



# 微算機實驗報告

Lab #07

姓名：洪巧芸

系級：資工 114

學號：110550143

上課時間：2023/11/7

## 一、實驗目的：

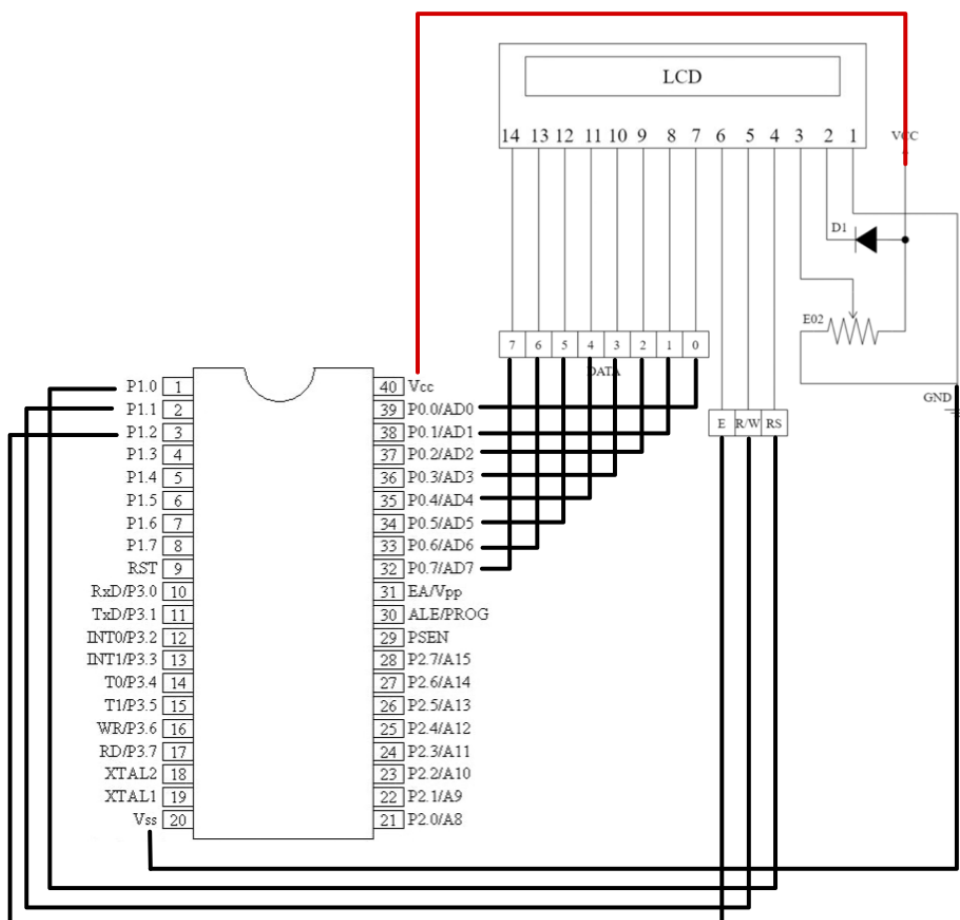
了解 Liquid Crystal Display (LCD) Module 工作原理以及相關的控制方法。

