

學生姓名：劉力仁      學號：B04901068      信箱：[b04901068@ntu.edu.tw](mailto:b04901068@ntu.edu.tw)

## 一：CirGate 設計：

最初我只用一個 class 也就是 CirGate 做完 hw6, fanin,fanout 則是用 vector<CirGate\*> 指到相對應的 gate, 然而記憶體使用量有點大, (有一次上課老師有說不要亂用 vector 跟 string, 很吃記憶體) 所以聽完老師上課講解後決定重新改架構, 所以寫成了繼承的 class, fanin 改成用 array, pi 沒有 po 一個 aig 兩個, 記憶體使用量從原本 ref 的三倍下降成差不多。然而, 好景不常, 當我繼續實做 final project 的時候, 到 optimize 就出現了許多問題, 最主要的兩個問題：

第一：如果 fanin 不是用 vector 要自己維持 array 麻煩很多！

第二：如果存 pointer 可能會產生 memory leak ,存到不該存的東西....而且很難找到 ORZ

所以結論是我又把 fanin 改成 vector 形式而且是 vector<size\_t> 存 gateid 而已, 這樣到 GateList 去找才不容易 segmemtation fault。

然後因為這樣我重寫了兩次 read 跟一堆 function.....所以知道一個好的架構很重要 T\_T

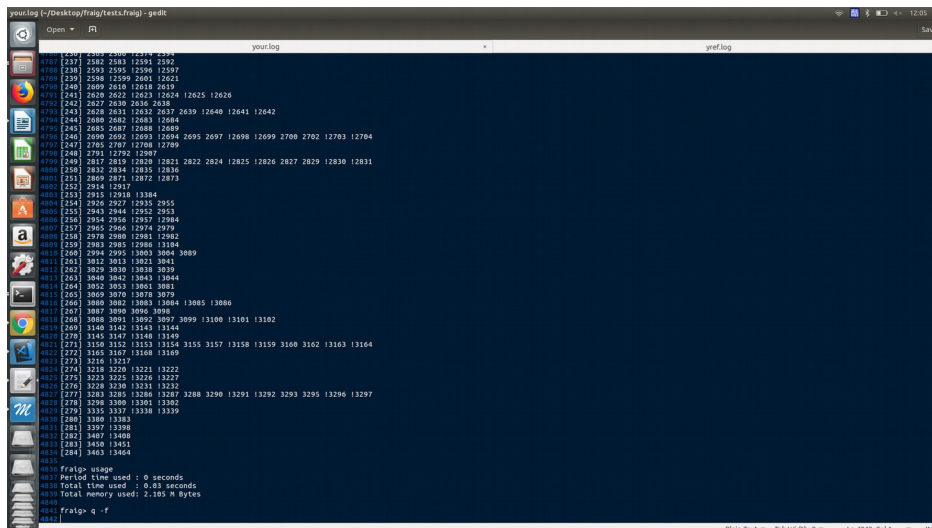
## 二：Simulation:

因為看不懂講義 Simulation 的 psuedo code, 所以就寫了一個很耗記憶體但邏輯比較簡單的方法。

也是用到 hash(hashset), 我是一次把所有 pattern 都讀進來, hash funtion 是數 1 的位元有幾個, \_numBuckets 是 pattern 數除二, 如果 1 的位元數大於 \_numBuckets, 就用兩倍的 \_numBuckets 扣掉 1 的位元數。這樣同一個 bucket 裡面才會有 IFEC。做完一次 hash 後就開始分 FECGroups。pattern 剛讀進來後, 每個 gate 會有一個 vector<size\_t> \_simValue 來存 simulation 的值, 而在同一個 bucket 裡的 gate 就可以抓出來比較, 第一層是跟這層 bucket 裡目前有的 FECGroups 比較, 如果有完全一樣或完全不一樣就加入, 如果都沒有就創建新的 FECGroup, 而一層 bucket 算完後把只有一個 member 的 FECGroup 刪掉, push\_back 進最大的 FECGroups。這樣把每層 bucket 都做完了後, 再排序 (我用 insertion sort) 就會得到正確的 FECGroups。

我後來知道老師的作法應該是一次 sim 一個 size\_t 然後不斷的把 FECGroups 分小, 這樣在記憶體上就不用每個 gate 存一個 vector<size\_t>, 明顯的節省空間, 至於速度的話以下有結果。

