PIC Microcontroller and Embedded Systems

PIC Architecture & Assembly Language Programming

1

Outline

- PIC Status Register
- PIC data format and directive
- Intro. To PIC assembly language
- The Program Counter and program ROM space in the PIC
- RISC Architecture in the PIC

2

The WREG Register

- Many registers for arithmetic and logic operation.
- The WREG (WORking Register) Register is one of the most widely used registers of the PIC
- 8-bit register
- Any data larger than 8 bits must be broken into 8-bits chunks before it is processed.
- There is only one .

3

MOVLW

- Moves 8-bit data into WREG
- MOVLW k ; move literal value k into WREG
- Example
  - MOVLW 25H
  - MOVLW A5H
- Is the following code correct?
  - MOVLW 9H
  - MOVLW A23H

4

ADDLW

ADDLW k

- ; Add literal value k to WREG (k +WREG)

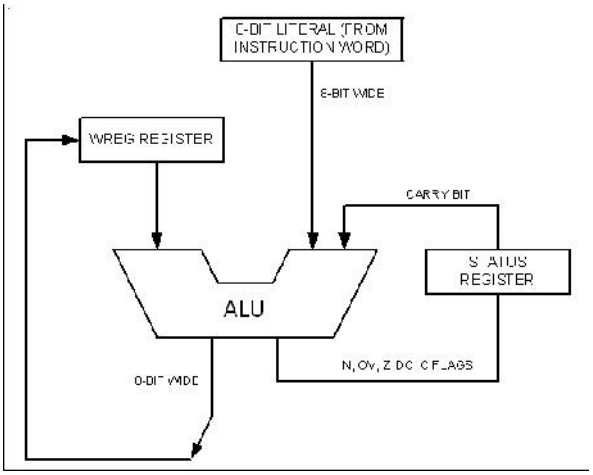
Example:

- MOVLW 12H ;
- ADDLW 16H ;
  - ADDKW 11H ;
  - ADDLW 43H ;

0	0	0	1	0	0	1	0
0	0	1	0	1	0	0	0
0	0	1	1	1	0	0	1
0	1	1	1	1	1	0	0

5

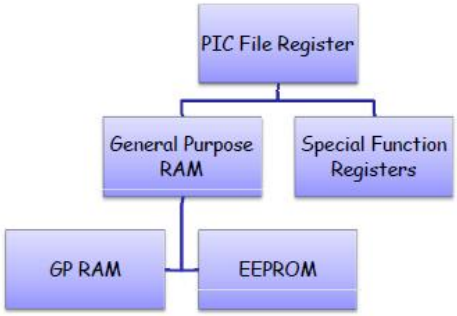
PIC WREG and ALU Using Literal Value



6

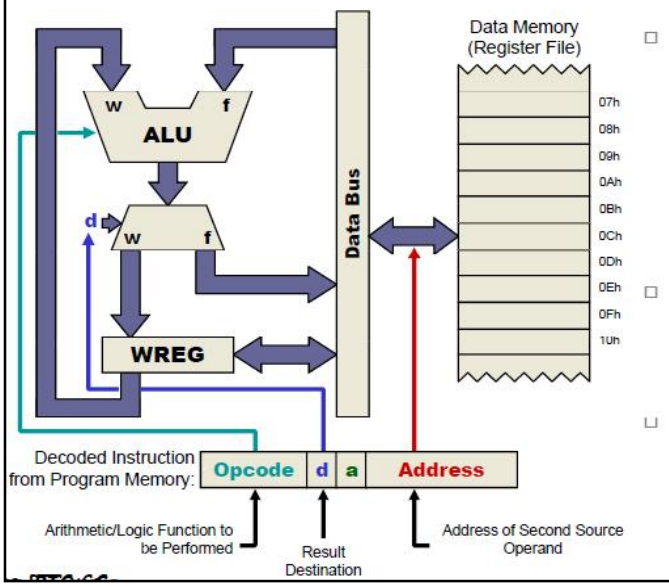
The PIC File Register

- It is the data memory.
- Read/Write → Static RAM
- Used for data storage, scratch pad and registers for internal use and function
- 8-bit width