在直接從 DDRAM 抓取原本就有的圖形資料外，學習自己設定要顯示的圖形

並將其寫入 CGRAM 顯示以及顯示模式的設定(e.g. 一行或兩行、游標控制)。

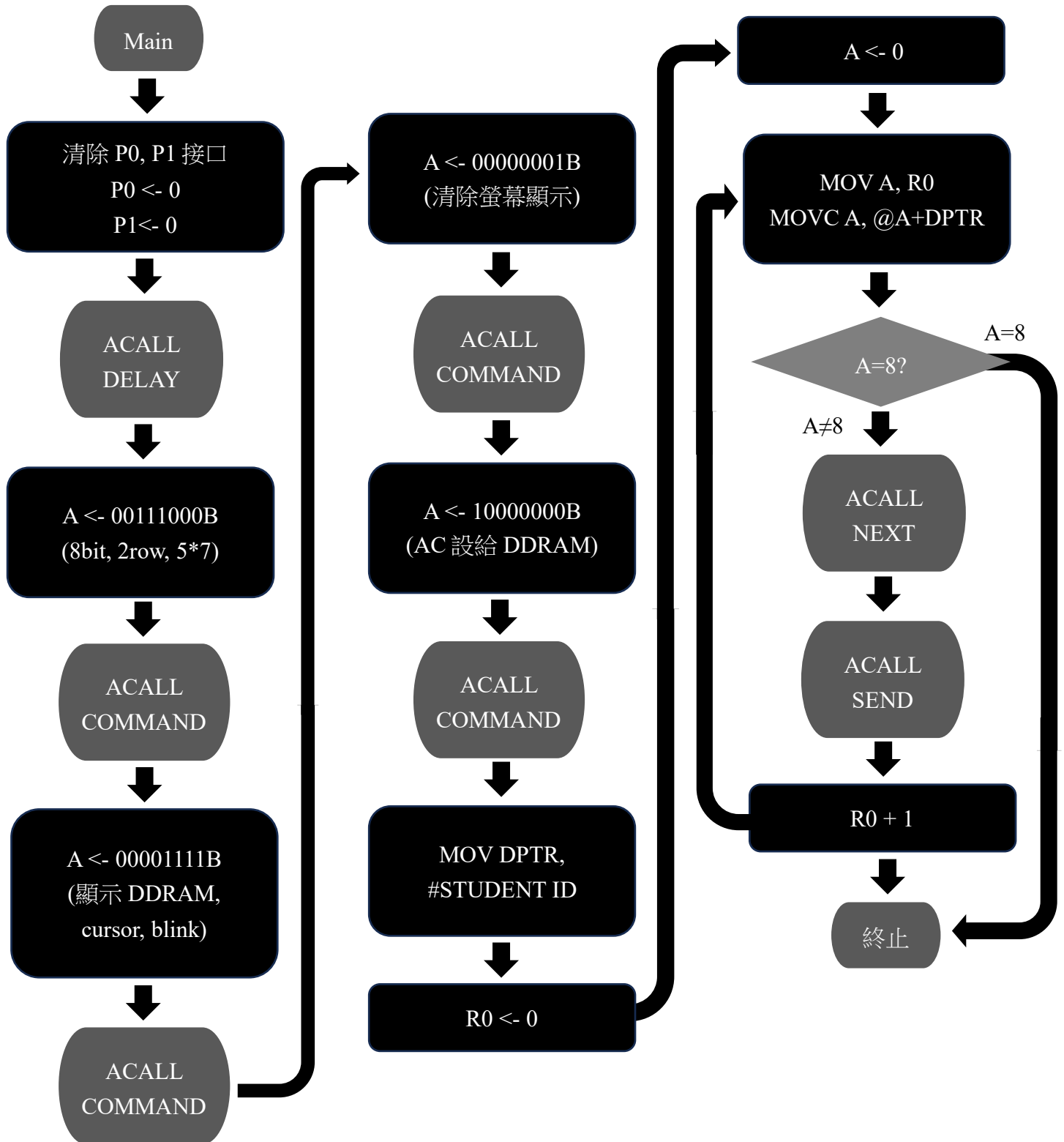
