實驗四 算術及邏輯運算指令

1. 學習重點

- 熟悉組合語言中算術運算指令，如：ADD、SUBB、MUL、DIV 等。
- 熟悉組合語言中邏輯運算指令，如：ANL、ORL、NOT 等。
- 了解 8051 與運算相關的特殊功能暫存器，如：PSW、ACC、B 等。

2. 元件原理

在 8051 的位址 0080H 到 00FFH 之間，有 21 個特殊功能暫存器（Special Function Register，簡稱 SFR），如 P0、P1、SBUF、IE 等，其中與算術及邏輯運算相關的為 PSW、ACC、B。

PSW（Program Status Word，程式狀態字組）

為 8 位元的暫存器，其位址在 00D0H，功能為顯示 8051 在執行過程中的狀態：

	7	6	5	4	3	2	1	0
PSW	CY	AC	F0	RS1	RS0	OV	–	P

圖 4-1、PSW 每個位元所代表的狀態

PSW.7

CY（Carry Flag，進位旗標）。在執行加法 / 減法運算時，若第 7 位元有進位 / 借位時，則 CY = 1；反之，沒有進位/借位，則 CY = 0。

PSW.6

AC（Auxiliary Carry Flag，輔助進位旗標）。在執行加法 / 減法運算時，若第 3 位元有進位 / 借位時，則 AC = 1；反之，AC = 0。

PSW.5

F0（Flag 0）。為使用者自訂旗標，由使用者自行設定的位元。

PSW.3 與 PSW.4

RS0（Register Bank Selector 0，暫存器庫選擇位元 0）與 RS1（Register Bank

Selector 1，暫存器庫選擇位元 1)。功能為選擇控制哪一組暫存器庫工作：

表 4-1、設定 RS0、RS1 所代表的暫存器庫

RS1	RS0	暫存器庫	位址
0	0	RB0	0000H ~ 0007H
0	1	RB1	0008H ~ 000FH
1	0	RB2	0010H ~ 0017H
1	1	RB3	0018H ~ 001FH

由圖 4-2 左邊黃框可知，PSW 的第 rs=0，因此系統原本的暫存器庫為 0，將數值 043H 存入 R1，會存入 RB0 的 R1，亦即位址 0x01H。

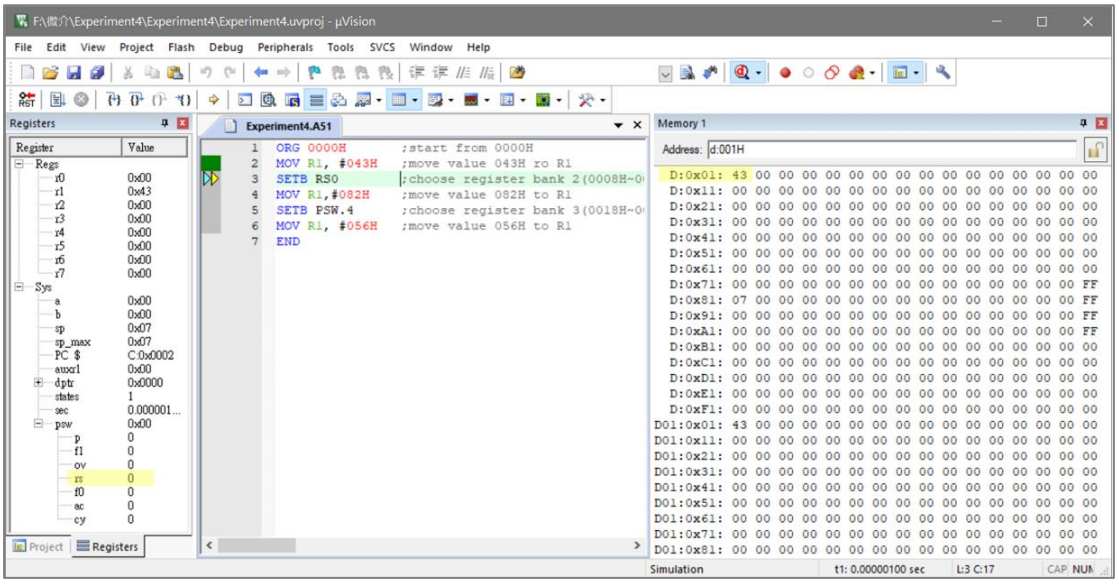

圖 4-2、系統的原始設定為 RB0

由圖 4-3 可知，將 RS0 設為 1，PSW 中的 rs = 1，也就是將工作的暫存器庫改為 RB1。由圖中的右邊黃框可驗證，將數值 082H 存入 R1，此數值將會存入 RB1 的 R1，亦即位址 0x09H。左邊 r1 = 0x82，是指目前暫存器庫的 R1，RB0 中 R1 (位址 0x01H) 的值仍然是 043H。