7

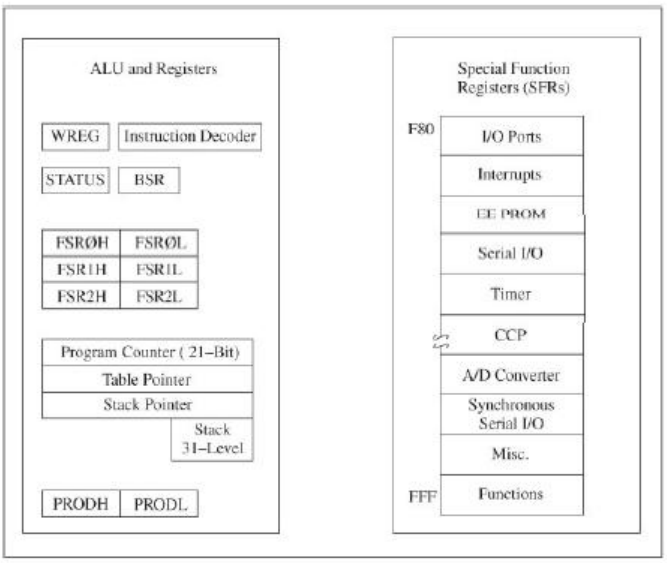
Register File Concept



- Register File Concept: All of data memory is part of the register file, so any location in data memory may be operated on directly
- All peripherals are mapped into data memory as a series of registers
- Orthogonal Instruction Set: ALL instructions can operate on ANY data memory location

8

PIC18F Programming Model



9

Special Function Registers

- Dedicated to specific functions such as ALU status, timers, serial communication, I/O ports, ADC,…
- The function of each SFR is fixed by the CPU designer at the time of design
  - it is used for control of the microcontroller or peripheral
- 8-bit registers
- Their numbers varies from one chip to another

10

General Purpose RAM

- Group of RAM locations
- 8-bit registers
- Larger than SFR
  - Difficult to manage them by using Assembly language
  - Easier to handle them by C Compiler.
- The microchip website provides the data RAM size, which is the same as GPR size.

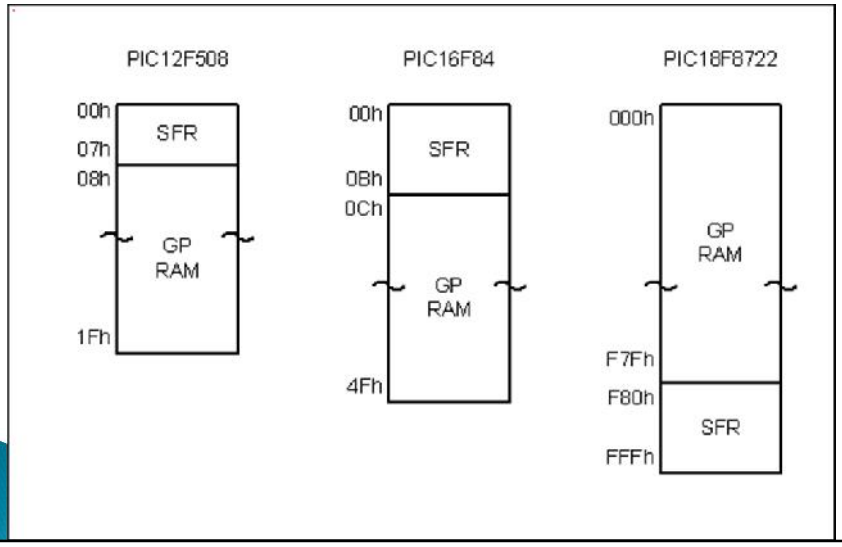
11

File Register Size

	File Register (Bytes)	=	SFR (Bytes)	+	GPR (Bytes)
PIC12F508	32		7		25
PIC16F84	80		12		68
PIC18F1220	512		256		256
PIC18F452	1792		256		1536
PIC18F2220	768		256		512
PIC18F458	1792		256		1536
PIC18F8722	4096		158		3938
PIC18F4550	2048		160		1888

12

File Registers of PIC12, PIC16, and PIC18



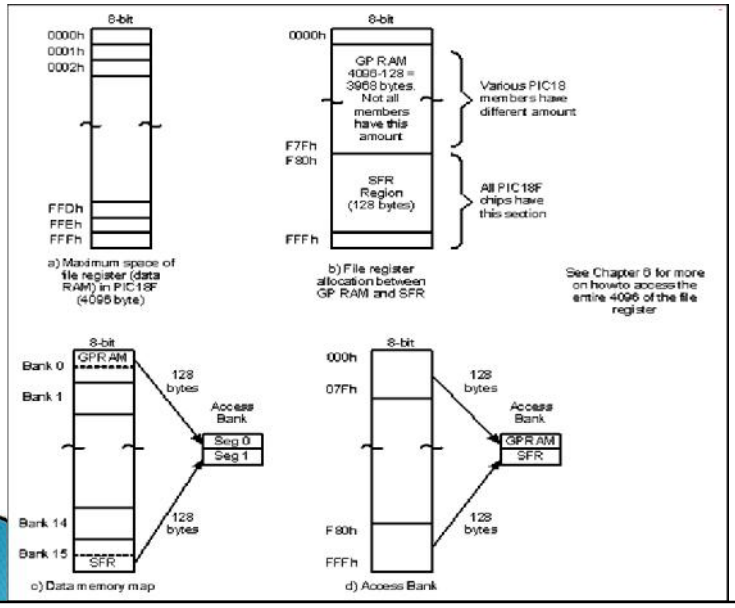
13

File Register and access bank in the PIC18

- The PIC18 Family can have a max. of 4096 Bytes.
- The File Register
  - has addresses of 000- FFFH
  - divided into 256-byte banks
  - Max. 16 banks (How?)
- At least there is one bank
  - Known as default access bank.
- Bank switching is a method used to access all the banks

