC	
Operation:	d>f <b< p=""></b<>	٨			Status Affected:			
Status Affected:	None				Encoding:	1110	0110 nnnn	nnnn
Encoding:	1001	bbba	ffff	ffff	Description:	If the Neg	- ! <u>-</u>	then the
Description:	Bit b in is 0, the	Bit 'b' in register 'f' is cleared. If 'a is 0, the Access Bank will be	is clear	ed. If 'a' be		program v The 2's co	program will branch. The 2's complement number '2n' is	umber '2n' is
	selected	selected, overriding the BSR value. If 'a' = 1, then the bank will be	g the B	SR value.		added to t	added to the PC. Since the PC will have incremented to fetch the next	e the PC wil
	selected (default).	selected as per the BSR value (default).	BSB	/alue		instruction PC+2+2n.	instruction, the new address will be PC+2+2n. This instruction is then	Idress will be
Words:	-					a two-cyc	a two-cycle instruction.	
Cvoles:	-				Words:	-		
Q Cycle Activity:					Cycles:	1(2)		
, Q	05	89		Q	Q Cycle Activity:	iy:		
Decode	Read	Process	S	Write	If Jumb:			
	register f	_	_	register "f	Ω	Q2	Q 3	Q4
					Decode	Read literal	Process	Write to PC
Example:	BCF	FLAG REG,	3, 7,	0		Ľ	Data	
made C	90,40	I			% N	%	8 S	°N
Before Instruction	re Instruction				operation	operation	operation	operation
After Instruction	ion - Carl				If No Jump:			
FLAG R	FLAG REG = 0x47				ρ	Q2	0 3	Q4
					Decode	Read literal	Process	No
						ŗ	Data	operation
					Example:	HERE	BN Jump	
					Before Instruction			
					PC	= ad	address (HERE)	

© 2006 Microchip Technology Inc. DS39564C-page 220

DS39564C-page 219

BNC		Branch	Branch if Not Carry		BNN		Branch if	Branch if Not Negative	ø	
Syntax:	ax:	[label] BNC	BNC n		Syntax:	ax:	[<i>label</i>] BNN	u NN		
Oper	Operands:	-128 ≤ n ≤ 127	า ≤ 127		Oper	Operands:	-128 ≤ n ≤ 127	127		
Oper	Operation:	if carry bit is '0' (PC) + 2 + 2n -	f carry bit is '0' PC) + 2 + 2n \rightarrow PC		Oper	Operation:	if negative bit is '0' (PC) + 2 + 2n \rightarrow P	f negative bit is '0' (PC) + 2 + 2n \rightarrow PC		
Statu	Status Affected:	None			Statu	Status Affected:	None			
Enco	Encoding:	1110	0011	uuuu uuuu	Enco	Encoding:	1110	0111 nnnn	uuuu ui	
Desc	Description:	If the Caprogram The 2's added to have inc instruction of the caprogram.	If the Carry bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC will have incremented to fetch the next instruction, the new address will be PC+2-P.	ien the umber '2n' is the PC will etch the next ddress will be ction is then	Desc	Description:	If the Negative bit is program will branch. The 2's complement added to the PC. Sin added to the PC. Sin have incremented to instruction, the new instruction, the new instruction.	If the Negative bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC+2-Pa. This instruction is then	If the Negative bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC+2-PC. This instruction is then	
,	<u>.</u>	אס- סיין	a two-cycle illistruction.	_			a two-cycle	e ilistraction.		
Words:	<u>8:</u>				Words:	JS:	_			
Cycles:	.se:	1(2)			Cycles:	es:	1(2)			
Q Cycle If Jump:	Q Cycle Activity: If Jump:				og <u>=</u>	Q Cycle Activity: If Jump:				
	8	02	ő	Q 4		ō	05	8	Q 4	
	Decode	Read literal	al Process Data	Write to PC	,	Decode	Read literal	Process Data	Write to PC	
	No	oN		No		No	No	No	No	
	operation	operation	operation	operation		operation	operation	operation	operation	
ĭ	If No Jump:				ž	If No Jump:				
	g	05	Q3	Φ		ō	Q2	O3	Φ	
	Decode	Read literal 'n'	al Process Data	No operation		Decode	Read literal	Process Data	No operation	
Example:	:ejdi	HERE	BNC Jump		Exar	Example:	HERE	BNN Jump		
_	Before Instruction		(mann) cockpto			Before Instruction		(mann) cockpto		
		l	aggices (HEIVE	,		0 - 24	I	(HENE)		
-	Arter Instruction If Carry PC	11 11	0; address (Tumb)			Arter Instruction If Negative PC		(mmilt) seat		
	If Carry PC	11 11	1; address (HERE+2)	+2)		If Negative PC	11 11	1; address (HERE+2)	+2)	

PIC18FXX2

Syntax:	BNOV	Branch if	Branch if Not Overflow	*	BNZ		Branch if	Branch if Not Zero	
if overflow bit is 0' (PC) + 2 + 2n → PC (PC) + 2n → PC	Syntax:	[<i>label</i>] Bl			Synt	ax:	[<i>label</i>] B		
if overflow bit is '0' (PC) + 2 + 2n → PC Sected: None I 110 0.101 mmm mmm mmm If the Overflow bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction. The new address will be PC+2+2n. This instruction is then a two-cycle instruction. Activity: O1	Operands:	-128 ≤ n ≤	127		Oper	ands:	-128 ≤ n ≤	127	
Siatus Affected: The color of t	Operation:	if overflow (PC) + 2 +	bit is '0' 2n → PC		Oper	ation:	if zero bit (PC) + 2 +	is '0' ⊦ 2n → PC	
1110 0101 mnnn mnnn mnnn program will branch. The 2's complement number '2n' is added to the new address will be added to the new address will be a two-cycle instruction, the new address will be a two-cycle instruction is then a two-cycle instruction is then a two-cycle instruction. This instruction is then a two-cycle instruction. Address with the population operation operat	Status Affected:	None			Statu	s Affected:	None		
Description: If the Overflow bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction is the new address will be protected instruction is then a two-cycle instruction. Activity: 1(2) Activity: 1(2) Activity: 1(2) Activity: 1(2) Activity: 1(2) Activity: 1(2) Activity: 1(3) Activity: 1(4) Activity: 1(5) Activity: 1(7) Activity: 1(8) Activity: Activity: 1(9) Activity: Activity: 1(10) Activity: Act	Encoding:	1110	H		Enco	ding:	1110	0001 m	uuuu uut
1	Description:	If the Over program w The 2's co added to th have incre instruction PC+2+2n	flow bit is 'o' ill branch. mplement numented to fe mented to fe, the new add	, then the nmber '2n' is e the PC will the next dress will be tion is then	Desc	rription:	If the Zerc gram will I The 2's cc added to t have incre instruction	bit is '0', the branch omplement in PC. Sin emented to on, the new a on, the new a This instruction of the present of the new and the new	ien the pro- number '2n' is ce the PC wil fetch the nex ddress will be
1(2)	146	a two-cycle	d IIIsii uciioii.		1,447		a two-cyc		_
1(2) 1(2)	Words:	_			Word	JS:	_		
Activity: 1 Jump: Activity: 1 Jump: Activity: 1 Jump: Activity: 1 Jump: Activity: 1 Jump: Activity: 1 Jump: Activity: 1 Jump: Activity: Activi	Cycles:	1(2)			Cyc	es:	1(2)		
Q1 Q2 Q3 Q4 Q1 Q2 Q3 code No No No No No No No No No No No No No No No norration operation operation operation operation operation operation norration norration If No Jump: Q1 Q2 Q3 code Read literal Process No No No No norration norration no No No No No norration no Data Q4 Decode Read literal Process no no No No No No No No no no Data Decode Read literal Process No No no no no No No No No No No no	Q Cycle Activity If Jump:				O 3 ±	ycle Activity: mp:			
code Read literal Increas Process Write to PC Decode Read literal Increas Process Process No No No No No No No No notation operation operation operation operation operation operation operation not Q2 Q3 Q4 Q1 Q2 Q3 code Read literal Process No Process Process Process re Instruction PC address (HBRE) PC address (HBRE) After Instruction PC address (HBRE) If Overflow If Zero PC address (Jump) After Instruction After Instruction After Instruction After Instruction	Ω	02	8	Φ		ō	05	89	8
No No No No No No No No	Decode	Read literal 'n'	Process Data	Write to PC		Decode	Read literal 'n'	Process Data	Write to PC
1	No	No	oN	No		No	No	No	No
Food	operation	operation	operation	operation		operation	operation	operation	operation
Order Read literal notation No Decode notation Read literal notation No Decode notation Read literal notation No Decode notation Read literal notation Process notation No Decode notation Result notation No Decode notation Result notation No Decode notati	If No Jump:				ž	o Jump:			
code Read literal Process No Decode Read literal Process 'n' Data Process 'n' Data Process 'n' Data Process 'n' Data 'n' Data Process Jump HERE BRIZ Jump Before Instruction Before Instruction PC address (HERE) After Instruction After Instruction If Overflow If Zero address (Jump) If Overflow If Zero address (Jump)	δ	Q2	Q3	Q4		δ	05	8	Φ4
HERE BNOV Jump Example: HERE HERE	Decode	Read literal 'n'	Process Data	No operation		Decode	Read literal 'n'	Process Data	No operation
Before Instruction	Example:	HERE	BNOV Jump		Exar	: <u>aldu</u>	HERE		Q.
After Instruction	Before Instri PC	П	dress (HERE)			Before Instru PC	11	Idress (HERE	
	After Instruc If Overflo PC If Overflo	11 11 11	dress (Jump)			After Instruc If Zero PC If Zero	11 11 11	dress (Jump	- (i

DS39564C-page 222 © 2006 Microchip Technology Inc.