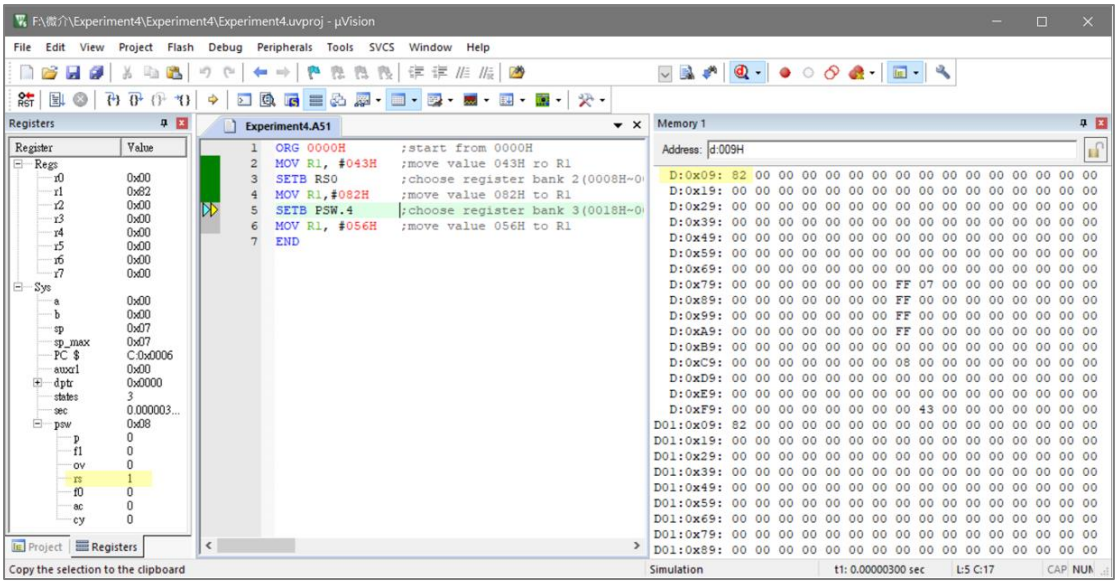

圖 4-3、選擇 RB1 作為工作的暫存器庫

由圖 4-4 可知，用另一種設置方式，將 PSW 的第 4 位元設為 1，此時 rs = 3，工作的暫存器庫改為 RB3。由右邊黃框可驗證，將數值 056H 存入 R1，此數值將會存入 RB3 的 R1，亦即位址 0x19H。

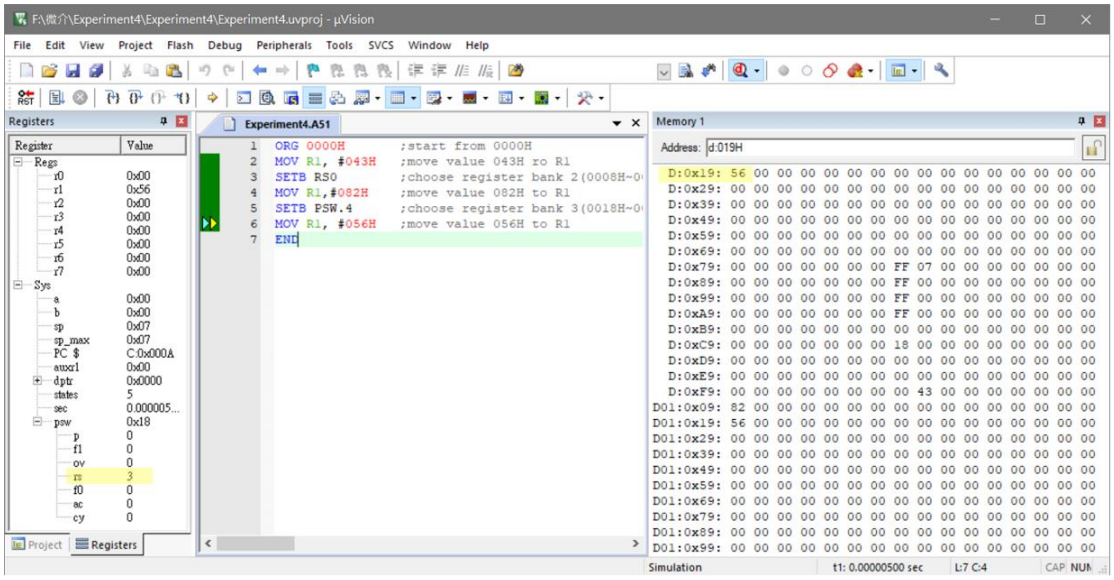

圖 4-4、選擇 RB3 作為工作的暫存器庫

PSW.2

OV（Overflow Flag，溢位旗標）。考慮有符號的運算，若運算結果大於 127 或小於-128，則 OV = 1；反之，則 OV = 0。

PSW.1

Reserved Flag，保留旗標。沒有提供服務。

PSW.0

P（Parity Flag，同位旗標）。8051 採偶同位，若 A 暫存器裡 1 的個數為奇數個，則 P = 1；反之，有偶數個，則 P = 0。

ACC（Accumulator）

又稱為 A 暫存器，為 8 位元暫存器，其位址在 00E0H。為 CPU 主要運作的位置，是使用頻率最高的暫存器。

B（Base Register）

為 8 位元暫存器，其位址在 00F0H。主要功能為乘法及除法運算時，和 A 暫存器搭配使用。若沒有執行乘法及除法運算，亦可作為一般暫存器使用。

在乘法運算中，將被乘數（無符號 8 位元）存入 A 暫存器，乘數（無符號 8 位元）存入 B 暫存器，執行乘法運算產生乘積（16 位元）。8051 會將乘積的高 8 位元存回 B 暫存器，低 8 位元存回 A 暫存器。

在除法運算中，將被除數（無符號 8 位元）存入 A 暫存器，除數（無符號 8 位元）存入 B 暫存器，執行除法運算。8051 會將餘數存回 B 暫存器，商數存回 A 暫存器。

算術運算指令

- ADD
  將 ACC 的值與 1 位元組的值相加，並且將結果存回 ACC。
- ADDC
  將 ACC 的值與 1 位元組的值，以及進位位元(CY)的值相加，並將結果存回 ACC。
- SUBB
  將 ACC 的值減去 1 位元組的值和進位位元(CY)的值，並將結果存回 ACC。

算術運算指令— 減法

- SUBB A, #0A2H

執行前 A = 033H、CY = 1 → 運算過程 A = A - CY - #0A2H → 執行後 A = 090H

CY	AC	OV	P
1	0	0	0

CY	AC	OV	P
1	0	1	0

借位 CY = 1

```
  0 0 1 1 | 0 0 1 1  → 033H
- 1 0 1 0 | 0 0 1 1  → 0A3H
---------------------------
  1 0 0 1 | 0 0 0 0  → 090H
```

註：這只是驗證旗標的狀態，實際上只需知道各個旗標的意思就可以了。

算術運算指令— 乘法

- MUL AB

```
          A   → 被乘數 (無符號8位元)
X         B   → 乘數 (無符號8位元)
-----------------------------------------------
   | B | A |  高8位元 → 存回B、低8位元 → 存回ACC
```