14

File Register for PIC18 Family



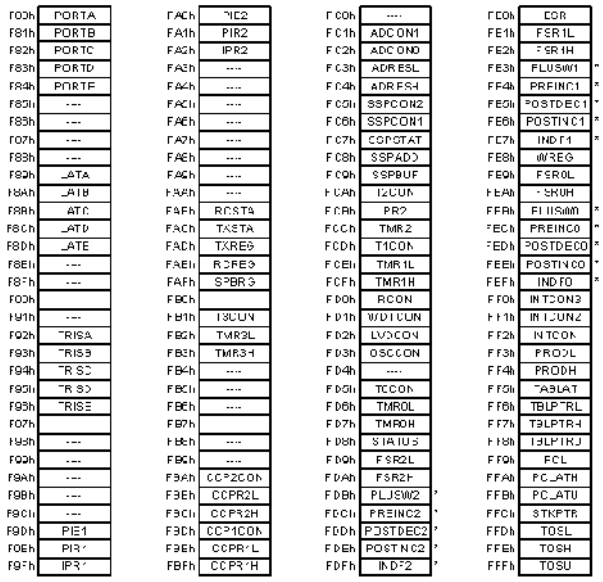
15

Access bank in the PIC18

- It is 256-Byte bank.
- Divided into equal two discontinuous sections (each 128 B).
- GP RAM, from 0 to 7FH
- SFR, from F80H to FFFH

16

SFR of the PIC18 Family



F80h	PORTA	FA0h	PIE2	FC0h	----	FE0h	EOR
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	TCCON	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.

17

Using instruction with the default access bank

- We need instruction to access other locations in the file register for ALU and other operations.
  - MOVWF
  - COMF
  - DECF
  - MOVF
  - MOVFF

18

MOVWF instruction

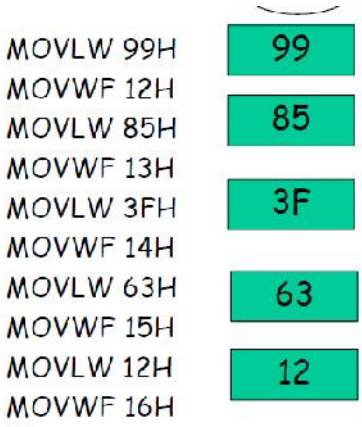
- F indicates for a file register

MOVWF Address

- It tells the CPU to copy the source register, WREG, to a destination in the file register.
  - A location in the SPR
  - A location in GP RAM

19

Example



Address	Data
012H	
013H	
014H	
015H	
016H	

Address	Data
012H	99
013H	85
014H	3F
015H	63
016H	12

We cannot move literal values directly into the general purpose RAM location in the PIC18. They must be moved there via WREG.

20

ADDWF

- Adds together the content of WREG and a file register location

  ADDWF File Reg. Address, D

- The result will be placed in either the WREG or in the file register location
  - D indicates the destination bit
- If D=0 or (D=w)
  - The result will be placed in the WREG
- If D=1 or (D=f)
  - The result will be placed in the file register

21

Example

State the content of file register location and WREG after the following program

```
MOVLW 0
MOVWF 12H
MOVLW 22H
ADDWF 12H, F
ADDWF 12H, F
ADDWF 12H, F
ADDWF 12H, F
```

0

22

22

COMF instruction

COMF File Reg. Address, D

- It tells the CPU to complement the content of fileReg and places the results in WREG or in fileReg.
- Write a simple program to toggle the SFR of Port B continuously forever

```
     MOVLW 55H
     MOVWF PORTB
B1   COMF PORTB,
     GOTO B1
```

23

DECF instruction

DECF File Reg. Address, D

- It tells the CPU to decrement the content of fileReg and places the results in WREG or in fileReg.
- Example:

```
MOVLW 3
MOVWF 20H
DECF 20H ,F
DECF 20H, F
DECF 20H, F
```

24

MOVF instruction

MOVF File Reg. Address, D

- It is intended to perform MOVFW
  MOVFW isn't existed
- If D=0
  Copies the content of fileReg (from I/O pin) to WREG
- If D=1
  The content of the fileReg is copied to itself

