

EE6094

CAD for VLSI Design

Final Project Report

CAD contest 2021

Problem F: Verilog Simulation Optimization via
Instruction Reduction

Team leader : 106501001 顏郁芩

Team member : 106501501 林妤蓓

106501506 吳唯甄

I. Abstract

為了要減少進行 Verilog Simulation 的時間，提高 simulator 的性能。主要是寫一個執行 Verilog-to-Verilog transformation 的程序，以盡量減少出於性能考慮的指令或分配。

因為直接指定一個位置之類的會比較耗費時間，所以在程式裡面盡量減少這個部分的指令就可以減少整體的 simulation 時間。

II. Problem Description

利用基本的 C++ 來撰寫程式，輸入是一個 Verilog 文件，開發一個程式來優化 Verilog 文件，包括正確性和 Verilog 文件中減少的連續賦值和向量位或部分選擇的數量，還有要注意一些在優化過程中不能違反的限制。

III. Program Structure

副程式：

- 儲存 tb(tb_save())

- 儲存 wire(wire_spilt())

- 儲存 assign 資料(assign_spilt())：

 - 一個輸入的 assign

 - 兩個輸入的邏輯閘

- 把一個輸入的 assign 的找出來，另外存成一個 vector(take_direct_out())

- 用數值取代掉所有單一 assign 的線(changeA())

- 刪掉已經做完的單一 assign 指令(delete_direct())

- 邏輯運算合併(logic_combine())

- 檢查還有沒有沒合併的邏輯運算(change_done_or_not())

- 把~的部分取代掉(change_logic())

- 把輸出有 out 的找出來存(find_out())

- 把可以合併的 out 找出來並儲存(teaming())

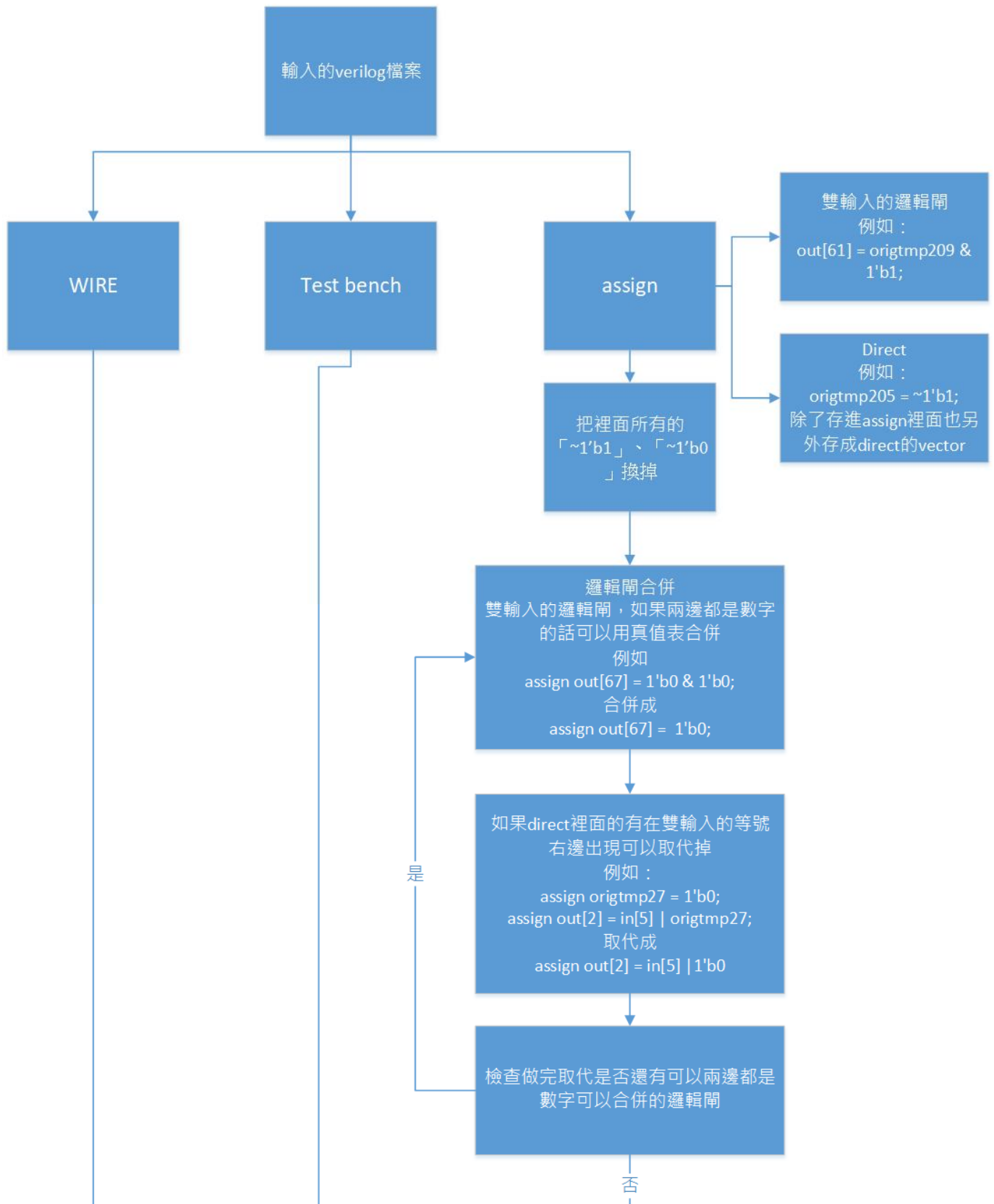
檢查該組合的 out 是不是已經儲存起來了(find_if_save())

