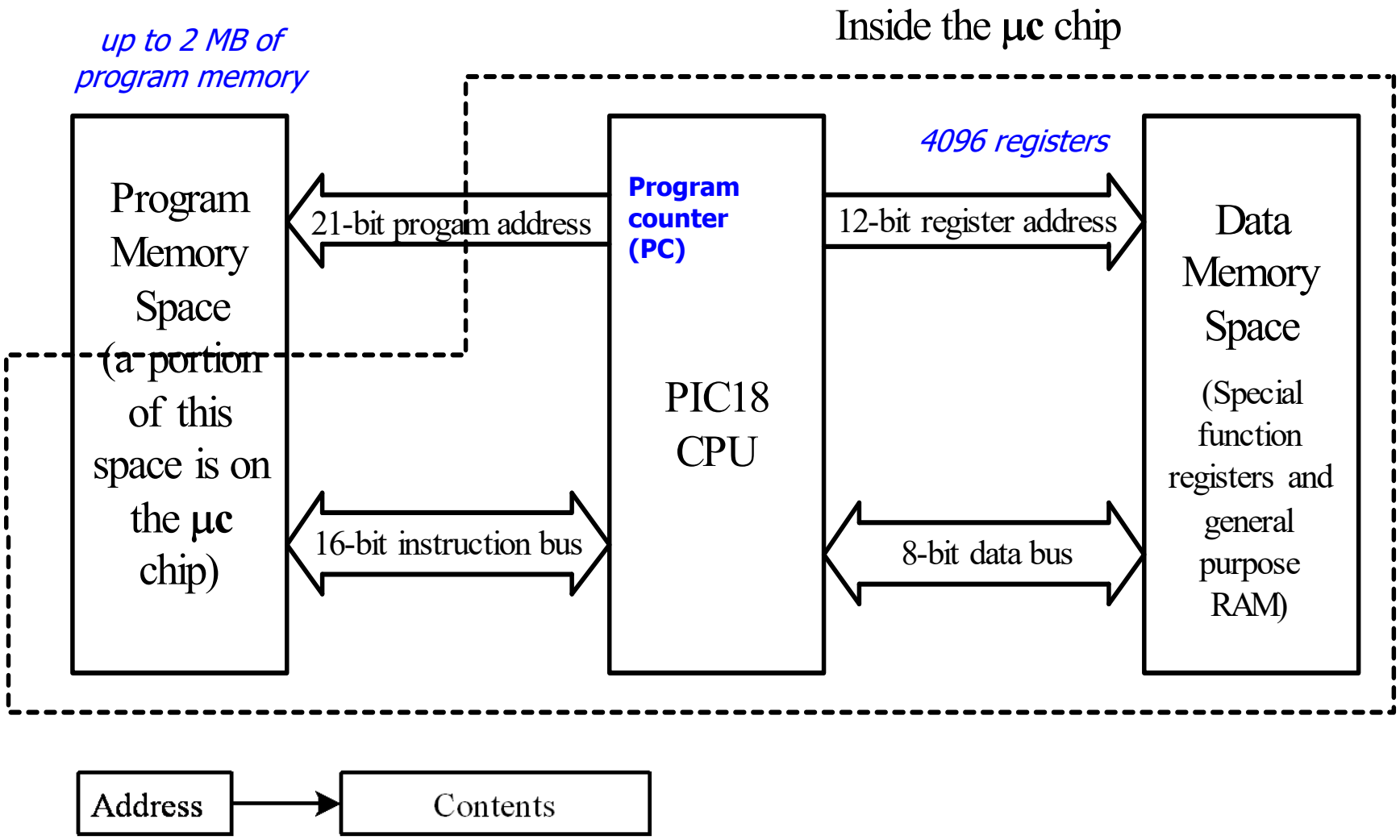
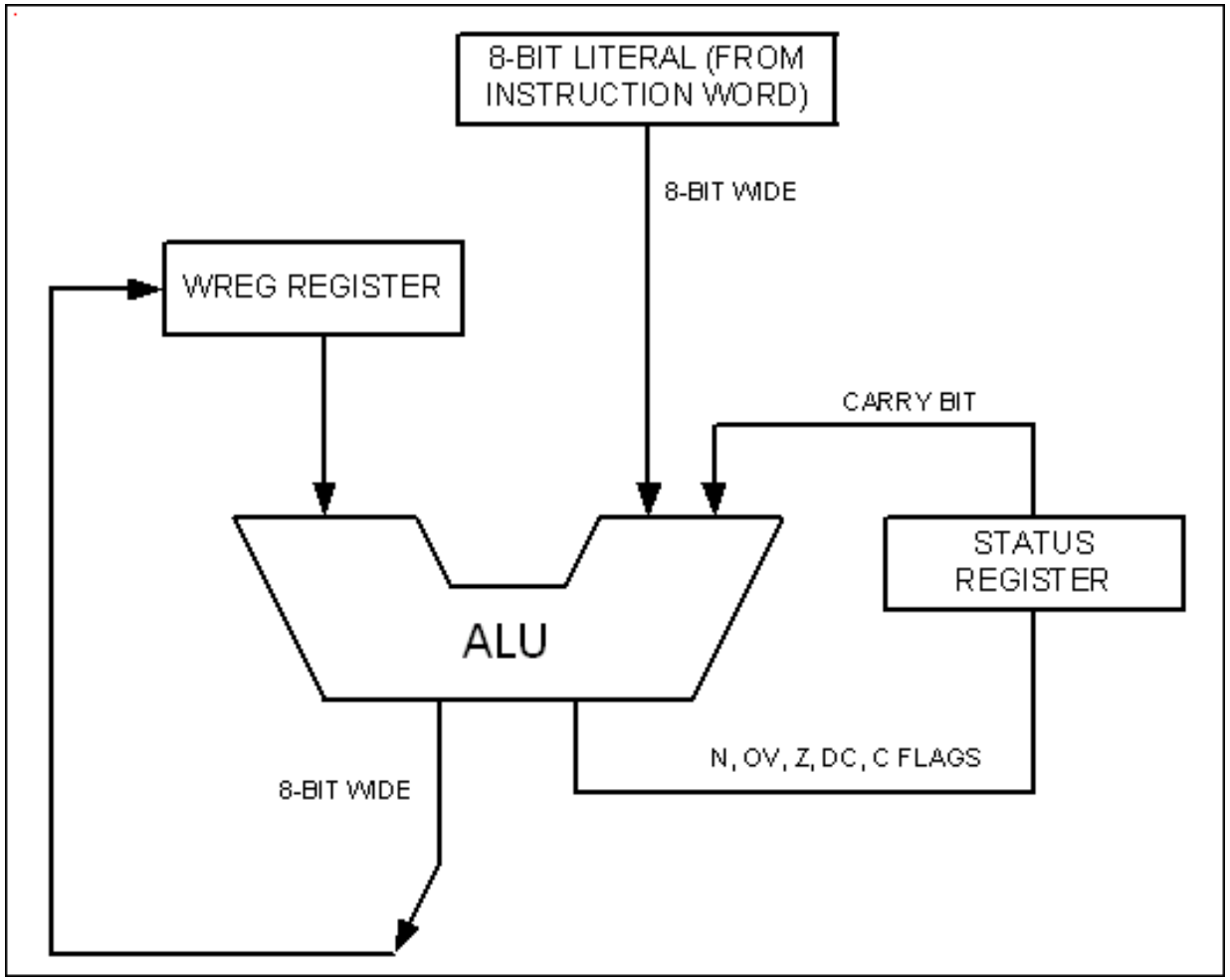
PIC Mov Instructions

