

資料結構與程式設計

Final Project Report

姓名：吳悠嘉

系級：電機三

學號：B06901144

聯絡 email：

b06901144@ntu.edu.tw

壹、資料結構設計

以下將分別介紹我對 CirMgr 以及 CirGate 的 class data member 的設計。

一、CirMgr

1. 建構電路

1. Gate 的儲存方式

- Gate 的數量：

在 aag 剛讀進來時，建立一個 `map<char,int> header`，其中

`header` 的 M、I、O、A 值分別儲存在 `header['M']`、

`header['I']`、`header['O']`、`header['A']` 裡。

以下動作皆在 `createGate` 中完成，且所有 gate（除了 UNDEF）皆

會存到已建立好的 private member `GateList ric` 中，儲存方式為：

`ric[gateld] = 此 gate 的指標`

- CONST0：

在 `createGate` 一開始即存入 `ric[0]` 中

- PI、PO：

除了存入 `ric`，也會分別將其 `gateld push_back` 進 private

member `vector<int> input` 及 `output` 中，以便之後 `print` 及 `write`

可利用。

- AIG、UNDEF：

依 aag 檔，將 gaet AIG 和其 fanin 的兩個 gate 的 id 存入

map<int,pair<int,int>> _and 中，讀完後再檢查 AIG 的 fanin gate

是否在 ric 中已存在，若不存在即是 UNDEF。

2. 接線

在 cirGate.h 建立 class Edge，此 class 用來儲存與現在所在的 gate

所連接的其他 gate，以及此條線是否有 invert。在 createGate 完之

後，在 connectGate 中掃過所有的 gate，並用(Edge*)儲存這個 gate

的 fanin 及 fanout。

2. FEC Groups

在 CirDef.h 中定義 typedef vector<CirGate*> FECGrp，並在 cirMgr.h

中箭 private member： vector< FECGrp> fecList，用以儲存每次 simulate

後的 FEC Groups。為了區別 gate 的 invert 已及原 gate 的不同，將進行

以下步驟：

若是 invert 則會先將 gate 的型態由 CirGate*的轉為 size_t 再-1，最後再

轉回 CirGate*。

之後判斷是否為 invert 即判斷 size_t(gate)是否被 2 整除，部整除即為此

gate 的 invert。

二、 CirGate

依每個 gate 的 type，分別建立：

CirPI、CirPO、CirAig、CirConst、CirUnDef

這五個 class 皆繼承 class CirGate。

1. 基本的 data member：

typeStr、gateId、lineNo、gateSymbol 分別儲存 gate 的種類、id、在 aag 檔所在的行數，以及 gate 的 symbol

2. Edge 相關：

每個 gate 都有 vector<Edge*> fanin 以及 fanout，用以儲存每個 gate 的 fanin 以及 fanout。

3. DFS 相關

color、pre、inDFS 分別儲存在 DFS 過程中 gate 的顏色、找到的 predecessor、以及是否在 dfsList 中，這三個變數在 gate 建立之初即會初始化為“white”、-1 以及 false。

4. Simulate 相關

simvalue 型態為 size_t 用於儲存一次讀進 64 個 bit 的模擬值。而 vector<size_t> simbit 則儲存 simvalue 的 0/1 版本，即是 64 次模擬的 0/1 值。最後 fecPartner 則是儲存與這個 gate 在同一個 fec group 的其他 gate。

貳、Hash 的運用

這次實作，我所用的為 myHashMap.h 中的 HashMap，以下將針對有用到 hashMap 的兩個功能 strash 以及區分 fecGroup 說明其 hash 資料結構的運用。

一、 Strash

在 cirDef.h 中定義一個 class 為 Fanins，作為 HashKey 使用，其目的是要將每個 gate 的兩條 fanin 合併，以方便找出每個 gate 所屬的 bucketnumber。

Fanins 只有兩個 private member 分別為 fanin1 以及 fanin2 (型態皆為 size_t)，合併方法為 overload()時，將 fanin1 以及 fanin2 分乘以 1.625，最後回傳 $(fanin1 * 1.625)^{(fanin2 * 1.625)}$ 。

