Introduction

As I was getting sick and tired of re-reading the instruction set information in the datasheet for the PIC16F8x microcontroller (very informational, but difficult to read, and spread over several pages), I decided to compile this easy to read, two page thing of all the instructions and register files. The information presented here has been compiled from the datasheet, but since typos may appear please use this document with appropriate care. Thank you!

//denMike, April 4th, 2002

Microchip PIC16F8x Instruction Set Summary

Instruction	Description	Example	Result	Flags affected
ADDLW	Add constant to W	ADDLW 0x15	W := W + 15h	C,DC,Z
ADDWF	Add W to register, store in register Add W to register, store in W	ADDWF reg,F ADDWF reg,W	Reg := Reg + W W := Reg + W	C,DC,Z C,DC,Z
ANDLW	AND constant with W	ANDLW b'01010101'	W := W and 01010101b	Z
ANDWF	AND W with register, store in register AND W with register, store in W	ANDWF reg,F ANDWF reg,W	Reg := Reg and W W := Reg and W	Z Z
BCF	Bit clear in register	BCF reg,7	Reg[bit 7] := 0	none
BSF	Bit set in register	BSF reg,7	Reg[bit 7] := 1	none
BTFSC	Bit test, skip if clear	BTFSC reg,7 next instruction	if Reg[bit 7] = 1 then NEXT INSTRUCTION	none
BTFSS	Bit test, skip if set	BTFSS reg,7 next instruction	if Reg[bit 7] = 0 then NEXT INSTRUCTION	none
CALL	Call subroutine	CALL label	TOS := PC+1; PC := label	none
CLRF	Clear register	CLRF reg	Reg := 0	Z := 1
CLRW	Clear W	CLRW	W := 0	Z := 1
CLRWDT	Clear watchdog timer	CLRWDT	WDT := 0	none
COMF	Complement register, store in register Complement register, store in W	COMF reg,F COMF reg,W	Reg := not Reg W := not Reg	Z Z
DECF	Decrement register, store in register Decrement register, store in W	DECF reg,F DECF reg,W	Reg := Reg - 1 W := Reg - 1	Z Z
DECFSZ	Decrement register, store in register, skip if 0	DECFSZ reg,F next instruction	Reg := Reg - 1 if Reg <> 0 then NEXT INSTRUCTION	none
	Decrement register, store in W, skip if 0	DECFSZ reg,W next instruction	W := Reg - 1 if W <> 0 then NEXT INSTRUCTION	none
GOTO	Unconditional branch	GOTO label	PC := label	none
INCF	Increment register, store in register Increment register, store in W	INCF reg,F INCF reg,W	Reg := Reg + 1 W := Reg + 1	Z Z
INCFSZ	Increment register, store in register, skip if 0	INCFSZ reg,F next instruction	Reg := Reg + 1 if Reg <> 0 then NEXT INSTRUCTION	none
	Increment register, store in W, skip if 0	INCFSZ reg,W next instruction	W := Reg + 1 if W <> 0 then NEXT INSTRUCTION	none
IORLW	Inclusive OR constant with W	IORLW d'137'	W := W or 137d	Z
IORWF	Inclusive OR W with register, store in reg Inclusive OR W with register, store in W	IORWF reg,F IORWF reg,W	Reg := Reg or W W := Reg or W	Z Z
MOVF	Move register to W Nothing moved, but Z flag updated	MOVF reg,W MOVF reg,F	W := Reg Z flag updated	Z Z
MOVLW	Move constant to W	MOVLW 13	W := 13h	none
MOVWF	Move W to register	MOVWF reg	Reg := W	none
NOP	No operation	NOP		none
RETFIE	Return from interrupt	RETFIE	PC := TOS; GIE enabled	GIE := 1
RETLW	Return from subroutine with constant in W	RETLW 0AAh	W := 0AAh; PC := TOS	none
RETURN	Return from subroutine	RETURN	PC := TOS	none
RLF	Rotate register left through carry, store in reg Rotate register left through carry, store in W	RLF Reg,F RLF Reg,W	Reg := RCL Reg W := RCL Reg	C C
RRF	Rotate register right through carry, store in reg Rotate register right through carry, store in W	RRF Reg,F RRF Reg,W	Reg := RCR Reg W := RCR Reg	C C
SLEEP	Enter power-down mode	SLEEP	WDT := 0	none
SUBLW	Subtract W from constant	SUBLW 4	W := 4 - W	C,DC,Z
SUBWF	Subtract W from register, store in register Subtract W from register, store in W	SUBWF reg,F SUBWF reg,W	Reg := Reg - W W := Reg - W	C,DC,Z C,DC,Z
SWAPF	Swap nibbles in register, store in register Swap nibbles in register, store in W	SWAPF reg,F SWAPF reg,W	Reg := RegHi xchg RegLo W := RegHi xchg RegLo	none none
XORLW	Exclusive OR constant with W	XORLW 137	W := W xor 137h	Z
XORWF	Exclusive OR W with register, store in reg Exclusive OR W with register, store in W	XORWF reg,F XORWF reg,W	Reg := Reg xor W W := Reg xor W	Z Z

