



# 微算機實驗報告

Lab #04

姓名：洪巧芸

系級：資工 114

學號：110550143

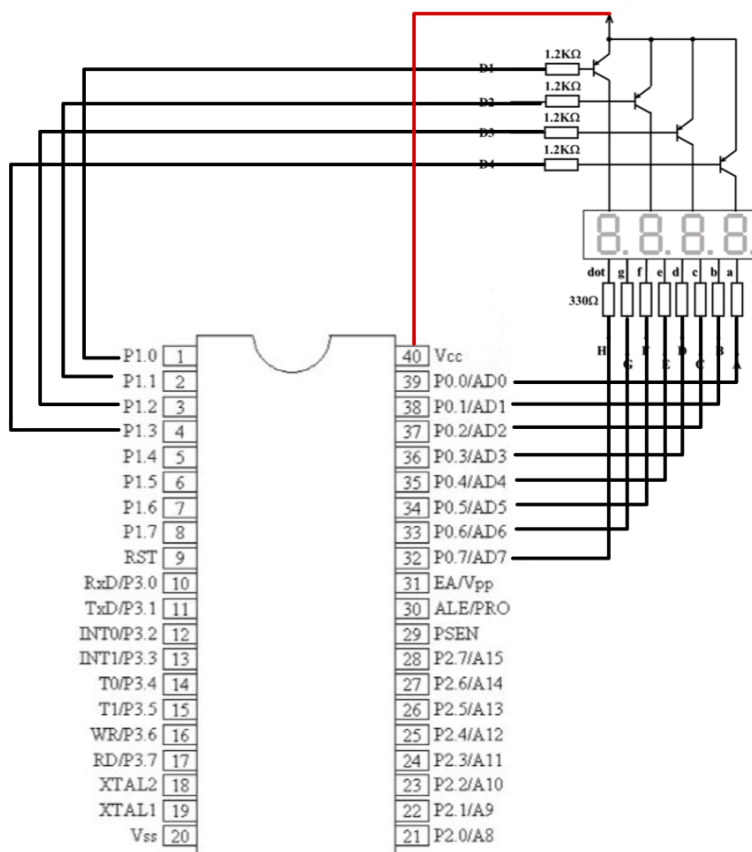
上課時間：2023/10/17

## 一、實驗目的：

了解 4 個七段顯示器的電路結構以及相關的控制方法。利用 D1~D4 接口控制七段顯示器的燈亮位置，A~H 接口控制燈亮位置要顯示的數值，並搭配程式設定延遲是人體產生視覺暫留，讓七段顯示器雖然每次只亮 1 個數字但肉眼卻可以同時看見 4 個位數。

