

DsnP Final Project Fraig

電機三 溫明浩

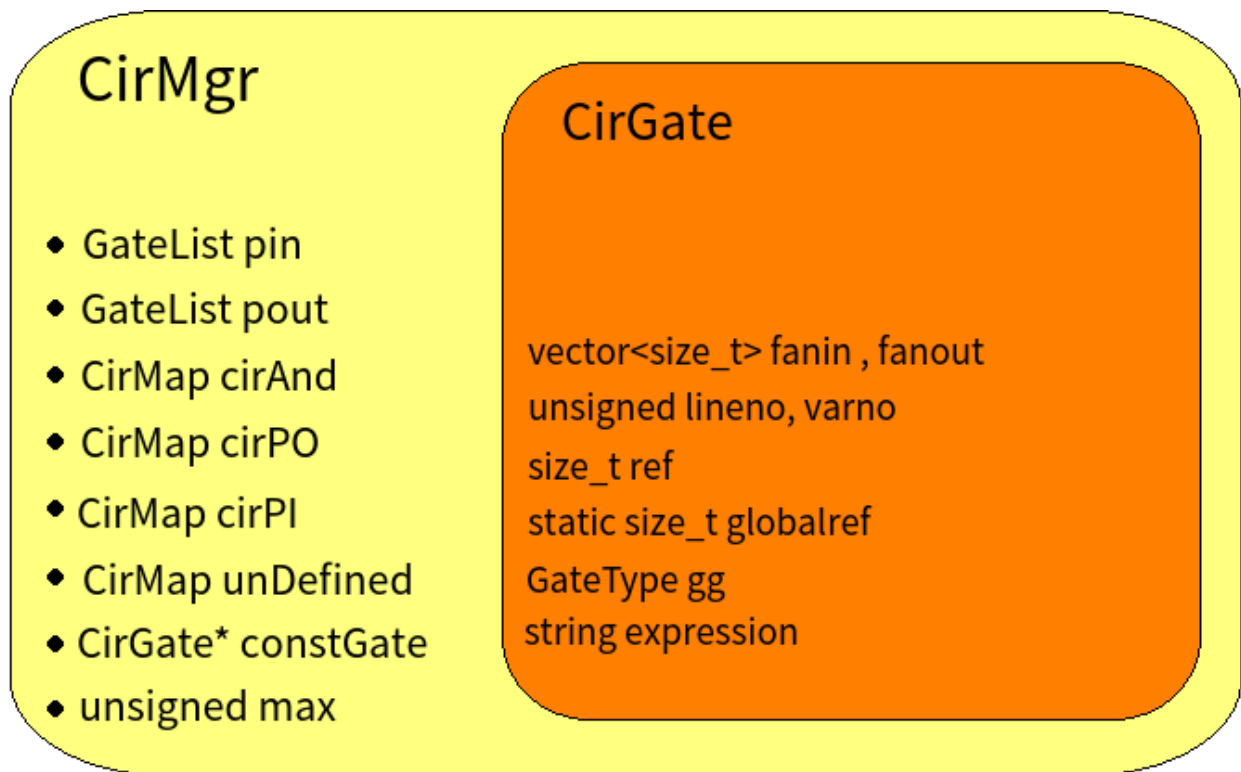
B03901179

Email: b03901179@ntu.edu.tw

Mobile: 0978821531

1. System Structure

※hw6 部份取自 江俊毅 b03901084 並修改過結構



○ CirGate

- fanin fanout 儲存連接的資訊, 是以 pointer 加上 invert 的資料並轉型成 *size_t* 儲存, 利用 *getGatePtr()*, *isInv()* 即可獲得資料.
- ref 是用於 dfs, 與 *globalref* 比較來判斷是否走過
- *GateType* gg 儲存 gate 的型態, 因此不需要繼承 *GateType* 為 enum.
- expression 儲存 symbol.

○ CirMgr

- 原先同學的只有 *cirmap*, 而我將它拆成 *cirAnd*, *cirPO*, *cirPI*, 並且獨立出 *constgate*, 其優點是對於後面的搜尋, 可以不用判別 *gateType*, 且增加搜索速度.

- pin, pout 主要是用以儲存 PI, PO 的順序,否則就要從 map 裏面搜索,比較 lineno 的大小,耗費過多時間反而得不償失.
- max 紀錄 gate 的最大數字.
 - ※關於 DFSList, 起初有想過儲存起來,之後就可以避免重複建立,但實際上除了 cirp -netlist 可以重複使用外, 其他 function 有的是透過 dfs 執行時完成, 有的只有用到部份 gate(i.e AND gate only), 有的再完成後還要 update DFSList, 因此建立一個 list 便不是這麼划算了.

