

35

PIC

小川 晃

PIC

PIC16F84
 35 (addressing)
 14
 (mnemonic)
 1 1
 + f, d, b, k
 (1).
 f , 8 가
 d , d=0
 W , d=1
 f 1
 b , 0 7 가
 k () , 8 가

. < > ()
 MOVF (move f)
 < > MOVF f, d (Z)
 f d=0 W
 , d=1 f Z 가
 【 】
 MOVF MEMORY, 0 : W
 MOVF MEMORY, 1 :
 MOVF MEMORY, 1
 , Z 가 0
 MOVLW (move literal to W)
 < > MOVLW k()
 8 k W
 MOVWF (move W to f)
 < > MOVWF f()
 W f
 【 】
 MOVLW n: n W



1. (3)

				14
ADDLW k	가 : (W)+k (W)	C, CD, Z	1	11 111x kkkk kkkk
ADDWF f, d	가 : (W)+f (dest)	C, DC, Z	1	00 0111 dfff ffff
ANDLW k	AND : (W).AND.(k) (W)	Z	1	11 1001 kkkk kkkk
ANDWF f, d	AND : (W).AND.(f) (dest)	Z	1	00 0101 dfff ffff
BCF f, b	: 0 f"b"		1	01 00bb bfff ffff
BSF f, b	: 1 f"b"		1	01 01bb bfff ffff
BTFSC f, b	: (f"b")=0		1(2)	01 10bb bfff ffff
BTFSS f, b	: (f"b")=1		1(2)	01 11bb bfff ffff
CALL k	(PC)+1 TOS, k PC"10: 0", (PCLATH"4: 3") PC(12: 11)		2	10 0kkk kkkk kkkk
CLRF f	: 00h f	Z(=1)	1	00 0001 lfff ffff
CLRWF	: 00h (W)	Z(=1)	1	00 0000 0xxx xxxx
CLRWDW	: 00h WDT, 0 WDT	PD(=1)	1	00 0000 0110 0100
COMF f, d	: (f) (dest)	T0(=1)	1	00 1001 dfff ffff
DECF f, d	: (f)-1 (dest)	Z	1	00 0011 dfff ffff
DECFSZ f, d	: (f)-1 (dest)	Z	1(2)	00 1011 dfff ffff
GOTO k	가 0 k PC"10: 0", PCLSTH"4: 3" PC"12: 11"		2	10 1kkk kkkk kkkk
INCF f, d	: (f)+1 (dest)	Z	1	00 1010 dfff ffff
INCFSZ f, d	: (f)+1 (dest), 가 0	Z	1(2)	00 1111 dfff ffff
IORLW k	OR : (W).OR.(k) (W)		1	11 1000 kkkk kkkk
IORWF f, d	OR : (W).OR.(f) (dest)	Z	1	00 0100 dfff ffff
MOVLW k	: k (W)		1	11 00xx kkkk kkkk
MOVF f, d	: (f) (dest)		1	00 1000 dfff ffff
MOVWF f	: (W) (f)		1	00 0000 1fff ffff
NOP	No operation		1	00 0000 0xx0 0000
RETFIE	: TOS PC, 1 GIE		2	00 0000 0000 1001
RETLW k	W : k W, TOS PC		2	11 01xx kkkk kkkk
RETURN	: TOS PC		2	00 0000 0000 1000
RLF f, d	: f"n" d"n+1", f"7" C, C d"0"	C	1	00 1101 dfff ffff
RLF f, d	: f"n" d"n-1", f"0" C, C d"7"	C	1	00 1100 dfff ffff
SLEEP	: 00h WDT	PD(=1)	1	00 0000 0110 0011
SUBLW k	:k-(W) (W)	TO(=0)	1	11 110x kkkk kkkk
SUBWF f, d	: f-(W) (dest)	C, DC, Z	1	00 0010 dfff ffff
SWAPF f, d	4 4 : f"0: 3" d"4: 7", f"4: 7" d"0: 3"		1	00 1110 dfff ffff
XORLW k	XOR : (W).XOR.(k) (W)	Z	1	11 1010 kkkk kkkk
XORWF f, d	XOR : (W).XOR.(f) (dest)	Z	1	00 0110 dfff ffff

MOVWF MEMORY : W
SWAPF (swap nibbles in f)

< > SWAPF f, d()
f 4 4

d=0 W
 d=1 f
【 】
 SWAPF MEMORY, 0 :
 W

ADDLW (add literal and W)
 < > ADDLW k(C, DC, Z)
 W 8 k 가 ,
 W
 C, DC, Z 가 .

【 】
 ADDLW H'30' : W 30h 가
 ADDWF (add W and f)
 < > ADDWF f, d(C, DC, Z)
 W f 가 ,
 d=0 W d=1
 C, DC,
 Z 가 .

