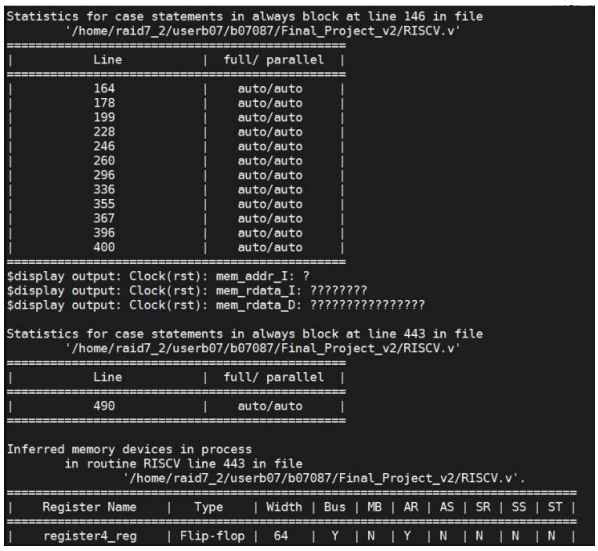
Final Project Report

Group: 19

Member: B07901093 蘇磐洋、B07901087 蔡沛洹

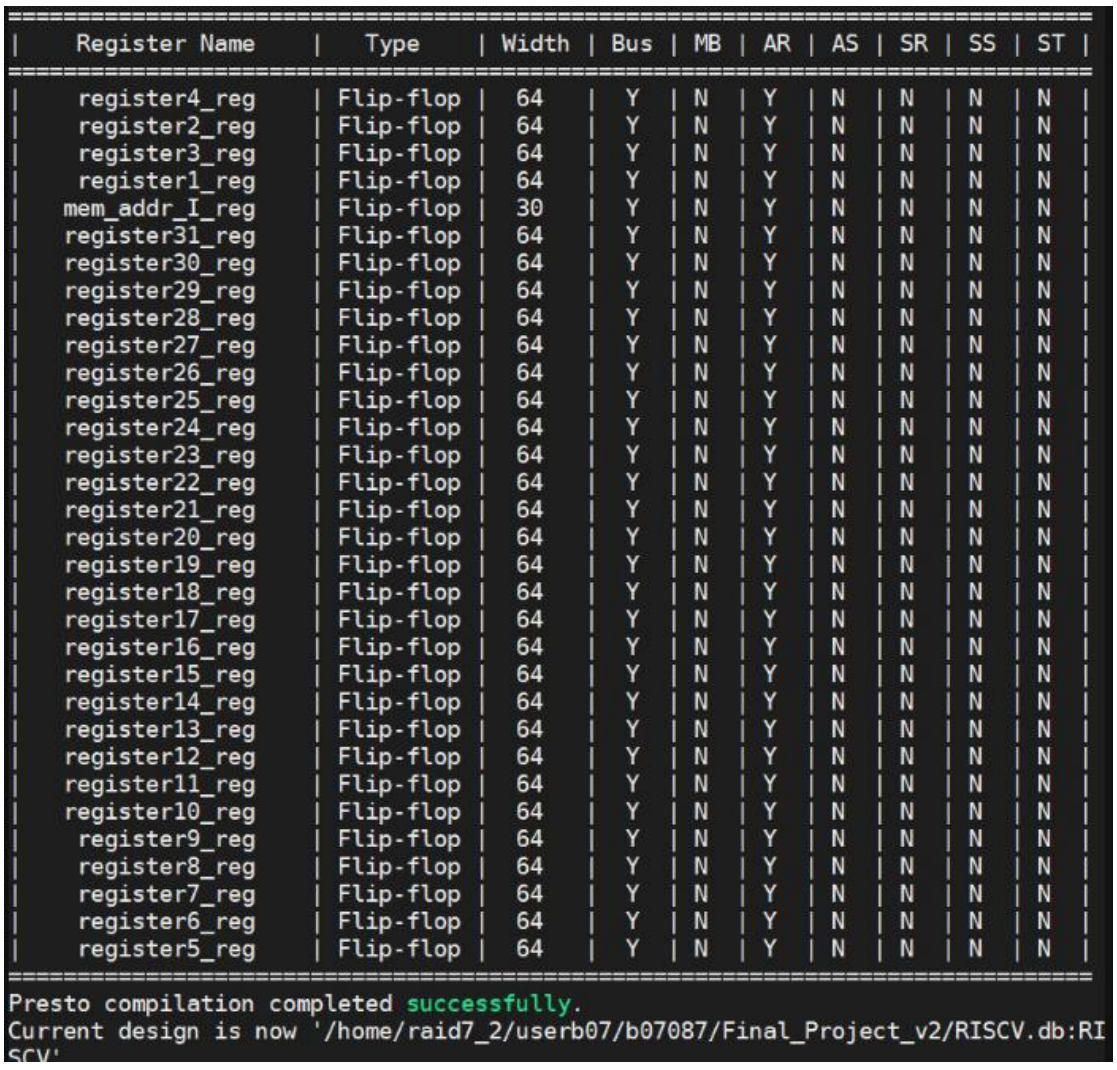
1. No latch



```
Statistics for case statements in always block at line 146 in file
        '/home/raid7_2/userb07/b07087/Final_Project_v2/RISCV.v'
===========================================
|           Line            |  full/ parallel  |
===========================================
|           164             |    auto/auto     |
|           178             |    auto/auto     |
|           199             |    auto/auto     |
|           228             |    auto/auto     |
|           246             |    auto/auto     |
|           260             |    auto/auto     |
|           296             |    auto/auto     |
|           336             |    auto/auto     |
|           355             |    auto/auto     |
|           367             |    auto/auto     |
|           396             |    auto/auto     |
|           400             |    auto/auto     |
===========================================
$display output: Clock(rst): mem_addr_I: ?
$display output: Clock(rst): mem_rdata_I: ????????
$display output: Clock(rst): mem_rdata_D: ????????????????

Statistics for case statements in always block at line 443 in file
        '/home/raid7_2/userb07/b07087/Final_Project_v2/RISCV.v'
===========================================
|           Line            |  full/ parallel  |
===========================================
|           490             |    auto/auto     |
===========================================

Inferred memory devices in process
        in routine RISCV line 443 in file
                '/home/raid7_2/userb07/b07087/Final_Project_v2/RISCV.v'.
=============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
=============================================================================
|    register4_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
```



```
=============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
=============================================================================
|    register4_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|    register2_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|    register3_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|    register1_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   mem_addr_I_reg    | Flip-flop |  30   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register31_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register30_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register29_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register28_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register27_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register26_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register25_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register24_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register23_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register22_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register21_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register20_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register19_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register18_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register17_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register16_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register15_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register14_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register13_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register12_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register11_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|   register10_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|    register9_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|    register8_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|    register7_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|    register6_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|    register5_reg    | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
=============================================================================
Presto compilation completed successfully.
Current design is now '/home/raid7_2/userb07/b07087/Final_Project_v2/RISCV.db:RI
SCV'
```

