**Dsnp Final Project**

**FRAIG**

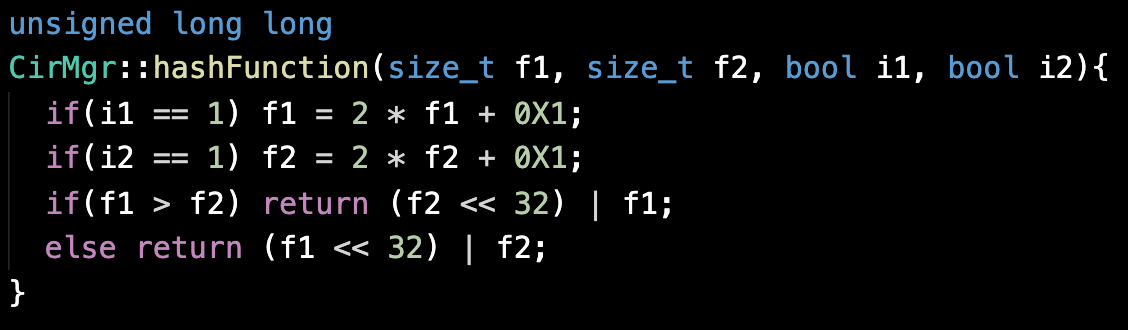
**(Functionally Reduced And-Inverter Graph)**

**B07901010**

**電機二 范詠為**

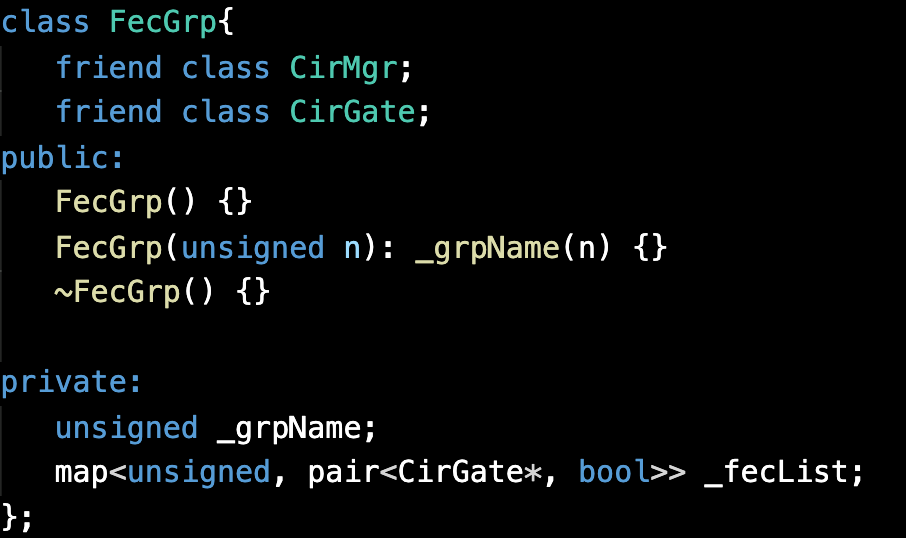
**contact inf. 0928900827**

1. **design of the data structure**
2. **strash---the unordered\_map**

在cirstrash中，為了判斷兩gate是否有一模一樣的input，但又不想用n^2的時間複雜度爆搜（兩個兩個都比過一次），而且順序不能亂掉，於是用了STL的unordered\_map去做hash，而這個map的key要能讓他分辨出input是否一致，所以我先做一次自己定義的hash如下（上網查的，return前先做排序是因為input正反是沒差的）：

1. **class FecGrp**

實作FEC的部分，我是用vector \_FecGrps<FecGrp\*>，也就是分很多組的感覺，用指標可以節省記憶體空間。每一個FecGrp\*指到一個FecGrp，也就是下面這個class，主要含有一個map，這個map有點類似\_gateList的map（<gateid, CirGate\*>）但多了最後那個布林值，為的是要判斷反向（相反simValue的會在同一組）。另外，在CirGate.cpp中，有新加一個data\_member FecGrp\*，目的在讓每個gate記住他所屬的FecGrp，在implement cirgate指令時會用得到。