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

PIC18 memory access



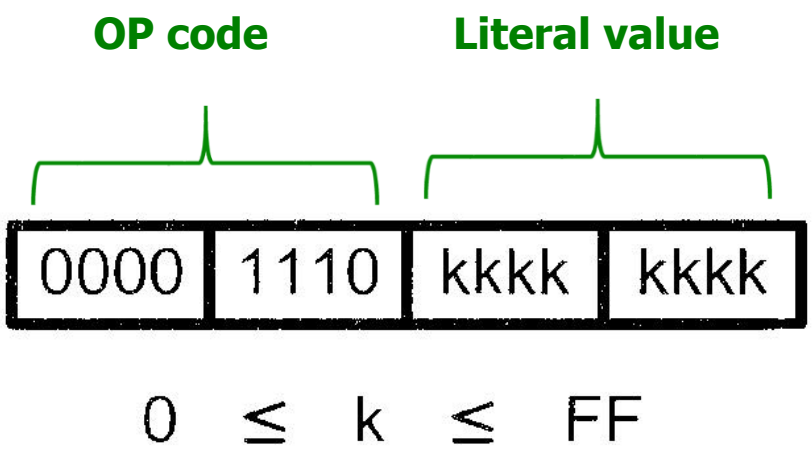
ALU with working register (WREG) and literal value