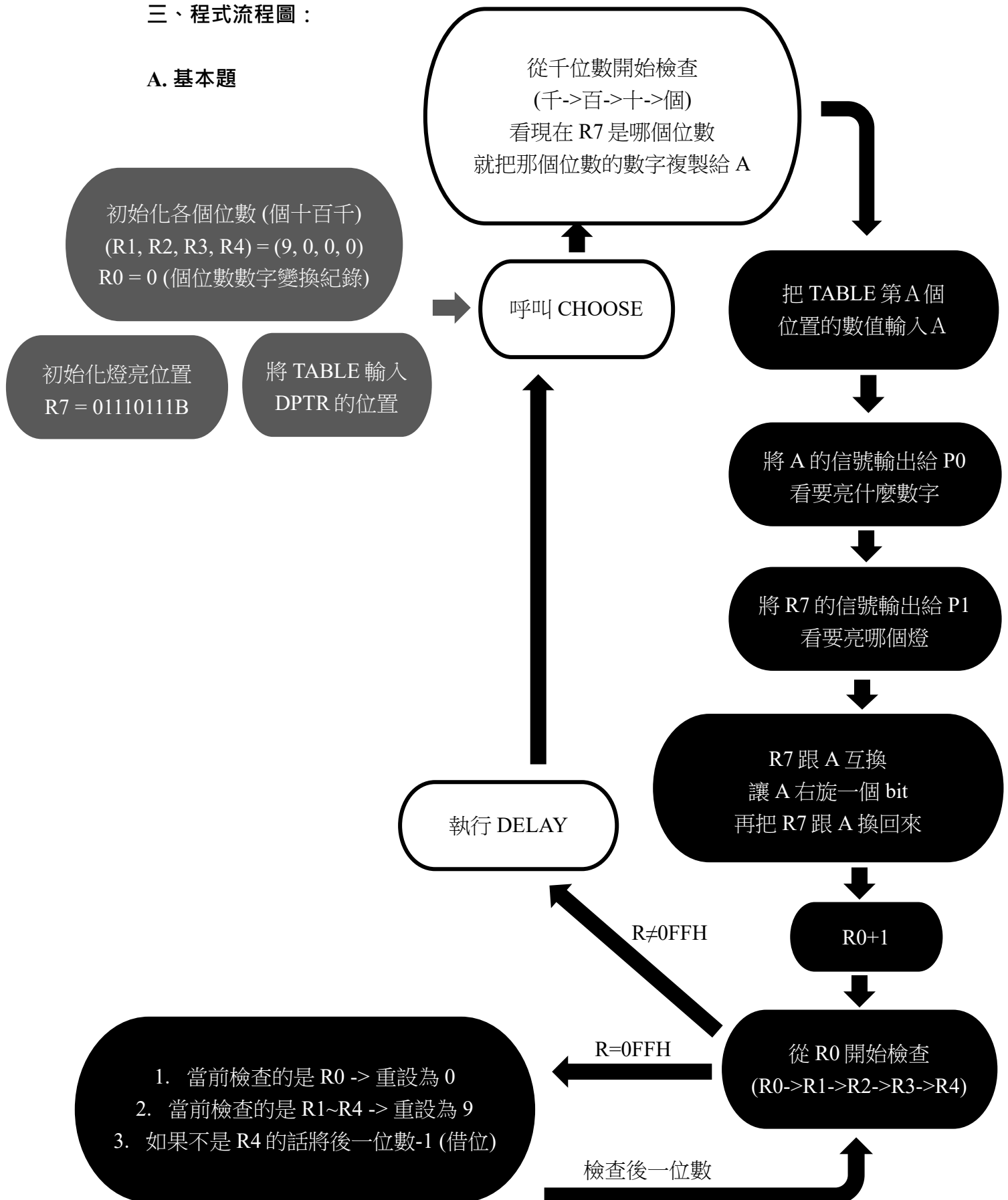
## 二、硬體架構：

P0 控制燈亮數字(A~H)，P1 控制燈亮位置(D1~D4)