## 二、硬體架構：

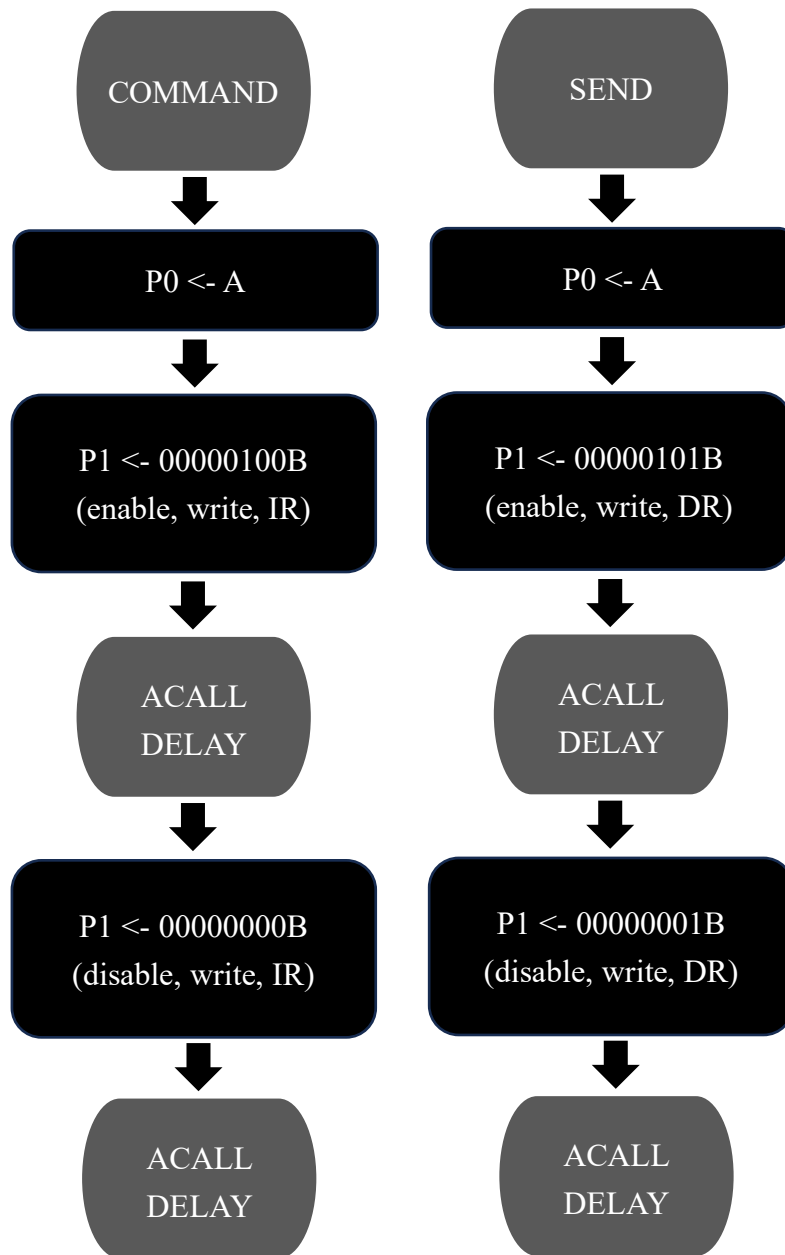
P0 連接 LCD 要顯示的數字(Data)，P1 連接 LCD 的控制項(RS, R/W, E)。