MOVLW instruction

- MOVLW K ; move literal value K into WREG
  - MOVLW 25H
  - MOVLW 7F2H ; illegal 7F2H > 8 bits (FFH), become F2H

- Instruction format (16 bits)



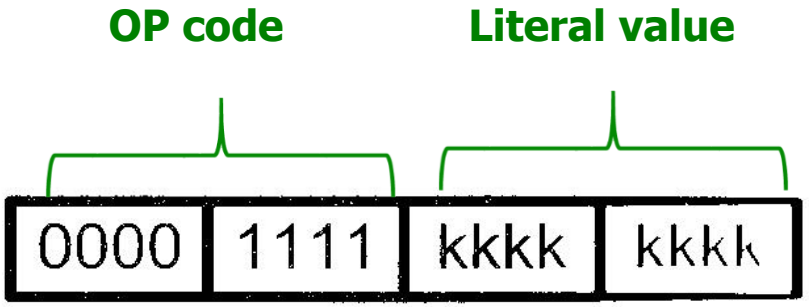
ADDLW instruction

- ADDLW K ; ADD literal value K to WREG

  MOVLW 25H

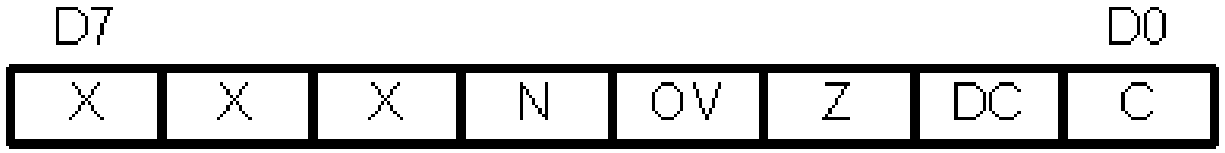
  ADDLW 34H

- Instruction format (16 bits)



$0 \leq k \leq FF$

PIC18 status register



C – Carry flag **(Carry from D7 bit during an ADD or SUB operation)**

DC – Digital Carry flag **(Carry from D3 to D4 bit)**

Z – Zero flag **(Binary coded decimal (BCD) operation)**

OV – Overflow flag **(for signed number operation)**

N – Negative flag **(for signed number operation)**

X – D5, D6, and D7 are not implemented, and reserved for future use.

Status register example

Show the status of the C, DC, and Z flags after the addition of 38H and 2FH in the following instructions:

```
MOVLW 38H
ADDLW 2FH          ;add 2FH to WREG
```

Solution:

	38H	0011 1000	
+	2FH	0010 1111	
	67H	0110 0111	WREG = 67H

C = 0 because there is no carry beyond the D7 bit.
DC = 1 because there is a carry from the D3 to the D4 bit.
Z = 0 because the WREG has a value other than 0 after the addition.

Status register example (cont.)

Show the status of the C, DC, and Z flags after the addition of 9CH and 64H in the following instructions:

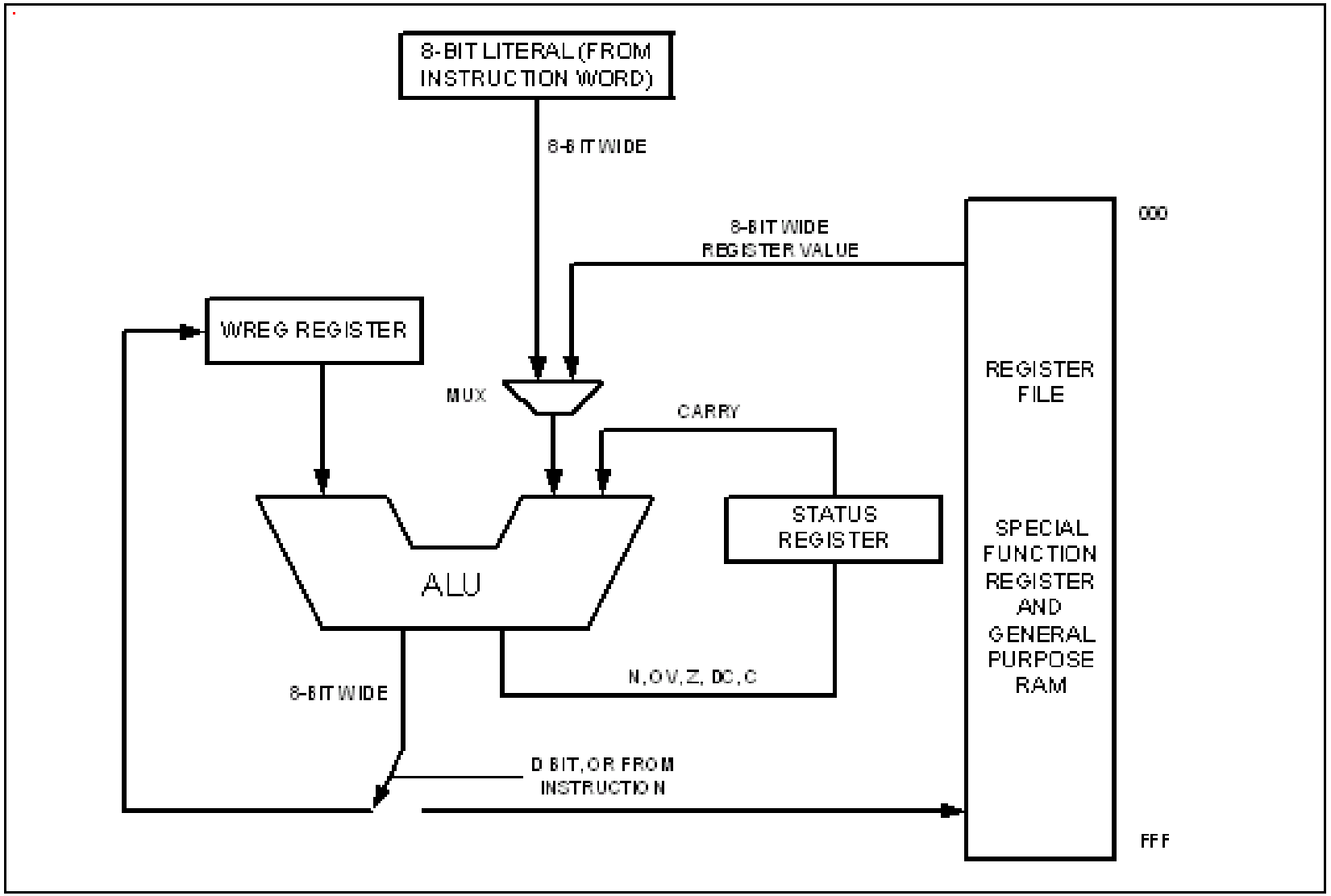
```
MOVLW  9CH
ADDLW  64H          ;add 64H to WREG
```

Solution:

9CH	1001 1100	
+ 64H	0110 0100	
100H	0000 0000	WREG = 00

C = 1 because there is a carry beyond the D7 bit.
DC = 1 because there is a carry from the D3 to the D4 bit.
Z = 1 because the WREG has a value 0 in it after the addition.

WREG, fileReg, and ALU in PIC18



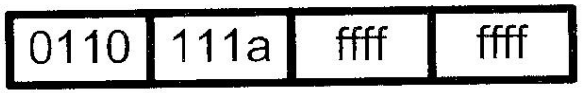
MOVWF instruction

- Copy WREG content to a file register
  - Ex. 1

```
MOVLW   55H
MOVWF   PORTB     ; copy WREG contents to port B
```

  - Ex. 2

```
MOVLW   99H
MOVWF   12H       ; copy WREG contents to location 12h
```