2. Timing report

```
clock CLK (rise edge)                    5.00       5.00
clock network delay (ideal)              0.50       5.50
clock uncertainty                       -0.10       5.40
mem_addr_I_reg_22_/CK (DFFRX4)           0.00       5.40 r
library setup time                      -0.04       5.36
data required time                                  5.36
-----------------------------------------------------------
data required time                                  5.36
data arrival time                                  -5.36
-----------------------------------------------------------
slack (MET)                                         0.00
```

3. Area report

```
Combinational area:          146702.885790
Buf/Inv area:                 22247.821702
Noncombinational area:        65360.082207
Macro/Black Box area:             0.000000
Net Interconnect area:      1898922.956390

Total cell area:             212062.967997
Total area:                 2110985.924387
```

4. Architecture and analysis

In general, our architecture is similar to that of the textbook. First, we read the instruction. Second, we give values for the control signals based on the instructions. Then, based on the control signals, we decide what operations the ALU should perform. Finally, we can write the data into the registers or the memory.

We adopt the following techniques to achieve better performance. First, we use case instead of if-else statements because cases use MUX while if-else statements use other gates. Second, we use wires instead of registers whenever possible, which may help the whole circuit run faster. Third, we combine the control signals, which at first are composed of 8 registers, into 1 register, which help lower the area. Finally, we remove some redundant parts.

5. Work distribution

蘇磐洋：基本架構、跑過 tb2 及 tb3、優化、寫報告

蔡沛洹：細部處理、跑過 tb1、優化、寫報告