我使用 tests.fraig/do.fsim 因為正確性已經測驗過無誤了, 所以第一筆我就放 sim09.aag 的結果



```
your log C:/Desktop/fraig/Tests/fraig - gedit
your log
your log
1780 12301 12302 12303 12304 12305
1781 12307 12308 12309 12310 12311
1782 12312 12313 12314 12315 12316
1783 12317 12318 12319 12320 12321
1784 12322 12323 12324 12325 12326
1785 12327 12328 12329 12330 12331
1786 12332 12333 12334 12335 12336
1787 12337 12338 12339 12340 12341
1788 12342 12343 12344 12345 12346
1789 12347 12348 12349 12350 12351
1790 12352 12353 12354 12355 12356
1791 12357 12358 12359 12360 12361
1792 12362 12363 12364 12365 12366
1793 12367 12368 12369 12370 12371
1794 12372 12373 12374 12375 12376
1795 12377 12378 12379 12380 12381
1796 12382 12383 12384 12385 12386
1797 12387 12388 12389 12390 12391
1798 12392 12393 12394 12395 12396
1799 12397 12398 12399 12400 12401
1800 12402 12403 12404 12405 12406
1801 12407 12408 12409 12410 12411
1802 12412 12413 12414 12415 12416
1803 12417 12418 12419 12420 12421
1804 12422 12423 12424 12425 12426
1805 12427 12428 12429 12430 12431
1806 12432 12433 12434 12435 12436
1807 12437 12438 12439 12440 12441
1808 12442 12443 12444 12445 12446
1809 12447 12448 12449 12450 12451
1810 12452 12453 12454 12455 12456
1811 12457 12458 12459 12460 12461
1812 12462 12463 12464 12465 12466
1813 12467 12468 12469 12470 12471
1814 12472 12473 12474 12475 12476
1815 12477 12478 12479 12480 12481
1816 12482 12483 12484 12485 12486
1817 12487 12488 12489 12490 12491
1818 12492 12493 12494 12495 12496
1819 12497 12498 12499 12500 12501
1820 12502 12503 12504 12505 12506
1821 12507 12508 12509 12510 12511
1822 12512 12513 12514 12515 12516
1823 12517 12518 12519 12520 12521
1824 12522 12523 12524 12525 12526
1825 12527 12528 12529 12530 12531
1826 12532 12533 12534 12535 12536
1827 12537 12538 12539 12540 12541
1828 12542 12543 12544 12545 12546
1829 12547 12548 12549 12550 12551
1830 12552 12553 12554 12555 12556
1831 12557 12558 12559 12560 12561
1832 12562 12563 12564 12565 12566
1833 12567 12568 12569 12570 12571
1834 12572 12573 12574 12575 12576
1835 12577 12578 12579 12580 12581
1836 12582 12583 12584 12585 12586
1837 12587 12588 12589 12590 12591
1838 12592 12593 12594 12595 12596
1839 12597 12598 12599 12600 12601
1840 12602 12603 12604 12605 12606
1841 12607 12608 12609 12610 12611
1842 12612 12613 12614 12615 12616
1843 12617 12618 12619 12620 12621
1844 12622 12623 12624 12625 12626
1845 12627 12628 12629 12630 12631
1846 12632 12633 12634 12635 12636
1847 12637 12638 12639 12640 12641
1848 12642 12643 12644 12645 12646
1849 12647 12648 12649 12650 12651
1850 12652 12653 12654 12655 12656
1851 12657 12658 12659 12660 12661
1852 12662 12663 12664 12665 12666
1853 12667 12668 12669 12670 12671
1854 12672 12673 12674 12675 12676
1855 12677 12678 12679 12680 12681
1856 12682 12683 12684 12685 12686
1857 12687 12688 12689 12690 12691
1858 12692 12693 12694 12695 12696
1859 12697 12698 12699 12700 12701
1860 12702 12703 12704 12705 12706
1861 12707 12708 12709 12710 12711
1862 12712 12713 12714 12715 12716
1863 12717 12718 12719 12720 12721
1864 12722 12723 12724 12725 12726
1865 12727 12728 12729 12730 12731
1866 12732 12733 12734 12735 12736
1867 12737 12738 12739 12740 12741
1868 12742 12743 12744 12745 12746
1869 12747 12748 12749 12750 12751
1870 12752 12753 12754 12755 12756
1871 12757 12758 12759 12760 12761
1872 12762 12763 12764 12765 12766
1873 12767 12768 12769 12770 12771
1874 12772 12773 12774 12775 12776
1875 12777 12778 12779 12780 12781
1876 12782 12783 12784 12785 12786
1877 12787 12788 12789 12790 12791
1878 12792 12793 12794 12795 12796
1879 12797 12798 12799 12800 12801
1880 12802 12803 12804 12805 12806
1881 12807 12808 12809 12810 12811
1882 12812 12813 12814 12815 12816
1883 12817 12818 12819 12820 12821