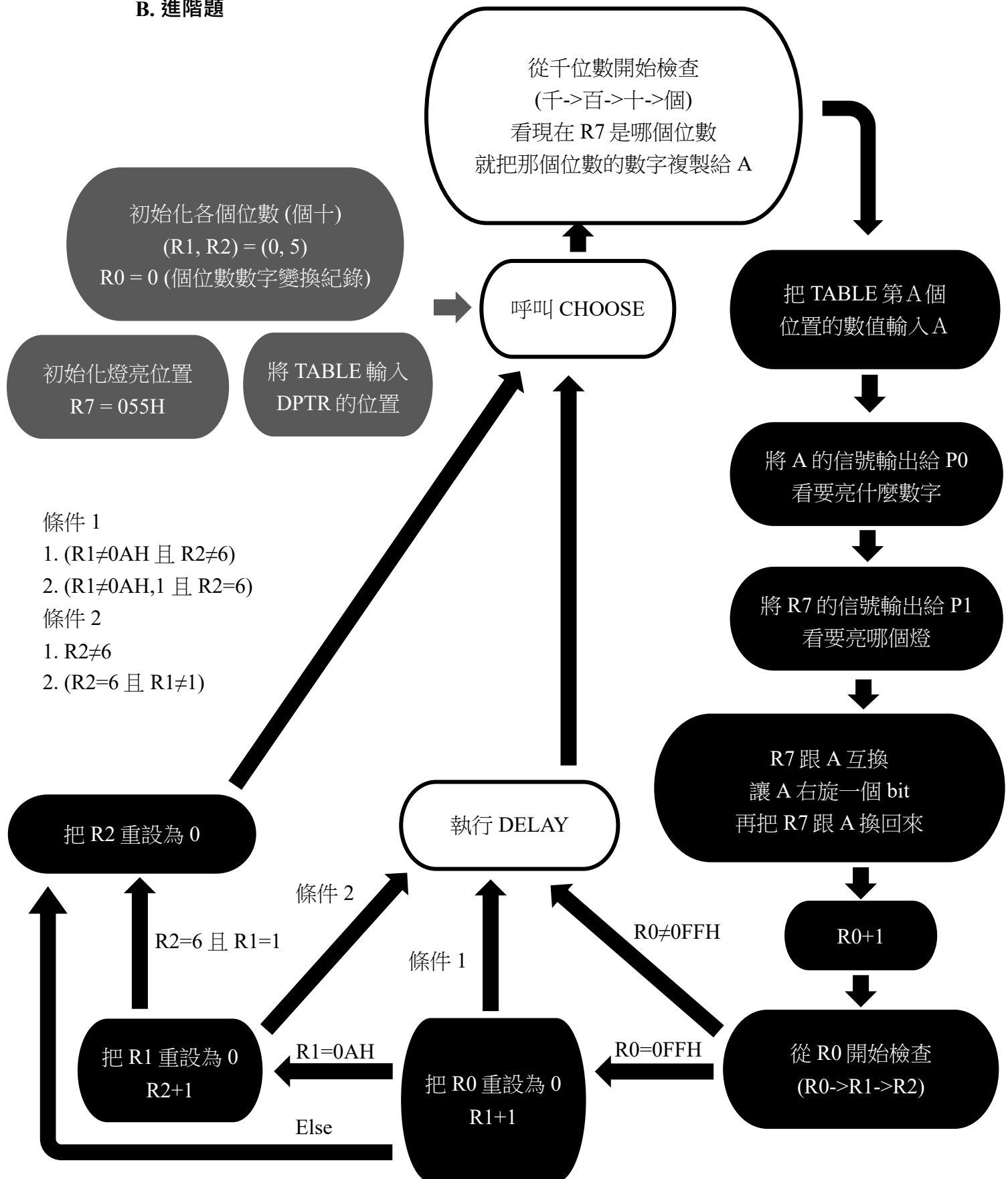


### 三、程式流程圖：

#### A. 基本題



## B. 進階題



#### 四、問題與討論：

(1) 若要使用一個開關來控制計數器 run 或 stop，要如何設計？(用文字及系統流程圖說明，不用附程式)

可以新增一個暫存器紀錄當前開關的狀態，並在 R0 (控制個位數字的變換的暫存器)改變前對開關做判斷，如果開關開的話就繼續讓 R0 改變(+1 或-1)，如果開關關的話就讓 R0 停止改變，如此一來七段顯示器就會一直顯示當前數值，不會有數字的跳動。

(2) 延續本實驗的基本題，若想在一秒鐘進行最少次數的燈號變換在確保視覺暫留的效果的同時令每顆七段顯示器的亮度一致，要如何設計？(用文字及系統流程圖說明，不用附程式)

要讓人體眼睛產生視覺暫留的時間大約是 0.1 秒到 0.4 秒之間，也就是一秒鐘要重複刷新 10-40 次才會產稱視覺暫留，不會讓數字暗掉，只看到其中幾個位數，因此需要調整 DELAY 的延遲時間，同時考慮程式本身的延遲。而選擇 0.1 秒可以讓七段顯示器用最少的次數變換來讓肉眼看見四位數字，且亮度相同。