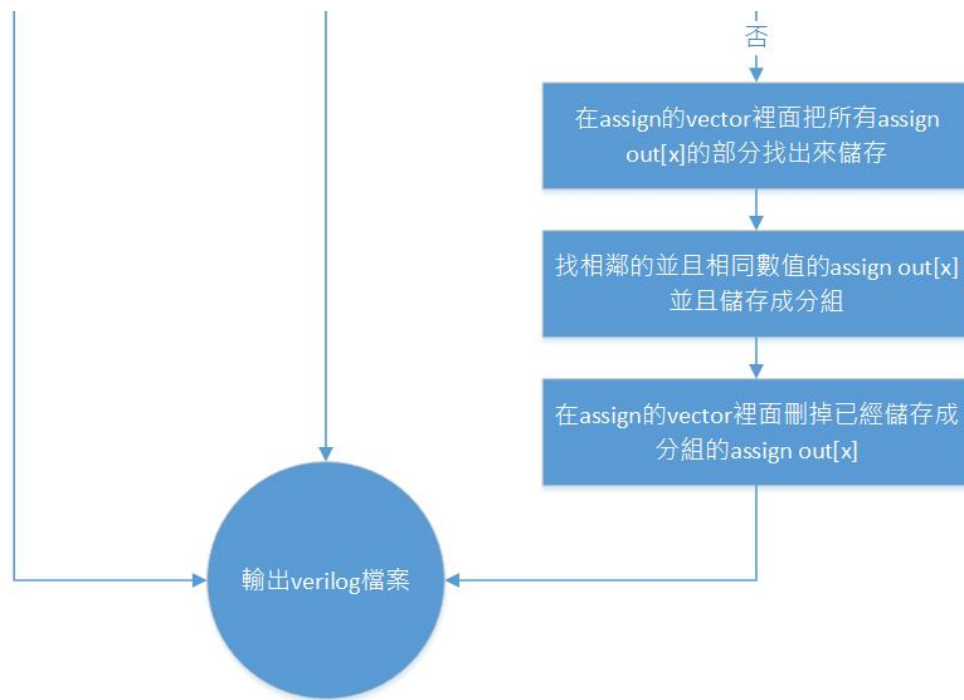
迭代刪除已經合併的 out 的 assign(del_out())