【 】
 ADDWF MEMORY, 0 : +W W
 SUBLW (subtract W from literal)
 < > SUBLW k(C, DC, Z)
 8 k W 2
 W
 C, DC, Z 가 .

【 】
 SUBLW H'30' : 30h W
 SUBWF (subtract W from f)
 < > SUBWF f, d(C, DC, Z)
 f W 2
 , d=0 W
 d=1 f
 C, DC, Z 가 .

【 】
 SUBWF MEMORY, 0 : W
 ANDLW (and literal with W)
 < > ANDLW k(Z)
 W 8 k AND

 , W
 Z 가 .

【 】
 ANDLW B'11110000' : 4 0
 ANDLW 0 : CLRW
 ANDWF (and W with f)
 < > ANDWF f, d(Z)
 W f AND
 , d=0 W
 D=1 f
 Z 가 .

【 】
 MOVLW B'1111 :
 ANDWF MEMORY, 1 :

IORLW (inclusive OR literal with W)
 < > IORLW k(Z)
 W 8 k OR
 , W
 Z 가 .

【 】
 IORLW B'11110000' : 1
 IORLW H'FF' : W 1
 IORWF (inclusive OR W with f)
 < > IORWF f, d(Z)
 W f OR
 , d=0 W
 d=1 f
 Z 가 .

【 】
 MOVLW B'1111' :
 IORWF MEMORY, 1 : 1

XORLW (exclusive OR with W)
 < > XORLW k(Z)
 W 8 k XOR
 , W
 Z 가 .

【 】
 XORLW B'11110000' :
 XORLW H'FF' : W (COMF



.)
 XORWF (exclusive OR W with f)
 < > XORWF f, d(Z)
 W, d=0 W, d=1 f Z 가 .
 XOR

【 】
 MOVLW B'1111' :
 XORWF MEMORY, 1 :
 DECF (decrement f)
 < > DECF f, d(Z)
 f (-1), d=0 W, d=1 f Z 가 .

【 】
 DECF MEMORY, 1 :
 INCF (increment f)
 < > INCF f, d(Z)
 f (+1), d=0 W, d=1 f Z 가 .

【 】
 INCF MEMORY, 1 :
 ()

RLF (rotate left through carry)
 < > RLF f, d(C)
 C 가 f 1, d=0 W, d=1 f . C
 C 가 .

【 】
 BCF STATUS, 0 : C (clear)
 RLF MEMORY, 1 :
 RRF (rotate right f through carry)
 < > RRF f, d(C)

C 가 f 1, d=0 W, d=1 f .
 C 가 .

【 】
 BCF STATUS, 0 : C
 RRF MEMORY, 1 :
 BCF (bit clear f)
 < > BCF f, b()
 f b .

【 】
 BCF PORTA, 2 : PORTA 2
 BCF STATUS, 0 : C
 BCF STATUS, 5 : 0 .
 BSF (bit set f)
 < > BSF f, b()
 f b 1 .

【 】
 BSF PORTA, 5 : PORTA 5
 BSF STATUS, 0 : C
 BSF STATUS, 5 : 1 .

BTFSF (bit test f, skip if set)
 < > BTFSF f, b()
 f b가 1, 0 . 2 .

【 】
 BTFSF STATUS, 0 : C 가 0
 GOTO LABEL : C 가 1
 BTFSS (bit test f, skip if set)
 < > BTFSS f, b()
 f b가 0, 1 . 2 .

【 】

BTFSS STATUS, 2 : Z 가 1
 GOTO LABEL : Z 가 0
 DECFSZ (decrement f, skip if 0)
 < > DECFSZ f, d()
 f (-1) , d=0
 W , d=1
 f 가 1
 , 0

【 】
 LOOP INCF MEMORY, 1
 DECFSZ CNT, 1 : CNT=0
 GOTO LOOP
 INCFSZ (increment f, skip if 0)
 < > INCFSZ f, d()
 f (+1) , d=0
 W , d=1
 f 가 1
 , 0

GOTO
 < > GOTO k()
 . 11 k
 . 2

CALL (call subroutine)
 < > CALL k()
 (待避) , 11 k
 . 2

RETURN (return from subroutine)
 < > RETURN ()
 . 2

【 】
 CALL ABC : ABC

GOTO LABEL :
 ABC INCF MEMORY, 1 : +1
 RETURN :
 RETLW (return with literal in W)
 < > RETLW k()
 8 k W
 . 가 ,
 2

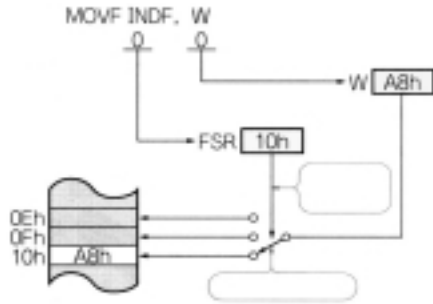
【 】
 BTFSC STATUS, 2 : Z
 RETLW 0 : 0 W=0
 RETLW 1 : 1 W=1
 RETFIE (return from interrupt)
 < > RETFIE ()
 . GIE
 . 2

NOP (no operation)
 < > NOP ()
 . 1

CLRF (clear f)
 < > CLRF f(Z=1)
 f . Z 1

CLRW (clear W)
 < > CLRW (Z=1)
 W . Z
 1

COMF (complement f)
 < > COMF f, d(Z)
 f (), d=0
 W , D=1
 f . Z
 가



1.