1884 12822 12823 12824 12825 12826
1885 12827 12828 12829 12830 12831
1886 12832 12833 12834 12835 12836
1887 12837 12838 12839 12840 12841
1888 12842 12843 12844 12845 12846
1889 12847 12848 12849 12850 12851
1890 12852 12853 12854 12855 12856
1891 12857 12858 12859 12860 12861
1892 12862 12863 12864 12865 12866
1893 12867 12868 12869 12870 12871
1894 12872 12873 12874 12875 12876
1895 12877 12878 12879 12880 12881
1896 12882 12883 12884 12885 12886
1897 12887 12888 12889 12890 12891
1898 12892 12893 12894 12895 12896
1899 12897 12898 12899 12900 12901
1900 12902 12903 12904 12905 12906
1901 12907 12908 12909 12910 12911
1902 12912 12913 12914 12915 12916
1903 12917 12918 12919 12920 12921
1904 12922 12923 12924 12925 12926
1905 12927 12928 12929 12930 12931
1906 12932 12933 12934 12935 12936
1907 12937 12938 12939 12940 12941
1908 12942 12943 12944 12945 12946
1909 12947 12948 12949 12950 12951
1910 12952 12953 12954 12955 12956
1911 12957 12958 12959 12960 12961
1912 12962 12963 12964 12965 12966
1913 12967 12968 12969 12970 12971
1914 12972 12973 12974 12975 12976
1915 12977 12978 12979 12980 12981
1916 12982 12983 12984 12985 12986
1917 12987 12988 12989 12990 12991
1918 12992 12993 12994 12995 12996
1919 12997 12998 12999 13000 13001
1920 13002 13003 13004 13005 13006
1921 13007 13008 13009 13010 13011
1922 13012 13013 13014 13015 13016
1923 13017 13018 13019 13020 13021
1924 13022 13023 13024 13025 13026
1925 13027 13028 13029 13030 13031
1926 13032 13033 13034 13035 13036
1927 13037 13038 13039 13040 13041
1928 13042 13043 13044 13045 13046
1929 13047 13048 13049 13050 13051
1930 13052 13053 13054 13055 13056
1931 13057 13058 13059 13060 13061
1932 13062 13063 13064 13065 13066
1933 13067 13068 13069 13070 13071
1934 13072 13073 13074 13075 13076
1935 13077 13078 13079 13080 13081
1936 13082 13083 13084 13085 13086
1937 13087 13088 13089 13090 13091
1938 13092 13093 13094 13095 13096
1939 13097 13098 13099 13100 13101
1940 13102 13103 13104 13105 13106
1941 13107 13108 13109 13110 13111
1942 13112 13113 13114 13115 13116
1943 13117 13118 13119 13120 13121
1944 13122 13123 13124 13125 13126
1945 13127 13128 13129 13130 13131
1946 13132 13133 13134 13135 13136
1947 13137 13138 13139 13140 13141
1948 13142 13143 13144 13145 13146
1949 13147 13148 13149 13150 13151
1950 13152 13153 13154 13155 13156
1951 13157 13158 13159 13160 13161
1952 13162 13163 13164 13165 13166
1953 13167 13168 13169 13170 13171
1954 13172 13173 13174 13175 13176
1955 13177 13178 13179 13180 13181
1956 13182 13183 13184 13185 13186
1957 13187 13188 13189 13190 13191
1958 13192 13193 13194 13195 13196
1959 13197 13198 13199 13200 13201
1960 13202 13203 13204 13205 13206
1961 13207 13208 13209 13210 13211
1962 13212 13213 13214 13215 13216
1963 13217 13218 13219 13220 13221
1964 13222 13223 13224 13225 13226
1965 13227 13228 13229 13230 13231
1966 13232 13233 13234 13235 13236
1967 13237 13238 13239 13240 13241
1968 13242 13243 13244 13245 13246
1969 13247 13248 13249 13250 13251
1970 13252 13253 13254 13255 13256
1971 13257 13258 13259 13260 13261
1972 13262 13263 13264 13265 13266
1973 13267 13268 13269 13270 13271
1974 13272 13273 13274 13275 13276
1975 13277 13278 13279 13280 13281
1976 13282 13283 13284 13285 13286
1977 13287 13288 13289 13290 13291
1978 13292 13293 13294 13295 13296
1979 13297 13298 13299 13300 13301
1980 13302 13303 13304 13305 13306
1981 13307 13308 13309 13310 13311
1982 13312 13313 13314 13315 13316
1983 13317 13318 13319 13320 13321
1984 13322 13323 13324 13325 13326
1985 13327 13328 13329 13330 13331
1986 13332 13333 13334 13335 13336
1987 13337 13338 13339 13340 13341
1988 13342 13343 13344 13345 13346
1989 13347 13348 13349 13350 13351
1990 13352 13353 13354 13355 13356
1991 13357 13358 13359 13360 13361