2. Function Implementation

○ CirSweep

1. 對每個 PO 執行 *dfsCheck()* 後, 所有在 dsf 裡的 gate, 其 ref 會與 globalref 相同.
2. 搜尋 cirAnd , 只要其 ref 與 globalref 不同, 就斷開魂結(*connectClean()*)並 delete.

※若使用 DFSList, 可以直接從 list 裏面 mark 每個 AND gate 的 ref, 在執行步驟 2.

○ CirOptimize

1. 執行 *getDList()* , 內部為對每個 PO 執行 *dfsAnd()*, 會得到一個只有儲存 AND gate pointer 的 list.

2. 分成兩部份:

- 兩個 fanin 的 gateType 皆相同者:

利用 char(省空間)來紀錄兩個 fanin 的 invert, 如為 1(一正一反), 用 const gate 去取代, 檢測 fanin 是否有 undef gate 且 fanout 為 empty; 如為 0 or 2, 用 fanin 去取代.

- 一為 const gate 一為 other gateType:

先判斷何者為 const, 再判斷 const 是 0 or 1, 0 就直接用 const 取代, 1 則用另一個 fanin 去取代. 其中 const 若為 0, 則要檢查另一個是否為 undef 且 fanout = empty.

✖這部份想滿久的, 這種排列組合的程式已經寫過不少遍, 但只要開頭沒想好, 就會是一團亂, 且 code 重複率極高; 對於第二項 1 const 1 gate, 只要善加利用 reference 的方式, 即可避免 if...else...的方式產生重複率超高的 code

○ CirStrash

○

1. 執行 *getDList()* 得到 AND gate list.
2. 個別加總 fanin 的兩個 gate 的 literalID + invert, 得到兩個 *size_t*.

3. 丟入 `HashKey` 裡, 在裏面會將數字小的乘上 2^{32} 再加上另一數字, 得到 `key`
4. 把 `key & gate` 丟入 `hashmap`, 若發現 `key` 相同則用前面的取代現在的.

※相較於 `optimize`, `strash` 反而容易許多

○ **CirWrite**

1. 對每個 PO 執行 `dfsAll()`, 得到 `AND & PI` 個別的 `list`, 以及 `maximum number`.
2. 後面的與 `hw6` 大同小異.

※起初是把 `AND PI` 放進同個 `list`, 但是這樣會導致還要寫個判別式去計算數量, 因此改以個別存, 而且直接透過 `dfs` 時取得最大 `varno`, 還可以避免多跑一次.

○ **CirSimulate**

○ **CirFraig**

3. Performance

Command	My program	Reference Program
cirr sim13.aag usage	Period time used : 0.07 seconds Total time used : 0.07 seconds Total memory used: 21.95 M Bytes	Period time used : 0.05 seconds Total time used : 0.05 seconds Total memory used: 12.44 M Bytes
cirsweep usage	Period time used : 0.01 seconds Total time used : 0.08 seconds Total memory used: 21.95 M Bytes	Period time used : 0 seconds Total time used : 0.05 seconds Total memory used: 12.44 M Bytes

ciroptimize usage	Period time used : 0.02 seconds Total time used : 0.1 seconds Total memory used: 23.07 M Bytes	Period time used : 0.01 seconds Total time used : 0.06 seconds Total memory used: 13.3 M Bytes
cirtrash usage	Period time used : 0.05 seconds Total time used : 0.15 seconds Total memory used: 26.17 M Bytes	Period time used : 0.02 seconds Total time used : 0.08 seconds Total memory used: 16.13 M Bytes
cirwrite 55660 usage	Period time used : 0 seconds Total time used : 0.15 seconds Total memory used: 26.17 M Bytes	Period time used : 0 seconds Total time used : 0.08 seconds Total memory used: 16.13 M Bytes

4. Discussion

就以上 function 而言, read 的 memory used 產生的差異最大, 再來就是 sweep, optimize 因為 sim13.aag 沒有多餘的且可以 optimize 的 gate, 因此造成的時間差異我想應該是 dfs 要重建的緣故.

另外遺憾的就是...simulation & fraig 沒有完成 Q Q

雖然這學期幾乎活在 coding 之中, 但還算滿享受的, 畢竟真的學到不少, 從以前只會想到什麼寫什麼的方式, 到現在懂的如何把 code 寫的簡潔, 有結構性, 以及探討速度與記憶體用量, 感謝 Ric 老師這學期的教導<(_ _)>