Note: TOS = Top of Stack

Microchip PIC16F8x Memory Layout

BANK0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	INDF	Indexed addressing data read/write (not a physical register)							
01h	TMR0	8-bit real-time clock/counter							
02h	PCL	Low order 8 bits of the Program Counter (PC)							
03h	STATUS	IRP	RP1	RP0	!TO	!PD	Z	DC	C
04h	FSR	Indexed addressing address pointer 0							
05h	PORTA	Unimplemented, read as ‘0’			RA4/TOCKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07h		Unimplemented location, read as ‘0’							
08h	EEDATA	EEPROM data register							
09h	EEADR	EEPROM address register							
0Ah	PCLATH	Unimplemented, read as ‘0’			Write buffer for upper 5 bits of the PC				
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

BANK1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
80h	INDF	Maps address 00h							
81h	OPTION_REG	!RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
82h	PCL	Maps address 02h							
83h	STATUS	Maps address 03h							
84h	FSR	Maps address 04h							
85h	TRISA	Unimplemented, read as ‘0’			PORTA data direction register (1=input (default), 0 = output)				
86h	TRISB	PORTA data direction register (1=input (default), 0 = output)							
87h		Unimplemented location, read as ‘0’							
88h	EECON1	Unimplemented, read as ‘0’			EEIF	WRERR	WREN	WR	RD
89h	EECON2	EEPROM control register 2 - used internally for EEPROM writes (not a physical register)							
8Ah	PCLATH	Unimplemented, read as ‘0’			Maps address 0Ah				
8Bh	INTCON	Maps address 0Bh							

STATUS REGISTER

bit	Name	Description	0	1
7	IRP	Not used - should be maintained clear	always	never
6	RP1	Not used - should be maintained clear	always	never
5	RP0	Register bank select bit	bank 0 (00h-7Fh)	bank 1 (80h-FFh)
4	!TO	Time-out bit	WDT time-out occured	after power-up, CLRWDT or SLEEP instr
3	!PD	Power-down bit	by execution of the SLEEP instr	after power-up or by the CLRWDT instr
2	Z	Zero bit	Last result was NOT zero	Last result was zero
1	DC	Digit carry/!borrow bit (for ADDWF and ADDLW instructions)	No carry-out	A carry-out from the 4th low order bit of the result
0	C	Carry/!borrow bit (for ADDWF, ADDLW, RRF, and RLF instructions)	No carry-out	A carry-out from the MSB of the result occured

INTCON REGISTER

bit	Name	Description	0	1
7	GIE	Global interrupt enable bit	Disable all interrupts	Enables all un-masked ints
6	EEIE	EE write complete interrupt enable bit	Disable	Enable
5	T0IE	TMR0 Overflow interrupt enable bit	Disable	Enable
4	INTE	RB0/INT interrupt enable bit	Disable	Enable
3	RBIE	PortB[7:4] Change interrupt enable bit	Disable	Enable
2	T0IF	TMR0 oferflow interrupt flag bit	No overflow occured	TMR0 has overflowed (must be cleared in software)
1	INTF	RB0/INT interrupt flag bit	No interrupt occured	An RB0/INT interrupt occured
0	RBIF	PortB[7:4] change interrupt flag bit	No pin have changed	At least one pin changed (must be cleared in software)

OPTION_REG REGISTER

bit	Name	Description	0	1
7	!RBPU	PortB pull-up enable bit	Disable pull-ups	Enable pull-ups
6	INTEDG	Interrupt edge select bit	RB0/INT intr on falling edge	RB0/INT intr on rising edge
5	T0CS	TMR0 clock source select pin	Internal (clk/4)	Transition on RA4/T0CLK pin
4	T0SE	TMR0 external (RA4/T0CLK) source edge select bit	Increment on low-to-high	Increment on high-to-low
3	PSA	Prescaler assignment bit	Assigned to TMR0	Assigned to watchdog timer
2:0	PS2:PS0	Prescaler rate select bits	See below	See below

Bit value	TMR0 rate	WDT rate	Bit value	TMR0 rate	WDT rate
000	1:2	1:1	100	1:32	1:16
001	1:4	1:2	101	1:64	1:32
010	1:8	1:4	110	1:128	1:64
011	1:16	1:8	111	1:256	1:128

EECON1

bit	Name	Description	0	1
7:5	-	Unimplemented read as ‘0’	Always	Never
4	EEIF	EE write operation interrupt flag bit	Not complete	The write operation completed (must be cleared in software)
3	WRERR	EE error flag bit	Write completed	Write didn’t complete
2	WREN	EE write enable bit	Inhibits writes	Allows writes
1	WR	Write control bit	Write complete	Initiates a write (cleared in hardware)
0	RD	Read control bit	Does nothing	Initiates an EE read (cleared in hardware)