1992 13362 13363 13364 13365 13366
1993 13367 13368 13369 13370 13371
1994 13372 13373 13374 13375 13376
1995 13377 13378 13379 13380 13381
1996 13382 13383 13384 13385 13386
1997 13387 13388 13389 13390 13391
1998 13392 13393 13394 13395 13396
1999 13397 13398 13399 13400 13401
2000 13402 13403 13404 13405 13406
2001 13407 13408 13409 13410 13411
2002 13412 13413 13414 13415 13416
2003 13417 13418 13419 13420 13421
2004 13422 13423 13424 13425 13426
2005 13427 13428 13429 13430 13431
2006 13432 13433 13434 13435 13436
2007 13437 13438 13439 13440 13441
2008 13442 13443 13444 13445 13446
2009 13447 13448 13449 13450 13451
2010 13452 13453 13454 13455 13456
2011 13457 13458 13459 13460 13461
2012 13462 13463 13464 13465 13466
2013 13467 13468 13469 13470 13471
2014 13472 13473 13474 13475 13476
2015 13477 13478 13479 13480 13481
2016 13482 13483 13484 13485 13486
2017 13487 13488 13489 13490 13491
2018 13492 13493 13494 13495 13496
2019 13497 13498 13499 13500 13501
2020 13502 13503 13504 13505 13506
2021 13507 13508 13509 13510 13511
2022 13512 13513 13514 13515 13516
2023 13517 13518 13519 13520 13521
2024 13522 13523 13524 13525 13526
2025 13527 13528 13529 13530 13531
2026 13532 13533 13534 13535 13536
2027 13537 13538 13539 13540 13541
2028 13542 13543 13544 13545 13546
2029 13547 13548 13549 13550 13551
2030 13552 13553 13554 13555 13556
2031 13557 13558 13559 13560 13561
2032 13562 13563 13564 13565 13566
2033 13567 13568 13569 13570 13571
2034 13572 13573 13574 13575 13576
2035 13577 13578 13579 13580 13581
2036 13582 13583 13584 13585 13586
2037 13587 13588 13589 13590 13591
2038 13592 13593 13594 13595 13596
2039 13597 13598 13599 13600 13601
2040 13602 13603 13604 13605 13606
2041 13607 13608 13609 13610 13611
2042 13612 13613 13614 13615 13616
2043 13617 13618 13619 13620 13621
2044 13622 13623 13624 13625 13626
2045 13627 13628 13629 13630 13631
2046 13632 13633 13634 13635 13636
2047 13637 13638 13639 13640 13641
2048 13642 13643 13644 13645 13646
2049 13647 13648 13649 13650 13651
2050 13652 13653 13654 13655 13656
2051 13657 13658 13659 13660 13661
2052 13662 13663 13664 13665 13666
2053 13667 13668 13669 13670 13671
2054 13672 13673 13674 13675 13676
2055 13677 13678 13679 13680 13681
2056 13682 13683 13684 13685 13686
2057 13687 13688 13689 13690 13691
2058 13692 13693 13694 13695 13696
2059 13697 13698 13699 13700 13701
2060 13702 13703 13704 13705 13706
2061 13707 13708 13709 13710 13711
2062 13712 13713 13714 13715 13716
2063 13717 13718 13719 13720 13721
2064 13722 13723 13724 13725 13726
2065 13727 13728 13729 13730 13731
2066 13732 13733 13734 13735 13736
2067 13737 13738 13739 13740 13741
2068 13742 13743 13744 13745 13746
2069 13747 13748 13749 13750 13751
2070 13752 13753 13754 13755 13756
2071 13757 13758 13759 13760 13761
2072 13762 13763 13764 13765 13766
2073 13767 13768 13769 13770 13771
2074 13772 13773 13774 13775 13776
2075 13777 13778 13779 13780 13781
2076 13782 13783 13784 13785 13786
2077 13787 13788 13789 13790 13791
2078 13792 13793 13794 13795 13796
2079 13797 13798 13799 13800 13801
2080 13802 13803 13804 13805 13806
2081 13807 13808 13809 13810 13811
2082 13812 13813 13814 13815 13816
2083 13817 13818 13819 13820 13821
2084 13822 13823 13824 13825 13826
2085 13827 13828 13829 13830 13831
2086 13832 13833 13834 13835 13836
2087 13837 13838 13839 13840 13841
2088 13842 13843 13844 13845 13846
2089 13847 13848 13849 13850 13851
2090 13852 13853 13854 13855 13856
2091 13857 13858 13859 13860 13861
2092 13862 13863 13864 13865 13866
2093 13867 13868 13869 13870 13871
2094 13872 13873 13874 13875 13876
2095 13877 13878 13879 13880 13881
2096 13882 13883 13884 13885 13886
2097 13887 13888 13889 13890 13891
2098 13892 13893 13894 13895 13896
2099 13897 13898 13899 1390
```