➢ 乘積大於0FFH時，產生溢位，OV = 1。

➢ 根據ACC的值，同位旗標 P 會有變化。

執行前：A = 030H、B = 012H ; A = 48D、B = 18D

執行後：A = 060H、B = 003H ; A = 96D、B = 03D (3×256=768)

A+B=96+768=864

				0	0	1	1	0	0	0	0
			x	0	0	0	1	0	0	1	0
			0	0	1	1	0	0	0	0	
+	0	0	1	1	0	0	0	0			
	0	0	1	1	0	1	1	0	0	0	0	0
		3				6				0	
		B				A					

OV	P
1	0

算術運算指令— 除法

- DIV AB

```
除數 (無符號8位元) ← B | A      → 被除數 (無符號8位元)
                 商數 ←   A ...B → 餘數
```

➢ 除數為0時，產生溢位，OV = 1。

➢ 根據ACC的值，同位旗標 P 會有變化。

執行前：A = 030H、B = 012H ; A = 48D、B = 18D

執行後：A = 002H、B = 00CH ; A = 2D、B = 12D

```
    2   → A
18 | 48
   36
   --
   12   → B
```

OV	P
0	1

邏輯運算指令

- ANL

將運算元 1 與運算元 2 執行 AND 邏輯運算，並將結果存回運算元 1。

- ORL

將運算元 1 與運算元 2 執行 OR 邏輯運算，並將結果存回運算元 1。

- CPL

將運算元執行 NOT 邏輯運算，並將結果存回運算元中。

3. 實驗內容

算術運算實驗

計算十六進制的（32H − 06H）× 07H 與（64H + 0DH）÷ 06H，觀察 Debug Mode 中的特殊功能暫存器，分別找出 A、B 暫存器中的值。

邏輯運算實驗

先使用 AND 邏輯運算將 #10111101B 中的第 4 位元變為 0，存入暫存器 B，再使用 OR 邏輯運算將 #01000010B 中的第 4 位元變為 1，最後透過 NOT 邏輯運算，使第二個值與第一個值相同。

4. 軟體流程圖

算術運算實驗

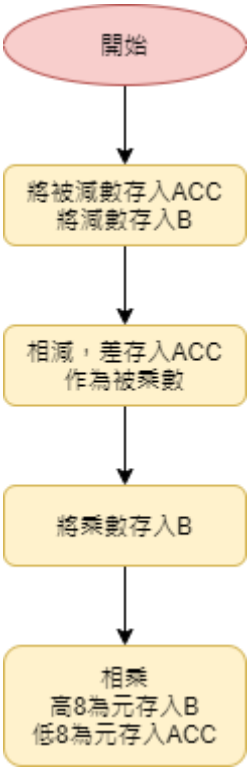

圖 4-5、(32H-06H) ×07H
運算流程圖

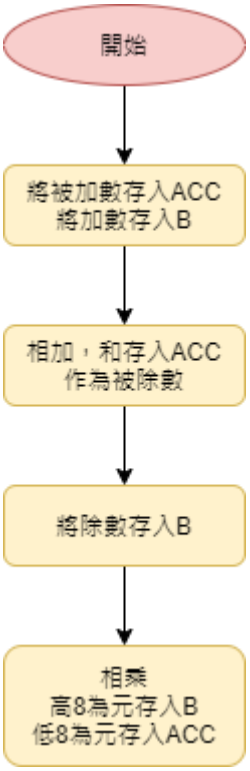

圖 4-6、(64H + 0DH) ÷ 06H
運算流程圖

邏輯運算實驗

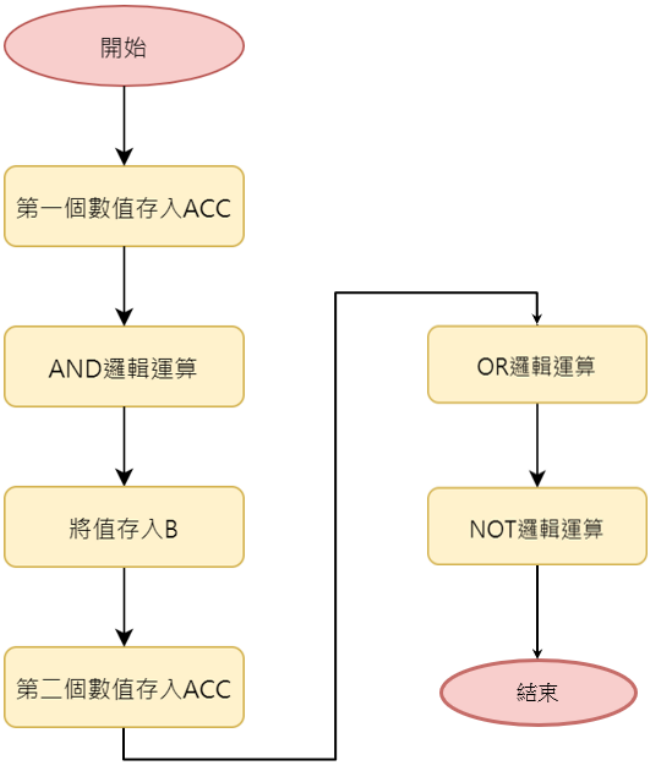

圖 4-7、邏輯運算實驗流程圖

5. 範例程式碼

算術運算實驗

（32H-06H）×07H

```
1  ORG 0000H          ; start from 0000H
2  MOV A, #032H       ; move 032H into ACC
3  MOV B, #006H       ; move 006H into B
4  SUBB A, B          ; A-B, then save into ACC
5  MOV B, #007H       ; move multiplier 007H into B
6  MUL AB             ; AxB
7  END
```

（64H+DH）÷6H

```
1  ORG 0              ; start from 0000H
2  MOV A, #064H       ; move 064H into ACC
3  MOV B, #00DH       ; move 00DH into B
4  ADD A, B           ; A+B, then save into ACC
5  MOV B, #006H       ; move divisor 006H into B
6  DIV AB             ; A/B
7  END
```

