

PIC16C63A/65B/73B/74B

4.0 MEMORY ORGANIZATION

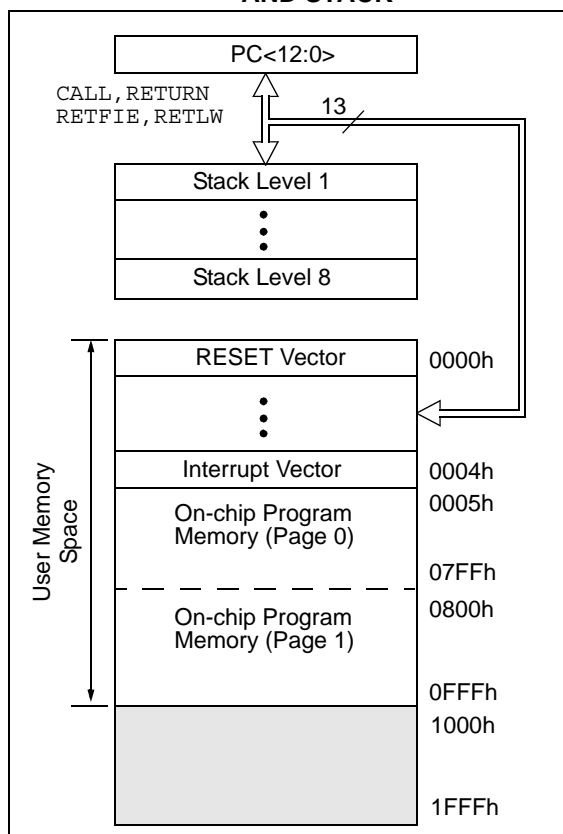
4.1 Program Memory Organization

The PIC16C63A/65B/73B/74B has a 13-bit program counter capable of addressing an 8K x 14 program memory space. All devices covered by this data sheet have 4K x 14 bits of program memory. The address range is 0000h - 0FFFh for all devices.

Accessing a location above 0FFFh will cause a wrap-around.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 4-1: PIC16C63A/65B/73B/74B PROGRAM MEMORY MAP AND STACK



4.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). Bits RP1 and RP0 are the bank select bits.

RP1:RP0 (STATUS<6:5>)

= 00 → Bank0

= 01 → Bank1

= 10 → Bank2

= 11 → Bank3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the SFRs. Above the SFRs are GPRs, implemented as static RAM.

All implemented banks contain SFRs. Frequently used SFRs from one bank may be mirrored in another bank for code reduction and quicker access.

Note: Maintain the IRP and RP1 bits clear in these devices.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly, through the File Select Register (FSR) (Section 4.5).

PIC16C63A/65B/73B/74B

4.2.2.1 STATUS Register

The STATUS register, shown in Register 4-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the \overline{TO} and \overline{PD} bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged).

It is recommended that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions be used to alter the STATUS register. These instructions do not affect the Z, C or DC bits in the STATUS register. For other instructions which do not affect status bits, see the "Instruction Set Summary."

Note 1: These devices do not use bits IRP and RP1 (STATUS<7:6>), maintain these bits clear to ensure upward compatibility with future products.

2: The C and DC bits operate as borrow and digit borrow bits, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 4-1: STATUS REGISTER (ADDRESS 03h, 83h)

	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
	IRP ⁽¹⁾	RP1 ⁽¹⁾	RP0	\overline{TO}	\overline{PD}	Z	DC	C ⁽²⁾
bit 7								bit 0

- bit 7 **IRP⁽¹⁾:** Register Bank Select bit (used for indirect addressing)
 1 = Bank 2, 3 (100h - 1FFh)
 0 = Bank 0, 1 (00h - FFh)
- bit 6-5 **RP1⁽¹⁾:RP0:** Register Bank Select bits (used for direct addressing)
 11 = Bank 3 (180h - 1FFh)
 10 = Bank 2 (100h - 17Fh)
 01 = Bank 1 (80h - FFh)
 00 = Bank 0 (00h - 7Fh)
 Each bank is 128 bytes
- bit 4 **\overline{TO} :** Time-out bit
 1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
 0 = A WDT time-out occurred
- bit 3 **\overline{PD} :** Power-down bit
 1 = After power-up or by the `CLRWDT` instruction
 0 = By execution of the `SLEEP` instruction
- bit 2 **Z:** Zero bit
 1 = The result of an arithmetic or logic operation is zero
 0 = The result of an arithmetic or logic operation is not zero
- bit 1 **DC:** Digit carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for borrow the polarity is reversed)
 1 = A carry-out from the 4th low order bit of the result occurred
 0 = No carry-out from the 4th low order bit of the result
- bit 0 **C⁽²⁾:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
 1 = A carry-out from the most significant bit of the result occurred
 0 = No carry-out from the most significant bit of the result occurred

Note 1: Maintain the IRP and RP1 bits clear.