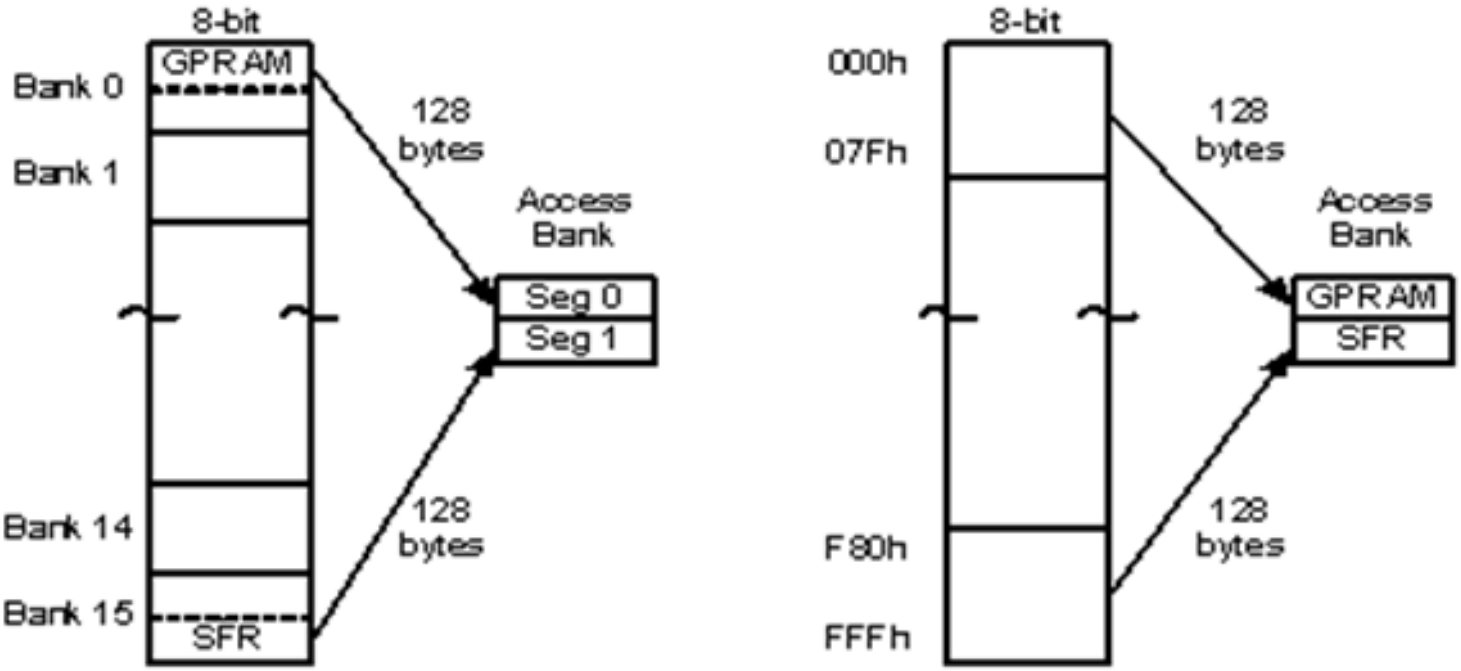
0 ≤ f ≤ FF

a = 0 : access bank is used. ← **default**

a = 1 : access bank is specified by the BSR register.

PIC18 file registers

- General-purpose registers (GPRs)
  - Data storage
- Special functional registers (SFRs)
  - ALU status, timers, serial communication, I/O ports, etc



SFRs in PIC18

F80h	PORTA	FA0h	PIE2	FC0h	----	FE0h	BSR
F81h	PORTB	FA1h	PIR2	FC1h	ADCON1	FE1h	FSR1L
F82h	PORTC	FA2h	IPR2	FC2h	ADCON0	FE2h	FSR1H
F83h	PORTD	FA3h	----	FC3h	ADRESL	FE3h	PLUSW1 *
F84h	PORTE	FA4h	----	FC4h	ADRESH	FE4h	PREINC1 *
F85h	----	FA5h	----	FC5h	SSPCON2	FE5h	POSTDEC1 *
F86h	----	FA6h	----	FC6h	SSPCON1	FE6h	POSTINC1 *
F87h	----	FA7h	----	FC7h	SSPSTAT	FE7h	INDF1 *
F88h	----	FA8h	----	FC8h	SSPADD	FE8h	WREG
F89h	LATA	FA9h	----	FC9h	SSPBUF	FE9h	FSR0L
F8Ah	LATB	FAAh	----	FCAh	T2CON	FEAh	FSR0H
F8Bh	LATC	FABh	RCSTA	FCBh	PR2	FEBh	PLUSW0 *
F8Ch	LATD	FACh	TXSTA	FCCh	TMR2	FECh	PREINC0 *
F8Dh	LATE	FADh	TXREG	FCDh	T1CON	FEDh	POSTDEC0 *
F8Eh	----	FAEh	RCREG	FCEh	TMR1L	FEEh	POSTINC0 *
F8Fh	----	FAFh	SPBRG	FCFh	TMR1H	FEFh	INDF0 *
F90h	----	FB0h	----	FD0h	RCON	FF0h	INTCON3
F91h	----	FB1h	T3CON	FD1h	WDTCON	FF1h	INTCON2
F92h	TRISA	FB2h	TMR3L	FD2h	LVDCON	FF2h	INTCON
F93h	TRISB	FB3h	TMR3H	FD3h	OSCCON	FF3h	PRODL
F94h	TRISC	FB4h	----	FD4h	----	FF4h	PRODH
F95h	TRISD	FB5h	----	FD5h	T0CON	FF5h	TABLAT
F96h	TRISE	FB6h	----	FD6h	TMR0L	FF6h	TBLPTRL
F97h	----	FB7h	----	FD7h	TMR0H	FF7h	TBLPTRH
F98h	----	FB8h	----	FD8h	STATUS	FF8h	TBLPTRU
F99h	----	FB9h	----	FD9h	FSR2L	FF9h	PCL
F9Ah	----	FBAh	CCP2CON	FDAh	FSR2H	FFAh	PCLATH
F9Bh	----	FBBh	CCPR2L	FDBh	PLUSW2 *	FFBh	PCLATU
F9Ch	----	FBCh	CCPR2H	FDCh	PREINC2 *	FFCh	STKPTR
F9Dh	PIE1	FBDh	CCP1CON	FDDh	POSTDEC2 *	FFDh	TOSL
F9Eh	PIR1	FBEh	CCPR1L	FDEh	POSTINC2 *	FFEh	TOSH
F9Fh	IPR1	FBFh	CCPR1H	FDFh	INDF2 *	FFFh	TOSU