邏輯運算實驗

```
1  ORG 0              ; start from 0000H
2  MOV A, #10111101B  ; move 10111101B into ACC
3  ANL A, #11101111B  ; use AND Gate to turn Bit 4 into 0
4  MOV B, A           ; move the value into B
5  MOV A, #01000010B  ; move 01000010B into ACC
6  ORL A, #00010000B  ; use OR Gate to turn Bit 4 into 1
7  CPL A              ; use NOT Gate
8  END
```

6. 模擬結果

算術運算實驗

計算 （32H-06H）×07H

ACC 中存入的值為 00110010B，1 的個數為奇數個，故圖 4-8 中 PSW 中 p = 1。

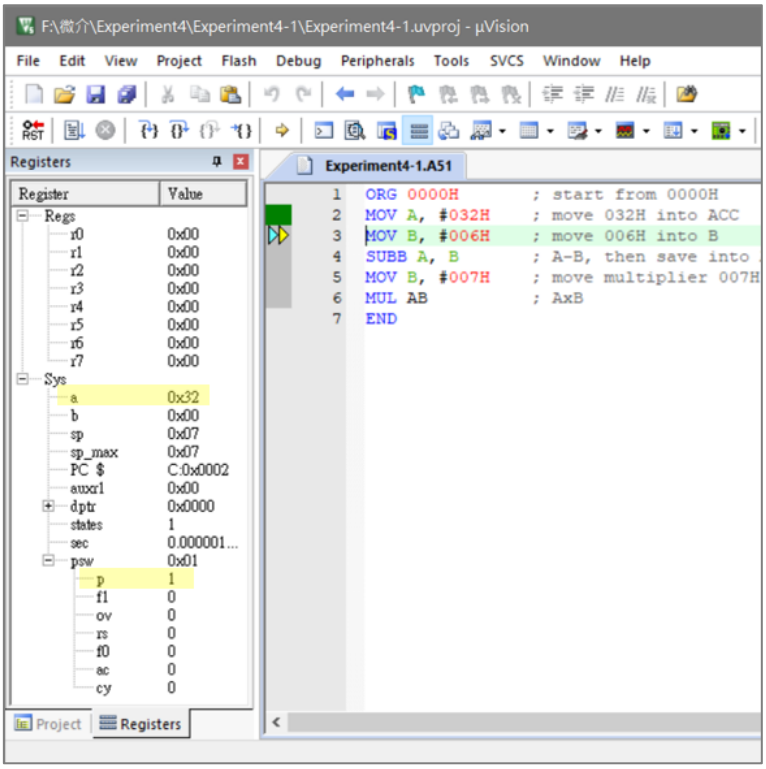

圖 4-8、032H 存入 ACC 後 PSW 的第 0 位元變化

ACC 的值和 B 的值執行減法運算後，由圖 4-9 可知運算結果存入 ACC。減法運算中第 3 位元有向第 4 位元借位，故 psw 中 ac = 1。

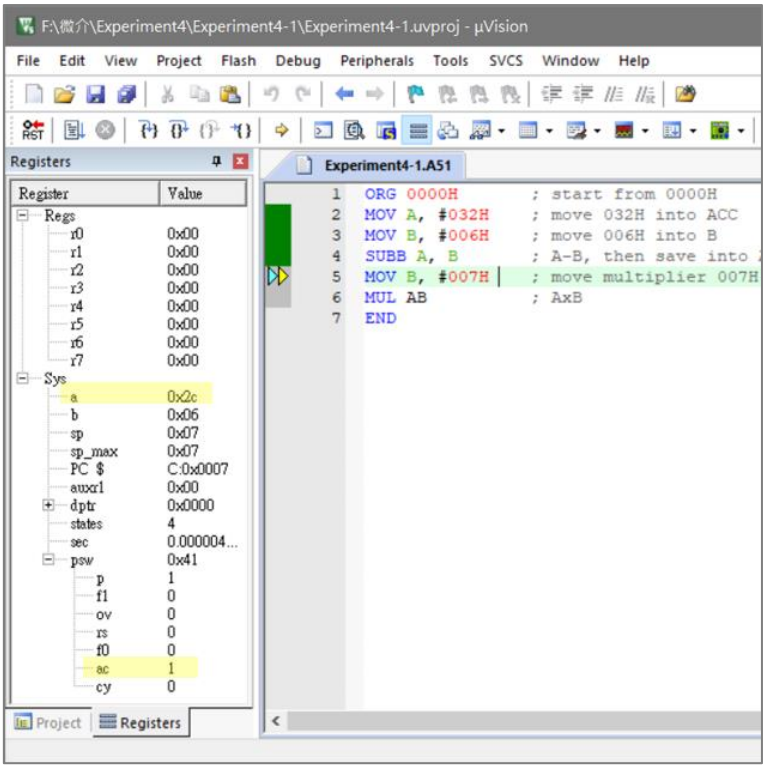

