

Arithmetic and Logic Instructions



Hsiao-Lung Chan
Dept Electrical Engineering
Chang Gung University, Taiwan
chanhl@mail.cgu.edu.tw

Objectives

- Arithmetic operations of unsigned numbers
- Arithmetic operations of signed numbers
- Logical instructions
- Compare instructions
- Rotation instructions and data serialization

Arithmetic operations of unsigned numbers

- Addition (Examples 5-2, 5-3)
 - Decimal adjust WREG (DAW) (Example 5-4)
- Subtraction (Examples 5-5, 5-6, 5-7)
- Multiplication (Page 163)
- Division (Page 163 & Example 5-8)

BCD (binary coded decimal) number system

- Unpacked BCD

- 59 is represented by "0000 0101" and "0000 1001"

- Packed BCD

- 59 is represented by 59H "0101 1001"
- Efficient in storing data

<i>Digit</i>	<i>BCD</i>
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Decimal adjust WREG (DAW) instruction

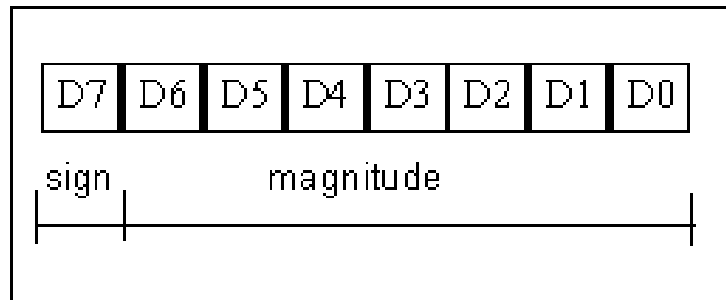
MOVLW	0x47	; WREG = 47H
ADDLW	0x25	; WREG = 6CH, DC = 0, C = 0
DAW		; Adjust for BCD addition by adding 6 (WREG=72H) C = 0
MOVLW	0x09	; WREG = 09H
ADDLW	0x08	; WREG = 11H, DC = 1, C = 0
DAW		; Adjust for BCD addition by adding 6 (WREG=17H) C = 0
MOVLW	0x52	; WREG = 52H
ADDLW	0x87	; WREG = D9H, DC = 0, C = 0
DAW		; Adjust for BCD addition by adding 60H (WREG=39H) C = 1
MOVLW	0x57	; WREG = 57H
ADDLW	0x77	; WREG = CEH, DC = 0, C = 0
DAW		; Adjust for BCD addition by adding 66H (WREG=34H) C = 1

Add 0110 to lower 4 bits if lower nibble > 9 or if DC = 1

Add 0110 to upper 4 bits if upper nibble >9 or if C = 1

Signed number arithmetic

- Positive vs. negative numbers



- Range for 8-bit number
 - Unsigned numbers: 0 ~ 255
 - Signed numbers: -128 ~ 127

Overflow in signed number operation

- OV is set if either two conditions occurs
 - A carry from D6 to D7 but no carry out of D7 ($C=0$)
 - A carry from D7 out ($C=1$) but no carry from D6 to D7
- Signed number operations (Examples 5-13,14,15 & 16)