* - These are not physical registers.

MOVF instruction

- MOVF filereg, d
  - Copy the content of fileReg to WREG or itself

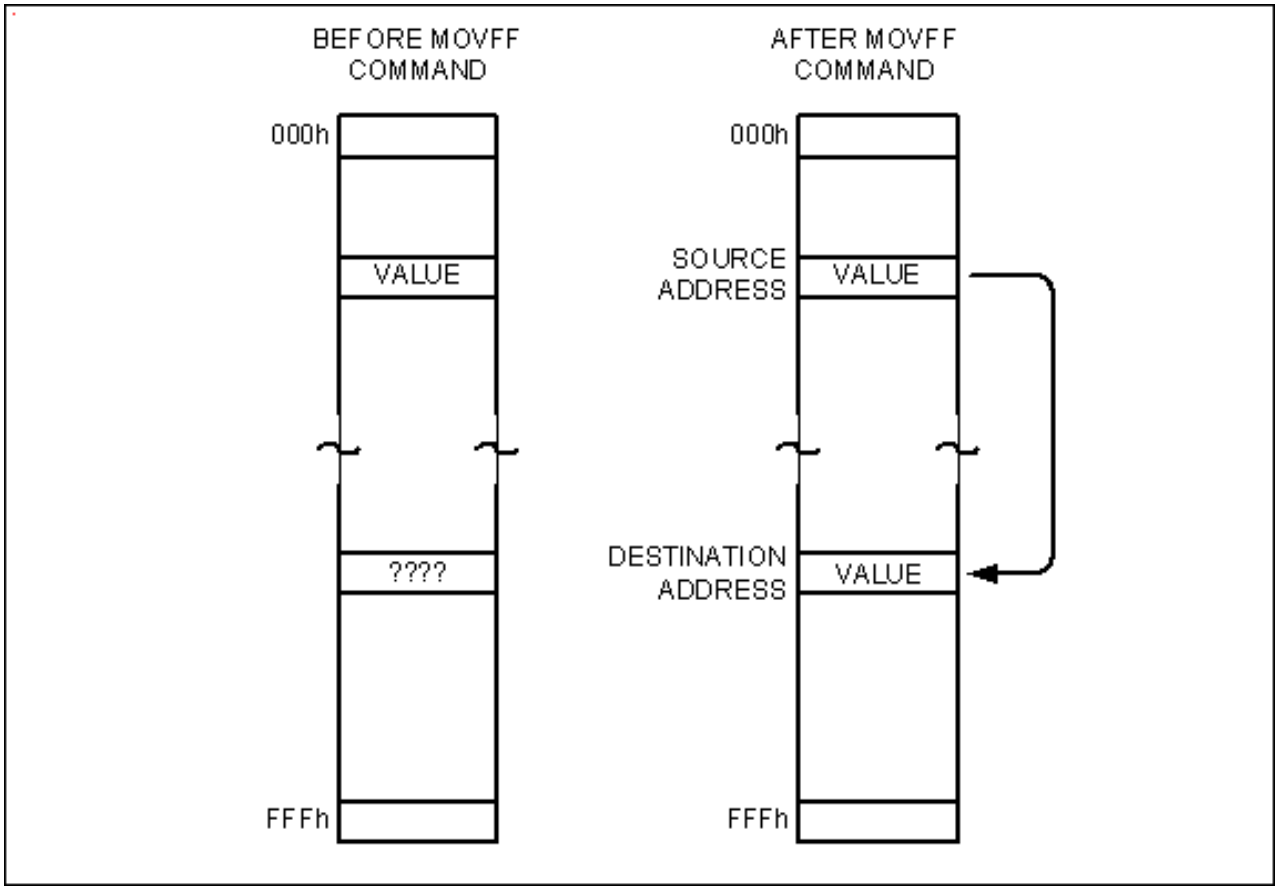
- Write a program to read data from PortB and send to PORTC continuously

