

Data Structure and Programming

Final Project Report

Name: 袁肇謙

ID: B06505022

Email: b06505022@ntu.edu.tw

- Data Structure

1. cirGate

- ✓ type: gate 類型
- ✓ line: 在 aag 檔中的行數
- ✓ fanin1, fanin2: gate 的輸入源 (輸入源 id * 2)
- ✓ fanout: gate 的輸出端 (輸出端 id * 2)
- ✓ ref: DFS 的標記
- ✓ symbol: gate 名稱
- ✓ sim: simulate pattern
- ✓ buc: FEC group 編號

這邊與我的 hw6 基本上一致，只是我這次 project 並沒有另外存是否 invert，這邊直接以 literal 判斷。

2. cirMgr

- ✓ _simLog: 判斷是否需要寫檔
- ✓ M, I, L, O, A: aag 檔的 header
- ✓ global_ref: 判斷 DFS 的標記
- ✓ PList: PI 的清單
- ✓ AIGList: 真正用到的 AIG
- ✓ DFSList:
- ✓ fecg: FEC group
- ✓ HashKey: 用在 strash 判斷是否等價

- Algorithms and Problems

1. READ

基本上與 hw6 完全一樣，大概就是資料結構上的小差異(fanin)，不過這邊我就沒另外作 error handling 了。

2. OPTIMIZE

- ✓ Sweep

只有 AIG 能被 sweep 掉，因此僅需判斷是否存在於 DFSList 中，不存在則 sweep 掉。這邊一開始遇到一個小問題就是不存在”的 gate 也會被 sweep 掉，因為我的 gate 一開始就依照 M 將所有 gate 建好，但這邊的問題是 $M > I + A$ 時，連本就不會存的 gate 也會被我建立，解決方式一開始要先判斷 gate type 與 fanout，若為 UNDEF_GATE 且 fanout 是 empty，則此 gate 不存在，不必理會，而被 sweep 掉的 gate 亦須將 gate type 改成 UNDEF_GATE 與將 fanout 清空，還須將以這些被清掉的 gate 為 fanout 的 gate 的 fanout 更新。這部分效能與 ref 差不多。

- ✓ Optimize

這部分分成 4 各 cases 討論: fanin 為 0、fanin 為 1、fanin 相同且同

向、fanin 相同但反向，並簡化電路。這邊較麻煩的部分為更新 fanin 與 fanout，我是將 fanin 與 fanout 相加檢查是否為偶數，若 true 則為正向，false 則為反向，更新結束後作 DFS。merge 的部分應該整理成 function 較好，不過當初憑著一股衝動就爆寫，造成 debug 上的困難。這部分效能與 ref 差不多。

3. STRASH

透過 hash function 丟入 bucket 並檢查 bucket 中是否已存在 gate，若是則可將二者 merge 起來。這邊的 hash function 必須符合 fanin1 和 fanin2 相反 hash value 仍需一致，起初我用較為廣泛使用的 $65536x + y$ 為 hash function，造成沒有 gate 能被 merge。這部分效能與 ref 差不多。

4. SIM

每 64 個 patterns 轉成 size_t 讀進電路，透過 bitwise operation 進行模擬，與此同時 simulate 後的 result 分成不同的 FEC group，每 64 個 patterns 就繼續細分，同時記錄每個 gate 所屬的 bucket，方便 cirg gateId 直接印出 FEC pair。randomSim 中所需的 64 bits unsigned number generator 則是參考自 stackoverflow。這部分效能我差了 ref 約 20 倍，據寫完的同學所述，FEC group 的分群應該用 hash set 效果較佳，不過我不太清楚如何以 hash set implement，因此我是自行開一個 vector<vector<unsigned>> 作為 group bucket，每一次 simulate 後都檢查同一個 vector<unsigned> 是否為 FEC pair，若否，開一個新 vector<vector<unsigned>> 進行分群，分完後將新舊 vector 連接。

5. FRAIG

沒能進行到這。

● 心得

很可惜沒能作完 project，雖然在期末前就把 SIM 之前的都寫完，但開始放假後看著 SIM 幾乎是毫無頭緒，直到同學投完票回來才開始教我寫，加上其他們刻亦有 final project，結果寫完的當天就是 deadline 了。

基本上到 hw6 時，我蠻意外我竟然還坐在課堂上(不是旁聽，是有學分壓力的)，畢竟我 hw1 就花了 14 小時左右，幾乎每次作業都話至少 20 個小時，作業出的當周的周五六基本上都是日出而睡的，加上能討論作業的只有一個同學，因此打 code 過程中真的是既好玩卻也是種折磨。幸好皇天不負苦心人，至少我 c++ 的能力的進步十分明顯，回頭看以前計程 100 多行的大作業，當初寫了一星期多，現在大概半天就能收工。教授說過這堂課的目標就是往後在面對大型專案時，幾千幾萬行的 code 也能 handle，但問自己真有達到如此目標嗎？其實我不敢肯定，也許還是會被這份量嚇到，但至少現在的我絕對能站更穩。DSnP 到此也告一段落，很榮幸能趕上這門課的末班車，成為我大學生涯的辛酸血淚史之一。