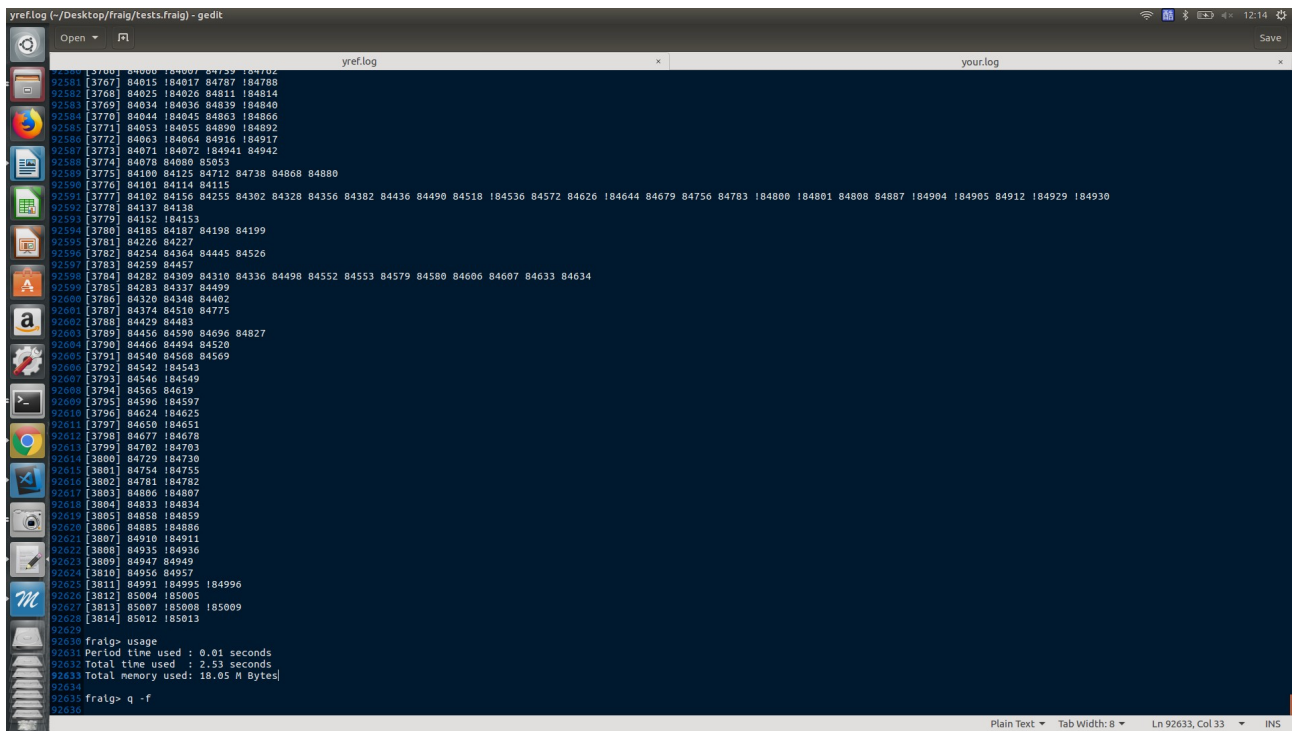
```
yref.log (-/Desktop/fraig/tests.fraig) - gedit
your.log
yref.log
4810 [230] 2305 2300 12374 2394
4810 [237] 2582 2583 12591 2592
4810 [238] 2593 2595 12596 12597
4810 [239] 2598 12599 2601 12621
4810 [240] 2609 2610 12618 2619
4810 [241] 2620 2622 12623 12624 12625 12626
4810 [242] 2627 2630 2636 2638
4810 [243] 2626 2631 2632 2637 2639 12640 12641 12642
4810 [244] 2688 2682 12683 12684
4810 [245] 2685 2687 12686 12689
4810 [246] 2688 2692 12693 12694 2695 2697 12698 12699 2700 2702 12703 12704
4810 [247] 2785 2787 12708 12709
4810 [248] 2791 12792 12907
4810 [249] 2817 2819 12820 12821 2822 2824 12825 12826 2827 2829 12830 12831
4810 [250] 2831 2834 12835 12836
4810 [251] 2869 2871 12872 12873
4810 [252] 2914 12917
4810 [253] 2915 12918 13384
4810 [254] 2926 2927 12935 2955
4810 [255] 2943 2944 12952 2953
4810 [256] 2954 2956 12957 12984
4810 [257] 2965 2966 12974 2979
4810 [258] 2978 2980 12981 12982
4810 [259] 2983 2985 12986 13104
4810 [260] 2994 2995 13003 3004 3089
4810 [261] 3012 3013 13021 3041
4810 [262] 3029 3030 13038 3039
4810 [263] 3040 3042 13043 13044
4810 [264] 3052 3053 13061 3081
4810 [265] 3069 3070 13078 3079
4810 [266] 3088 3082 13083 13084 13085 13086
4810 [267] 3087 3090 3096 3098
4810 [268] 3088 3091 13092 3097 3099 13100 13101 13102
4810 [269] 3140 3142 13143 13144
4810 [270] 3145 3147 13148 13149
4810 [271] 3150 3152 13153 13154 3155 3157 13158 13159 3160 3162 13163 13164
4810 [272] 3165 3167 13168 13169
4810 [273] 3216 13217
4810 [274] 3218 3220 13221 13222
4810 [275] 3223 3225 13226 13227
4810 [276] 3228 3230 13231 13232
4810 [277] 3283 3285 13286 13287 3288 3290 13291 13292 3293 3295 13296 13297
4810 [278] 3298 3300 13301 13302
4810 [279] 3335 3337 13338 13339
4810 [280] 3380 13383
4810 [281] 3397 13398
4810 [282] 3407 13408
4810 [283] 3450 13451
4810 [284] 3463 13464
4810
4810 fraig> usage
4810 Period time used : 0 seconds
4810 Total time used : 0.01 seconds
4810 Total memory used: 0.2656 M Bytes
4810
4810 fraig> q -f
4810
4810
```

