FRAIG

DSnP Final Project Report

B00201037 / 數學四 / 張志瑋

Email: jrweizhang@gmail.com
Mobile: 0988-876-576

資料結構

HashMap

這次的 HashMap 我只實作了 check 和 insert，但是其實我的 check 是和 query 類似的 — 若查詢的 key 已存在，則 update data reference。在 strash 和 simulation FEC groups 的地方用到時，我分別建立了兩個不同的 key class，對於 strash，我利用 (fanin1 + fanin2)*(fanin1 + fanin2 + 1) 的方式作爲 hash value；對於 FEC groups，我利用 simVal * ~simVal 作爲 hash value，這樣的好處是對於 inverted FEC groups 我也能有同樣的 hash value。

---

cirMgr

在 cirMgr 中我只另外新增了 vector<IdList*> _fecGrps 來儲存所有的 FEC groups，其中 IdList 是 vector<unsigned>。

---

cirGate

在 cirGate 中我新增了 _simResult 用於存放最後一次的 simulation 結果、_fecGrpIdx 用於存放所屬的 frcGrp index，若不屬於任何一個 group 則設爲 -1。

演算法實作

Sweep

我 iterate 了全部的 gate，若當前的 gate 不在 DFS list 中、且不是 PI/CONST、也不會直接連到 PO 的話，就將其清除。

Optimize

按照 DFS order 來做，首先略過所有 PI/PO/CONST/UNDEF，接着按照 spec 中的要求將等價的 gate 合併，合併的具體做法是將要留下來的 gate 連接到每個要被取代的 gate->Fanout，接着將這個 gate 從所有個 list 中刪除。一開始我用 iterator 來遍歷 DFS，但是這樣若要在 iterating 的過程中將 iterator 移出 dfs list 會造成 iterator 不 consistent，這似乎要 C++11 才有好的處理方式，於是我後來便改用 for size_t i = 0 的方式來 iterate，我發現其實在 iterate vector 的時候這種方式不見得比 ::iterator 來得麻煩，更多數時候這樣反而更好用。

Strash

Strash 的實作上比較特別的只有用於 hash 的 key；另外在 merge gate 的時候，我實作了一個 function 可以把 gate 所有的 fanin 都連到 fanout 上，這樣我就可以放心的將這個 gate 刪除。

Simulation

爲了要完成 simulation，我首先在每個 gate 上都新增了兩個 function，一個是設定 _simResult，一個是 simulate，前者用於直接指定 simulation result，可以用在 PI；後者用於實際 simulate，用於 AIG，而在 simulate 的時候我都是直接 simulate 32 bit unsigned integer，所以搭配 fanin 的 inverse phase 以及 bit-wise 的 &, ~ 即可。

接着在 simulation 的時候，每做完一輪 simulation，就直接檢查一次 FEC groups、寫入 simLog，這樣的好處是不需要儲存每次的 simulation 結果，只需要儲存前一次的就好，這裏碰到的麻煩是，我原本的架構都是直接吃 32 bit integer，但是在 sim file 的時候，是 line by line 的讀取，所以若要能變成 parallel simulation 的話，就要另外將每 32 行切成一個 chunk，而我在處理這裏的時候出了點問題，不知道在死線前能不能處理好。