DS39564C-page 221

BRA	Unconditi	Unconditional Branch	_	BSF	Bit Set f		
Syntax:	[label] BRA	RA n		Syntax:	[<i>label</i>] B	[label] BSF f,b[,a]	
Operands:	$-1024 \le n \le 1023$	≤ 1023		Operands:	$0 \le f \le 255$	10	
Operation:	(PC) + 2 +	(PC) + 2 + 2n \rightarrow PC			0 ≤ b ≤ 7		
Status Affected:	None				α (O, 1)		
Encoding:	1101	Onnn nm	uuuu uuuu	Operation:	^ <u>1</u>		
Occupation.	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	owolawoo o	- daile	Status Affected:	None		
Describinori.	2n' to the	20, to the PC. Since the PC will	ne PC will	Encoding:	1000	bbba ffff	JJJJ J
	have incre	mented to fe	nave incremented to fetch the next	Description:	Bit b in re	Bit b in register f is set. If 'a' is 0	at. If 'a' is 0
	instruction PC+2+2n	instruction, the new address w	instruction, the new address will be PC+2+2n. This instruction is a		Access Ba	Access Bank will be selected, over-	lected, over
	two-cycle	two-cycle instruction.	5		the bank v	he bank will be selected as per the	d as per the
Words:	_				BSR value	oi.	
Cvcles:	7			Words:	-		
Q Cycle Activity:				Cycles:	-		
ō	02	8	Q 4	Q Cycle Activity:	į.		
Decode	Read literal	Process	Write to PC	۵1	Q2	O 3	Φ
	ļu,	Data		Decode	Read	Process	Write
9	9	oN	No		register f	Data	register f
operation	operation	oneration	operation				

			<u>Example:</u>	BSF	
Example: HERE	RE BRA	dmu√.	Before Instruction	struction	
Before Instruction	entre constant	1	FLAG_REG After Instruction	FLAG_REG After Instruction	II
After Instruction		(neke)	FLAG	FLAG_REG	Ш
S	= address (Jump)	(dunn)			

FLAG REG, 7, 1

Unconditi	Unconditional branch	_	150	ם
[label] BRA	u ∀⊱		Syntax:]
-1024 ≤ n ≤ 1023	≤ 1023		Operands:	0
(PC) + 2 + 2n \rightarrow PC	$2n \to PC$			0 (
None			:	٠ -
1101	unnn nnno	nunn	Operation:	-
20 Od tho 2's	<u> </u>	ot primppor	Status Affected:	Ż
'2n' to the	2n' to the PC. Since the PC will	ie PC will	Encoding:	
have incre	have incremented to fetch the next	tch the next	Description:	m
instruction	, the new ad	instruction, the new address will be		ď
PC+2+2n. two-cycle i	PC+2+2n. This instruction is a two-cycle instruction.	tion is a		££
				m
2			Words:	_
			Cycles:	-
05	93	Ω4	Q Cycle Activity:	
Read literal	Process	Write to PC	ß	
ŗ	Data		Decode	_
%	No	№		Je
operation	operation	operation		

PIC18FXX2

Syntax:	[label] BTFSC	TFSC f,b[,a]	J.		Syntax:	×	[label] BTFSS f,b[,a]	TFSS f,b	,a]	
Operands:	$0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$	2			Operands:	ands:	$0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$	10		
Operation:	skip if $(f < b >) = 0$	0= (<c< td=""><td></td><td></td><td>Operation:</td><td>Operation:</td><td>skip if (f) = 1</td><td>>) = 1</td><td></td><td></td></c<>			Operation:	Operation:	skip if (f) = 1	>) = 1		
Status Allecter	Z [H			olatus	, Allected.	200	Ī		
Encoding:	1011	pppa t	titi	ffff	Encoding:	:Buit	1010	pppa	ffff	ffff
Description:	If bit 'b' in next instrainstr	If bit 'b' in register 'f' is 0, then the next instruction is skipped. If bit 'b' is 0, then the next instruction fetched during the current instruction execution is discarded, and a NoP is executed instead, making this a two-cycle instruction. If 'a is 0, the Access Bank will be selected, overriding the BSR value. If 'al = 1, then the bank will be selected as per the BSR value (default).	0, then bed. ext instance ins	n the struction struction a NOP is a NOP is so a two-e e l., over-1, then oer the	Descr	Description:	If bit 'b' in register "f' is 1, then the next instruction is skipped. If bit 'b' is 1, then then ext instructior fetched during the current instruction execution, is discarded and a NOP is executed instraed, making this a two-cyloel instruction. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).	If bit 'b' in register "f' is 1, then the next instruction is skipped. If bit 'b' is 1, then the next instruction letched during the current instruction execution, is discarded and a NOP is executed instead, making this a two-cyole in instruction. If "a is 0, the Access Bank will be selected, overriding the BSR value. If "a! = 1, then the bank will be selected as per the BSR value (default).	is 1, the tipped. I next instruction in scarded (acad, ma con. If 'a' selected (action). If 'a' = selected (acted as	or the struction struction and a sing this is 0, the 1, over-1, then oer the
Words:	-				Words:	ió	-			
Cycles:	1(2) Note: 3	3 cycles if skip and followed by a 2-word instruction.	and fo	ollowed on.	Cycles:	ï	1(2) Note: 3 by	3 cycles if skip and followed by a 2-word instruction.	kip and ! instruct	ollowed on.
Q Cycle Activity:					Q Cy	Q Cycle Activity:				
ā	Q2	O3		4		ō	O2	Ö		۵ 4
Decode	Read register "	Process Data		No operation		Decode	Read register If	Process Data		No operation
lf skip:					If skip:	ä				
δ	05	පි	-	8	ı	ō	05	Ö		8
No operation	Νο οperation	No operation	do	No operation		No operation	No operation	No operation		No operation
If skip and foll	If skip and followed by 2-word instruction:	rd instruction			If skij	If skip and followed by 2-word instruction:	ed by 2-wor	d instruction	:uc	
ā	Q2	0 3	-	Q4 [ō	02	ဗ		Q 4
No operation	No n operation	No operation	8	No operation		No	No	No		No
No operation	No operation	No operation	g	No operation		No	No	No		No
Example:	HERE JEALSE TRUE	BIFSC FLA	FLAG, 1,	1, 0	Example:	inde:		BTESS	AG,	0
Before Instruction	1	address (HRDE)			ш	Before Instruction	1		ĵ	
After Instruction If FLAG<1> PC If FLAG<1>		0; address (TRUE)	_		∢	After Instruction If FLAG<1> PC If FLAG<1>		address (HEKE) 0; address (FALSE) 1;	KE)	
		CHO FEET OF CASE	í			5	ı			

© 2006 Microchip Technology Inc. DS39564C-page 224

DS39564C-page 223

вта	Bit Toggle f	-		BOV	Branch if Overflow	
Syntax:	[<i>label</i>] B	[label] BTG f,b[,a]		Syntax:	[label] BOV n	
Operands:	$0 \le f \le 255$	10		Operands:	-128 ≤ n ≤ 127	
	0≤b≤7 a∈ [0,1]			Operation:	if overflow bit is '1' (PC) + 2 + 2n → PC	
Operation:	$\langle d \rangle \rightarrow f \langle d \rangle$	⟨ q>		Status Affected:	None	
Status Affected:	None			Encodina:	1110 0100	uuuu uuuu
Encoding:	0111	bbba	ffff ff	ffff Description:	erflow bit is	1. then the
Description:	Bit b in d	Bit b in data memory location f is	ocation f	1	program will branch.	
	inverted. I	inverted. If 'a' is 0, the Access Bank	Access Ba	ank	The 2's complement number '2n' is	number '2r
	will be se	will be selected, overriding the BSR	ding the B	SR	added to the PC. Since the PC will	nce the PC
	value. If 's	value. If 'a' = 1, then the bank will be	e bank wil	lbe	have incremented to fetch the next	fetch the n
	selected a	selected as per the BSR value	R value		instruction, the new address will be	address will
Words.	(deladii).				a two-cycle instruction.	nction is the
: : : : : : : : : : : : : : : : : : :				Words:	-	
Q Cycle Activity:	-			Cycles:	1(2)	
٩	OZ	0 3	8	Q Cycle Activity:		
Decode	Read	Process	Write	If Jump:		(
	register f	Data	register f	. 6	Q2 Q3	Ø

Decode	Read	Process	Š
	register f	Data	regis

BTG PORTC, 4, 0

Before Instruction:
PORTC = 0111 0101 [0x75]
After Instruction:
PORTC = 0110 0101 [0x65]

Ø	Write to PC	8	operation	
8	Process Data	No	operation	
05	Read literal	No	operation	
g	Decode	8	operation	f No.lumn
				Z

	Ω	No	operation
	O3	Process	Data
	Q2	Read literal	ļu,
T No Jump:	۵ م	Decode	
_			

BOV Jump		address (HERE)			address (Jump)		address (HERE+2)
BOV		address		÷	address	ö	address
Example: HERE	Before Instruction	PC =	After Instruction	If Overflow =	PC =	If Overflow =	PC =

PIC18FXX2

	BZ	Branch if Zero	Zero		CALL		Subroutine Call	le Call	
if Zero bit is '1' (PC) + 2 + 2n → PC (PC) + 2n → PC (Syntax:				Syntax				
if Zero bit is '1' (PC) + 2 + 2n → PC None 1110	Operands:	-128 ≤ n ≤	; 127		Operar	:spu	0 ≤ k ≤ 10	148575	
None	Operation:	if Zero bit	is '1'				s ∈ [0,1]		
1110		(PC) + 2 +	- 2n → PC		Operat	ion:	(PC) + 4	→ TOS,	
1110	Status Affected:	None					K → PC<		
If the Zero bit is '1', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fatch the next instruction, the new address will be protected instruction. 1	Encoding:	1110					(W) (W)	ý.	
Activity: 23	Description:	If the Zerc gram will b	bit is '1', the branch.	en the pro-			(STATUS) (BSR) →) → STATUS BSRS	ý,
Activity: Activity: Code Read literal Process No		The 2's cc	omplement no	umber '2n' is	Status	Affected:	None		
14 15 15 15 15 15 15 15		have incre	mented to fe	atch the next	Encodi	ng:			
1 (2)		instructior PC+2+2n.	, the new ad This instruc	dress will be tion is then	1st wo	rd (k<7:0>) ord(k<19:8>	1110		
1(2)		a two-cycl	le instruction	٠	Descri	otion:	Subroutin	e call of enti	re 2 Mbyte
Activity: 1(2) Activity: Code Read literal Process Write to PC	Words:	-					memory r	ange. First,	return
STATUS and BSR regis also pushed into their regis and bushed into their registers and bushed into their regi	Cycles:	1(2)					return sta	ck. If 's' = 1,	the W,
Standard	Q Cycle Activity:						STATUS also push	and BSR reg	listers are
According Process Write to PC		Ö	Ö	0			shadow re	egisters, WS	, STATUSS
No	Decode	Read literal	Process	Write to PC			and BSR8 occurs (de	S. If S = 0, r efault). Then	no update , the 20-bit
10 10 10 10 10 10 10 10	No.	No.	No.	No.			value 'K' is	s loaded into	PC<20:1>
Code Read literal Process No	operation	operation	operation	operation	Words		2	200	
Process No	<u>:</u>	ò	Ö	04	Cycles		2		
HERE Frocess No O Cycle Activity	3	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3	5 2	2006	. <u>-</u>	1		
HERE BZ Jump Decode Read literal Push PC to stack	Decode	head literal	Process	operation	S S S S	le Activity: Q1	02	03	04
No	Example:	HERE				Decode	Read literal 'k'<7:0>,	Push PC to stack	Read literal 'k'<19:8>,
Operation Oper	Before Instru PC	11	dress (HERE	~		N _o	N _o	S _O	Write to PC No
= 1: = address (Jump)	After Instruct	ion				peration	operation	operation	operation
= address (HERE+2) Before Instruction PC = address After Instruction PC = address PC = BSH BSHS = BSH STATUSS= STATUS	If Zero PC	11 11 1	ldress (Jump		Examp	<u>:</u>	HERE		RE, 1
## After Instruction After Instruction From a address TOS = address WS = address WS = WHINS BSRS = BSR STATUSS = STATUS	Od BZ	1 11	Idress (HERE	+2)	B	efore Instru			
address address W BSR STATUS					74	PC		S (HERE)	
					₹	TOS WS		S (THERE) S (HERE + 4	<u></u>
						BSRS STATUS		S	