0.01s+0.2656M 的記憶體。  
看目前來看速度差了三倍，記憶體約 7~8 倍（那這個差異是不 const? 還是正比 n 呢？）

### 測量 sim13.aag

```
your.log (-/Desktop/fraig/tests.fraig) - gedit
yref.log
your.log
92222 [3760] 84000 184007 84739 184702
92222 [3767] 84015 184017 84787 184788
92222 [3768] 84025 184026 84811 184814
92222 [3769] 84034 184036 84839 184840
92222 [3770] 84044 184045 84863 184866
92222 [3771] 84053 184055 84890 184892
92222 [3772] 84063 184064 84916 184917
92222 [3773] 84071 184072 184941 84942
92222 [3774] 84078 84080 85053
92222 [3775] 84108 84125 84712 84738 84868 84880
92222 [3776] 84101 84114 84115
92222 [3777] 84102 84156 84255 84302 84328 84356 84382 84436 84490 84518 184536 84572 84626 184644 84679 84756 84783 184800 184801 84808 84887 184904 184905 84912 184929 184930
92222 [3778] 84137 84138
92222 [3779] 84152 184153
92222 [3780] 84185 84187 84198 84199
92222 [3781] 84226 84227
92222 [3782] 84254 84364 84445 84526
92222 [3783] 84259 84457
92222 [3784] 84282 84309 84310 84336 84498 84552 84553 84579 84580 84606 84607 84633 84634
92222 [3785] 84283 84337 84489
92222 [3786] 84320 84348 84402
92222 [3787] 84374 84510 84775
92222 [3788] 84429 84483
92222 [3789] 84456 84500 84606 84827
92222 [3790] 84466 84494 84520
92222 [3791] 84540 84568 84569
92222 [3792] 84542 184543
92222 [3793] 84540 184549
92222 [3794] 84565 84619
92222 [3795] 84596 184597
92222 [3796] 84624 184625
92222 [3797] 84650 184651
92222 [3798] 84677 184678
92222 [3799] 84702 184703
92222 [3800] 84729 184730
92222 [3801] 84754 184755
92222 [3802] 84781 184782
92222 [3803] 84806 184807
92222 [3804] 84833 184834
92222 [3805] 84858 184859
92222 [3806] 84885 184886
92222 [3807] 84910 184911
92222 [3808] 84935 184936
92222 [3809] 84947 84949
92222 [3810] 84950 84957
92222 [3811] 84991 184995 184996
92222 [3812] 85004 185005
92222 [3813] 85007 185008 185009
92222 [3814] 85012 185013
92222
92222 fraig> usage
92222 Period time used : 0 seconds
92222 Total time used : 10.12 seconds
92222 Total memory used: 627.5 M Bytes
92222
92222 fraig> q -f
92222
92222
```

我花了 10.12s+627M 的記憶體



ref 則是 2.53s+18.05M

速度差約為四倍，記憶體使用量三十倍 xDD

的確是蠻驚人的數字~~

sim09 178 個 PI

sim13 3464 個 PI

差異二十倍，但我的速度是從三倍變四倍，所以我的方法跟老師的演算法級數應該沒有差，是 const 的差異。而記憶體使用量，我想最大的地方沒意外就是存\_simValue，因為我是一次行比對，所以會跟 pattern 數成正比，而實驗結果也的確如此。

### CIRSIM -R:

我是依照 PI 的數量做直覺的計算，因為 pattern 的組合是 exponential 比上 input 的數量，而跟 AIGgate 的數量沒關係，所以我用 if else 訂了簡單的公式（也是參考 pattern01~13 的 PI 數以及 pattern 數訂出的）

我在想一次把全部的 simValue 做 hash 然後分開有什麼好處？

如果\_numBucket 夠大，那每個\_bucket 裡面有的 gate 就會很少，就不會需要太多兩兩比較，然後創建 FECGroup 的問題，這樣分 FEC 應該會比較快，但如果要更快應該要做兩層 hash，也許我可以再定義第二個 hash function 去把同個\_buckets 裡的東西再丟一個 hash 去找，但這次限於時間，就先做到這裡了。

### 三：Fraig

很急的趕出來了簡單的版本，我也不是照 DFS 的順序去 merge 的，而是照 FECGroups 的順序去找。（我將 FECGroups 存在 CirMgr 裡，一個 Group 有 FECPair 的 gateid,invert 與否,以及首項的 sim 值。所以把所有 FEC 組合都算過了。就是把每個 FECGroup 裡頭的值兩兩丟進 SAT 驗證一次(除了已經確定可以跟別人 merge 的 gate 不用在驗證) 有的就 merge。簡單的電路圖跑出來沒問題，複雜的像是 sim09 跟 sim13 則是電路圖長的會跟 ref 差蠻多的。

