

# EDA 導論 Final report

電機三 B02901059 吳昱陞

電機三 B02901108 吳省澤

## A. 實作方法

參照投影片的圖，這裡會以文字描述來解釋，首先程式碼主要分成 Gate.h、CirMgr.h、fsim.cpp，分別是建立 gate，整個電路的相關 function，程式 interface 的檔案。這裡就暫且略過讀檔的部分，主要是針對 simulation 的部分進行探討：

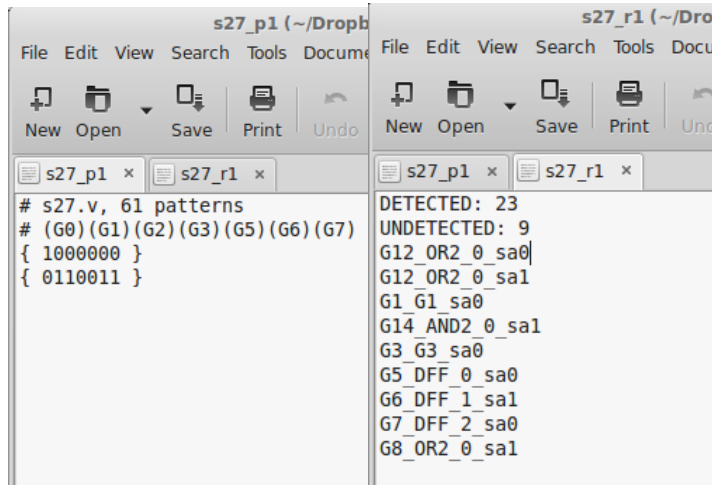
- (1) 我們會先讀進一個 stuck-at-fault 及一組 test pattern，傳給電路的每個 Input，接著會跑 CirMgr.h 裡的 cir\_sim()對整個電路進行 simulation。
- (2) cir\_sim()裡面，首先會把 stuck-at-fault 標記在對應的 gate 的 input 或 output 上，接著對每個 gate 進行 simulate。
- (3) simulate 的順序首先是電路的 Input，再來就是檢查 output 的 gate 中誰的 input 都已經有值會把它推進 sim\_id\_list。
- (4) 針對一個 gate 而言，先檢查每條 input 有沒有 fault 的標記，如果有會把標記丟給 output，接著把 input 的 true\_value 跟 faulse\_value 運算後傳給 output。
- (5) 如果 output 有標記(input 傳過來的或是 output 本身的)，就需要對 output 的 true\_value 跟 faulse\_value 進行比較，兩者相異就代表是活路，會把 output 對應到的 gate 推進 marked\_id\_list 中，並把自己從 marked\_id\_list 排除，反之(兩者相同)是死路，只會把自己從 marked\_id\_list 排除。當 simulation 到最初的 fault 標記時，就會開始檢查 marked\_id\_list 是否 empty，當 marked\_id\_list 是空的，代表現在都是死路，這時 simulation 就終止，並回傳失敗的參數。當有任意一條活路傳到電路的 Output，就回傳成功的參數。

## B. 測試結果

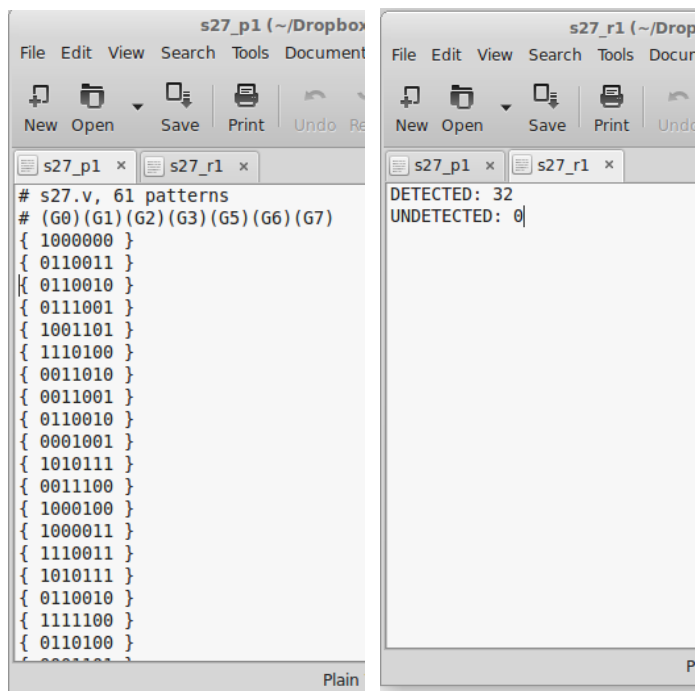
在 command line 輸入

```
./fsim s27.v s27_f1 s27_p1 s27_r1
```

這次在實作上雖然把 simulation 完成，但是仍有一些 bug 尚未解決，目前在測資上只有通過 s27.v 的測試，我們有減少 test pattern 的數量以檢驗是否有正確的 performance。但是在 s298.v 的測試中出現 segmentation fault，為此想了許多可能的問題但仍無法解決，所以在此只能針對不同的 test pattern 展示 s27.v 的結果。



(incomplete)



(complete)

### C. 未來方向

這次因為 testbench 沒有全部通過，所以在接下來首先必須把這個問題先解決，再來是演算法的優化，我認為可以把 gate 先進行排序可以減少判斷不需要 simulation 的時間，以及把之前的 simulation 結果學習起來，若是之後新的錯誤剛好在之前的 simulation 路線上出現代表一定可以偵測到，就不用針對這個新的錯誤重跑 simulation。