© 2006 Microchip Technology Inc. DS39564C-page 226

DS39564C-page 225

CLRF	Clear f			CLRWDT	Clear Wat	Clear Watchdog Timer	÷
Syntax:	[<i>label</i>] CI	[<i>label</i>] CLRF f [,a]		Syntax:	[label] CLRWDT	CLRWDT	
Operands:	$0 \le f \le 255$			Operands:	None		
	a∈ [0,1]			Operation:	000h → WDT.	/DT.	
Operation:	$\begin{array}{c} 000h \rightarrow f \\ 1 \rightarrow Z \end{array}$			-	000h → V 1 → <u>TO</u> ,	$000h \xrightarrow{\rightarrow} WDT \ postscaler, \\ 1 \xrightarrow{TO},$	er,
Status Affected:	Z				1 → PD		
Encoding:	0110	101a ffff	£ ££££	Status Affected:	<u>TO, PD</u>		
Description:	Clears the	contents of t	Clears the contents of the specified	Encoding:	0000	0000 0000	0010
-	register. If	'a' is 0, the ∕	egister. If 'a' is 0, the Access Bank	Description:	CLRWDT ir	CLRWDT instruction resets the	ets the
	will be sele	will be selected, overriding the BS	will be selected, overriding the BSR		Watchdog	Watchdog Timer. It also resets the	o resets the
	be selecte	be selected as per the BSR value	BSR value		TO and PD are set.	D are set.	Status Dits
	(default)			Words:	-		
Words:	-			Cvcles:	-		
Cycles:	-			Q Cvcle Activity:			
Q Cycle Activity:				, S	05	ဗ	Q
ō	05	Ö	Q	Decode	oN N	Process	ž
Decode	Read	Process	Write		operation	Data	operation
	register f	Data	register f				

FLAG REG, 1

CLRF

Example:

Before Instruction FLAG_REG = 0x5A After Instruction FLAG_REG = 0x00

PIC18FXX2

Syntax: Operands: Operation: Status Affected: Encoding:	[lahal]						ı	ادءا
Operands: Operation: Status Affected: Encoding:		SOM F	f [,d [,a]		Syntax:	[<i>label</i>]	CPFSEC	<u>ਰ</u>
Operation: Status Affected: Encoding:	$0 \le f \le 255$ $d \in [0,1]$	10			Operands:	0≤f≤255 a∈[0,1]	ю	
Status Affected: Encoding:	$a \in [0,1]$ $(\overline{f}) \rightarrow dest$	sst			Operation:	(f) - (W), skip if $(f) = (W)$	(W) =	
Encoding:	N, Z					_	(unsigned comparison)	<u> </u>
00000	0001	11da	ffff	ffff	Status Affected:	None	-	_
Description.	The contents of register If are com-	nts of rec	jister 🖁	are com-	Encoding:	0110	001a ff	ffff ffff
	plemented. If d' is 0, the result is stored in W. If d' is 1, the result is	A. If dis. N. If dis	0, the re 1. the r	ssult is esult is	Description:	Compares memory Ic	Compares the contents of data memory location 1f to the conte	Compares the contents of data memory location If to the contents
	stored back in register 17 (default). If	ok in regis	ter II (d	efault). If		of W by p	of W by performing an unsigned	unsigned
	a Is 0, the Access Bank will be selected, overriding the BSR value.	e Access overriding	bank w the BS	ill be R value.		subtraction. If 'f' = W, the	subtraction. If 'f' = W, then the fetched instruc-	hed instruc-
	If 'a' = 1, then the bank will be selected as per the BSR value	hen the t is per the	ank wil. BSR va	l be alue		tion is disc cuted inst	tion is discarded and a NOP is excuted instead, making this a two-	tion is discarded and a NOP is executed instead, making this a two-
	(default).					cycle insti	cycle instruction. If 'a' is 0, the Access Bank will be selected.	cycle instruction. If 'a' is 0, the Access Bank will be selected over-
Words: Cycles:						riding the	BSR value.	riding the BSR value. If 'a' = 1, then the bank will be selected as per the
Q Cycle Activity:						BSR value	BSR value (default).	
D	Q2	Q3		Q4	Words:	-		
Decode	Read register f*	× = 1		Write to destination	Cycles:	1(2) Note: 3 0	cycles if skip	3 cycles if skip and followed
Example:	COMF	REG, 0,	0 ,				by a 2-word instruction.	struction.
Before Instruction REG =	ction = 0x13				Q Cycle Activity:		(Ċ
After Instruction					6	05	පි	8
REG	= 0x13				Decode	Read register f	Process Data	No operation
3					If skip:	,	,	,
					5	Q 2	පි	8
					No operation	No	No operation	No
					If skip and followed by 2-word instruction:	wed by 2-wor	d instruction	
					ğ	05	ဗ	8
					No	No	No	No
					Operation	No	No	No
					operation	operation	operation	operation
					Example:	HERE NEQUAL EQUAL	CPFSEQ REG,	g, 0
					Before Instruction PC Address W	11 11	HERE	
					REG After Instruction	П		
					If REG	П	, w.	į
					# REG	II 14	iguess (EQUA	(1)
					ď	II	Address (NEQUAL)	JAL)

DS39564C-page 228 © 2006 Microchip Technology Inc.

DS39564C-page 227

CPFSGT	Сотра	Compare f with W, skip if f > W	W, ski	p if f > W	CPFSLT	Compare	Compare f with W, skip if f < W	kip if f < W
Syntax:	[label]	[label] CPFSGT	T f[,a]		Syntax:	[label] CPFSLT		f [,a]
Operands:	$0 \le f \le 255$ $a \in [0,1]$	255]			Operands:	$0 \le f \le 255$ $a \in [0,1]$	2	
Operation:	(f) – (W), skip if (f) (unsigned	(f) – (W), skip if (f) > (W) (unsigned comparison)	arison)		Operation:	(f) - (W), skip if $(f) < (W)$ (unsigned com	(f) – (W), skip if (f) < (W) (unsigned comparison)	ē
Status Affected:	None				Status Affected:	None		
Encoding:	0110	010a	ffff	IJJJ J	Encoding:	0110	000a ff	IIII IIII
Description:	Compa memory of the V unsigne If the cc the conf	Compares the contents of data memory location if to the conten of the W by performing an unsigned subtraction. If the contents of if are greater the the contents of WHEG, then the the contents of whe was a proper of the subtraction is discontent of the contents of where, then the contents of where, then the	T to the trop or trop	Compares the contents of data memory location ¹ fo the contents of the W by performing an unsigned subtraction. If the contents of ¹ f are greater than the contents of WEG, then the forched negative finds of the contents of WEGs, then the forched negative finds of the contents of WEGs.	Description:	Compares to memory loc of W by persubtraction. If the content the content instruction instruction is	Compares the contents of data memory location if to the content of W by performing an unsigned subtraction. If the contents of if are less than the contents of W, then the fetch intercentant is a discoursed.	Compares the contents of data memory location # to the contents of W by performing an unsigned subtraction. If the contents of \(Y \) are less than the contents of \(W \) then the fetched in the contents of \(W \) then the fetched and a transfer is disconded and a transfer is d
	a NOP is this a tw 0, the A selected If 'a' = 1	a not present a construction of a not be a not b	d insternations and instructions will instructions will into the bank he BSF	Accession and a second of a second of a second of a second of instead, making this a two-cycle instruction. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value.	Words:	is execute two-cycle Access Bi is 1, the B (default).	ad instead, nistraction. instruction. ank will be siSR will not k	is executed instead, making this as is executed instead, making this a two-cycle instruction. If 'a' is 0, the Access Bank will be selected. If 'a' is 1, the BSR will not be overridden (default).
Words:	(default)	œ.			Cycles:		20 m	- T
Cycles:	1(2)					Note: 3	3 cycles II skip and rolld by a 2-word instruction.	3 cycles II skip and followed by a 2-word instruction.
	Note:	3 cycles if skip and follo by a 2-word instruction.	if skip a ord inst	3 cycles if skip and followed by a 2-word instruction.	Q Cycle Activity:		(
O Cycle Activity:					5	05	8	Ğ :
a cycle returns.	8	ď	ဗ	Q 4	Decode	Read register f	Process Data	No operation
Decode	Read register 'f	Process Data	ess	No operation	If skip:	S	Ö	S
If skip:		•			o N	N S	S N	S N
5 2	S 2	ဗို	S 2	2 2	operation	operation	operation	operation
operation	operation	do	ation	operation	If skip and followed by 2-word instruction: Q1 Q2 Q3	ved by 2-wor O2	d instruction O3	:: 04
If skip and followed by 2-word instruction:	ed by 2-w	ord instru	ction:		8	N N	S S	8
ē	8	g	ဗွ	8	operation	operation	operation	operation
No operation	No operation	No n operation	o ation	No operation	No operation	No operation	No operation	No operation
No operation	No operation	No n operation	o ation	No operation	Example:	HERE	CPFSLT REG.	
Example:	HERE	CPFS	CPFSGT REG,	0 '5		NLESS		
	NGREATER GREATER				Before Instruction PC	II	dress (HERI	(3)
Before Instruction	ction				N -	Ш	5	ì
S ≽	11 11	Address (HERE) ?	(HERE)		Alter Instruction If REG	٧		
After Instruction	ion				PC		Address (LESS)	3)
If REG PC	∧ II	dress	(GREATER)	ER)	P = O	. N Adv.	vv, Address (NLESS)	ss)
# REG PC	VI II	W; Address (NGREATER)	(NGREA	TER)				

© 2006 Microchip Technology Inc.

© 2006 Microchip Technology Inc.

