## 一、設計方法:

我這次設計的方式是採用字典排序法,dic 的順序一路從[1~8]跑到[8~1],每次改動順序都會改到下一個字典序,大致演算法如以下的實作,硬體的部分,為了防止 latch 跟 03 error 的發生,每個變數在 combinational blk 一開始有給好初始值,然後在交換的部分就比較笨,我寫了如下的巨型巢狀判斷去取代 c++的 for 迴圈的使用,然後用一個 wire 叫 en 判斷目前 dic list 跑到哪,如果跑到 8~1 就進到下個 fsm 的 output stage

```
for(int i = change_point+1;i<=(change_point+1)+(6-change_point)/2;++i){
    int temp = dic_list[i];
    dic_list[i] = dic_list[8-i+change_point];
    dic_list[i] = dic_list[8-i+change_point];

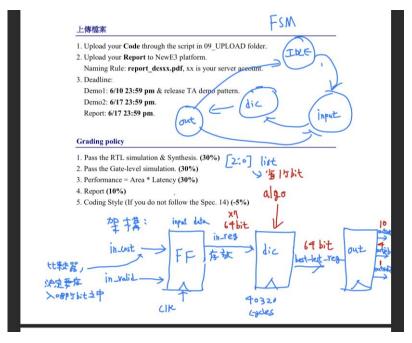
dic_list[8-i+change_point] = temp;

}
// go through every case
if((dic_list[0]==7) && (dic_list[1]==6) && (dic_list[2]==5) && (dic_list[3]==4) && (dic_list[4]==3) && (dic_list[5]=
    keep_exchange = false;
}
</pre>
```

以上為 C++軟體想法

以上為軟體想法轉換成硬體 verilog

## 二、架構圖:



## 三、心得報告:

這次的作業我大概開始的日期是 6/5 號,我首先比對了一下兩個演算法 ic contest pdf 檔的描述,發現一個超多頁一個不到一頁,就隨即決定要使用暴力字典法,給自己一個輕鬆快樂的暑假。首先開始實作軟體,實作完軟體開始照 spec 寫 pattern,再來開始 design,全部過程兩天結束!開心快樂~我其實之前寫 hw 跟 lab 心中都沒有電路架構都直接隨寫隨想,然後發現這樣效能真的很差,hw5 開始嘗試要先畫架構圖,但是就是一看就知道很大,然後那次的效能超級差。但我都完全懶的思考要怎麼壓電路…我覺得下學期修研究所課真的要多想看看!

然後再來分享一下這學期的心得,我覺得助教們真的都很用心在這堂課上,修課的感受跟之前在電機上其他課不太一樣,有的課的助教可能你問他問題他自己也不會,但這堂我真的覺得很用心,而且助教回覆都超級快也感覺超級厲害!

## 四、困難與解決:

這學期學完我發現 verilog 會遇到的問題好像都很類似,像是 01 可能就是功能不對、程式語法不對、軟體驗證不對,02 是 latch 跟 timing,可以試著找 critical path 擋 register 或是使用 pipeline 的設計手法,03 就是 unknown 的問題,可能是陣列取值不明確、invalid 沒使用之類。這次我 01 遇到的問題就是因為直接複製上次作業 code 然後發現忘了改 counter 沒動,不過改完 01 沒問題之後發現 02 跟 03 居然都跟著一起過,真的太快樂了,不過我這次就沒有用太多測資,好像只有 300,因為他會跑到天荒地老…cycle 數太多,希望 demo 的時候不會出錯!