  Ans:

```
AGAIN  MOVF    PORTB, W
       MOVWF   PORTC
       GOTO    AGAIN
```

MOVFF instruction

- Move data directly among fileReg locations
- 32-bit instruction



MOVFF instruction (cont.)

- Write a program to read data from PortB and send to PORTC continuously

  Ans:

```
AGAIN   MOVFF       PORTB, PORTC
        GOTO        AGAIN
```

MOVFF (32-bit instruction)

1100	ssss	ssss	ssss
1111	dddd	dddd	dddd

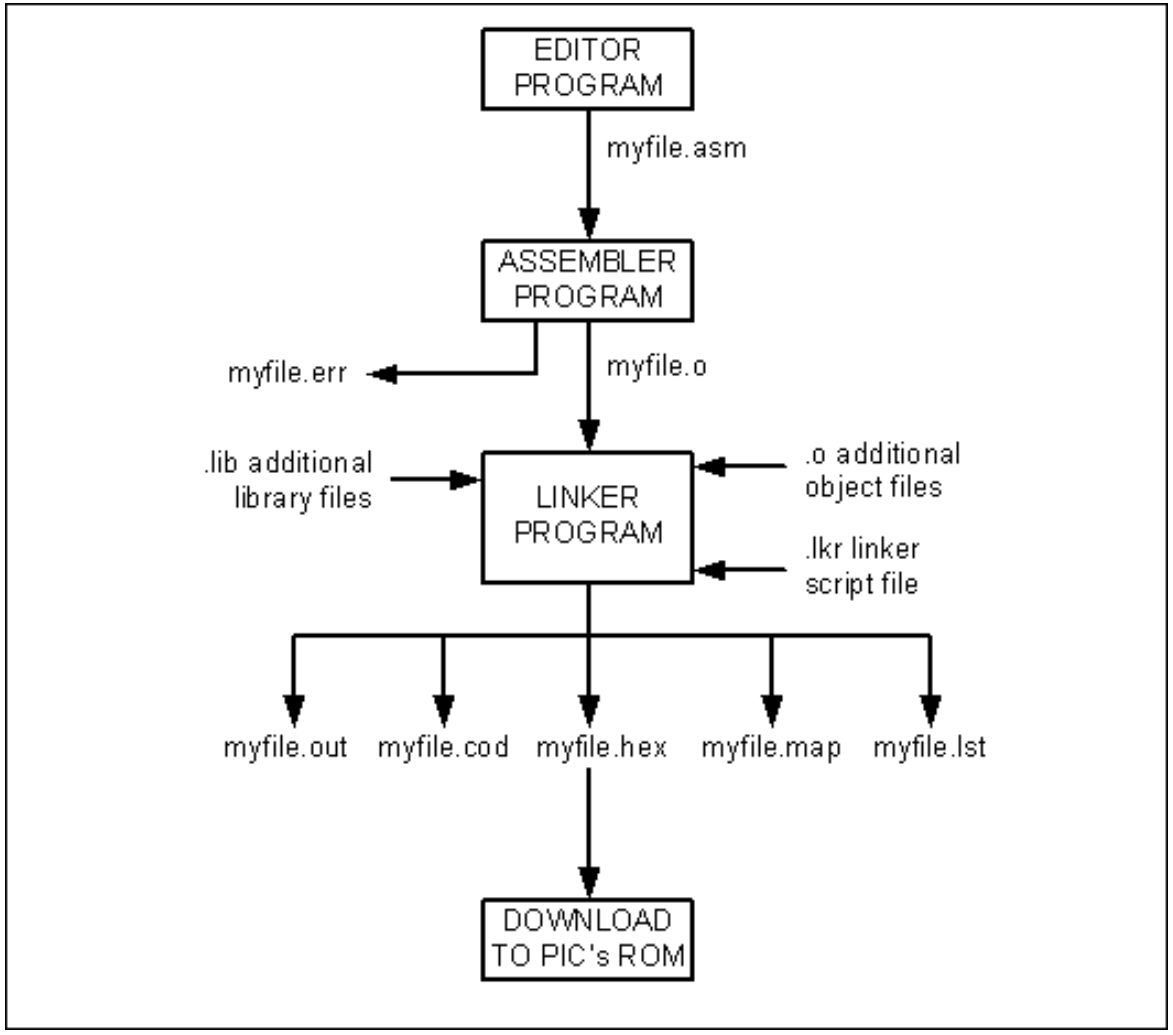
Source (f_s)