## 五、程式碼與註解：

### A、基本題

```
1  # 初始設定
2  ORG 0000H;
3  AJMP MAIN;
4  ORG 0050H;
5
6  MAIN :
7      MOV R0, #00H //記錄個位數數字變換的暫存器
8      //目標：初始化各個位數數值 0009
9      MOV R1, #09H //個位數 9
10     MOV R2, #00H //十位數 0
11     MOV R3, #00H //百位數 0
12     MOV R4, #00H //千位數 0
13     MOV DPTR, #NUM; //將NUM的位置存入DPTR
14     MOV R7, #01110111B; //燈亮的位置
15
16     LOOP:
17         ACALL CH00SE; //跑到CH00SE這個副程式
18         MOVC A, @A+DPTR; //將Table(DPTR)的第A個數值輸給A
19         MOV P1, R7; //把R7的信號輸入D1-D4當中
20         MOV P0, A; //把A的信號輸入A-H當中
21
22         //目標：讓R7右移一bit
23         XCH A, R7; //交換A跟R7的數值
24         RR A; //讓A的數值右移一個bit，原本在最右邊的移到最左邊
25         XCH A, R7; //交換A跟R7的數值
26
27         INC R0; //R0的值+1
28
29         //目標：計算當前要顯示的數字
30         CJNE R0, #0FFH, RESET0; //如果R0≠0FFH的話跳到RESET0副程式
31         MOV R0, #00H; //把R0重新設為0
32         DEC R1; //R1-1
33
34         CJNE R1, #0FFH, RESET0; //如果R1≠0FFH的話跳到RESET0副程式
35         MOV R1, #09H; //把R1重新設為9
36         DEC R2; //R2-1 當R1(個位數)不夠的時候要借位
37
38         CJNE R2, #0FFH, RESET0; //如果R2≠0FFH的話跳到RESET0副程式
39         MOV R2, #09H; //把R2重新設為9
40         DEC R3; //R3-1 當R2(十位數)不夠的時候要借位
41
42         CJNE R3, #0FFH, RESET0; //如果R3≠0FFH的話跳到RESET0副程式
43         MOV R3, #09H; //把R3重新設為9
44         DEC R4; //R4-1 當R3(百位數)不夠的時候要借位
```

```

46      //因為只顯示到千位數,所以不夠的時候不需要考慮借位,只要重設為9就可以了
47      CJNE R4, #0FFH, RESET0;
48      MOV R4, #09H; //把R4重新設為9
49
50  RESET0:
51      ACALL DELAY; //跑到DELAY副程式
52      JMP LOOP; //跳回去LOOP重新執行
53
54      //目標：看現在亮的是哪個燈的位置
55  CHOOSE:
56      CJNE R7, #11101110B, NUM4; //如果是千位數的話
57      MOV A, R4; //複製R4給A
58  NUM4:
59      CJNE R7, #11011101B, NUM3; //如果是百位數的話
60      MOV A, R3; //複製R3給A
61  NUM3:
62      CJNE R7, #10111011B, NUM2; //如果是十位數的話
63      MOV A, R2; //複製R2給A
64  NUM2:
65      CJNE R7, #01110111B, NUM1; //如果是個位數的話
66      MOV A, R1; //複製R1給A
67  NUM1:
68      RET

```

```

70  /*"DELAY": 同Lab1, 為避免LED的數值跳動太快,
71      導致我們肉眼無法判讀, 因此需要延遲LED每次的顯示時長*/
72  DELAY:
73      MOV R5, #07FH; //令R5初始值為07FH
74
75  DELAY1:
76      MOV R6, #05CH; //令R6初始值為05CH
77
78  DELAY2:
79      DJNZ R6, DELAY2;
80      DJNZ R5, DELAY1;
81      RET;
82
83      //NUM的Table
84  NUM:
85      DB 1000000B; //0
86      DB 11111001B; //1
87      DB 10100100B; //2
88      DB 10110000B; //3
89      DB 10011001B; //4
90      DB 10010010B; //5
91      DB 10000010B; //6
92      DB 11111000B; //7
93      DB 10000000B; //8
94      DB 10011000B; //9
95      DB 00;
96
97  END;