DS39564C-page 230

PIC18FXX2

DAW	Decimal Adjust W Register	DECF	Decrement f	
Syntax:	[label] DAW	Syntax:	[label] DECF f[,d[,a]	ı
Operands:	None	Operands:	$0 \le f \le 255$	
Operation:	If [W<3:0> >9] or [DC = 1] then $(W<3:0>) + 6 \rightarrow W<3:0>$:		d∈ [0,1] a∈ [0,1]	
	ese	Operation:	(f) $-1 \rightarrow dest$	
	$(W<3:0>) \to W<3:0>;$	Status Affected:	C, DC, N, OV, Z	
	If [W<7:4> >9] or [C = 1] then	Encoding:	0000 01da ffff ffff	
	$(W<7:4>) + 6 \rightarrow W<7:4>;$ else	Description:	Decrement register If. If 'd' is 0, the result is stored in W. If 'd' is 1, the	Φ.
	$(W<7:4>) \rightarrow W<7:4>;$		result is stored back in register "	
Status Affected:	O		(default). If 'a' is 0, the Access Book will be coloofed everyiding	
Encoding:	0000 0000 0111		the BSR value. If $a' = 1$, then the	
Description:	DAW adjusts the eight-bit value in W, resulting from the earlier addi-		bank will be selected as per the BSR value (default).	
	tion of two variables (each in	Words:	-	
	packed BCD formal, and produces a correct packed BCD result.	Cycles:	-	
Words:	-	Q Cycle Activity:		
Cvcles:	-	δ	Q2 Q3 Q4	٦
Q Cycle Activity		Decode	Read Process Write to	
Ø	Q2 Q3 Q4			1
Decode	Read Process Write	Example:	DECF CNT, 1, 0	
- Foliamora	3	Before Instruction	tion	
EXAMIPIEL: DAY Refore Instruction	DAW Iotion	ONT Z	= 0x01 = 0	
		After Instruction		
≥ o o o	= 0xA5 = 0 = 0	CNT	= 0x00 = 1	
After Instruction	tion			
W C DC Example 2:	= 0005 = 0			
Before Instruction	uction			
≥ 00	= 0xCE = 0 = 0			
After Instruction	tion			
≥ 00	= 0x34 = 1 = 0			

DECFSZ	Decremen	Decrement f, skip if 0		DCFSNZ	Z	Decremer	Decrement f, skip if not 0	ot 0
Syntax:	[<i>label</i>] D	[d [,a]]	Syntax:		[label] DCFSNZ		f [,d [,a]
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$			Operands:	g::	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	10	
Operation:	(f) $-1 \rightarrow \text{dest}$, skip if result = 0	est, It = 0		Operation:	ou:	(f) $-1 \rightarrow dest$, skip if result $\neq 0$	lest, ult ≠ 0	
Status Affected:	None			Status A	Status Affected:	None		
Encoding:	0010	11da ffff	jjjj j	Encoding:	:Bi	0100	11da ffff	f ffff
Description:	The contents of regiremented. If 'd' is 0, placed in W. If 'd' is placed in W. If 'd' is placed in W. If 'd' is blaced in W. If 'd' is blaced had back in registion, which is already discarded, and a No instruction. If 'a is 0, Bank will be selected the SEN value. If 'a' the BSR value. If 'a' the BSR value (I' a' the Selected BSR value (I' a' the SEN value (I' a' t	The contents of register II are decremented. If 'd' is 0, the result is placed in W. If 'd' is 1, the result is placed back in register II (default). If the result is 0, the next instruction, which is already fatched, is discarded, and a voro is executed instruction. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the Bank will be selected, as per the BSR value (default).	The contents of register II are decremented. If 'd' is 0, the result is blaced in W. If 'd' is 1, the result is blaced back in register I' (default). If the result is 0, the next instruction, which is already fatched, is ilsoarded, and a NoP is executed nstruction. If 'd' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the BSR value (default).	Description:	tion:	The conterted placed in V placed bad placed bad placed bad placed bad in the result in struction fetched, is executed is executed is executed in struction with the horizon part the BS per the BS	The contents of register # are decremented. If d' is 0, the result is placed in W. If d' is 1, the result is placed back in register # (default). If the result is not 0, the next instruction, which is already eletched, is discarded, and a NOP is executed instead, making it a two-cycle instruction. If a' is 0, the Access Bank will be selected. Access Bank will be selected, overriding the BSR value. If a' = 1, then the bank will be selected.	r If are dec- s result is the result is a next eady und a NOP is ing it a two- s 0, the lected, lected, a. If a' = 1, alected as ault).
Words:	-			Words:		-		
Cycles:	1(2) Note: 3 cy by <i>a</i>	3 cycles if skip and follo by a 2-word instruction.	3 cycles if skip and followed by a 2-word instruction.	Cycles:		1(2) Note: 3 c by	3 cycles if skip and follo by a 2-word instruction.	3 cycles if skip and followed by a 2-word instruction.
Q Cycle Activity:				Q Cyd	Q Cycle Activity:			
ö	02	03	Q 4		5	Q2	89	Q4
Decode	Read register 'f	Process Data	Write to destination		Decode	Read register f*	Process Data	Write to destination
If skip:	S	3	5	If skip:	5	S	Š	5
3 2	3 :	3 :	ţ :	L	3 2	3 2	3 2	į :
No operation	No operation	No operation	No operation	5	No operation	No operation	No operation	No operation
If skip and followed by 2-word instruction:	ed by 2-word	I instruction:		If skip	and follow	ed by 2-word	If skip and followed by 2-word instruction:	
ğ	Q2	ဗ	Q4		ō	05	89	Q4
No operation	No operation	No operation	No operation	5	No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation	6	No operation	No operation	No operation	No operation
<u>Example</u> :	HERE	DECFSZ	CNT, 1, 1	<u>Example:</u>	öί	HERE I	DCFSNZ TEMP,	P, 1, 0
	CONTINUE		1000			NZERO		
Before Instruction PC =	ction = Address	(HERE)		Bef	Before Instruction TEMP	ction =	<i>د.</i>	
After Instruction				Affe	After Instruction	on		
CNT If CNT PC	= CNT-1 = 0; = Address	(CONTINUE)			TEMP If TEMP PC	11 11 11	TEMP - 1, 0; Address (ZERO)	ERO)
If CNT PC	* II	(HERE+2)			If TEMP PC	* II	0; Address (NZERO)	IZERO)

© 2006 Microchip Technology Inc.

PIC18FXX2

INCF Increment f	Syntax: [label] INCF f[,d[,a]	Onerands: 0 < f < 255			_	kkkk ₀ Status Affected: C, DC, N, OV, Z	kkkk ₈ Encoding: 0010 10da ffff ffff	Description:	ire incremented. If 'd' is 0, the result is e 20-bit placed in W. If 'd' is 1, the result is		selected, overriding the BSR value.	If 'a' = 1, then the bank will be	selected as per the BSR value	(default).	Words: 1	Q4 Cycles:			No Decode Read Process Write to	eration register Y Data destination	Example: INCF CNT, 1, 0	truction	" " " ruction	$\begin{array}{rcl} CNT & = & 0x00 \\ Z & = & 1 \end{array}$
Unconditional Branch	OTO k	575	S			1111 k ₇ kkk kk	k ₁₉ kkk kkkk kk	GOTO allows an unconditional	branch anywhere within entire 2 Mbyte memory range. The 20-bit	value 'k' is loaded into PC<20:1>	iys a two-cycle					Q3 Q4	_	operation K < 19:8>, Write to PC	No	operation operation		E C	(RP)	
ondition	[label] GOTO	0 < k < 1048575	k → PC<20:1>	None			1111	GOTO allows	branch anyw 2 Mbyte mer	value 'k' is lo	instruction.	2	1 1	2		Q2	=	K :U ,	No	operation	GOTO THERE	After Instruction PC - Address (THEPE)	GUT) SSS	
n	_	Ĭ				Ist word (k<7:0>)	2nd word(k<19:8>)								Q Cycle Activity:									

DS39564C-page 232 © 2006 Microchip Technology Inc.

INCFSZ	ncrement	Increment f, skip if 0		INFSNZ	NZ	Incremen	Increment f, skip if not 0	o t 0
Syntax:	[<i>label</i>]	INCFSZ f[,	f [,d [,a]	Syntax:	:XE	[label]	INFSNZ f[f [,d [,a]
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	10		Oper	Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	10	
Operation:	(f) + 1 \rightarrow dest, skip if result = 0	dest, ult = 0		Oper	Operation:	(f) + 1 \rightarrow dest, skip if result \neq 0	test, ult ≠ 0	
Status Affected:	None			Statu	Status Affected:	None		
Encoding:	0011	11da ffff	IIII I	Enco	Encoding:	0100	10da ffff	III EEEE
Description:	The conterincement placed in V placed in V placed bad lif the result iton, which discarded, instead, m instruction Bank will the BSP w will be BSR value	The contents of register if are incremented. If 'd' is 0, the result is placed in W. If 'd is 1, the result is placed back in register if. (default) if the result is 0, the next instruction, which is already fetched, is alksady fetched, is alksady fetched, is discarded, and a NOP is executed instruction. If 'd' is 0, the Access Bank will be selected, overriding the BSR walle. If 'd' = 1, then the BSR value. If 'd' = 1, then the BSR value (default).	in the result is X instruction at the result is a secured a sexecuted a sexecuted a sexecuted be Access overriding in the the sexecuted is the result in the	Desc	Description:	The conterior and a second of the second of	The contents of register if are incremented. If 'd' is 0, the result is placed in W. If 'd' is 1, the result is placed back in vegister if (default). If the result is not 0, the next instruction, which is already etched, is discarded, and a NOP is executed instead, making it a two-cycle instruction. If 'a' is 0, the Access Bank will be selected, over riding the BSR value. If 'a' is 0, the Access Bank will be selected, over the bank will be selected as per the BSR value (default).	The contents of register " are incremented, If "d" is 0, the result is placed in W. If "d" is 1, the result is placed back in register " (default). If the result is not 0, the next instruction, which is already eletched, is discarded, and a NOP is executed instead, making it a two-cycle instruction. If "a" is 0, the Access Bank will be selected, over-riding the BSR value. If "a" = 1, then the bank will be selected as per the BSR value (default).
Words:	-			Words:	<u>s:</u>	-		
Cycles:	1(2) Note: 3 cy by 8	3 cycles if skip and followed by a 2-word instruction.	and followed ruction.	Cycles:	:se	1(2) Note: 3 c by	3 cycles if skip and follc by a 2-word instruction.	3 cycles if skip and followed by a 2-word instruction.
Q Cycle Activity:				O O	Q Cycle Activity:			
20	05	O3	Φ4	'	Ω	Q2	G3	Φ4
Decode	Read register 'f'	Process Data	Write to destination		Decode	Read register f*	Process Data	Write to destination
If skip:				If skip:				
ō	8	ප	Q 45		8	Ø5	පි	φ
No operation	No operation	No operation	No operation		No operation	No operation	No operation	No operation
If skip and followed by 2-word instruction:	ed by 2-word	d instruction:		# Sk	ip and follow	ed by 2-wor	If skip and followed by 2-word instruction:	
ō	Q2	83	Q 4	ı	Q	Q2	ဗ	Q4
No operation	No operation	No operation	No operation		No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation		No operation	No operation	No operation	No operation
Example:	HERE I NZERO : ZERO :	INCFSZ CN:	CNT, 1, 0	Example:	: <u>aldr</u>	HERE ZERO NZERO	INFSNZ REG,	1, 1, 0
Before Instruction PC =	ction = Address	s (HERE)			Before Instruction PC =	ction = Address	(HERE)	
After Instruction CNT = If CNT = PC = If CNT ≠ If CNT ← If CNT	ion = CNT+1 = 0; = Address = Address	1 s (ZERO) s (NZERO)			After Instruction REG = If REG ≠ PC = If REG =		REG + 1 0; Address (NZERO) 0; Address (ZERO)	