以下為實測內容：

cirr sim09.aag

cirstrash

cirp -n

cirsim -f pattern.09

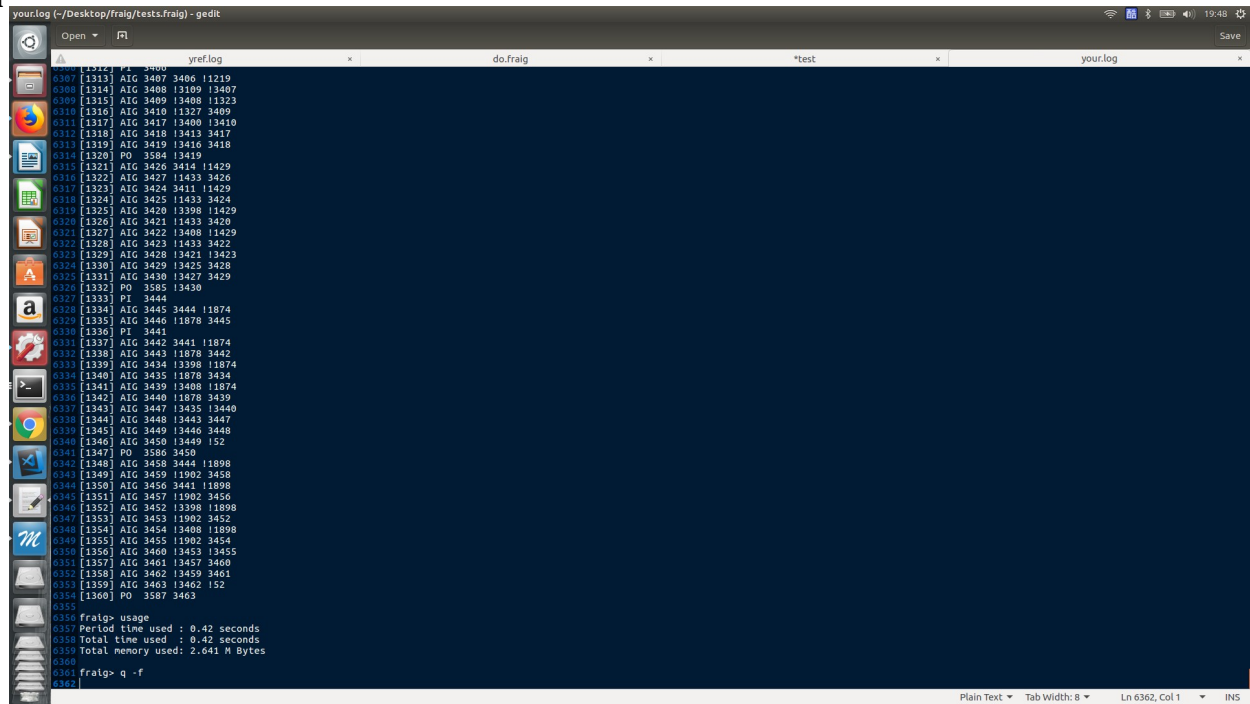
cirp -fec

cirfraig

cirp -n

usage

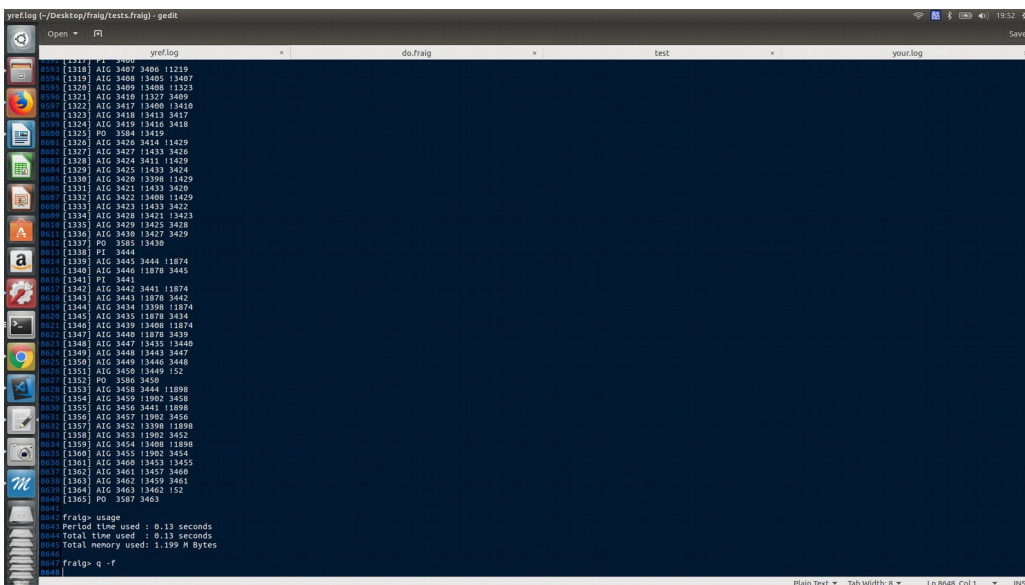
q -f



```
your.log (~/.Desktop/fraig/tests.fraig) - gedit
Open  x  yref.log  x  do.fraig  x  *test  x  your.log  x  Save

6307 [1312] AIG 3407 3406 11219
6308 [1314] AIG 3408 13109 13407
6309 [1315] AIG 3409 13408 11323
6310 [1316] AIG 3410 11327 3409
6311 [1317] AIG 3417 13400 13410
6312 [1318] AIG 3418 13413 3417
6313 [1319] AIG 3419 13416 3418
6314 [1320] PO 3584 13419
6315 [1321] AIG 3426 3414 11429
6316 [1322] AIG 3427 11433 3426
6317 [1323] AIG 3424 3411 11429
6318 [1324] AIG 3425 11433 3424
6319 [1325] AIG 3420 13398 11429
6320 [1326] AIG 3421 11433 3420
6321 [1327] AIG 3422 13408 11429
6322 [1328] AIG 3423 11433 3422
6323 [1329] AIG 3428 13421 13423
6324 [1330] AIG 3429 13425 3428
6325 [1331] AIG 3430 13427 3429
6326 [1332] PO 3585 13430
6327 [1333] PT 3444
6328 [1334] AIG 3445 3444 11874
6329 [1335] AIG 3446 11878 3445
6330 [1336] PT 3441
6331 [1337] AIG 3442 3441 11874
6332 [1338] AIG 3443 11878 3442
6333 [1339] AIG 3434 13398 11874
6334 [1340] AIG 3435 11878 3434
6335 [1341] AIG 3439 13408 11874
6336 [1342] AIG 3440 11878 3439
6337 [1343] AIG 3447 13435 13440
6338 [1344] AIG 3448 13443 3447
6339 [1345] AIG 3449 13446 3448
6340 [1346] AIG 3450 13449 152
6341 [1347] PO 3586 3450
6342 [1348] AIG 3458 3444 11898
6343 [1349] AIG 3459 11902 3458
6344 [1350] AIG 3456 3441 11898
6345 [1351] AIG 3457 11902 3456
6346 [1352] AIG 3452 13398 11898
6347 [1353] AIG 3453 11902 3452
6348 [1354] AIG 3454 13408 11898
6349 [1355] AIG 3455 11902 3454
6350 [1356] AIG 3460 13453 13455
6351 [1357] AIG 3461 13457 3460
6352 [1358] AIG 3462 13459 3461
6353 [1359] AIG 3463 13462 152
6354 [1360] PO 3587 3463
6355
6356 fraig> usage
6357 Period time used : 0.42 seconds
6358 Total time used : 0.42 seconds
6359 Total memory used: 2.641 M Bytes
6360
6361 fraig> q -f
6362
```