1. **algorithms**
2. **cirsweep**

為了要知道誰是不在\_dfsList中需要被移除的gate，且不想用n^2時間複雜度去找，於是只好在每一個gate中記一個\_toDel = true，在跑一遍\_dfsList的過程中，有跑到的改成false代表之後要把它移除，接著跑一次\_gateList

做移除的動作。

1. **ciroptimize**

\_dfsList中是AIG\_GATE的要去檢查是否那四種情況，再重接，最後做移除，沒什麼特別的做法便不再贅述。

1. **cirstrash**

在1-a其實大概講過了一遍，同樣也是搜\_dfsList，如果即將新增的key在unordered\_map有找到的話就表示找到同樣input的兩個gate，接下來就要進行merge，再刪除，merge的作法也跟opt類似。

1. **cirsimulate**

一開始先從FileSim() 做起因為較好跟ref比對，而且做好的話random應該也不難。我的作法原本是全部的pattern都cin進來，再64個64個去做模擬，但後來因為使用的記憶體是ref的五倍，所以改成一次cin 64個，不但節省了記憶體空間，也增加了一點速度，這部分後面會討論。每一次的模擬分成三個部分，simulate(patterns), genFec(), genLog(patterns, num) 三個部分，simulate就是給每個PI\_GATE剛吃進去的input size\_t，從PO\_GATE跑過recursive後更新每個gate的simvalue值。genLog則是outputfile的部分。

最後最花時間的就是genFec了，首先要開始想FecGrp的資料型態(1-b)，決定好後實作才不至於遇到太多問題。

接下來我的作法是先intialize第0個FecGrp的東西，再從\_FecGrps開始跑起，每一個進去後要開一個map<simValue, FecGrp\*> newFecGrps，有點似hash的感覺，再來先判斷其他的simvalue是否跟base（每個grp中的第一個，也就是一定不會被移出grp的元素）ㄧ樣，ㄧ樣的話就不用管去看下一個（當然ㄧ樣是指正向反向都不同），不ㄧ樣的話就去搜newFecGrps看有沒有存在，有的話就根據newFecGrps的second搬到剛剛有人先搬去的新家，沒有的話就要新開一個FecGrp\*存進去，並insert他進去newFecGrps。

過程中搬家的人最後都要記得自己新的FecGrp\*並把舊的FecGrp內的自己刪掉。結束後再用newFecGrps的second把size>1的加入\_FecGrps中，完成genFec()的過程。

1. **cirfraig**

fraig很可惜的是我沒完整做完的部分，我試過跑\_FecGrps跟\_dfsList

，前者是可以成功的但是因為不是從input端證出來每個satprove要跑很久，所以sim13.aag大概要跑幾十分鐘。但跑\_dfsList的話雖然速度很快，但我simulate到最後會進入無窮回圈，用了cout大法後發現是跑到後面有可能發生進去\_dfsList之內的每個gate存的FecGrp\* \_fecgrp都是0。想了很久我推測應該是在merge的過程中砍掉的gate的fanincone有一個grp的成員都在裡面，造成其實已經結束了但\_FecGrps.size()卻不是零的狀況。因為真的沒時間了不然其實應該是想得到解決方法的。

至於我的實作方法是先設定完satsolver後，最外層的迴圈就是if(\_FecGrps.size()不是0)，也就是裡面還有東西的話就要繼續sat、merge、sim。然後每個grp（從\_dfsList的gate中進去的）都拿第一個(base)跟其他去比，如果比完這個grp內都是UNSAT的話就代表可以進行merge，接著因為merge後\_dfsList會變，我會跳出去genDfsList()再進入迴圈繼續跑。而如果不是每個都UNSAT，就把SAT的反例收集起來，每64個就simulate()一次，同時genFec()，就會有新的FecGrp\*產生，而且會分得更開而有機會裡面再次prove會全部UNSAT就進行merge這樣一直下去直到沒有grp了。最後把initFec設成false，這樣接著cirsimulate之後就會重新初始化。

1. **results and analysis**

因為fraig沒有很完整只能跑sim01~sim06的測資，這邊就放cirsimulate在sim13.aag跟ref的效能比較結果，左邊是mine，右邊是ref。