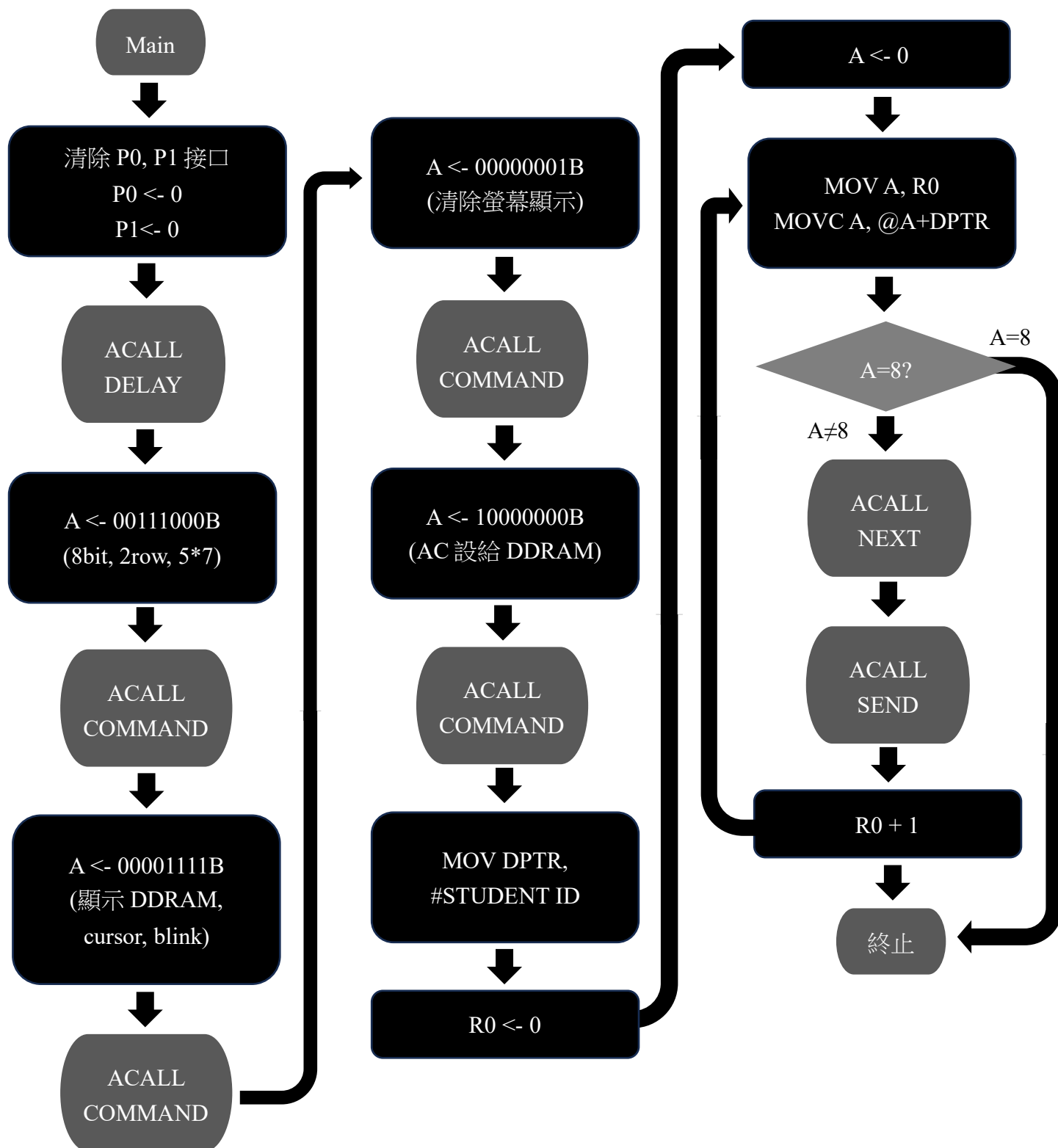
### 三、程式流程圖：

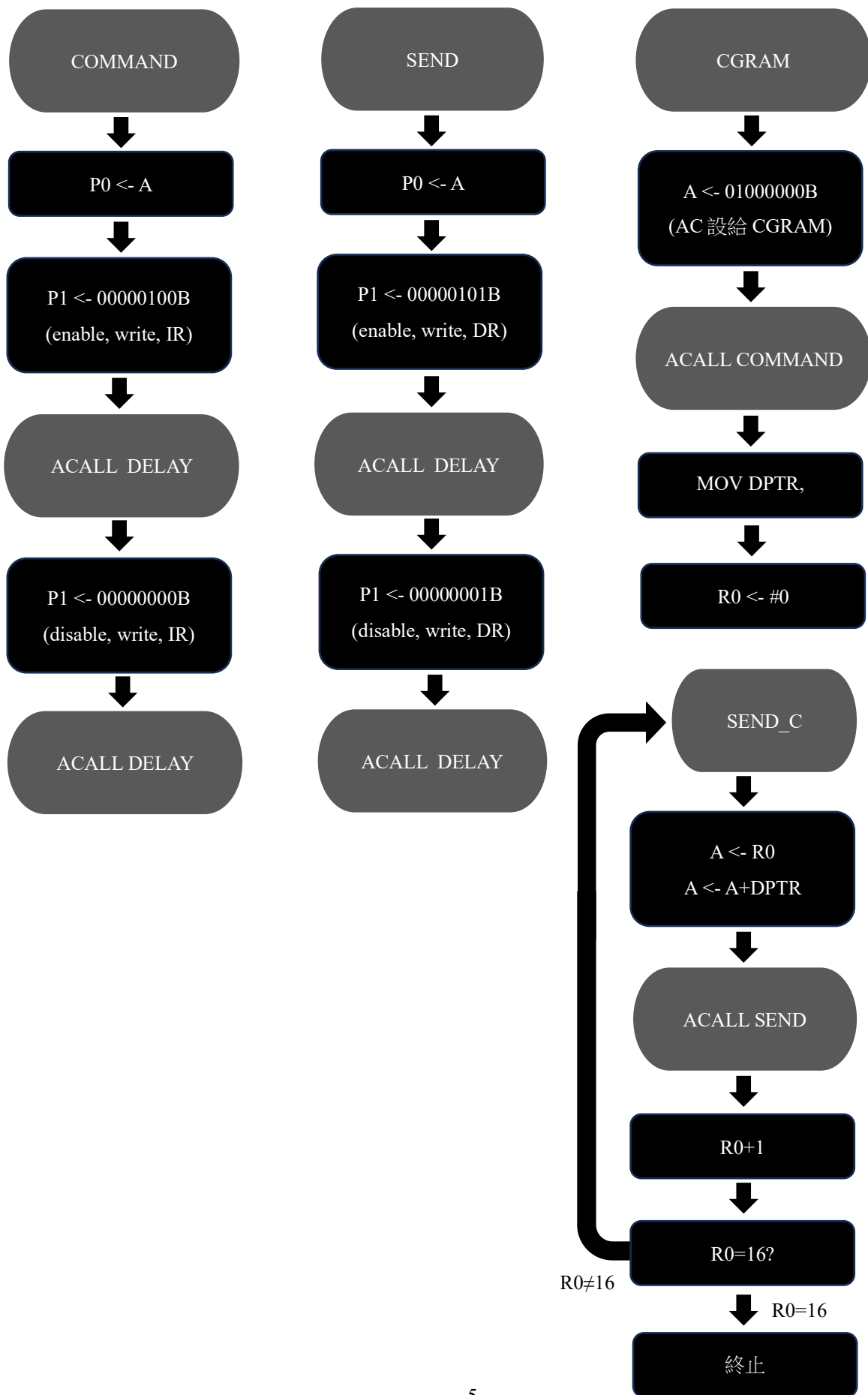
#### A. 基本題





## B. 進階題





#### 四、問題與討論：

(1) 在講義中提到，在進行模組間通訊的資料傳輸時，在讀寫資料或指令需要一定的延遲時間。假設：今天一個系統以 1ms 的取樣週期由 LCM 讀取資料，如果在讀取資料時，延遲時間過短，會發生什麼問題？

沒有辦法去確保每次的輸入都會被寫入微算機中。如果輸入指令所間隔的時間很短，在還沒有處理完之前的輸入時就輸入新的指令就會寫入失敗。舉第二題為例，如果延遲時間不夠長，然後動畫的組成頁面更多，那就會有跳頁顯示的情形。

#### 五、程式碼與註解：

##### A、基本題

```
1  ORG    0000H
2  AJMP   MAIN
3  ORG    0050H
4
5  MAIN:
6      MOV    P0,#0 //清除P0接口
7      MOV    P1,#0 //清除P1接口
8      ACALL  DELAY
9      MOV    A,#00111000B //8BIT, 2ROW, 5X7
10     ACALL  COMMAND //寫入指令
11     MOV    A,#00001111B //DISPLAY DDRAM, CURSOR, BLINK
12     ACALL  COMMAND //寫入指令
13     MOV    A,#0000001B //清除顯示
14     ACALL  COMMAND //寫入指令
15     MOV    A,#10000000B //DDRAM歸0
16     ACALL  COMMAND //寫入指令
17     MOV    DPTR,#STUDENT_ID //把寫好的Table移到DPTR的位置
18     MOV    R0,#0 //從學號的第一位開始顯示
19
20  LOOP:
21     CLR    A //清除A
22     MOV    A,R0
23     MOVC   A,@A+DPTR //將學號的第A(R0)位數字輸入A
24     CJNE   A,#8,NEXT //如果不是第8位數，則跳轉到NEXT(因為學號有8位)
25     AJMP   $
26  NEXT:
27     ACALL  SEND
28     //R0+1再跳回圈，以顯示下一位的學號
29     INC    R0
30     AJMP   LOOP
31
