

PIC16F84A

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.

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.

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.

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.

BTFSC Bit Test, Skip if Clear

Syntax: [label] 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 2Tcy instruction.

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'.

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'.

BTFSS Bit Test f, Skip if Set

Syntax: [label] 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.

CALL Call Subroutine

Syntax: [label] CALL k

Operands: $0 \leq k \leq 2047$

Operation: $(PC)+1 \rightarrow \text{TOS}$,

$k \rightarrow \text{PC}<10:0>$,

$(\text{PCLATH}<4:3>) \rightarrow \text{PC}<12:11>$

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.

CLRF Clear f

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

CLRWDT Clear Watchdog Timer

Syntax: [label] CLRWDT
Operands: None
Operation: $00h \rightarrow WDT$
 $0 \rightarrow WDT$ prescaler,
 $1 \rightarrow TO$
 $1 \rightarrow PD$
Status Affected: TO, PD
Description: CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits TO and PD are set.

DECF Decrement f

Syntax: [label] 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'.

GOTO Unconditional Branch

Syntax: [label] GOTO k
Operands: $0 \leq k \leq 2047$
Operation: $k \rightarrow PC<10:0>$
 $PCLATH<4:3> \rightarrow PC<12:11>$
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 twocycle instruction.

CLRW Clear W

Syntax: [label] CLRW
Operands: None
Operation: $00h \rightarrow (W)$
 $1 \rightarrow Z$
Status Affected: Z
Description: W register is cleared. Zero bit (Z) is set.

COMF Complement f

Syntax: [label] COMF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(f) \rightarrow (\text{destination})$
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'.

DECFSZ Decrement f, Skip if 0

Syntax: [label] 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 2TCY instruction.

INCF Increment f

Syntax: [label] 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'.

INCFSZ Increment f, Skip if 0

Syntax: [label] 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 2TCY instruction.

IORWF Inclusive OR W with f

Syntax: [label] IORWF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(W) .OR. (f) \rightarrow (\text{destination})$
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'.

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.

NOP No Operation

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

RETLW Return with Literal in W

Syntax: [label] RETLW k
Operands: $0 \leq k \leq 255$
Operation: $k \rightarrow (W)$;
 $TOS \rightarrow PC$
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.

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.

MOVF Move f

Syntax: [label] MOVF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: $(f) \rightarrow (\text{destination})$
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.

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'.

RETFIE Return from Interrupt

Syntax: [label] RETFIE
Operands: None
Operation: $TOS \rightarrow PC$,
 $1 \rightarrow GIE$
Status Affected: None

RETURN Return from Subroutine

Syntax: [label] RETURN
Operands: None
Operation: $TOS \rightarrow PC$
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.

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'.

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'.

SLEEP

Syntax: [label] SLEEP

Operands: None

Operation: $00h \rightarrow$ WDT,

$0 \rightarrow$ WDT prescaler,

$1 \rightarrow$ TO,

$0 \rightarrow$ PD

Status Affected: TO, PD

Description: The power-down status bit, PD is cleared. Time-out status bit, TO is set. Watchdog Timer and its prescaler are cleared.

The processor is put into SLEEP mode with the oscillator stopped.

Register f C

Register f C

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.

SUBWF Subtract W from f

Syntax: [label] SUBWF f,d

Operands: $0 \leq f \leq 127$

$d \in [0,1]$

Operation: $(f) - (W) \rightarrow$ (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'.

SWAPF Swap Nibbles in f

Syntax: [label] SWAPF f,d

Operands: $0 \leq f \leq 127$

$d \in [0,1]$

Operation: $(f<3:0>) \rightarrow$ (destination<7:4>),

$(f<7:4>) \rightarrow$ (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'.

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.

XORWF Exclusive OR W with f

Syntax: [label] XORWF f,d

Operands: $0 \leq f \leq 127$

$d \in [0,1]$

Operation: $(W) .XOR. (f) \rightarrow$ (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'.