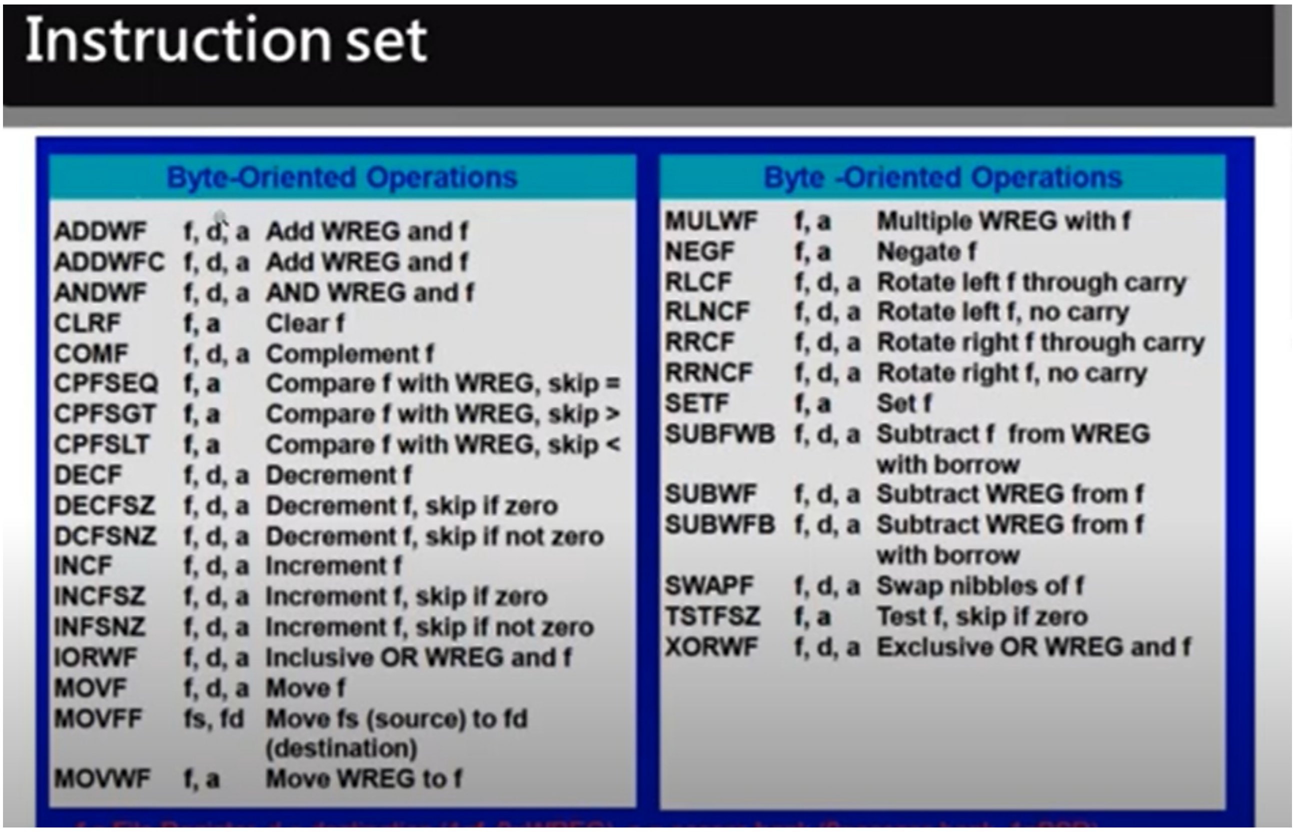
Lab1: Introduction to Instruction Set

目錄

- PIC18、MPLAB、實驗課規劃
- Instruction set
- WREG
- 補充

Lab1 主要在介紹如何開始在MPLAB寫組合語言並且執行

Instruction set



Instruction set

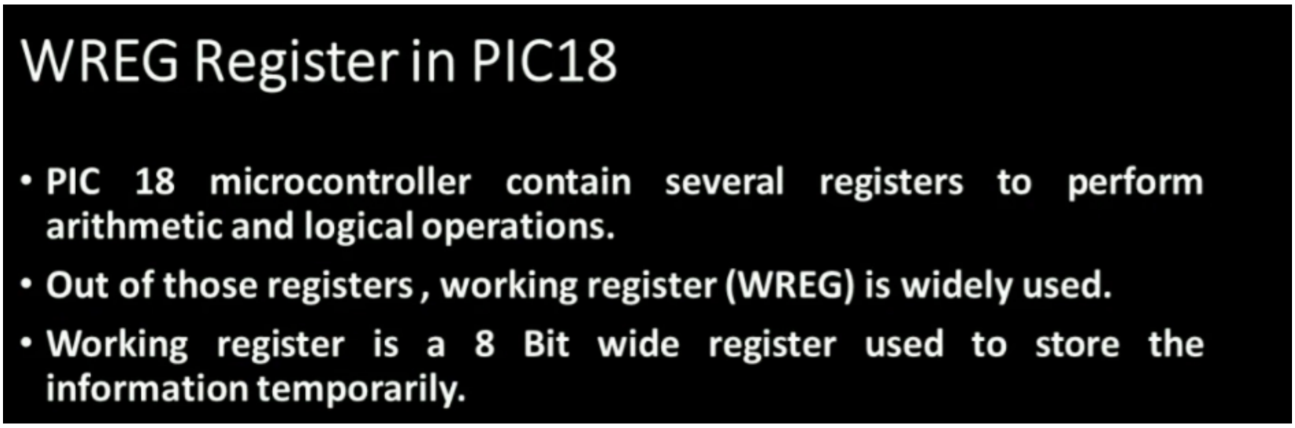
Byte-Oriented Operations		
ADDWF	f, d, a	Add WREG and f
ADDWFC	f, d, a	Add WREG and f
ANDWF	f, d, a	AND WREG and f
CLRF	f, a	Clear f
COMF	f, d, a	Complement f
CPFSEQ	f, a	Compare f with WREG, skip =
CPFSGT	f, a	Compare f with WREG, skip >
CPFSLT	f, a	Compare f with WREG, skip <
DECF	f, d, a	Decrement f
DECFSZ	f, d, a	Decrement f, skip if zero
DCFSNZ	f, d, a	Decrement f, skip if not zero
INCF	f, d, a	Increment f
INCFSZ	f, d, a	Increment f, skip if zero
INFSNZ	f, d, a	Increment f, skip if not zero
IORWF	f, d, a	Inclusive OR WREG and f
MOVF	f, d, a	Move f
MOVFF	fs, fd	Move fs (source) to fd (destination)
MOVWF	f, a	Move WREG to f

Byte -Oriented Operations		
MULWF	f, a	Multiple WREG with f
NEGF	f, a	Negate f
RLCF	f, d, a	Rotate left f through carry
RLNCF	f, d, a	Rotate left f, no carry
RRCF	f, d, a	Rotate right f through carry
RRNCF	f, d, a	Rotate right f, no carry
SETF	f, a	Set f
SUBFWB	f, d, a	Subtract f from WREG with borrow
SUBWF	f, d, a	Subtract WREG from f
SUBWFB	f, d, a	Subtract WREG from f with borrow
SWAPF	f, d, a	Swap nibbles of f
TSTFSZ	f, a	Test f, skip if zero
XORWF	f, d, a	Exclusive OR WREG and f

↓連結為PIC18指令集介紹，要懂得如何使用這些指令。
http://technology.niagarac.on.ca/staff/mboldin/18F_Instruction_Set/
(http://technology.niagarac.on.ca/staff/mboldin/18F_Instruction_Set/)

WREG

What is WREG? (https://www.microchip.com/forums/m150959.aspx)



WREG Register in PIC18

- PIC 18 microcontroller contain several registers to perform arithmetic and logical operations.
- Out of those registers , working register (WREG) is widely used.
- Working register is a 8 Bit wide register used to store the information temporarily.

- The W register is a special register in the PIC architecture
- It used as one of the 2 operands for ALU operations
- It can be the destination for any ALU operation.