PIC18FXX2

Syntax:	[label] IORLW k	Syntax:	[label] IORWF f [,d [,a]	f [,d [,a]
Operands:	$0 \le k \le 255$	Operands:	$0 \le f \le 255$	
Operation:	(W) OR $k \rightarrow W$		d∈ [0,1]	
Status Affected:	N, N		a = [0,1]	
Encoding:	0000 1001 KKK KKK	Operation:	(w) .HO. (w) → dest	
Dosoription.	The contents of W are OB'ed with	Status Affected:	N, Z	
Description:	the eight-bit literal 'k'. The result is	Encoding:	0001 00da	ffff ffff
	placed in W.	Description:	Inclusive OR W with register 'f'. If 'd'	register f'. If 'd'
Words:	-		is 0, the result is placed in W. If d	ced in W. If d
Cycles:	-		is 1, the result is placed back in register If (default) If 'a' is 0, the	ced back In If 'a' is 0 the
Q Cvcle Activity:			Access Bank will be selected, over-	selected, over-
δ	Q2 Q3 Q4		riding the BSR value. If 'a' = 1, then	. If 'a' = 1, then
Decode	Read Process Write to W		the bank will be selected as per the BSR value (default).	cted as per the
		Words:	-	
Example:	IORLW 0x35	Cycles:	-	
Before Instruction	ion	Q Cycle Activity:		
X	= 0x9A	ō	Q2 Q3	Q
After Instruction	C	Decode	_	
M	= 0xBF		register 🖁 💮 Data	destination
		<u>Example</u> :	IORWF RESULT, 0, 1	1
		Before Instruction	ction	
		RESULT = W	= 0x13 = 0x91	
		After Instruction RESULT =	on = 0x13 = 0x03	

DS39564C-page 234 © 2006 Microchip Technology Inc.

DS39564C-page 233

LFSR	Load FSR	~			MOVF	Move f			
Syntax:	[<i>label</i>]	[label] LFSR f,k			Syntax:	[label] MOVF f [,d [,a]	JVF	f [,d [,a]	
Operands:	$0 \le f \le 2$ $0 \le k \le 4095$	95			Operands:	$0 \le f \le 255$ $d \in [0,1]$			
Operation:	k → FSRf					a∈ [0,1]			
Status Affected:	None				Operation:	f → dest			
Encoding:	1110	1110	3300	k ₁₁ kkk	Status Affected:	N, N			
ı	1111	0000	k_7kkk	kkkk	Encoding:	0101 00	00da	ffff	ffff
Description:	The 12-bit the file se	The 12-bit literal 'k' is loaded into the file select register pointed to	s loade er point	ed into ted to	Description:	The contents of register If are moved to a destination dependent	of reg	jister 1° a	re
	by f".					upon the status of 'd'. If 'd' is 0, the	ns of	'd'. If 'd' i	s 0, the
Words:	2					result is placed in W. If d is 1, the	ed in V	N. If d is	1, the
Cycles:	2					result is placed back in register #	ed bac	X in regi	ster †
Q Cycle Activity:						where in the 256 byte bank. If 'a' is	256 b	yte bank	fiyis Ifa'is
ē	02	ဗ		Q		0, the Access Bank will be	s Bank	will be	
Decode	Read literal	Process		Write		selected, overriding the BSR value.	ərridinç	the BSF	s value.
	k MSB	Data	≛	iteral 'k'		If 'a' = 1, then the bank will be	n the b	ank will	96
			≥ 11	MSB to FSRfH		selected as per the BSR value (default).	er the	BSR va	en
Decode	Read literal	Process		Write literal	Words:	-			
	k LSB	Data	χ Σ	'k' to FSRfL	المامان.	•			

LFSR 2, 0x3AB

After Instruction FSR2H = 0x03 FSR2L = 0xAB

Q3 Process Data Cycres.
Q Cycle Activity:
Q1 Q2 C
Decode Read Pr

MOVF REG, 0, 0

| Example: MOVF REG, 0, Before Instruction REG = 0x22 W = 0xFF After Instruction REG = 0x22 W = 0x22

Q4 No operation

Q3 Process Data

Decode 5

Write register 'f' (dest)

No operation

Read register II (src) No operation No dummy read

Example: MOVFF REG1, REG2

 Before Instruction
 6x33

 REG1
 6x11

 After Instruction
 6x11

 REG1
 6x33

 REG2
 6x33

PIC18FXX2

MOVFF	Move f to f	ţ,			MOVLB		Move liter	Move literal to low nibble in BSR	ibble in E	SSR
Syntax:	[<i>label</i>]	MOVFF f _s ,f _d	f _s ,f _d		Syntax:		[label]	MOVLB k		ı
Operands:	$0 \le f_s \le 4095 \\ 0 \le f_d \le 4095$	1095 1095			Operands:	ds:	0 ≤ k ≤ 255 k → BSB	10		
Operation:	$(\mathfrak{f}_s) \to \mathfrak{f}_d$				Status A	Status Affected:	None			
Status Affected:	None				Encoding:		0000	0001 K	kkk kk	kkkk
Encoding: 1st word (source) 2nd word (destin.)	1100	ffff	ffff ffff	fiffs fiffd	Description:	tion:	The 8-bit II	The 8-bit literal 'K' is loaded into the Bank Select Register (BSR).	ster (BSF	ج ج
Description:	The con	The contents of source register ¹ s are moved to destination register	ource reg	jister f _s egister	Words: Cycles:					
	f _d '. Loci anywher	f _d : Location of source f _s can be anywhere in the 4096 byte data	ource f _s .096 byte	can be data	Q Cycle	Q Cycle Activity:	S	č	5	
	space (0	space (000h to FFFh), and location	Fh), and	location be seed		Decode	Read literal	Process	Write	
	where fr	where from 000h to FFFh.	to FFFh.	ne ally-			<u>-</u> 7	Data	literal 'k' to BSR	<u>و</u>
	Either so	Either source or destination can be	estinatio	n can be]
	W (a use	vv (a userul special situation). MOVFF is particularly useful for	al situatio arly usefu	on). Il for	Example:	öi	MOVLB 5			
	transferr to a peri	transferring a data memory location to a peripheral register (such as the	memory	location	Bef	Before Instruction BSR register	ction ster = 0x02	25		
	transmit	transmit buffer or an I/O port)	an I/O po	ort).	Afte	After Instruction				
	The MOV the PCL the desti	The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.	ction can OSH or gister	not use TOSL as		BSR register	ster = 0x05	92		
	Note:	The MOVFF instruction should not be used to r	F instruct	The MOVFF instruction should not be used to mod-						
		ify interrupt settings while any interrupt is enabled. See Section 8.0 for more information.	pt setting upt is en ion 8.0 fc	s while abled. r more						
Words:	8									
Cycles:	2 (3)									
Q Cycle Activity:	į	i								

© 2006 Microchip Technology Inc. DS39564C-page 236

DS39564C-page 235

MOVLW	Move literal to W	ral to W		MOVWF	Move W to f	
Syntax:	[<i>label</i>]	[label] MOVLW k		Syntax:	[label] MOVWF f[,a]	I
Operands:	$0 \le k \le 255$	55		Operands:	$0 \le f \le 255$	
Operation:	k ↓ W				a∈ [0,1]	
Status Affected:	None			Operation:	$(W) \to f$	
Encoding:	0000	1110 kkk	kk kkk	Status Affected:	None	
Description:	The eight	The eight-bit literal 'k' is loaded	s loaded	Encoding:	0110 111a ffff ffff	9
-	into W.			Description:	Move data from W to register If.	
Words:	-				Location If can be anywhere in the	the
Cycles:	-				Access Bank will be selected, over-	ver-
Q Cycle Activity:					riding the BSR value. If 'a' = 1, then	neu
6	02	0 3	Q 4		the bank will be selected as per the	the
Decode	Read	Process	Write to W		BSR value (detault).	
	literal 'k'	Data		Words:	-	
				-		

words.

Cycles:

Q Cyde Activity:

Q1

Decode

Read

Decode

Read

MOVLW 0x5A

After Instruction W = 0x5A

 Q1
 Q2
 Q3

 Decode
 Read
 Process

 register **
 Data

Q4 Write register 14

PIC18FXX2

MULLW	Multiply Li	Multiply Literal with W	~	MULWF	Multiply W with f	W with f			_
Syntax:	[label] MULLW	NULLW k		Syntax:	[label] MULWF	MULWF	f [,a]		
Operands:	$0 \le k \le 255$			Operands:	$0 \le f \le 255$	2			
Operation:	$(W) \times k \rightarrow I$	(W) x k → PRODH:PRODL	IODL		a ∈ [0,1]				
Status Affected:	None			Operation:	(W) × (f)	$(W) \times (f) \rightarrow PRODH:PRODL$:PROL	7	
Encoding:	0000	1101 kk	kkkk kkkk	Status Affected:	None				_
Description:	An unsigne	An unsigned multiplication is car-	tion is car-	Encoding:	0000	001a f	ffff	ffff	
	ried out bei	ried out between the contents of	ontents of	Description:	An unsign	An unsigned multiplication is car-	cation	is car	
	16-bit resul	w and the o-bit literal k. 16-bit result is placed in	- - - -		Wand the	N and the register file location F.	e locat	tion F.	
	PRODH:PF	PRODH:PRODL register pair.	ter pair.		The 16-bi	The 16-bit result is stored in the	stored i	in the	
	W is unchanged.	nged.	igii byte.		PRODH	PRODH contains the high byte.	e high	byte.	
	None of the	None of the status flags are	s are		Both W ar	Both W and If are unchanged.	nchang	ged.	
	Note that n	Note that neither overflow nor	low nor		affected.			,	
	carry is pos	carry is possible in this opera-	s opera-		Note that	Note that neither overflow nor	erflow his one	nor	
	not detected	d.	nagiologic		tion. A zer	tion. A zero result is possible but	dissod	le but	
Words:	-				not detect	not detected. If 'a' is 0, the	0, the	7	
Cycles:	-				overriding	Access ballk will be selecte overriding the BSR value. If	select	, = ec,	
Q Cycle Activity:					'a' = 1, the	a' = 1, then the bank will be	k will b	Ф	
Ω	Q2	Q3	Φ4		selected	selected as per the BSR value	BSR v	alne	
Decode	Read literal 'k'	Process Data	Write	Words:	(derault).				
			PRODH:	Cvcles:	-				
			PHODL	Q Cycle Activity:					
Example:	MULLW 0:	0xC4		Ω	Q2	Q3		Q 4	
Before Instruction	ction			Decode	Read	Process	- 1	Write	
W PRODH	= 0xE2 = ?	N			register i	Data	- E E	PRODH: PRODL	
After Instruction									
W PRODH	= 0xE2 = 0xAD	ρ.ς		Example: MUI Before Instruction	IME	REG, 1			
PRODL	= 0x08	∞		>		2			
				REG PRODH PRODL	1 11 11 11	0xB5			
				Arrer Instruction		į			
				W REG PRODH PRODL	8888	0xC4 0xB5 0x8A 0x94			

DS39564C-page 238 © 2006 Microchip Technology Inc.