2: For borrow and digit borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

PIC16C63A/65B/73B/74B

14.0 INSTRUCTION SET SUMMARY

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

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

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

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

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

TABLE 14-1: OPCODE FIELD DESCRIPTIONS

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

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

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

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

Table 14-2 lists the instructions recognized by the MPASM™ assembler.

Figure 14-1 shows the general formats that the instructions can have.

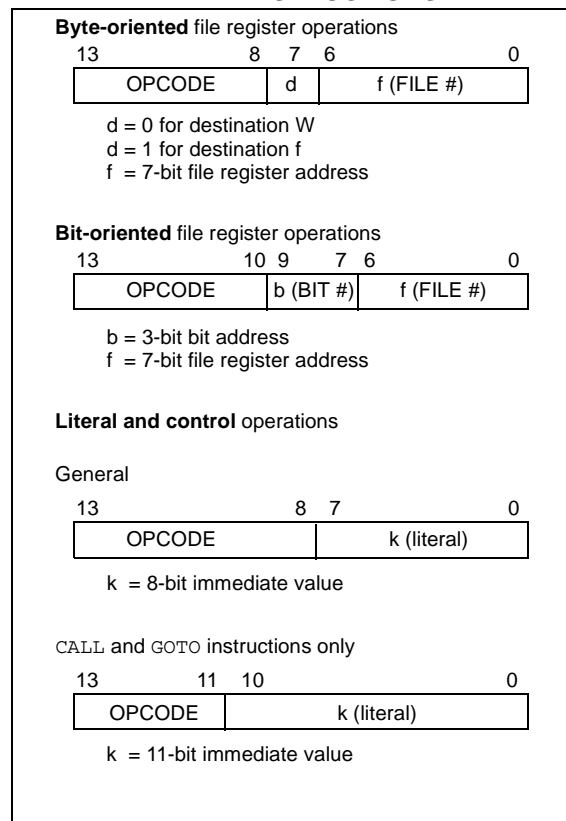
Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

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

0xhh

where h signifies a hexadecimal digit.

FIGURE 14-1: GENERAL FORMAT FOR INSTRUCTIONS



PIC16C63A/65B/73B/74B

TABLE 14-2: PIC16CXX INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode			Status Affected	Notes		
			MSb		LSb				
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRWF	-	Clear W	1	00	0001	0000	0011	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECWF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOF	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDI	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

- Note 1:** When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.
- 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOF.

Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

PIC16C63A/65B/73B/74B

14.1 Instruction Descriptions

ADDLW **Add Literal and W**

Syntax: *[label]* ADDLW *k*
Operands: $0 \leq k \leq 255$
Operation: $(W) + k \rightarrow (W)$
Status Affected: *C*, *DC*, *Z*
Description: The contents of the *W* register are added to the eight bit literal '*k*' and the result is placed in the *W* register.

ANDWF **AND W with f**

Syntax: *[label]* ANDWF *f,d*
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(W) .AND. (f) \rightarrow (\text{destination})$
Status Affected: *Z*
Description: AND the *W* register with register '*f*'. If '*d*' is 0, the result is stored in the *W* register. If '*d*' is 1, the result is stored back in register '*f*'.

ADDWF **Add W and f**

Syntax: *[label]* ADDWF *f,d*
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(W) + (f) \rightarrow (\text{destination})$
Status Affected: *C*, *DC*, *Z*
Description: Add the contents of the *W* register with register '*f*'. If '*d*' is 0, the result is stored in the *W* register. If '*d*' is 1, the result is stored back in register '*f*'.

BCF **Bit Clear f**

Syntax: *[label]* BCF *f,b*
Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
Operation: $0 \rightarrow (f)$
Status Affected: *None*
Description: Bit '*b*' in register '*f*' is cleared.

ANDLW **AND Literal with W**

Syntax: *[label]* ANDLW *k*
Operands: $0 \leq k \leq 255$
Operation: $(W) .AND. (k) \rightarrow (W)$
Status Affected: *Z*
Description: The contents of *W* register are AND'ed with the eight bit literal '*k*'. The result is placed in the *W* register.

BSF **Bit Set f**

Syntax: *[label]* BSF *f,b*
Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
Operation: $1 \rightarrow (f)$
Status Affected: *None*
Description: Bit '*b*' in register '*f*' is set.