```

```

32 //傳指令到Data Bus -> Instruction Register
33 COMMAND:
34     MOV     P0, A
35     MOV     P1, #00000100B //ENABLE,WRITE,IR
36     ACALL   DELAY
37     MOV     P1, #00000000B //DISABLE,WRITE,IR
38     ACALL   DELAY
39     RET
40
41 //傳資料到Data Bus -> Data Register
42 SEND:
43     MOV     P0,A
44     MOV     P1, #00000101B //ENABLE, WRITE, DR
45     ACALL   DELAY
46     MOV     P1, #00000001B //DISABLE, WRITE, DR
47     ACALL   DELAY
48     RET
49
50 // 延遲
51 DELAY:
52     MOV     R5, #080H
53 DELAY1:
54     MOV     R6, #080H
55 DELAY2:
56     DJNZ    R6, DELAY2
57     DJNZ    R5, DELAY1
58     RET
59
60 STUDENT_ID:
61     DB      "110550143", 8
62     END

```

## B、進階題

```

1  ORG     0000H
2  AJMP    MAIN
3  ORG     0050H
4
5  MAIN:
6      MOV     P0,#0 //清除P0接口
7      MOV     P1,#0 //清除P1接口
8      ACALL   DELAY
9      ACALL   CGRAM
10     MOV     A, #00111000B //8BIT,2ROW,5X7
11     ACALL   COMMAND //寫入指令
12     MOV     A, #00001100B //DISPLAY DDRAM,CURSOR,BLINK
13     ACALL   COMMAND //寫入指令
14     MOV     A, #00000001B //清除顯示
15     ACALL   COMMAND //寫入指令
16     MOV     A, #10000000B //把AC設給DDRAM
17     ACALL   COMMAND //寫入指令
18