主程式：

- 1.在讀取的時候判斷是哪一種種類的輸入(wire、assign 或是 tb 部分)，呼叫對應副程式儲存
- 2.先把程式裡面所有的「~1'b1」、「~1'b0」換掉
- 3.合併邏輯閘之後，把現有所有的單一 assign 指令找出來，把所有的取代掉，之後把已經處理完的 assign 的指令刪除掉。
4. 檢查是否已經沒有可以合併的邏輯閘，有的話重複步驟 3

5.檢查可以合併的 out 部分(條件：相鄰而且數值相同)，合併之後把已經合併完的 assign out[x]刪掉





IV. Data Structures / Algorithms Used

我們整個程式是利用 vector 搭配 struct 來儲存資料，個別的 struct 結構如下：

主要儲存 assign 的 struct 是這樣：

```
typedef struct {
    string input1;
    string input2;
    string access;
    string output;
}ASSIGN;
```

有兩個輸入的話就依順序存進去 input1 和 input2 裡面，然後 access 放兩個輸入之間的關係(例如 and 或是 or 等等)，只有一個輸入的話只存放在 input1 裡面，access 放 direct。

Wire 還有 testbench 的儲存方式如下：

```
typedef struct {
    string name;
}WIRE;

typedef struct {
    vector <string>code;
}TESTBENCH;
```

分別把 direct 的 assign 指令(例如：assign origtmp188 = 1'b0;)還有輸出是 out 的單輸入指令(例如：assign out[15] = 1'b1;)另外儲存一份起來做處理。

```
typedef struct {
    string name;
    string data;
}DIRECT;
```

```
typedef struct {
    string out_name;
    string data;
    int num;
}OUT;
```

檢查 OUT 的 vector 裡面的內容然後把連續的相同的數值的輸出部分找出來，存在以下這樣的 struct vector 裡面

```
typedef struct {
    int first;
    int last;
    string data;
}TEAM;
```

V. How to Execute

Compile

因為有利用到 C++11 裡面的指令所以原始的 cpp 要利用 C++11 的編譯器來編譯

```
g++ -std=c++0x cadb0068.cpp cadb0068
```

Use

將要轉換的 original.v 和.exe 放在同一個資料夾裡面

./cadb0068 原始檔案名稱.v 新輸出的檔名.v

(ex:./cadb0068 original.v optimized.v)

checker

1. /bin/tcsh
2. source /project/cad/cad50/synopsys/CIC/vcs.cshrc
3. tcsh
4. vcs -sverilog 原始檔案名稱.v -o 輸出執行檔的名稱
(ex:vcs -sverilog optimized.v -o optimized)
5. ./輸出執行檔的名稱
(ex:./optimized)
6. diff simorig.txt result.txt

因為原始的 original.v 裡面的 tb 限制了輸出的結果命名會是 result.txt，所以是拿 result.txt 和大會提供的 testbench 中的 simorig.txt 做比較，如果沒有跑出任何輸出表示兩個檔案相等。

VI. Difficulty Encountered

1. 最一開始先遇到要怎麼把 verilog 程式碼裡面的一些邏輯閘轉換成 0101 的形式，還有資料存取該用什麼資料結構等等的問題。
2. 邏輯閘的化簡：將可以合併的邏輯合併起來
3. 在 alpha version 繳交的時候最主要就是工作站上的使用比較有問題，因為對於指令的操作不熟悉，導致驗證過程中一直出現 error，就沒辦法做驗證這個步驟：最後有解決
4. 做 Alpha version 的時候遇到消除掉已經取代過的直接 assign 指令會出現錯誤的狀況：因為測資 2 和 3 有我們沒有注意到的跟測資 1 不一樣的指令。
5. 要重複做幾次才能結束取代交換這個步驟：檢查程序。
6. 如何合併 output 的部分變成多 bit assign。