圖 4-9、執行減法運算後 PSW 的變化

乘法運算中，被乘數存入 ACC，乘數存入 B。前段減法運算的結果為此次實驗的被乘數，故存入 ACC 中。乘數 007H 則存入 B，如圖 4-10。

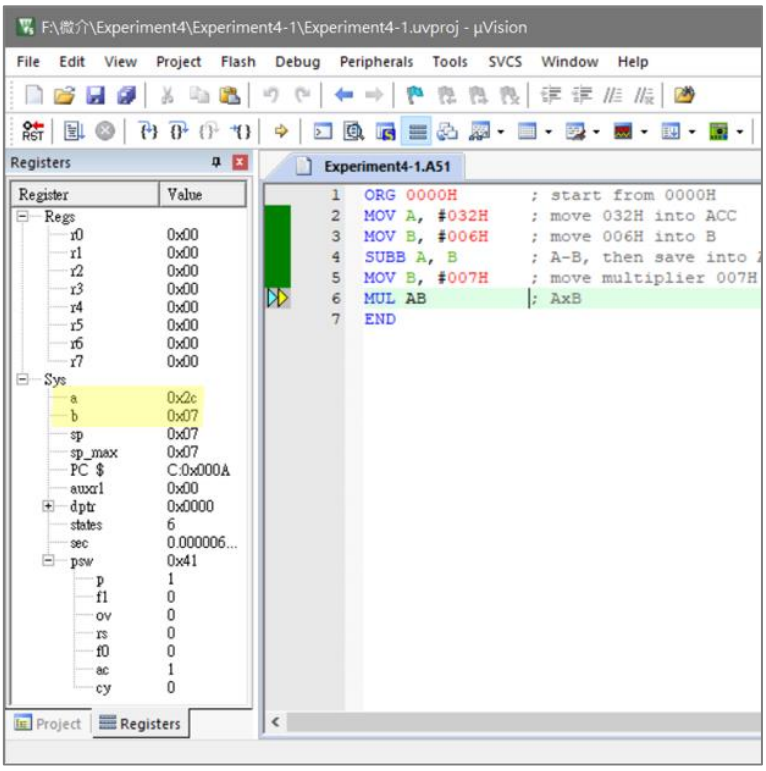

圖 4-10、被乘數存入 ACC，乘數存入 B

執行乘法運算後，乘積的高 8 位元存入 B，低 8 位元存入 ACC。故可知算式 $(32H-6H)\times 7H$ 的答案為 0134H，如圖 4-11。將其換算十進制為 308。因為乘積大於 0FFH，有發生溢位，所以 psw 中 ov = 1。

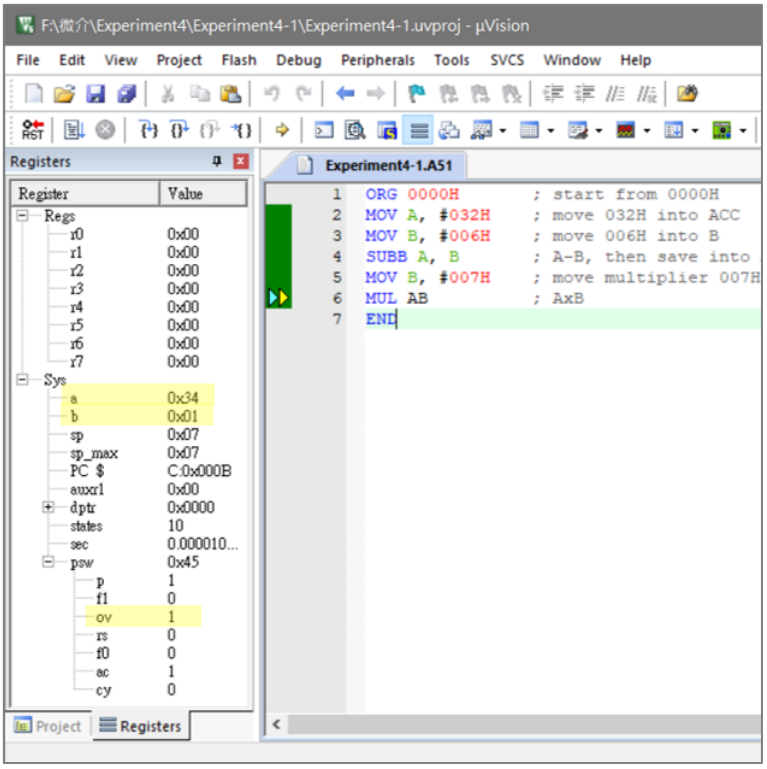

圖 4-11、乘法運算後的結果

計算（64H+DH）÷6H

ACC 中存入的值為 01100100B，1 的個數為奇數個，故下圖 4-12 中 psw 中 p = 1。

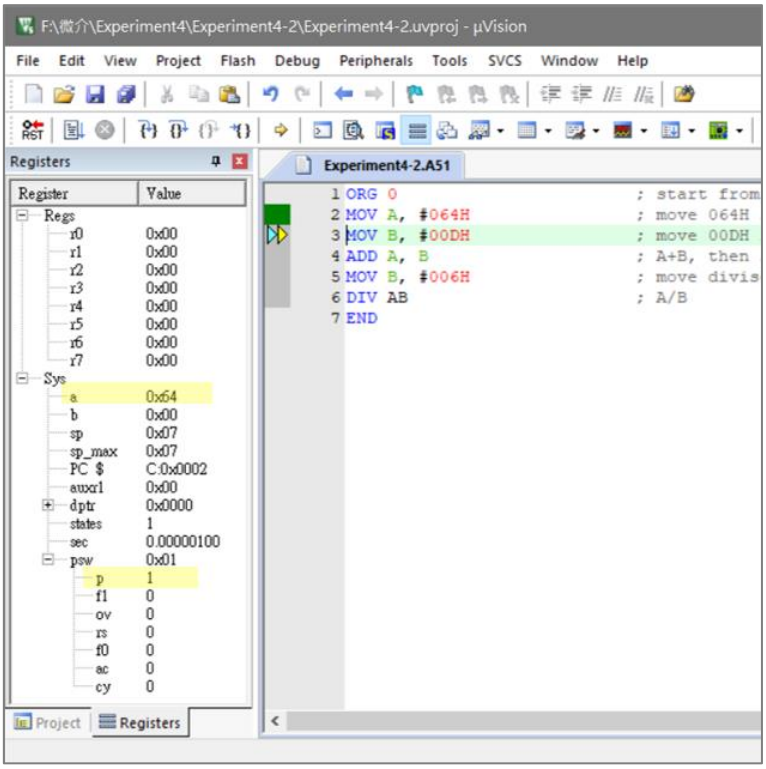

圖 3、064H 存入 ACC 後 PSW 的第 0 位元變化

ACC 的值和 B 的值執行加法運算後，由圖 4-13 可知運算結果存入 ACC。此時 ACC 中的值為 01110001B，1 的個數為偶數個，故 psw 中 p = 0，且加法運算中第 3 位元有向第 4 位元進位，故 psw 中 ac = 1。