DS39564C-page 237

NEGF	Negate f			NOP	No Operation	ıtion		
Syntax:	[label] NEGF f[,a]	ìF f[,a]		Syntax:	[label] NOP	NOP		
Operands:	$0 \le f \le 255$			Operands:	None			
	a∈ [0,1]			Operation:	No operation	tion		
Operation:	$(\overline{\mathfrak{f}})$ + 1 \rightarrow f			Status Affected:	None			
Status Affected:	N, OV, C, DC, Z	Z		Encodina:	0000	0000	0000	0000
Encoding:	0110 110a	la ffff	ffff	ò	1111	xxxx	xxxx	XXXX
Description:	Location 'f' is negated using two's	negated us	sing two's	Description:	No operation.	tion.		
	complement. The result is placed in	The result i	s placed in	Words:	-			
	the data memory location 'f'. If 'a' is 0 the Access Bank will be	ory location Bank will b	ا". اجزیع/is	Cycles:	-			
	selected, overriding the BSR value.	riding the E	3SR value.	Q Cycle Activity:				
	If 'a' = 1, then the bank will be	the bank v	vill be	Ø	Ø	ဗ		8
	selected as per the BSR value.	er the BSR	value.	Decode	N _o	8		9
Words:	-				operation	operation	-	operation
Cycles:	-							
Q Cycle Activity:				<u>Example:</u>				
ō	Q2	83	Q 4	O CON				
Decode	Read	Process	Write	Olio				
	register f	Data	register f					

Example: NEGF REG, 1

Before Instruction
REG = 011 1010 [0x3A]

After Instruction
REG = 1100 0110 [0xC6]

PIC18FXX2

POP	Po	Pop Top of Return Stack	Return S	tack		PUSH		Push To	p of Re	Push Top of Return Stack	×
Syntax:	9]	[label] POP	d.			Syntax:		[<i>label</i>]	PUSH		
Operands:	No	None				Operands:	ids:	None			
Operation:	Ë	$(TOS) \rightarrow bit bucket$	bucket			Operation:	ion:	$(PC+2) \rightarrow TOS$	→ T0S		
Status Affected:		None				Status	Status Affected:	None			
Encoding:	Ĺ	0000	0000	0000	0110	Encoding:	ığ:	0000	0000	0000	0101
Description:	,	The TOS value is pulled off the eturn stack and is discarded. The FOS value than becomes the providence that the providence th	ue is pull	ed off	the d. The	Description:	otion:	The PC+	2 is pus n stack.	thed onto	The PC+2 is pushed onto the top of the return stack. The previous TOS
	z is te	ous value that was pushed onto the return stack.	t was pu:	shed o	nto the			This instra	ruction a	allows to in by modify	Value is pusified down of the seach. This instruction allows to implement a software stack by modifying TOS.
	± €	This instruction is provided to	on is provent	vided t	0.0000000000000000000000000000000000000			and then	push it	and then push it onto the return	return
	5 ₽	the return stack to incomorate a	ork to inc	ornora	4 a	;					
	los	software stack.	, X	200	3	Words:		-			
Words.	•					Cycles:		-			
; ; ;						a Cyc	Q Cycle Activity:	۲			
Cycles.	_						8	Q2	U	83	8
Q Cycle Activity:	stivity:						Decode	PUSH PC+2		No.	No
δ		Q 2	0 3		04			onto return		u _o	operation
Decode		No P	POP TOS value	- edo	No operation			stack			
						Example:	.: 	PUSH			
Example:	TOD	0	NEW			Be	Before Instruction	uction	ı	0094EAB	
Before	Before Instruction	_					<u> </u>		1 11	000124h	
SS	TOS Stack (1 level down)	own)	= 0031A2h = 014332h	42h 32h		Aft	After Instruction	ction	1	0001 26h	
After Inst TOS PC	After Instruction TOS PC	" "	= 014332h = NEW	32h			TOS Stack (1	TOS Stack (1 level down)	1 11 11	000126h 000345Ah	

© 2006 Microchip Technology Inc. DS39564C-page 240

DS39564C-page 239

RCALL	Relative Call			RESET	Reset		
Syntax:	[label] RCALL n	_		Syntax:	[label] RESET	RESET	
Operands:	$-1024 \le n \le 1023$			Operands:	None		
Operation:	$(PC) + 2 \rightarrow TOS,$ $(PC) + 2 + 2n \rightarrow PC$	S		Operation:	Reset all are affect	Reset all registers and flags that are affected by a MCLR Reset.	flags that R Reset.
Status Affected:	None			Status Affected:	₹		
Encoding:	1101 lnnn	nnnn	uuuu	Encoding:	0000	0000	1111
Description:	Subroutine call with a jump up to 1K from the current location. First,	ith a jump int locatio	up to n. First,	Description:	This instri execute a	This instruction provides a way to execute a MCLR Reset in software.	es a way to
	return address (PC+2) is pushed	C+2) is p	nshed	Words:	-		
	complement number '2n' to the PC.	her '2n' to	the PC.	Cycles:	-		
	Since the PC will have incremented	have incre	emented	Q Cycle Activity:			
	to fetch the next instruction, the	nstruction	ı, the	9	02	Q3	Q 4
	This instruction is a true graft.	be PC+2-	-2n.	Decode	Start	S	2
	I nis instruction is a two-cycle	a two-cy	e <u>e</u>		reset	operation	operation
	instruction.						

After Instruction Registers = Reset Value Flags* = Reset Value

Example: RESET

Words:
Cycles:
Q Cycle Activity:

Q	Write to PC			å	operation
Q3	Process Data			No	operation
0 5	Read literal	Push PC to	stack	No	operation
ō	Decode			No	operation

HERE RCALL Jump

Before Instruction
PC = Address (HERE)
After Instruction
PC = Address (Jump)
TOS = Address (HERE+2)

PIC18FXX2

בוווב						
Syntax:	[label] RETFIE	[8]	Syntax:	[label]	RETLW k	
Operands:	s ∈ [0,1]		Operands:	$0 \le k \le 255$	55	
Operation:	(TOS) \rightarrow PC, 1 \rightarrow GIE/GIEH or PEIE/GIEL, if s = 1	PEIE/GIEL,	Operation:	k → W, (TOS) → PC, PCLATU, PCI	PC, PCLATH are	k → W, (TOS) → PC, PCLATU, PCLATH are unchanged
	(WS) → W,	<u>v</u> II.	Status Affected:	None		
	(BSRS) → BSR,	, ,	Encoding:	0000	1100 kk	kkkk kkkk
	PCLATU, PCLATH are unchanged.	are unchanged.	Description:	W is loade	ed with the	W is loaded with the eight-bit literal
Status Affected:	GIE/GIEH, PEIE/GIEL	EL.		'k'. The pr	k'. The program counter is loaded	ter is loaded
Encoding:	0000 0000	0001 000s		from the to address).	from the top of the stack (the return address). The high address latch	from the top of the stack (the return address). The high address latch
Description:	Return from Interrupt. Stack is	pt. Stack is		(PCLATH)	(PCLATH) remains unchanged.	changed.
	popped and Top-of-Stack (TOS) is	-Stack (TOS) is	Words:	-		
	enabled by setting either the high	either the high	Cycles:	2		
	or low priority global interrupt	al interrupt	Q Cycle Activity:			
	enable bit. If 's' = 1, the contents of	, the contents of	۵1 م	Q2	89	Φ
	the shadow registers WS, STATUSS and BSRS are loaded	rs WS, 3S are loaded	Decode	Read	Process	pop PC from
	into their corresponding registers,	iding registers,		literal n	Dala	to W
	W, STATUS and BSR. If 's' = 0, no	SR. If 's' = 0, no	No	N _o	ž	8
	update of these registers occurs (default).	jisters occurs	operation	operation	operation	operation
Words:	-		Example:			
Cydes:	2					
Q Cycle Activity:			CALL TABLE	; W contains t : offset value	W contains table offset value	
ğ	Q2 Q3	Φ4		; W now has	as	
Decode	No No operation	pop PC from stack Set GIEH or	: TABLE ADDWF PCL	; table value ; W = offset	alue set	
No operation	No No operation	8	RETLW KO RETLW K1 :	; Begin table	able	
Example:	RETFIE 1		: RETLW kn	; End of table	table	
After Interrupt	"	SOL	Before Instruction	ction		
W BSR STATUS GIE/GIE	W = WS BSR = BSF STATUS = STA GIF/GIEH PFIE/GIEI = 1	WS BSRS STATUSS 1	W = After Instruction	= 0x07 on		
			/ / /			

