

Digital logic design Project 2

State Minimization Report

Input4 Report

```
.start_kiss
.i 3
.o 1
.p 24
.s 3
.r S1
000 S1 S1 0
001 S1 S2 1
010 S1 S3 0
011 S1 S4 0
100 S1 S2 0
101 S1 S1 1
110 S1 S3 1
111 S1 S2 0
000 S2 S1 1
001 S2 S2 0
010 S2 S3 1
011 S2 S4 1
100 S2 S2 0
101 S2 S1 1
110 S2 S1 1
111 S2 S2 0
000 S3 S1 0
001 S3 S2 1
010 S3 S3 0
011 S3 S4 0
100 S3 S2 0
101 S3 S3 0
110 S3 S1 1
111 S3 S2 0
.end_kiss
```

Input4 kiss file

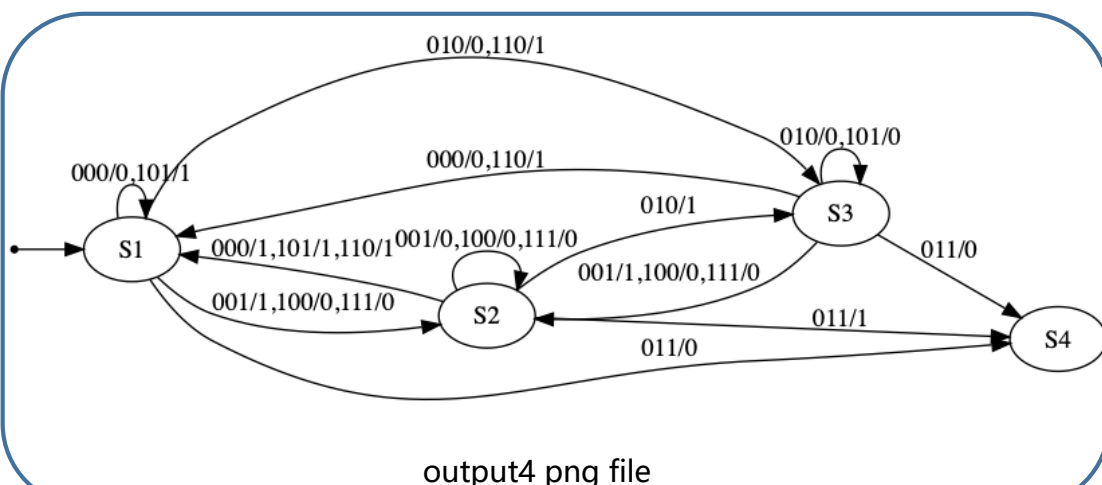
```
.start_kiss
.i 3
.o 1
.p 24
.s 3
.r S1
000 S1 S1 0
001 S1 S2 1
010 S1 S3 0
011 S1 S4 0
100 S1 S2 0
101 S1 S1 1
110 S1 S3 1
111 S1 S2 0
000 S2 S1 1
001 S2 S2 0
010 S2 S3 1
011 S2 S4 1
100 S2 S2 0
101 S2 S1 1
110 S2 S1 1
111 S2 S2 0
000 S3 S1 0
001 S3 S2 1
010 S3 S3 0
011 S3 S4 0
100 S3 S2 0
101 S3 S3 0
110 S3 S1 1
111 S3 S2 0
.end_kiss
```

output4 kiss file

```
digraph STG {
    rankdir = LR;

    INIT [shape = point]
    S1[label="S1"];
    S2[label="S2"];
    S3[label="S3"];
    INIT ->S1
    S1->S1[label="000/0,101/1"];
    S1->S2[label="001/1,100/0,111/0"];
    S1->S3[label="010/0,110/1"];
    S1->S4[label="011/0"];
    S2->S1[label="000/1,101/1,110/1"];
    S2->S2[label="001/0,100/0,111/0"];
    S2->S3[label="010/1"];
    S2->S4[label="011/1"];
    S3->S1[label="000/0,110/1"];
    S3->S2[label="001/1,100/0,111/0"];
    S3->S3[label="010/0,101/0"];
    S3->S4[label="011/0"];
}
```

output4 dot file



output4 png file

Process & Problem solving

State Minimization 可以減少不必要的 state，藉此可以達到減少電路成本的功效。

老師還沒教到該章節及概念時，我便嘗試先照著投影片的邏輯，把變化重複出現的 state 刪除，創出新的 dictionary，但最後發現，當輸入、輸出為 3 個變數時，我的程式一定會掛掉，於是我回歸最初的邏輯問題，我刪掉所有程式碼並上網查了許多資料，也想起老師之前在彈性自學群的專案中有提到過 Partition 演算法，先求出 Partition0 list，再利用迴圈找出 Final Partition。

其中遇到的第一個困難便是，要如何處理輸入、輸出值可能是 1bit 或是多個 bits？我利用 2 進制轉 10 進制的概念把每個輸入、輸出都改成有意義的數字，如此便可以用一個很大的值，當作迴圈中 range 中的參數，去跑所有可能的結果。

再來便是 Partition 後面分組的問題，我把每個輸入對應的 state 的字串跟自己相加，便會讓每個 state 跑出自己特有的標籤，再把有一樣標籤的 state 合併，最後利用 while 迴圈，如果 $p_next = p_present$ 便會終止迴圈，並找到 Final partition。

My Experience

在做完 Quine-McCluskey 專案後，我覺得這次的專案並沒有想像中的困難，我用的招式也很簡單，只用了簡單的for迴圈比較，還有簡單的list、dictionary便完成了，這種感覺令我十分振奮。這是我今年第一次修資工的課，所以利用之前管院所學的python程式基礎去完成會有些吃力，但也因為進步空間很大，我能感受這學期實力的變化，這也讓我在下學期即將面對到的大魔王「物件導向程式設計」有了些微信心，我覺得寫程式很好玩，尤其是完成後，可以很瀟灑的上傳作業更是心情的一大抒發。