VII. Experimental Results

alpha version 的時候我們完成的只有：

1. 把 wire、assign 還有 tb 分開儲存
2. 把單一輸入的 assign 部分找出來取代掉
3. 還有簡單的邏輯閘化簡

本來還有做到要把已經做完的單一輸入的 assign 刪除掉，但是找不出錯誤只好暫時先放著。

從那之後到現在的進度，主要是把當時繳出去的程式碼做修正。

1. 針對當初沒有完成的刪除部分找出原因並修正
2. 對於沒有完整寫完的邏輯閘合併部分做完善
3. 加上自動檢查系統判斷是否要繼續做化簡與合併。
4. 合併連續的有相同數值的 out 部分
5. 增加 xformtmp

對於 alpha version 沒有解決掉的問題，為什麼把做完的單一輸入的 assign 刪除掉會出錯這一點，後來仔細研究了一下 2 和 3 兩筆測資之後發現兩者裡面有類似這樣的程式碼：

```
assign out[78] = ~origtmp220; assign out[39] = origtmp197 ^ origtmp197;
```

因為沒有把這兩者的可能性考慮進去所以導致取代的時候不完全，並誤刪了需要的資料，導致結果出錯。

最後利用工作站測試三筆測資之後得到以下的結果，可以看到在執行 diff simorig.txt result.txt 的時候並沒有出現其他的訊息，表示我們所變換出來的 verilog 檔案在邏輯上是與原本的相同的。

```
[cadb0068@cad22 ~]$ ./tc1
Chronologic VCS simulator copyright 1991-2012
Contains Synopsys proprietary information.
Compiler version G-2012.09-SP1; Runtime version G-2012.09-SP1; Jun 28 21:12 2021
data = [10110100101001001011]
results = [1000011001]
VCS Simulation Report
Time: 1
CPU Time: 0.380 seconds; Data structure size: 0.0Mb
Mon Jun 28 21:12:28 2021
[cadb0068@cad22 ~]$ diff simorig.txt results.txt
[cadb0068@cad22 ~]$

[cadb0068@cad22 ~]$ ./tc2
Chronologic VCS simulator copyright 1991-2012
Contains Synopsys proprietary information.
Compiler version G-2012.09-SP1; Runtime version G-2012.09-SP1; Jun 28 21:14 2021
data = [11110001011010111110010000000001111100000010110000]
results = [100101010101001011101001011011]
VCS Simulation Report
Time: 1
CPU Time: 0.390 seconds; Data structure size: 0.0Mb
Mon Jun 28 21:14:30 2021
[cadb0068@cad22 ~]$ diff simorig.txt results.txt
[cadb0068@cad22 ~]$

[cadb0068@cad22 ~]$ ./tc3
Chronologic VCS simulator copyright 1991-2012
Contains Synopsys proprietary information.
Compiler version G-2012.09-SP1; Runtime version G-2012.09-SP1; Jun 28 21:18 2021
data = [111000010000110101000100000000101001111111101111111111110000001001000000100100000010000
000000010000111111111110110000100111100000100000011111111001]
results = [10110101101100101010011001001000001011000010010000110011000111100110101001001010]
VCS Simulation Report
Time: 1
CPU Time: 0.410 seconds; Data structure size: 0.0Mb
Mon Jun 28 21:18:10 2021
[cadb0068@cad22 ~]$ diff simorig.txt results.txt
[cadb0068@cad22 ~]$
```

VIII. Alpha 繳交證明以及一點 reverse



Dear contestants,

Thanks for joining the 2021 CAD contest .
Topic Chairs already complete your alpha test result evaluation.
Please refer to the test report as follows:

Score	testcase1	testcase2	testcase3	testcase4	testcase5	testcase6	Total Score
cadb0068	0	0	0	0	0	0	0
Notes Compile error Pass: Allowed for Beta							

Best Regards,

2021 CAD Contest
cad.contest.iccad@gmail.com

收到這封 mail 之後我們討論之後覺得可能發生的問題是因為 README 裡面指令有寫錯所以導致 compile 出現 error 。

IX. Reference

1. <https://ieeexplore.ieee.org/document/5577890>
2. https://onlinecourses.nptel.ac.in/noc20_cs63/preview
3. <https://ieeexplore.ieee.org/document/724551>
4. <https://www.synopsys.com/verification/simulation/vcs.html>