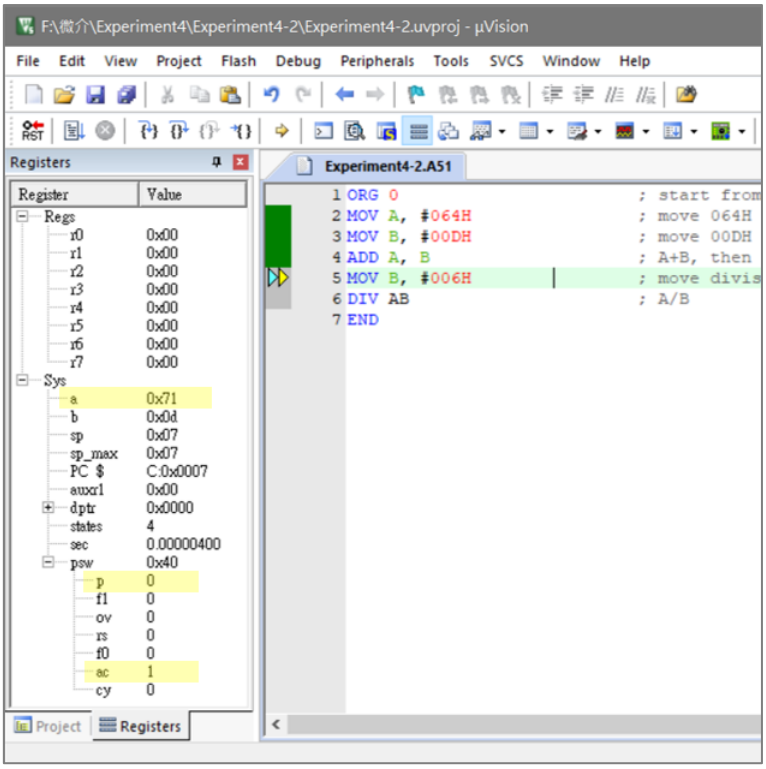

圖 4、加法運算後 PSW 的變化

除法運算中，被除數存入 ACC，除數存入 B。前段加法運算的結果為此次實驗的被除數，故存入 ACC 中。除數 006H 則存入 B，如圖 4-14。

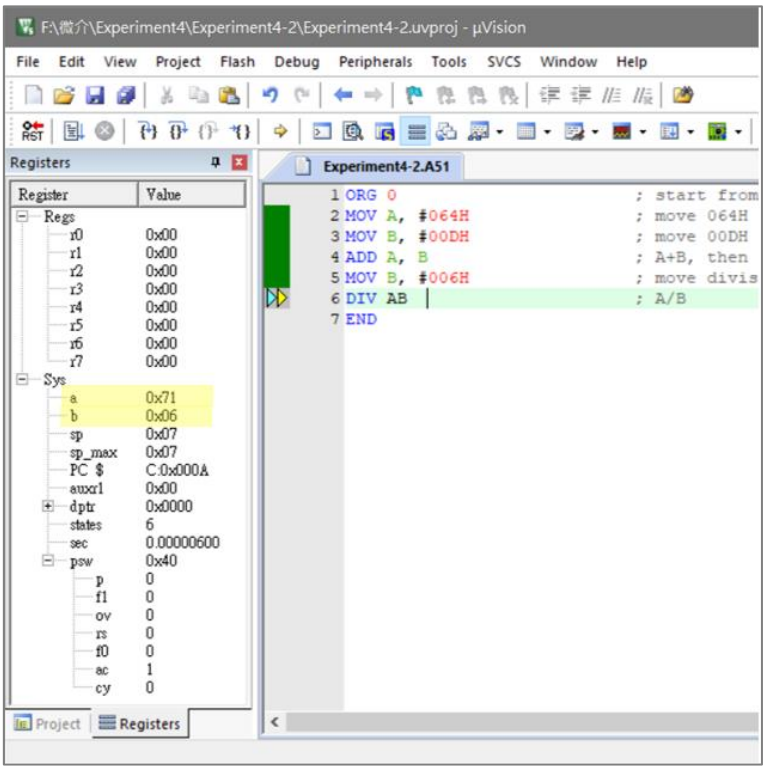

圖 4-14、被除數存入 ACC，除數存入 B

執行除法運算後，商數存入 ACC，餘數存入 B。故可知算式 $(64H + DH) \div 6H$ 的商為 012H，餘數為 005H，如圖 4-15。將其換算十進制為商 18、餘數 5。

