

## 群組作業, Week 5

組別: 18 參與的成員: 簡蔚驛、鄧暉宣

- 利用RTL寫出RET 指令的完整動作。(參考8051 micro-architecture, 使用PC[7:0]代表PC的低位元組, PC[15:8]則是高位元組。請注意PC與BUFFER都是16位元, 堆疊在哪?)

```
STEP1. PAR = PC  
STEP2. PC = PC + 1  
STEP3. IR = EPROM[PAR]  
STEP4. RAR = SP  
STEP5. BUFFER[15:8] = RAM[RAR]  
STEP6. SP = SP - 1  
STEP7. RAR = SP  
STEP8. BUFFER[7:0] = RAM[RAR]  
STEP9. SP = SP - 1  
STEP10. PC = BUFFER
```

- (From Lab 4)表格(TABLE)存放在[Code Memory | Data Memory]  
**ans: Code Memory**
- (From Lab 4)演算法的課程會教導使用較為優雅的方式來計算出燈號的內容,原因何在?  
**ans: 本來的方式太暴力, 演算法能訓練邏輯**
- 使用標準的時脈, 12MHz, 請計算下列的DELAY常式會造成多久的延遲。  
(Hint: 請先查表, 將各指令的執行需要多少機器週期, 填入週期欄位。次數則是填入執行的次數, 2次就填2, 次數等於R1的值就填R1。最後將各指令的"週期 \* 次數"欄位加總)

```
DELAY:  
    MOV  R0, #60H  
DELAY1:  
    MOV  R7,#80H  
    DJNZ R7,$  
    DJNZ R0,DELAY1  
    RET
```

| 指令             | 週期 | 次數      | 週期 * 次數     |
|----------------|----|---------|-------------|
| DELAY:         |    |         |             |
| MOV  R0, #60H  | 1  | 1       | 1           |
| DELAY1:        |    |         |             |
| MOV  R7,#80H   | 1  | R0      | R0          |
| DJNZ R7,\$     | 2  | R7 * R0 | 2 * R7 * R0 |
| DJNZ R0,DELAY1 | 2  | R0      | 2 * R0      |
| RET            | 2  | 1       | 2           |

$$3 + 3 * R0 + 2 * R7 * R0 = 24867$$

**ans: 2.07225 ms**