我的花了 0.42 秒+2.641M



```
yref.log (~/.Desktop/fraig/tests.fraig) - gedit
Open  x  yref.log  x  do.fraig  x  test  x  your.log  x  Save

6359 [1312] AIG 3407 3406 11219
6360 [1314] AIG 3408 13109 13407
6361 [1315] AIG 3409 13408 11323
6362 [1316] AIG 3410 11327 3409
6363 [1317] AIG 3417 13400 13410
6364 [1318] AIG 3418 13413 3417
6365 [1319] AIG 3419 13416 3418
6366 [1320] PO 3584 13419
6367 [1321] AIG 3426 3414 11429
6368 [1322] AIG 3427 11433 3426
6369 [1323] AIG 3424 3411 11429
6370 [1324] AIG 3425 11433 3424
6371 [1325] AIG 3420 13398 11429
6372 [1326] AIG 3421 11433 3420
6373 [1327] AIG 3422 13408 11429
6374 [1328] AIG 3423 11433 3422
6375 [1329] AIG 3428 13421 13423
6376 [1330] AIG 3429 13425 3428
6377 [1331] AIG 3430 13427 3429
6378 [1332] PO 3585 13430
6379 [1333] PT 3444
6380 [1334] AIG 3445 3444 11874
6381 [1335] AIG 3446 11878 3445
6382 [1336] PT 3441
6383 [1337] AIG 3442 3441 11874
6384 [1338] AIG 3443 11878 3442
6385 [1339] AIG 3434 13398 11874
6386 [1340] AIG 3435 11878 3434
6387 [1341] AIG 3439 13408 11874
6388 [1342] AIG 3440 11878 3439
6389 [1343] AIG 3447 13435 13440
6390 [1344] AIG 3448 13443 3447
6391 [1345] AIG 3449 13446 3448
6392 [1346] AIG 3450 13449 152
6393 [1347] PO 3586 3450
6394 [1348] AIG 3458 3444 11898
6395 [1349] AIG 3459 11902 3458
6396 [1350] AIG 3456 3441 11898
6397 [1351] AIG 3457 11902 3456
6398 [1352] AIG 3452 13398 11898
6399 [1353] AIG 3453 11902 3452
6400 [1354] AIG 3454 13408 11898
6401 [1355] AIG 3455 11902 3454
6402 [1356] AIG 3460 13453 13455
6403 [1357] AIG 3461 13457 3460
6404 [1358] AIG 3462 13459 3461
6405 [1359] AIG 3463 13462 152
6406 [1360] PO 3587 3463
6407
6408 fraig> usage
6409 Period time used : 0.13 seconds
6410 Total time used : 0.13 seconds
6411 Total memory used: 1.199 M Bytes
6412
6413 fraig> q -f
6414
```

ref 則花了 0.13s+1.199M

速度跟記憶體慢的比值都跟做 sim 差不多 ORZ 至於跑 sim13，我的演算法時間就爆炸了....  
不過從 sim09 可以看出，我從 FEC 找到可以 merge 的 gate 似乎比 ref 多！（ref cirp -n 有 1365 個，  
我只有 1360 個 XD 只能說這是  $n^2$  算 SAT 的唯一好出了吧。

總結：這次是在期末結束後才開始實做，optimize 寫完後只剩四天，simulation 又卡很久，幾乎沒有時間改良 fraig 覺得很可惜，但要怪也得怪我期末考前沒有先做，也應該說當初沒想好架構，改了架構兩次，幾個基本的 function 也改了好幾次，剛踏進 fraig 這有趣的地方，下課鐘聲卻響了，意猶未盡的感覺。真希望能再多個一兩天，想一些特別的方法來改進 fraig 的速度，而不是只是剛好做出功能而已。還有助教辛苦了==