```

```

19  /*目標：一頁一頁顯示動畫(2個圖像)，
20      每次先把cursor移到左上角讓LCM顯示後再寫入資料，
21      不斷重複以達到動畫的效果*/
22  LOOP:
23      //顯示第一個圖案
24      MOV     A, #10000000B
25      ACALL   COMMAND //寫入指令
26      MOV     A, #00H
27      ACALL   SEND //寫入資料
28      ACALL   POSTPONE
29
30      //顯示第二個圖案
31      MOV     A, #10000000B
32      ACALL   COMMAND //寫入指令
33      MOV     A, #01H
34      ACALL   SEND //寫入資料
35      ACALL   POSTPONE
36      AJMP    LOOP
37
38  //傳指令到Data Bus -> Instruction Register
39  COMMAND:
40      MOV     P0, A
41      MOV     P1, #00000100B //ENABLE,WRITE,IR
42      ACALL   DELAY
43      MOV     P1, #00000000B //DISABLE,WRITE,IR
44      ACALL   DELAY
45      RET
46
47  //傳資料到Data Bus -> Data Register
48  SEND:
49      MOV     P1, #00000101B //ENABLE,WRITE,DR
50      MOV     P0, A
51      ACALL   DELAY
52      MOV     P1, #00000001B //DISABLE,WRITE,DR
53      ACALL   DELAY
54      RET
55
56  //小延遲
57  DELAY:
58      MOV     R6, #0FFH
59  DELAY1:
60      MOV     R7, #0FFH
61  DELAY2:
62      DJNZ    R7, DELAY2
63      DJNZ    R6, DELAY1
64      RET
65

```



```

65 //大延遲
66 POSTPONE:
67     MOV     R1, #10
68 POSTPONE1:
69     ACALL   DELAY
70     DJNZ    R1, POSTPONE1
71     RET
72
73 //設定CGRAM位置
74 CGRAM:
75     MOV     A, #01000000B //把AC設給CGRAM
76     ACALL   COMMAND //寫入指令
77     MOV     DPTR, #GRAPH //把動畫的Table存到DPTR的位置
78     MOV     R0, #0 //從第0列開始印
79
80 //傳資料到CGRAM
81 SEND_C:
82     MOV     A, R0
83     MOVC    A, @A+DPTR
84     ACALL   SEND //寫入資料
85     INC     R0
86     /*因為動畫總共有兩個圖案，每次都是8列，一共16列，
87     | 所以還沒有顯示完16列之前要一直循環這個迴圈*/
88     CJNE    R0, #16, SEND_C
89     RET
90

```

```

92 //小人在跑的圖案
93 GRAPH:
94     DB     10100B
95     DB     10100B
96     DB     11111B
97     DB     00101B
98     DB     01110B
99     DB     01011B
100    DB     11000B
101    DB     11111B //地板
102
103    DB     00101B
104    DB     00101B
105    DB     11111B
106    DB     10100B
107    DB     01110B
108    DB     11010B
109    DB     00011B
110    DB     11111B //地板
111    END

```

## 六、心得：

### 1. 對於上課內容的心得感想：

之前看零件盒的時候看到 LCD 板子只覺得他就是單存拿來顯示數字或文字的板子，沒有想到它其實是一台微算機，能夠藉由不同接口去控制不同的輸出。且相較於之前所使用的七段顯示器，LCD 能夠更加直觀地顯示數字(DGRAM)相比七段顯示器在想要顯示數字時需要以轉換過的數字顯示(e.x. 要顯示 1 需要輸入 11111001B)，LCD 可以直接用 ASCII code 表示，甚至還可以顯示自己設計的動畫(CGRAM)。

### 2. 對於實驗內容的心得感想：

我覺得在寫這次的程式碼時很需要將流程具象化，或者說先把每個部分的 RS, E, R/W 狀態都寫出來，在要 send DDRAM 的時候三個信號分別要是什麼，改成處理 CGRAM 的時候又是什麼，DDRAM 的 read 跟 send 和 CGRAM 的寫法又有什麼不同，在沒有先寫出來的情況下我覺得很容易混淆。