WREG=working register

- PIC18中,可以經常用來當作運算元的register
- 運算時常用來暫時存放data

補充

（指令集中有些參數是關於access bank、BSR等往後實驗會再詳細介紹,有興趣的同學可以先參考以下的資料）

描述PIC18的memory架構

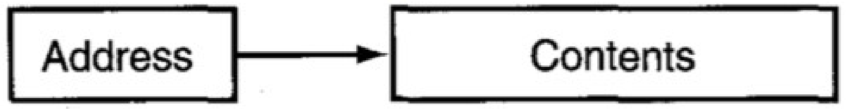
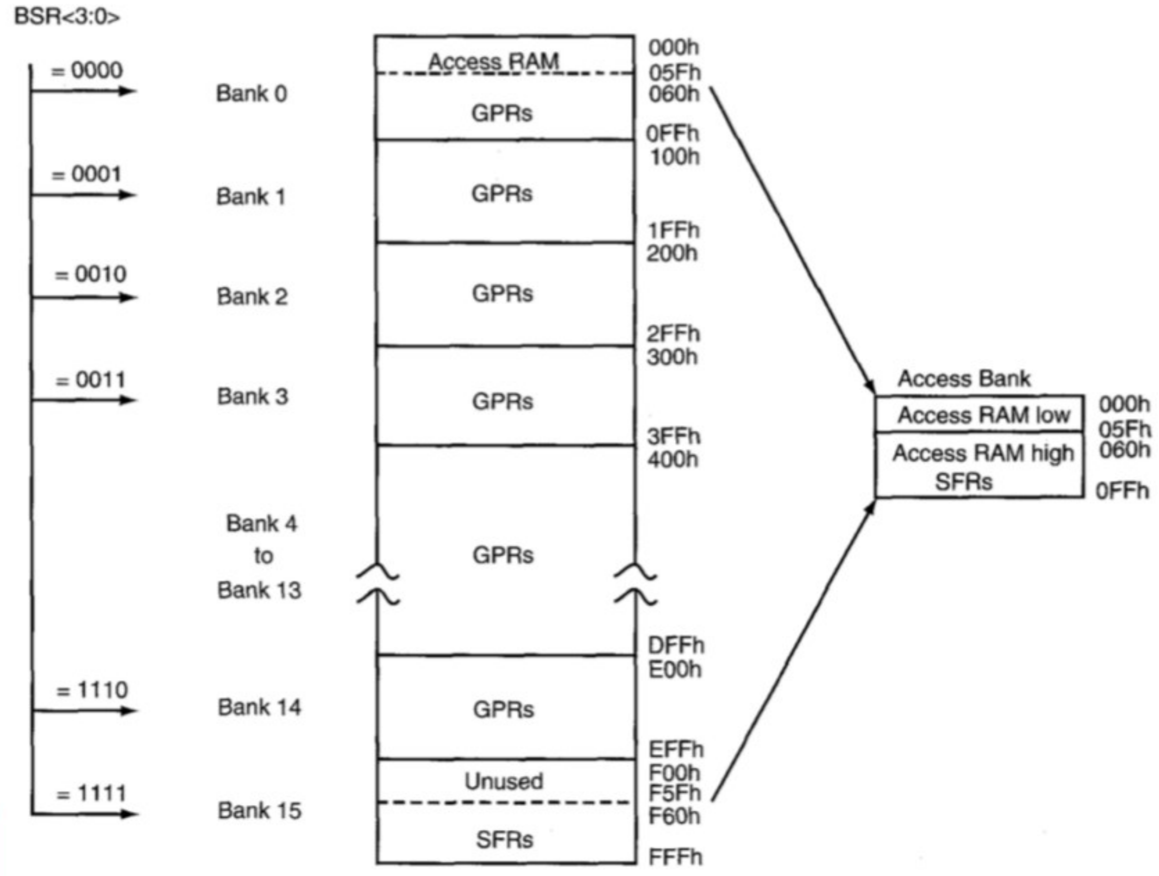
- The PIC18 Memory Organization
  - A memory location is referred to as an information unit.
  - A memory location in the PIC18 holds **eight bits** of information.
  - An information unit has two components: its address and its contents
  - 

    Figure 1.2 ■ The components of a memory location
- Separation of **Data Memory** and **Program Memory**
  - The PIC18 MCU assigns **data** and **program** to different memory spaces

- PIC18 **Data Memory**
  - Each location in the data memory is also referred to as a **register** or **file register**
  - Supports 4096 bytes(8 bits) of data memory. It requires 12 bits of address to select one of the data registers. (要用12bits才能分辨現在用的是哪個register)
  - Because the limited length of the PIC instruction,**only eight bits** of the PIC18 instruction are used **to specify the file register**.
  - As a result, the PIC designers divided the 4096 file registers into 16 banks. Only one bank of 256 file registers is active at any time.
  - An additional four bits are placed in a special register called bank select register (**BSR**) to **select the bank to be active.**
  - 如果沒有指定BSR,通常就是預設access bank的register
  - 

    Note. 1. **BSR** is the 4-bit bank select register.

    Figure 1.4 ■ Data memory map for PIC18 devices (redraw with permission of Microchip)

- Registers可以分成兩個種類:
  - General-purpose registers (**GPR**s)
    hold dynamic data when the CPU is executing a prog.
    (運算的時候可以用來存放值、讀值...等等)

- Specialfunction registers (**SFR**s)
  control the desired operation of the MCU
  (就是可以有一些特殊用途,往後lab會慢慢去用到這些比較特別的register)

程式初始化code

```
List p=18f4520
    #include<p18f4520.inc>
        CONFIG OSC = INTIO67
        CONFIG WDT = OFF
        org 0x00
```