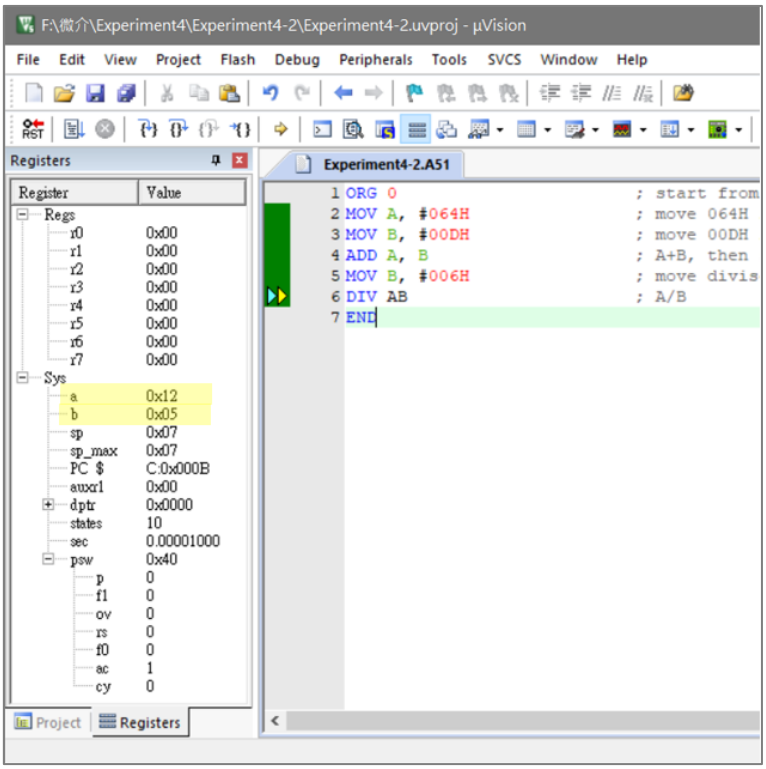

圖 5-15、除法運算後的結果

邏輯運算實驗

將 10111101B 與 11101111B 執行 AND 邏輯運算，過程如下：

$$\begin{array}{rr} & 10111101B \\ AND & 11101111B \\ \hline & 10101101B \end{array}$$

運算完後第 5 位元變為 0，值變為 10101101B，換算成十六進制為 0ADH，存入 ACC 中，如圖 4-16。

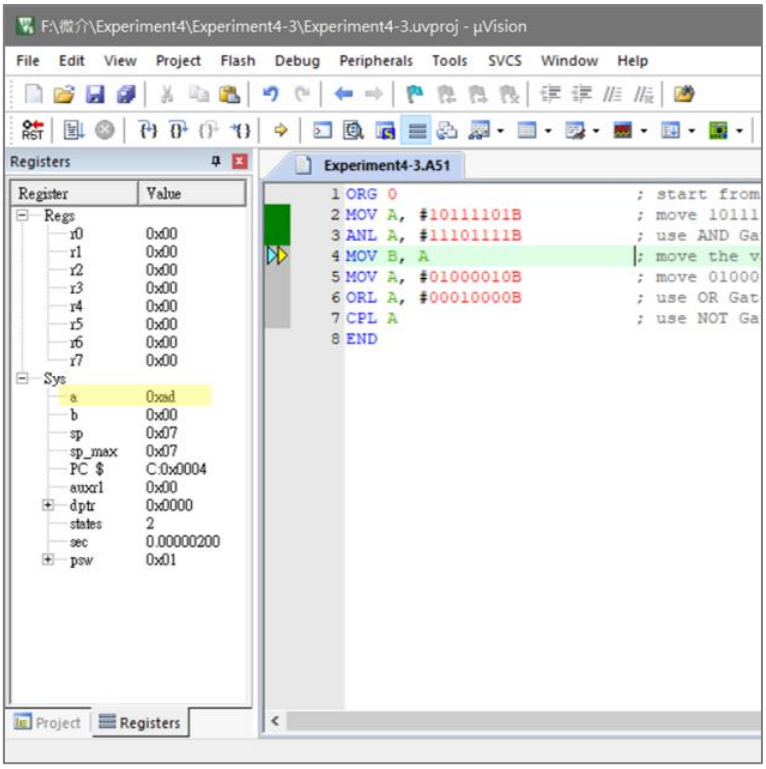

圖 4-16、AND 邏輯運算後的結果

將 01000010B 與 00010000B 執行 OR 邏輯運算，過程如下：

$$\begin{array}{rr} & 01000010B \\ OR & 00010000B \\ \hline & 01010010B \end{array}$$

運算完後第 5 位元變為 1，值變為 01010010B，換算成十六進制為 052H，存入 ACC 中，如圖 4-17。

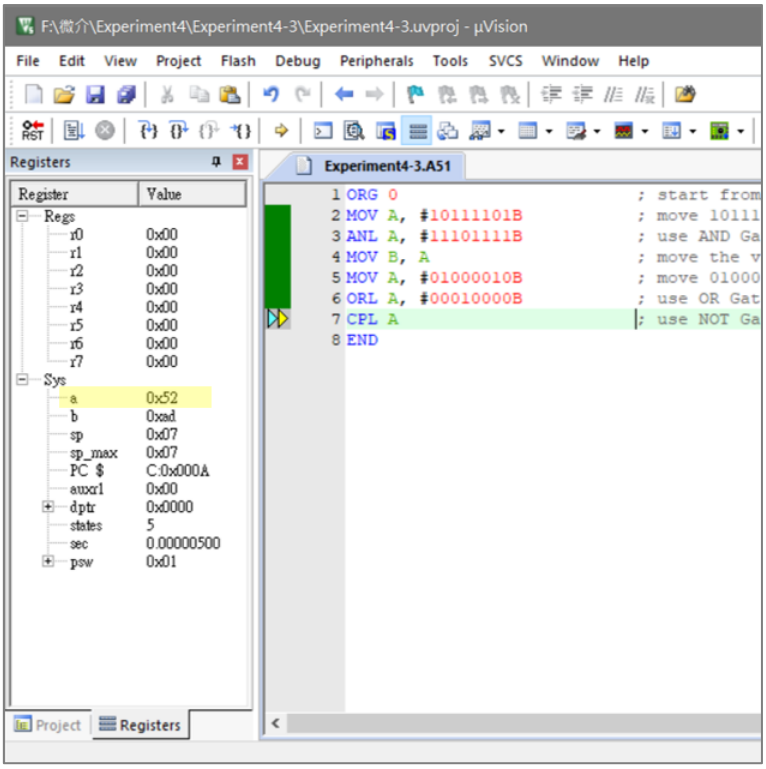

圖 4-17、OR 邏輯運算後的結果

將前段 OR 運算後的結果 01010010B 執行 NOT 邏輯運算，過程如下：

$$\begin{array}{ll} \text{NOT} & 01010010B \\ \hline & 10101101B \end{array}$$

ACC 中的值 0、1 對調，值變為 10101101B，換算成十六進制為 0ADH，存入 ACC 中，如圖 4-18。可看出暫存器 ACC 中的值與 B 中的值一樣即為完成實驗。