© 2006 Microchip Technology Inc. DS39564C-page 242

DS39564C-page 241

RETURN	Return fro	Return from Subroutine	ine	RLCF	ų.	Rotate Lo	Rotate Left f through Carry	h Carry	
Syntax:	[<i>label</i>]	[label] RETURN [s]		Syntax:	ax:	[<i>label</i>]	[label] RLCF f [,d [,a]	,d [,a]	
Operands:	$s \in [0,1]$			Ope	Operands:	$0 \le f \le 255$	5		
Operation:	(TOS) → PC, if s = 1	Ď,				d∈ [0,1] a∈ [0,1]			
	(WS) → W,	(WS) → W,	ď	Ope	Operation:	(f <n>) → de</n>	(f <n>) → dest<n+1>,</n+1></n>		
	(BSRS) → BSR	bSR,	'n			(I) → C, (C) → dest<0>	st<0>		
	PCLATU, F	>CLATH are	PCLATU, PCLATH are unchanged	Statı	Status Affected:	C, N, Z			
Status Affected:	None			Enc	Encoding:	0011	01da f	ffff ffff	
Encoding:	0000	0000	0001 001s	Desc	Description:	The confe	The contents of register IT are	ter If are	
Description:	Return fror is popped	Return from subroutine. The stack is popped and the top of the stack	The stack of the stack		-	rotated or the Carry	rotated one bit to the left through the Carry Flag. If a' is 0, the resu	rotated one bit to the left through the Carry Flag. If d is 0, the result	
	(TOS) is lo counter. If '	(TOS) is loaded into the program counter. If 's'= 1. the contents of the	(TOS) is loaded into the program counter. If 's'= 1. the contents of the			is placed	s placed in W. If d' is 1, the s stored back in register f	s placed in W. If d' is 1, the result s stored back in register If	
	shadow re	shadow registers WS, STATUSS	STATUSS			(default).	(default). If 'a' is 0, the Access	e Access	
	and BSRS	are loaded i	and BSRS are loaded into their cor-			Bank will	Bank will be selected, overriding	, overriding	
	responding	responding registers, W, STALUS	v, SIAIUS			the BSH	value. If 'a' =	the BSR value. If 'a' = 1, then the	
	and BSR. I these regis	and BSR. If 's' = 0, no update of these registers occurs (default).	update of (default)			bank will BSR valu	bank will be selected as per the BSR value (default).	as per the	
Words:	-					CO L	register f	Ţ	
Cycles:	2								
Q Cycle Activity:				Words:	is:	_			
δ.	Q 2	Q3	Q4	Cycles:	es:	-			
Decode	S	Process	pop PC from	Ø	Q Cycle Activity:		ć	č	
	operation	Data	stack		5	QZ	83	Ω 42	
No	No	No	No		Decode	Read	Process Data	Write to	
ioimodo	- Composition	obologo	polario			i picificii i	Dala	desiliation	

Example: RLCF REG, 0, 0

Before Instruction

After Instruction

REG = 0110 0110

After Instruction

REG = 1110 0110

W = 1100 1100

C = 1

Example: RETURN
After Interrupt
PC = TOS

PIC18FXX2

RLNCF	Rotate Left f (no carry)	RRCF	Rotate Right f through Carry
Syntax:	[<i>label</i>] RLNCF f[,d[,a]	Syntax:	[label] RRCF f[,d[,a]
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
Operation:	$(f) \rightarrow dest,$ $(f<7>) \rightarrow dest<0>$	Operation:	$(fcn) \rightarrow destcn-1>,$ $(fc0>) \rightarrow C,$
Status Affected:	N, N		$(C) \rightarrow dest < / >$
Encoding:	0100 01da ffff ffff	Status Affected:	C, N, Z
Description:	The contents of register # are	Encoding:	0011 00da ffff ffff
	rotated one bit to the left. If 'd' is 0,	Description:	The contents of register I' are
	the result is placed in W. If d' is 1, the result is stored back in register		rotated one bit to the right through the Carry Flag. If 'd' is 0, the result
	f' (default). If 'a' is 0, the Access		is placed in W. If d' is 1, the result
	Bank will be selected, overriding		is placed back in register "
	the bor value: If a 1s 1, then the bank will be selected as per the		(delault). If a 1s 0, trie Access Bank will be selected, overriding
	BSR value (default)		the BSR value. If 'a' is 1, then the
	register f		bank will be selected as per the BSR value (default).
Words:	-		C register f
Cycles:	-		
O Cycle Activity:		Words:	
0	02 03 04	Cycles:	-
Decode	Process	Q Cycle Activity:	
	register II Data destination	ā	Q2 Q3 Q4
Example:	RINCF REG, 1, 0	Decode	Read Process Write to register T Data destination
Before Instruction		Example.	0 0 084 8044
5	= 1010 1011	Lyaniple.	'o 'pay
After Instruction REG =	tion = 0101 0111	Before Instruction REG = C =	ction = 1110 0110 = 0
		After Instruction	ion
		REG	
		≊ C	= 0111 0011 = 0
		,	

DS39564C-page 244 © 2006 Microchip Technology Inc.

DS39564C-page 243

Syntax.		notate night I (no carry)	SEIF) aer			
	[<i>label</i>] RRNCF f [,d [,a]	f[,d[,a]	Syntax:	[<i>label</i>] SE	[label] SETF f [,a]		
Operands: 0 ≤ d ∈	$0 \le f \le 255$ d $\in [0,1]$		Operands:	$0 \le f \le 255$ $a \in [0,1]$	10		
ae	a∈ [0,1]		Operation:	FFh → f			
Operation: (f <r< td=""><td>(f<n>) → dest<n-1>,</n-1></n></td><td>Ň.</td><td>Status Affected:</td><td>None</td><td></td><td></td><td></td></r<>	(f <n>) → dest<n-1>,</n-1></n>	Ň.	Status Affected:	None			
Status Affected: N, Z	Z / / / / / / Z		Encoding:	0110	100a ff	ffff	ffff
Encoding: 0	0100 00da	fiff fiff	Description:	Ine conte ter are set	I ne contents of the specified regis- er are set to FFh. If 'a' is 0, the	pecified a' is 0, i	regis-
Description: The rota	he contents of register If are otated one bit to the right. If d	The contents of register If are otated one bit to the right. If I'd' is 0,		Access Bariding the	Access Bank will be selected, over- riding the BSR value. If 'a' is 1, then	selectec If 'a' is	d, over- 1, then
the	result is placed	the result is placed in W. If 'd' is 1,		the bank v	the bank will be selected as per the	ted as	er the
) <u> </u>	The result is placed back in regist	The Acces		DON Value	(deladit).		
) - Bar	l (deradii): II a is 0, iiie Access Bank will he selected overriding	o, une Access	Words:	-			
the	BSR value. If	the BSR value. If 'a' is 1, then the	Cycles:	-			
bar	bank will be selected as per the	ed as per the	Q Cycle Activity:	÷			
BSI	BSR value (default).	t).	ø	Q2	ဗ	J	Q 40
	ē´	register f	Decode	Read register #	Process Data	v regis	Write register f
Words: 1) -			
Cycles: 1			Example:	SETF	REG,1		
Q Cycle Activity:			Before Instruction				
0	Q2 Q3	Φ	REG	п	0x5A		
Decode Re	Read Process register If Data	ss Write to destination	Atter Instruction REG	П	0xFF		

Example 1: RRNCF REG, 1, 0 Before Instruction REG = 1101 0111 After Instruction REG = 1110 1011

© 2006 Microchip Technology Inc. DS39564C-page 245

PIC18FXX2

SLEEP	Enter SLEEP mode	SUBFWB	Subtract f from W with borrow	borrow
Syntax:	[label] SLEEP	Syntax:	[label] SUBFWB f[,c	f [,d [,a]
Operands:	None	Operands:	0 < f < 255	
Operation:	TOW - 400	<u>.</u>	d∈ [0,1]	
	0 → WDT postscaler.		a∈ [0,1]	
		Operation:	$(W) - (f) - (\overline{C}) \rightarrow dest$	
	0 → PD	Status Affected:	N, OV, C, DC, Z	
Status Affected:	<u>제</u> , 한	Encoding:	0101 01da ffff	ffff
Encodina:	0000 0000 0001		M notoin to	200
Description:	wer-down s	Description:	(borrow) from W (2's complement	ny nag olement
-	cleared. The time-out status bit		method). If 'd' is 0, the result is	u l tis
	(TO) is set. Watchdog Timer and		stored in W. If 'd' is 1, the result is	result is
	its postscaler are cleared		stored in register "f (default). If 'a' is	t) If a is
	The processor is put into SLEEP mode with the oscillator stonned		0, the Access Bank will be selected overriding the BSB value. If 'a' is 1	selected, If 'a' is 1
Mordo.			then the bank will be selected as	ted as
words.	_		per the BSR value (default).	÷.
Cycles:		Words:		
Q Cycle Activity:	į	Cvcles:	•	
δ	03	O Chale Activities		
Decode	No Process Go to	C Cycle Activity.	03	Q
	Date	Decode	Process	Write to
Example:	4 日本日本		F Data	destination
Before Instruct	ion	Example 1:	SUBFWB REG, 1, 0	
<u>70</u> = 2		notes and or		
= 0		Berore Instruct REG =	Jon 3	
tructi	Ē	M		
= = BB		U :		
1	-	ucti		
† If WDT causes	If WDT causes wake-up, this bit is cleared.	g g		
		= = Z	00	
		Example 2:	SUBFWB REG, 0, 0	
		Before Instruction	ion	
		■	. 2	
		1		
		REG =		
			ဗ	
		" " Z		
			0 ; result is positive	
		Example 3:	SUBFWB REG, 1, 0	
		Before Instruction	ion	
		REG =	-	
			2	
		U -		
		Arrer Instruction BEG -		
			۰ ۵	
			1 —	
		Z N	1 ; result is zero	

DS39564C-page 246 © 2006 Microchip Technology Inc.

SUBLW	Subtra	Subtract W from literal	eral	SUBWF	Subtract W from f
Syntax:	[label	[label] SUBLW k		Syntax:	[label] SUBWF fl,d [,a]
Operands:	0 ≤ k ≤ 255	255		Operands:	
Operation:	k – (W	$k - (W) \rightarrow W$		-	d∈ [0,1]
Status Affected:	N, OV,	N, OV, C, DC, Z			A (A) . deep
Encoding:	0000	1000	kkkk kkk	Operation.	$(1) = (M) \rightarrow (M) = (1)$
Description:	W is si literal '	W is subtracted from the eight-bit literal 'k'. The result is placed in W	the eight-bit s placed	Status Attected. Encoding: Description:	N, OV, C, DC, Z 0101 11da ffff ffff Subtract W from register # (2's
Words:	-				complement method). If 'd' is 0,
Cvoles:	-				the result is stored in W. If 'd' is 1,
Q Cycle Activity:					tile lesuit is stored back in regis- ter ff (default). If 'a' is 0, the
. δ	Ø5	03	Q4		Access Bank will be selected,
Decode	Read literal 'k'	Process Data	Write to W		overriding the BSR value. If a' is 1, then the bank will be selected
Example 1:	SUBLW	0×02		Words:	as per me bort value (deraun). 1
Before Instruction	ıction			: ::::::::::::::::::::::::::::::::::::	
≥ (- 0			O Cycle Activity:	-
≜ O	 			Q1	02 03
×	-			Decode	Process
ON	- 0	; result is positive	Ф		register 1' Data destination
ız	0 =			Example 1:	SUBWF REG, 1, 0
Example 2:	SUBLW	0x02		Before Instruction	ıction
Before Instruction	ction			5 0	
≥ ∪	- 2			= CO 440	
After Instruction	ion			Aller Instruct REG	1
*	0 =			Μ	8
ONZ	0	; result is zero		ONZ	= 1 ; result is positive = 0 = 0
Example 3:	SUBLW	0×02		Example 2:	SUBWF REG, 0, 0
Before Instruction	ıction			Before Instruction	ıction
≥ (0 0			REG W	α α
After Instruction				O -	
Μ	倠	; (2's complement)	rf)	After Instruction REG =	ion = 2
OΝ		result is negativ	Ф		
z)NZ	
				Example 3:	SUBWF REG, 1, 0
				Before Instruction	ction
) 	
				= C = After Instruction	<u>.</u>
				REG W	= FFh ;(2's complement) = 2
				NC	= 0 ; result is negative = 0
				Z	

© 2006 Microchip Technology Inc.