在實作 strash 時，利用 HashMap<Fanins, CirGate*> 判別使否有 gate 需要合併。

二、 divideFECGroups

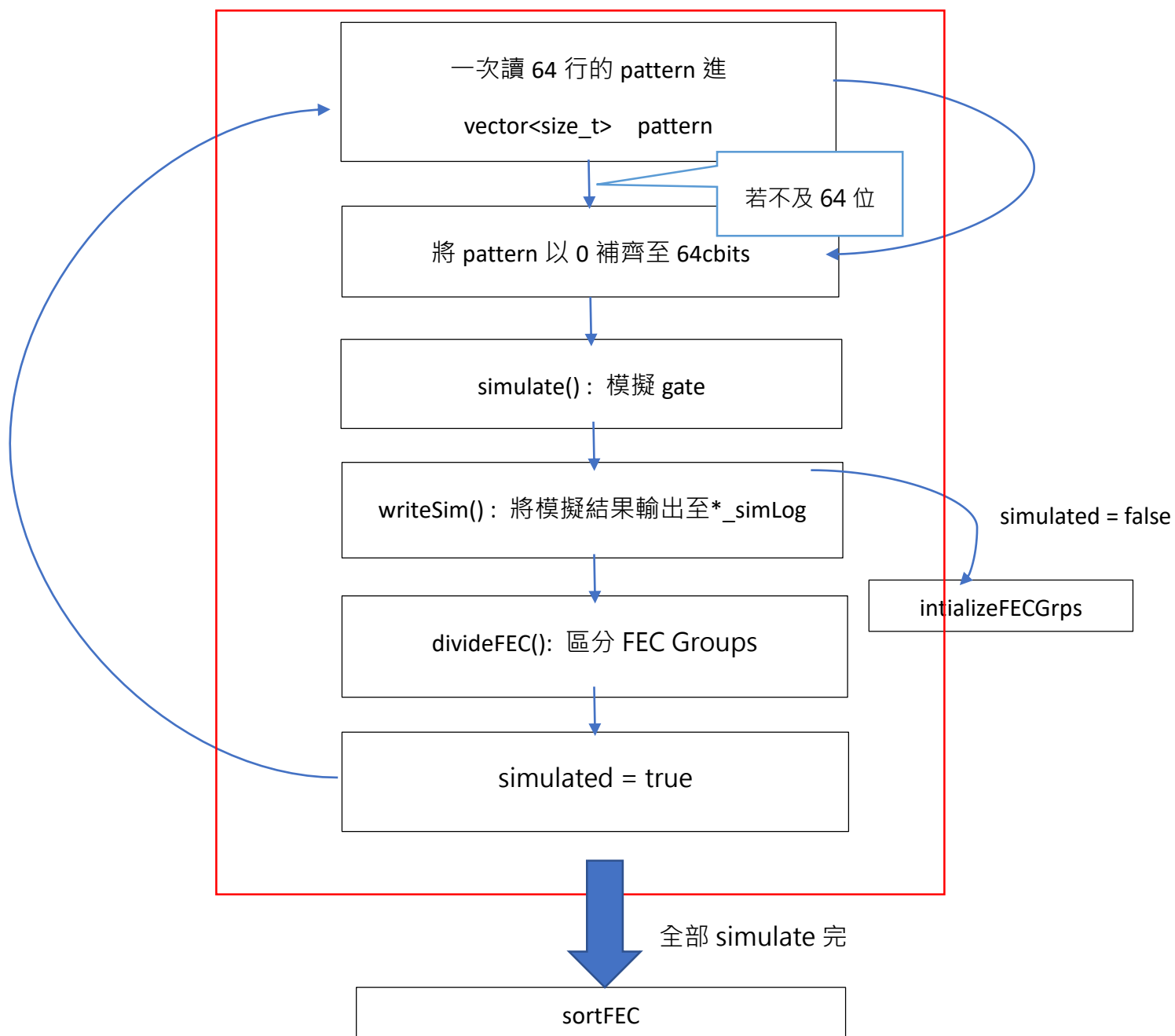
在 cirDef.h 中定義一個 class 為 Moni，overload()時只須回傳 moni (即模擬值) 即可。故在區分 FEC Groups 時，建立 HashMap<Moni, FECGrp*>，如此在區分時即不斷利用 query 來尋找是否已有模擬值為 Moni 的 FECGrp 存在即可。

• 小心得：一開始其時是用 hashset<FECGrp>，但發現過程中大量 query 都是針對模擬值，寫法上較 map 稍麻煩一些，也容易出錯。

參、Simulate

很遺憾地這次只完成到了 fileSim，故以下 simulation 的說明皆針對 fileSim。

其實作流程圖如下：



肆、心得

這真的是我有史以來寫過最多跟複雜的 code，其實有在期末考結束前完成期末考進度，把前三個都寫完了，可是到了 sim 就突然卡住，還因為有些 class 設計太爛或資料結構使用不當，把一天寫好的進度全部砍掉，所以最後只完成了 fileSim，覺得非常的可惜，實在很想把 fraig 也寫完。雖然沒有像期待一樣交出完整的 final project，但是在不斷砍掉 code 跟 debug 的過程中，真的明白了「心中要有記憶體」這句話的重要性，我因為沒有管好記憶體，發生了超多靈異事件。算是對一整個學期的總結吧，經過這次失敗心中真的比較有記憶體了，然後終於可以把這分難產的作業交出去，真的是太爽了。

伍、Reference

myHashMap 以及 HashSet 皆是使用去年修課的楊軒(b06901019)hw7 的內容