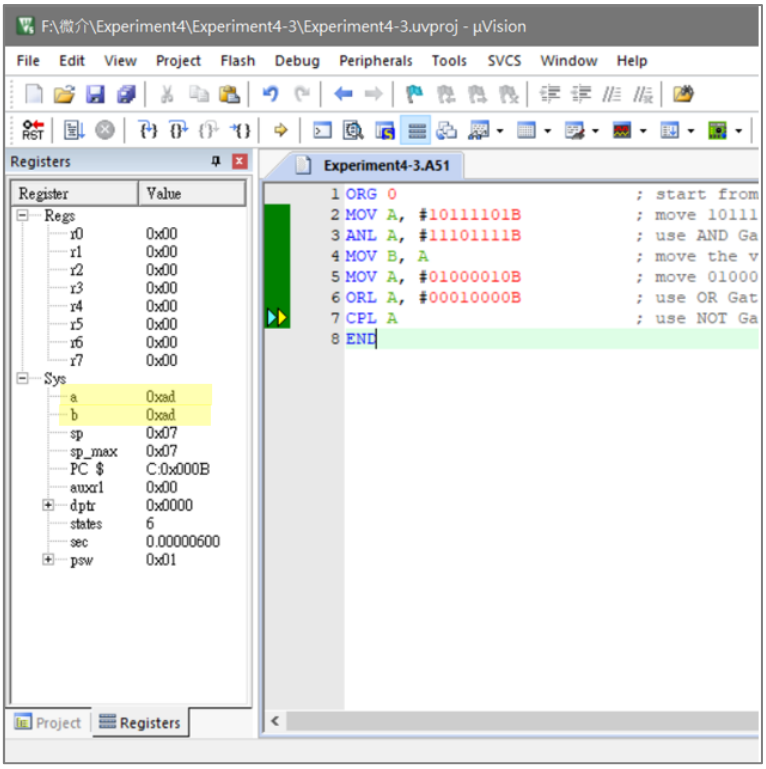

圖 4-18、NOT 邏輯運算後的結果

7. 整理的題目，選擇/是非題

（　）若第 6 位元與第 7 位元同時有進位 / 借位，表示產生溢位。

附錄

算術運算指令

ADD

此指令不考慮進位位元 CY。將 1 位元組的值與 A 暫存器的值相加，並且將結果存回 A 暫存器。

- 範例：*ADD A, #02BH*

  執行前：A = 055H

  運算過程：A = A + #02BH

  執行後：A = 080H

  PSW 旗標狀態：

CY	AC	OV	P
0	1	1	1

ADDC

此指令考慮進位位元 CY。將 1 位元組的值、進位位元的值、A 暫存器的值

相加，並將結果存回 A 暫存器。

- 範例：*ADDC A, 40H*

  執行前：A = 013H、CY = 1、(40H) = 005H

  PSW 旗標狀態：

CY	AC	OV	P
1	0	0	1

  運算過程：A = A + CY + (40H)

  執行後：A = 019H

  PSW 旗標狀態：

CY	AC	OV	P
0	0	0	1

SUBB

此指令考慮進位位元 CY。將 A 暫存器的值減去 1 位元組的值和進位位元的值，並將結果存回 A 暫存器。

- 範例：*SUBB A, @R0*

  執行前：A = 033H、CY = 1、R0= 023H、(23H) = 0A2H

  運算過程：A = A - CY - (R0)

  執行後：A = 090H

  PSW 旗標狀態：

CY	AC	OV	P
1	0	1	0

MUL

以 A 暫存器的值 (無符號 8 位元) 為被乘數，以 B 暫存器的值 (無符號 8 位元) 為乘數，執行乘法運算，產生乘積 (16 位元)，並將結果的高 8 位元存回 B 暫存器，將低 8 位元存回 A 暫存器。在旗標部分，當乘積大於 0FFH 時，產生溢位，OV = 1；同位旗標 P 則會根據 ACC 的值變化。

		A	→ 被乘數 (無符號 8 位元)
x		B	→ 乘數 (無符號 8 位元)
	B	A	高 8 位元 → 存回 B；低 8 位元 → 存回 ACC

- 範例：*MUL AB*

  執行前：A = 030H、B = 012H

  執行後：A = 060H、B = 003H

PSW 旗標狀態：

CY	OV	P
0	1	0

DIV

以 A 暫存器的值 （無符號 8 位元） 為被除數，以 B 暫存器的值 （無符號 8 位元） 為除數，執行除法運算，並將結果的餘數存回 B 暫存器，將商數存回 A 暫存器。在旗標部分，當除數為 0 時，產生溢位，OV = 1；同位旗標 P 則會根據 ACC 的值變化。

除數（無符號 8 位元）← B | A → 被除數（無符號 8 位元）

商數 ← A …B → 餘數

- 範例：*DIV AB*

  執行前：A = 030H、B = 012H

  執行後：A = 002H、B = 00CH

  PSW 旗標狀態：

CY	OV	P
0	0	1

邏輯運算指令

ANL

將運算元 1 與運算元 2 執行 AND 邏輯運算，並將結果存回運算元 1。

- 範例：*ANL A, R2*

  執行前：A = 011H、R2 = 10000011B

  運算過程：

```
  00010001B
AND 10000011B
----------
  00000001B
```

  執行後：A = 001H、R2 = 10000011B

  PSW 旗標狀態：

P
1

ORL

將運算元 1 與運算元 2 執行 OR 邏輯運算，並將結果存回運算元 1。

- 範例：*ORL A, 41H*

  執行前：A = 03BH、(41H) = 067H

  運算過程：

	00111011B
OR	01100111B
	01111111B

  執行後：A = 07FH、(41H) = 067H

  PSW 旗標狀態：

P
1

CPL

將運算元執行 NOT 邏輯運算，並將結果存回運算元中。

- 範例：*CPL A*

  執行前：A = 0C2H

  運算過程：

NOT	11000010B
	00111101B

  執行後：A = 03DH

  PSW 旗標狀態：

P
1