Destination (f_d)

$$0 \leq f_s \leq FFF$$

$$0 \leq f_d \leq FFF$$

Assembling and linking a PIC program

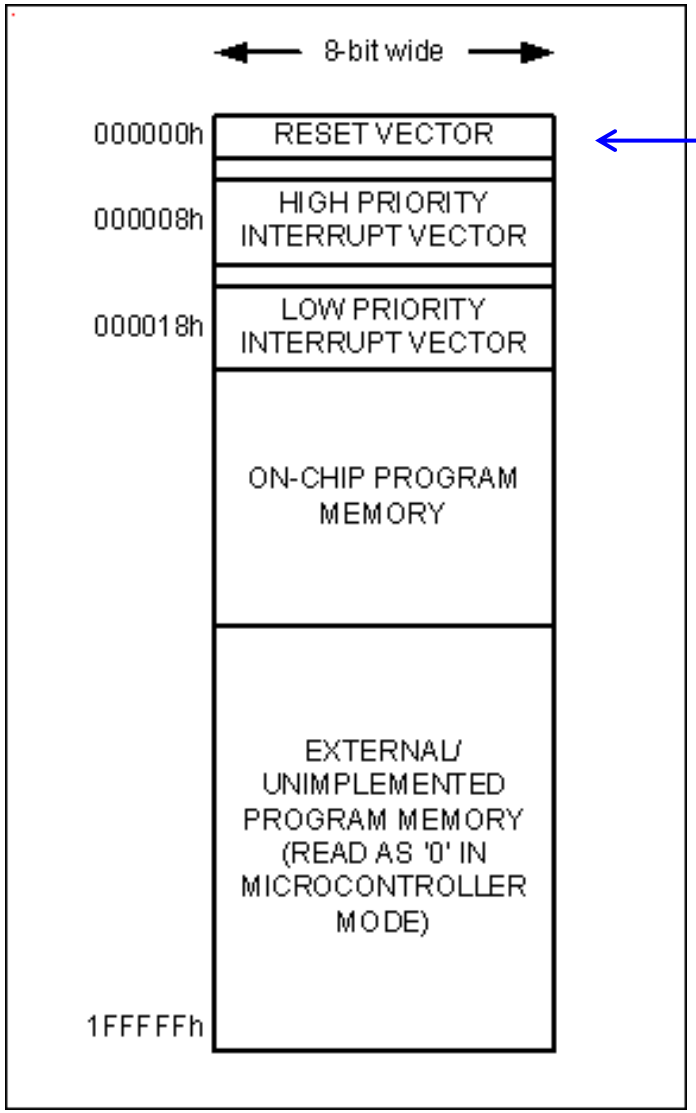


List file

```
LOC    OBJECT CODE LINE SOURCE TEXT          VALUE

                   00001
                   00002 ;PIC Asm Language Program To Add Some Data
                   00003 ;store sum in fileReg location 10H
  00000010         00004 SUM EQU 10H         ;RAM loc 10H for sum
                   00005
000000             00006 ORG 0H              ;start at address 0
000000 0E25        00007 MOVLW 25H           ;WREG = 25
000002 0F34        00008 ADDLW 0x34          ;add 34H to WREG
000004 0F11        00009 ADDLW 11H           ;add 11H to WREG
000006 0F12        00010 ADDLW D'18'         ;W = W + 12H = 7CH
000008 0F1C        00011 ADDLW 1CH           ;W = W + 1CH = 98H
00000A 0F06        00012 ADDLW B'00000110'   ;W = W + 6 = 9EH
00000C 6E10        00013 MOVWF SUM           ;save the sum in loc 10H
00000E EF07 F000   00014 HERE GOTO HERE      ;stay here forever
                   00015 END                 ;end of asm source file
```

PIC18 Program ROM space



Wake up at memory address 0000 when PIC is powered up