```

## B、進階題

```
1  #初始設定
2  ORG 0000H ;
3  AJMP MAIN ;
4  ORG 0050H;
5
6  MAIN :
7      MOV R0, #00H; //記錄個位數數字變換的暫存器
8      //目標：初始化各個位數數值 05
9      MOV R1, #00H; //個位數0
10     MOV R2, #05H; //十位數5
11     MOV DPTR, #NUM; //將NUM的位置存入DPT.、(13)
12     MOV R7, #055H; //看哪個燈要亮
13 LOOP:
14     ACALL CH00SE; //跑到CH00SE這個副程式
15     MOVC A, @A+DPTR; //將Table(DPTR)的第A個數值輸給A
16     MOV P1, R7; //把R7的信號輸入D1-D4當中
17     MOV P0, A; //把A的信號輸入A-H當中
18
19     //目標：讓R7右移一bit
20     XCH A, R7; //交換A跟R7的數值
21     RR A; //讓A的數值右移一個bit，原本在最右邊的移到最左邊
22     XCH A, R7; //交換A跟R7的數值
23
24     INC R0; //R0的值+1
25
26     //目標：計算當前要顯示的數字
27     CJNE R0, #0FFH, RESET; //如果R0≠0FFH的話跳到RESET副程式
28     MOV R0, #00H; //把R0重新設為0
29     INC R1; //R1+1
30
31     //為了檢查是不是61，所以跳到另一個副程式
32     CJNE R1, #0AH, SIX; //如果R1≠0AH的話跳到SIX副程式
33     MOV R1, #00H; //把R1重新設為0
34     INC R2; //R2+1 因為是上數所以要進位
```

```

36      CJNE R2, #06H, RESET; //如果R2≠6的話跳到RESET副程式
37      CJNE R1, #01H, RESET; //如果R1≠1的話跳到RESET副程式->為了顯示60
38      MOV R2, #00H; //把R2重新設為0
39  RESET:
40      ACALL DELAY;
41      JMP LOOP;
42  SIX:
43      CJNE R2, #06H, RESET; //如果R2≠6的話跳到RESET副程式
44      CJNE R1, #01H, RESET; //如果R1≠1的話跳到RESET副程式
45      MOV R2, #00H; //如果現在的顯示數字是61(R2=6,R1=1)就把R2歸0->顯示01
46      JMP LOOP; //跳回去LOOP重新執行
47
48  CHOOSE:
49      CJNE R7, #10101010B, NUM2; //燈亮位置是十位數
50      MOV A, R2; //把R2的值輸給A
51  NUM2:
52      CJNE R7, #01010101B, NUM1; //燈亮位置是個位數
53      MOV A, R1; //把R1的值輸給A
54  NUM1:
55      RET
56
57  /*"DELAY": 同Lab1, 為避免LED的數值跳動太快,
58  | 導致我們肉眼無法判讀, 因此需要延遲LED每次的顯示時長*/
59  DELAY:
60      MOV R5, #07FH; //令R5初始值為07FH
61  DELAY1:
62      MOV R6, #05CH; //令R6初始值為05CH
63  DELAY2:
64      DJNZ R6, DELAY2;
65      DJNZ R5, DELAY1;
66      RET;

```

```

68  //NUM的Table
69  NUM:
70      DB 1000000B; //0
71      DB 11111001B; //1
72      DB 10100100B; //2
73      DB 10110000B; //3
74      DB 10011001B; //4
75      DB 10010010B; //5
76      DB 10000010B; //6
77      DB 11111000B; //7
78      DB 10000000B; //8
79      DB 10011000B; //9
80      DB 00;
81
82  END;

```



## 六、心得：

### 1. 對於上課內容的心得感想：

這次的課程用到了七段顯示器，跟之前的不同，有兩組接口相互配合產生顯示的數字，D1~D4 控制燈亮的位置，A~H 則是控制要顯示的數字，同時還要考慮到讓肉眼可以同時看到 4 個數字的視覺暫留，比次之間都需要相互配合，因此在上課時聽講就相較之前吃力，對實驗內容的實行毫無頭緒。

### 2. 對於實驗內容的心得感想：

這次實驗比起前幾次難度有大幅度的提升，是第一次在上課沒有實作出來的實驗。尤其進階題要在顯示 60 後回到 01 讓我卡了很久，原本的程式碼只能從 59 跳到 01，或是會一直從 60 往上數然後壞掉，在詢問助教過後才找到程式的問題所在，是因為忘記考慮重新進入迴圈時會先滿足前面的條件，因此不會進入到我新增的條件的地方。有時候自己看自己的程式很容易被自己的固有思想侷限住，debug 不出個所以然，這時候尋求同學或助教的協助就顯得更為重要。