CLRWDWT (clear watching timer)

< > CLRWDWT ()
(WDT)

WDT
TO, PD

SLEEP

< > SLEEP ()

PIC PD 0, TO
1 WDT

PIC

가

가

MPASM

LIST

< > LIST P=device_name
PIC

【 】

LIST P=16F84

END

< > END

EQU (equate)

1.

```

BTFSK STATUS, 2
GOTO Label ;BZ
BTFSK STATUS, 2
GOTO Label ;BNZ
BTFSK STATUS, 0
GOTO Label ;BC
BTFSK STATUS, 0
GOTO Label ;BNC
    
```

2.

```

BZ Label ;
BNZ Label ; 가
BC Label ;C=1
BNC Label ;C=0
    
```

3.

```

SUBLW n ;n-W
BTFSK STATUS, 2 ;BZ
GOTO Label ;W = n

(a) W n

SUBLW n ;n-W
BTFSK STATUS, 0 ;BC
GOTO Label ;W < n

(b) W n

SUBLW n ;n-W
BTFSK STATUS, 0 ;BNC
GOTO Label ;W > n

(c) W n
    
```

4.

```

WSAVE EQU nn
CCRSAVE EQU nn
PUSH MOVWF WSAVE ;W
      SWAPF STATUS, 0 ;
      MOVWF CCRSAVE ;

POP SWAPF CCRSAVE, 0 ;
   MOVWF STATUS ;
   SWAPF WSAVE, 1 ;W
   SWAPF WSAVE, 0 ;
   RETFIE ;
    
```

5.

```

LOOPCNT EQU nn ;LOOPCNT
MEMCLR MOVLW H'20' ; 20h
        MOVWF FSR ;FSR = 20h
        MOVLW 16 ;16
        MOVWF LOOPCNT ;LOOPCNT = 16
NEXT CLR FSR ;
      INCF FSR ; +1
      DECFSZ LOOPCNT ;LOOP ?
      GOTO NEXT ;
    
```

PIC

6.

TABLE	ADDWF	PC, 1	;PC+W	jump
RETLW	B'10000000'		;W=0	80h Get
RETLW	B'01000000'		;W=1	40h Get
RETLW	B'00100000'		;W=2	20h Get
RETLW	B'00010000'		;W=3	10h Get
RETLW	B'00001000'		;W=4	8 Get
RETLW	B'00000100'		;W=5	4 Get
RETLW	B'00000010'		;W=6	2 Get
RETLW	B'00000001'		;W=7	1 Get

7.

TABLE	ADDWF	PC, 1	;PC+W	jump
GOTO	TASK 1		;W=0	TASK 1
GOTO	TASK 2		;W=1	TASK 2
GOTO	TASK 3		;W=2	TASK 3
GOTO	TASK 4		;W=3	TASK 4

< > Name EQU nn
()

【 】
Label EQU H'F0'
PORTA EQU 5

PIC

PIC 가

Z C

. PIC 가

Z C

1

, MPASM

2 BZ,

BNZ, BC, BNC가

, PIC

. SUBLW

3

W
(待避)

. PIC

PUSH, POP

4

WSAVE CCRSAVE

W

(FSR)

PIC

(FSR, 4)

(INDF, 0)

1

FSR

FSR

INDF

5

20h

16

0

FSR

가

, ROM

ROM

W

RETLW

, RETLW

가

6

W

0 7

. W

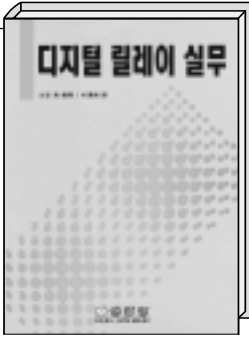
RETLW

GOTO

가가 (7).



PICSTAR Plus 가 가
 가
 MPLAB
 . PIC
 Windows
 MPLAB , PIC
 가 Window
 PIC
 PIC , I/O
 (1) PIC 가
 ASCH PIC
 (2)
 HEX PIC C . C 가
 (3) HEX , PIC PIC . PIC C
 E E
 PRO MATE



- 1 , 2
 , 3 , 4
 5 , 6 , 7

(:三谷 泉/ : / : / 180 / 가 10,000)