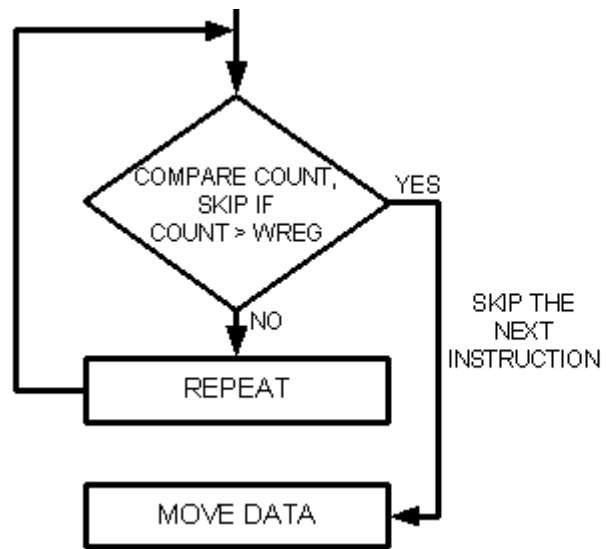
Logic operations

- `ANDLW K`
- `IORLW K ; Inclusive OR`
- `XORLW K ; Exclusive OR`

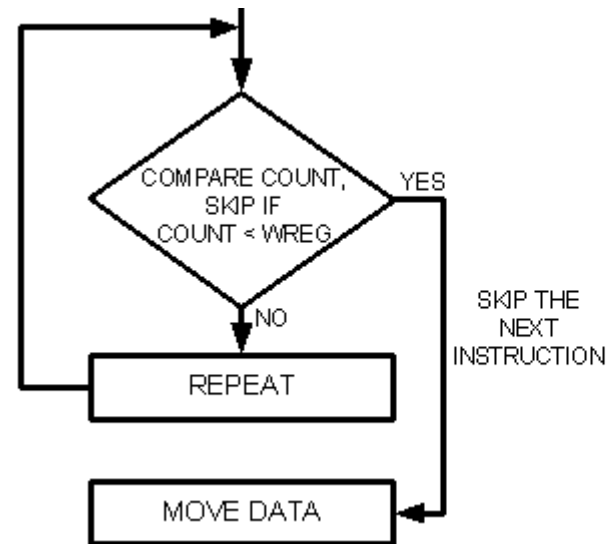
- Applications
 - Set bits using `IORLW` (Example 5-18)
 - Clear bits using `ANDLW` (Example 5-18)
 - Test a value using `XORLW` (Example 5-21)

- **What are `ANDWF`, `IORWF`, `XORWF`?**

Compare instructions



AGAIN CPFSGT COUNT
GOTO AGAIN
MOVWF PORTB



AGAIN CPFSLT COUNT
GOTO AGAIN
MOVWF PORTB

Compare examples

- Test a value using CPFSEQ (Example 5-24)
 - What's the difference between XORLW and CPFSEQ
- Compare a value using CPFSGT & CPFSLT (Example 5-26)

Rotate instructions

Mnemonic, operator	Description	16-bit instruction word	Status affected
RLCF f, d, a	Rotate left f through carry	0011 01da ffff ffff	C, Z,N
RLNCF f, d, a	Rotate left f (no carry)	0100 11da ffff ffff	Z,N
RRCF f, d, a	Rotate right f through carry	0011 00da ffff ffff	C, Z,N
RRNCF f, d, a	Rotate right f (no carry)	0100 00da ffff ffff	Z, N

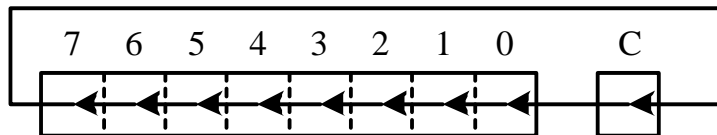


Figure 2.17 Operation performed by the **rlcf f,d,a** instruction

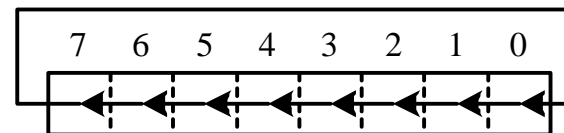


Figure 2.18 Operation performed by the **rlncf f,d,a** instruction

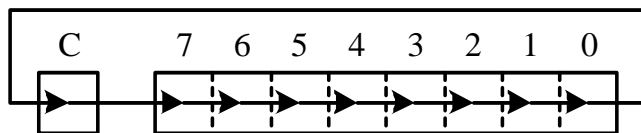


Figure 2.19 Operation performed by the **rrcf f,d,a** instruction

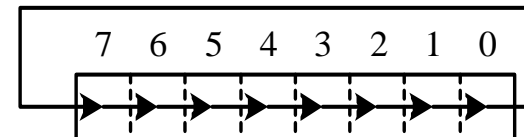


Figure 2.20 Operation performed by the **rrncf f,d,a** instruction

Data serialization via rotation instructions

- Output data bits serially (Example 5-28)
- Input data bits serially (Example 5-29)
- Test the bits in a value (Example 5-30)

SWAPF fileReg, d



```
MYREG EQU          0x20
    MOVLW          0x72
    MOVWF          MYREG
    SWAPF          MYREG, F
```

Reference

- M.A. Mazidi, R.D. Mckinlay, D Causey, PIC Microcontroller and Embedded Systems Using Assembly and C for PIC18, Pearson Education Inc., 2008.
- Han-Way Huang, PIC Microcontroller: An Introduction to Software and Hardware Interfacing, Thomson Delmar Learning, 2005.