25

Example

- Write a simple program to get data from the SFRs of Port B and send it the SFRs of PORT C continuously

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

- Write a simple program to get data from the SFRs of Port B Add the value 5 to it and send it the SFRs of PORT C

```
        MOVF PORTB,W
        ADDLW 05H
        MOVWF PORTC
```

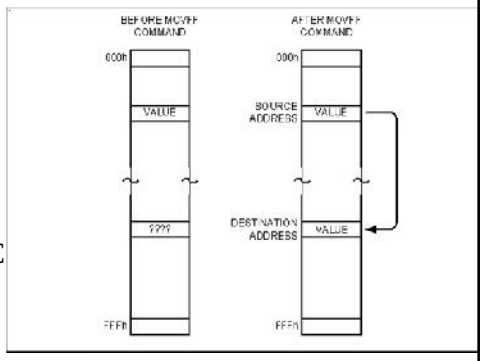
26

MOVFF instruction

- It copies data from one location in FileReg
- to another location in FileReg.

  MOVFF Source FileReg, destination FileReg

- Write a simple program to get data from the SFRs of Port B and send it the SFRs of PORT C continuously




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

27

PIC Status Register

D7							D0
X	X	X	N	OV	Z	DC	C

C – Carry flag

DC – Digital Carry flag

Z – Zero flag

OV – Overflow flag

N – Negative flag

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

28

Example

- Show the status of the C, DC, Z flags after the following addition instruction

```
MOVLW 38H
ADDLW 2FH
```

- Solution
  - 38H + 2FH = 67H
  - WREG=67H
- C=0, DC=1, Z=0

29

Example

- Show the status of the C, DC, Z flags after the following addition instruction

```
MOVLW 9CH
ADDLW 64H
```

- Solution
  - 9CH + 64H = 100H
  - WREG= 00H
- C=1, DC=1, Z=1

30

PIC Data Format

- There is one data type
  - 8 bits
  - It is the job of the programmer to break down data larger 8 bits
- Data type can be positive or negative
- Data format are
  - Hex (default in PIC) 12 or 0x12 or H'12' or 12H
  - Binary B'00010010'
  - Decimal .12 or D'12'
  - ASCII A'c' or a'c‘

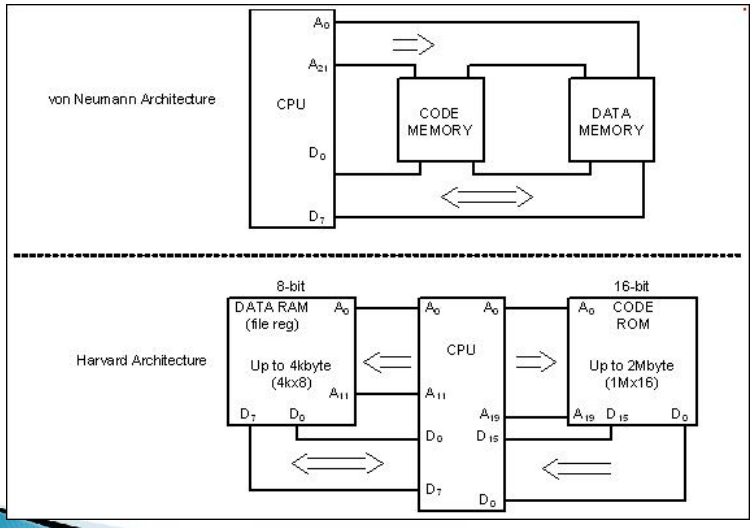
31

Harvard Architecture in the PIC

- Von Neumann Architecture: uses the same bus for accessing both the code and data memory.
- Slow down the processing speed
- Get in each other’s way
- Harvard Architecture: uses separate buses for accessing the code and data memory.
- Inexpensive for a chip

32

Vonneuman VS Harvard



33

RISC Architecture in the PIC

- To increase the processing power of the CPU

1. Increase the clock frequency of the chip

2. Use Harvard architecture

3. Change the internal architecture of the CPU and use what is called RISC architecture

34

RISC Architecture in the PIC

- **RISC**
- Simple and Small instruction set
- Regular and fixed instruction format
- Simple address modes
- Pipelined instruction execution --> 95% executed in one cycle

- **CISC**
- Complex and large instruction set
- Irregular instruction format
- Complex address modes
- May also pipeline instruction execution

35

RISC Architecture in the PIC

- **RISC**
- Provide large number of CPU registers
- Separated data and program memory
- Most operations are register to register
- Take shorter time to design and debug

- **CISC**
- Provide smaller number of CPU registers
- Combined data and program memory
- Most operations can be register to memory
- Take longer time to design and debug

36