PIC18FXX2

					5		
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]			Operands:	40 — —	10	
Operation: Status Affected:	(f) $-$ (W) $-$ (\overline{C}) $-$ N, OV, C, DC, Z	φ	+	Operation: Status Affected:	(f<3:0>) - (f<7:4>) - None	(f<3:0>) → dest<7:4>, (f<7:4>) → dest<3:0> None	
Encoding:	0101	Olol loda ffff ffff Cubtoot W and the committee / box	if fiff	Encoding:	0011	10da ff	ffff ffff
	row) from row) from row) from row from row from row. If 'd' i back in reg the Access overriding then the bath a BSR variation of the BSR variation of the bath a BSR variation of the BSR variation of the bath a BSR variation of	Southard water the Carry head tool- row) from register *f (2's complement method). If 'v' is 0, the result is stored in W. If 'v' is 1, the result is stored back in register *f' (default). If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, then the bank will be selected as per the BSR value (default).	y riag (tot)- complement soult is stored is stored it. If a' is 0, selected, e. If 'a' is 1, lected as per	Description:	The upper ister "Fare result is presult is presult is presult is presult is presult." Bank will the BSR v bank will	The upper and lower nibbles of register if are exchanged. If d'is 0, the result is placed in W. If d'is 1, the result is placed in register if (default). If a' is 0, the Access Bank will be selected, overriding the BSR value. If a' is 1, then the bank will be selected as per the	ibbles of registrations of registrations of the ster o
Words:	-			-	BSH valu	BSR value (detault).	
Cycles:	_			Words:			
Q Cycle Activity:				Cycles:	_		
٥	02	03	90	Q Cycle Activity:	S	Ö	S
Decode	Head register f	Process Data	Write to destination	Decode	Read	Process	Write to
Example 1:	SUBWFB	REG, 1, 0			register T	Data	destination
Before Instruction	tion			Example:	SWAPF	REG, 1, 0	
EG.			1001)	Before Instruction	rction		
≽ O	= 0x0D	(11 0000)	1101)	REG	= 0x53		
ructi				After Instruction	tion		
<u></u>		(0000 10	1011)	פֿיי	ccxo =		
	= 0X0D	(TOTT 0000)	OT.)				
ΝZ	00	; result is positive	ositive				
Example 2:	SUBWFB	REG, 0, 0					
struc	tion						
REG	= 0x1B	(0001 10	1011)				
itolitio		10000	(0)				
Aitel Histincil	= 0x1B	(1101 1000)	11)				
ONZ	0	; result is zero	ero				
Example 3:	SUBWFB	REG, 1, 0					
struc	tion						
S H ⊠	= 0x03	(1000 0000)	0011)				
itoria			i				
REG	= 0xF5	(1111 0100)	(00				
>	= 0x0F	;[Z's comp]	01)				
: On		0000					
) -	; result is negative	egative				

Table Read (cont'd)

TBLRD	Table Read	TBLRD Tabl	Table Read
Syntax:	[label] TBLRD (*; *+; *-; +*)	Example1: TBLE	TBLRD *+;
Operands:	None	Before Instruction	
Operation:	if TBLRD *,	TABLAT TRI DTD	
	(Prog Mem (TBLPTR)) → TABLAT;	MEMORY (0x00A356)	356)
	TBLPTR - No Change;	After Instruction	
	if TBLRD *+,	TABI AT	,
	(Prog Mem (TBLPTR)) \rightarrow TABLAT;	TBLPTR	"
	(TBLPTR) +1 → TBLPTR;		
	if TBLRD ⁴-,	Example2:	· *+
	(Prog Mem (TBLPTR)) → TABLAT;	Before Instruction	
	(TBLPTR) -1 → TBLPTR;	TABLAT	"
	if TBLRD +*,	TBLPTR	
	(TBLPTR) +1 → TBLPTR;	MEMORY(0x01A	357) =
	(Prog Mem (TBLPTR)) → TABLAT:	MEMORY(0x01A358)	358) =
	(i		

0x55 0x00A356 0x34

0x34 0x00A357

TBLRD +*;

Status Affected: None

0x34 0x01A358

After Instruction TABLAT TBLPTR

0xAA 0x01A357 0x12 0x34

Encoding:	0000	0000	0000	10nn
				* 0=uu
				=1 *+
				=2 *-
				=3 +*
:] -	:	

This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used. The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2 Mbyte address range. Description:

TBLPTR[0] = 0: Least Significant
Byte of Program
Memory Word

TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLRD instruction can modify the value of TBLPTR as follows:

- no change
 post-increment
 post-decrement
 pre-increment
- Cycles: Words:

Q Cycle Activity:

8 8 ٥

8

oN.	No operation	(Write TABLAT)	
No	Operation	operation	
No	No operation	(Read Program	Memory)
Decode	S	operation	

PIC18FXX2

TBLWT	Table Write	ite		TBLWT Table Write (Co	(Continued)
Syntax:	[label]	TBLWT (*; *+;	(*+;**;*;*)	Example1: TBLWT *+;	
Operands:	None				r L
Operation:	if TBLWT* (TABLAT) TBLPTR -	! ~	→ Holding Register; No Change;	R NG REGISTER 356) =	0x00A356 0xFF
	if TBLWT*+, (TABLAT) → (TRI PTR) +	f TBLWT*+, TABLAT) → Holding Register; TRI PTR) +1 → TRI PTR:	g Register; I PTR·	rite co	mpletion) 0x55 0x004357
	if TBLWT		: : :		OKSS
	(TBLPTR)	↑ —	Holding Hegister; → TBLPTR;	*	3
	if IBLWI+*, (TBLPTR) +1 (TABLAT) →	고도조	→ TBLPTR; Holding Register:	uction =	0x34
Status Affected: None	ed: None				OX01309A
Encoding:	0000	0000	0000 11nn nn=0 *	(0x01389B) = 0x (0x01389B) = 0x	OXFF
			# * * * # # # # # # # # # # # # # # # #	Atter Instruction (table write completion) TABLAT TBLPTR = 0x013891	npletion) 0x34 0x01389B
Description:	This instr	uction uses	This instruction uses the 3 LSbs of the	П	0×FF
	TBLPTR holding re	to determir saisters the	TBLPTR to determine which of the 8 nolding registers the TABLAT data is	П	0x34
	written to	The 8 hol	written to. The 8 holding registers are		
	used to p gram Mer	rogram the mory (P.M.)	used to program the contents of Pro- gram Memory (P.M.). See Section 5.0		
	for inform	ation on w	riting to FLASH		
	The TBLF	PTR (a 21-l	The TBLPTR (a 21-bit pointer) points		
	to each b	yte in the p has a 2 ME	to each byte in the program memory. TBLPTR has a 2 MBtye address		
	range. Th which byt	ne LSb of th te of the pro	range. The LSb of the TBLPTR selects which byte of the program memory		
	location to access	o access.	- House		
	181	.i = [0] = 0:	BLP H[U] = U: Least Significant Byte of Program Memory Mord		
	TRIE	TRI PTRÍOI = 1.	Most Significant		
	<u>.</u>	I	Byte of Program Memory Word		
	The TBLV	The TBLWT instruction can value of TBI PTR as follows:	The TBLWT instruction can modify the value of TBI PTB as follows:		
	 no change 	nge			
	 post-in 	post-increment			
	• post-de	post-decrement			
1000000))) •				
, odgo.	- ი				
O Cholo Activity:	ivija.				
2 cycle 20	02	03	04		
Decode	oN.	No	No		
2	operation	operation	operation		
operation	operation	operation	operation		
	(Read TABLAT)		(Write to Holding Register or Memory)		

DS39564C-page 249 © 2006 Microchip Technology Inc.

© 2006 Microchip Technology Inc.

DS39564C-page 250

TSTFSZ	Test f, skip if 0	XORLW	Exclusive OR literal with W	iteral with W	
Syntax:	[label] TSTFSZ f[,a]	Syntax:	[label] XORLW k	× /	
Operands:	$0 \le f \le 255$	Operands:	$0 \le k \le 255$		
	a∈ [0,1]	Operation:	W) XOR K→W	>	
Operation:	skip if $f = 0$	Status Affected:	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \		
Status Affected:	None	Propoling:	_	10000	2
Encoding:	0110 0113 6666 6666	Ellcodiilg.	0000 1010	KKKK	KKKK
		Description:	The contents of W are XORed	W are XORed	~
Description:	If $f' = 0$, the next instruction,		with the 8-bit literal 'k'. The result	eral 'k'. The re	sult
	fetched during the current instruc-		is placed in W		
	tion execution, is discarded and a	Worde.			
	NOP is executed, making this a two-				
	cycle instruction. If 'a' is 0, the	Cycles:	-		
	Access Bank will be selected, over-	Q Cycle Activity:			
	riding the BSR value. If 'a' is 1,	ō	Q2 Q3	3 Q4	
	men the bank will be selected as	Decode	Read Process	ess Write to W	3
	per lire bon value (uerauli).		literal 'k' Data		
Words:	-				
Cycles:	1(2)	Example:	XORLW 0xAF		
	Note: 3 cycles if skip and followed by a 2-word instruction.	Before Instruction	stion		
	•	141	20.0		

W = 0xB5After Instruction W = 0x1A

§ 8

Process Data Q3

Q2 Read register "

Q Cycle Activity: Q1 Decode Example: HERE TSTFSZ CNT, 1 NZERO : ZERO : Before Instruction PC = Address (HERE)

= 0x00, = Address (ZERO) \$\neq\$ 0x00, = Address (NZERO)

PIC18FXX2

Exclusive OR W with f	[label] XORWF f[,d[,a]	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]	(W) XOR. (f) → dest	NI	0001 10da fiff fiff	Exclusive OR the contents of W with register "I if d' is 0, the result is stored in W. If d' is 1, the result is stored back in the register "I (default). If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, then the bank will be selected, overriding the SBR value. If a' is 1, then the BSR value (default).	Q2 Q3 Q4	Read Process Write to
XORWF Ex	Syntax: [//	Operands: 0 de	Operation: (M	Status Affected: N, Z	Encoding:	Description: EX wif will will will will will will be started to the will be sent that the base of the words: 1 Cycles: 1 Cycles: 1	D T	Decode

register f Data destination XORWF REG, 1, 0

Before Instruction
REG = 0xAF
W = 0xB5
After Instruction
REG = 0x1A

© 2006 Microchip Technology Inc. DS39564C-page 252

DS39564C-page 251