PIC16C63A/65B/73B/74B

BTFSS	Bit Test f, Skip if Set
Syntax:	[<i>label</i>] BTFSS f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b < 7$
Operation:	skip if (f) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead making this a 2TCY instruction.

CLRF	Clear f
Syntax:	[<i>label</i>] CLRF f
Operands:	$0 \leq f \leq 127$
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

BTFSC	Bit Test, Skip if Clear
Syntax:	[<i>label</i>] BTFSC f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if (f) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2 TCY instruction.

CLRW	Clear W
Syntax:	[<i>label</i>] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CALL	Call Subroutine
Syntax:	[<i>label</i>] CALL k
Operands:	$0 \leq k \leq 2047$
Operation:	(PC)+ 1 → TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRWDT	Clear Watchdog Timer
Syntax:	[<i>label</i>] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → \overline{TO} 1 → \overline{PD}
Status Affected:	\overline{TO} , \overline{PD}
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits \overline{TO} and \overline{PD} are set.

PIC16C63A/65B/73B/74B

COMF	Complement f
Syntax:	[<i>label</i>] COMF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(\bar{f}) \rightarrow (\text{destination})$
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.

GOTO	Unconditional Branch
Syntax:	[<i>label</i>] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	$k \rightarrow \text{PC}<10:0>$ $\text{PCLATH}<4:3> \rightarrow \text{PC}<12:11>$
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

DECF	Decrement f
Syntax:	[<i>label</i>] DECF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination})$
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

INCF	Increment f
Syntax:	[<i>label</i>] INCF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination})$
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

DECFSZ	Decrement f, Skip if 0
Syntax:	[<i>label</i>] DECFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination});$ skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead making it a 2 T _{cy} instruction.

INCFSZ	Increment f, Skip if 0
Syntax:	[<i>label</i>] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination}),$ skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead making it a 2 T _{cy} instruction.

PIC16C63A/65B/73B/74B

IORLW **Inclusive OR Literal with W**

Syntax: [*label*] IORLW k
Operands: $0 \leq k \leq 255$
Operation: (W) .OR. k \rightarrow (W)
Status Affected: Z
Description: The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k
Operands: $0 \leq k \leq 255$
Operation: k \rightarrow (W)
Status Affected: None
Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: (W) .OR. (f) \rightarrow (destination)
Status Affected: Z
Description: Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f
Operands: $0 \leq f \leq 127$
Operation: (W) \rightarrow (f)
Status Affected: None
Description: Move data from W register to register 'f'.

MOVF **Move f**

Syntax: [*label*] MOVF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: (f) \rightarrow (destination)
Status Affected: Z
Description: The contents of register f are moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

NOP **No Operation**

Syntax: [*label*] NOP
Operands: None
Operation: No operation
Status Affected: None
Description: No operation.

PIC16C63A/65B/73B/74B

RETFIE **Return from Interrupt**

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → PC,
 1 → GIE

Status Affected: None

RLF **Rotate Left f through Carry**

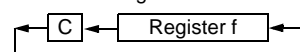
Syntax: [*label*] RLF f,d

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.



RETLW **Return with Literal in W**

Syntax: [*label*] RETLW k

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow (W)$;
 TOS → PC

Status Affected: None

Description: The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

RRF **Rotate Right f through Carry**

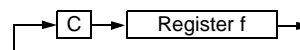
Syntax: [*label*] RRF f,d

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
 0 → WDT prescaler,
 1 → \overline{TO} ,
 0 → \overline{PD}

Status Affected: \overline{TO} , \overline{PD}

Description: The power-down status bit, \overline{PD} is cleared. Time-out status bit, \overline{TO} is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped. See Section 13.8 for more details.

PIC16C63A/65B/73B/74B

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k
Operands: $0 \leq k \leq 255$
Operation: $k - (W) \rightarrow (W)$
Status Affected: C, DC, Z
Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

XORLW **Exclusive OR Literal with W**

Syntax: [*label*] XORLW k
Operands: $0 \leq k \leq 255$
Operation: $(W) .XOR. k \rightarrow (W)$
Status Affected: Z
Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(f) - (W) \rightarrow (\text{destination})$
Status Affected: C, DC, Z
Description: Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(W) .XOR. (f) \rightarrow (\text{destination})$
Status Affected: Z
Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(f<3:0>) \rightarrow (\text{destination}<7:4>),$
 $(f<7:4>) \rightarrow (\text{destination}<3:0>)$
Status Affected: None
Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.