CA FINAL PROJECT REPORT

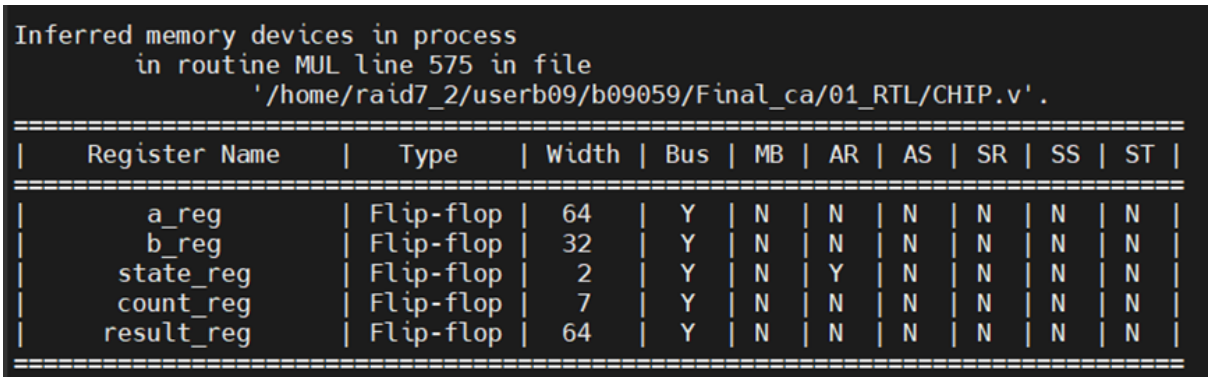
B09901059 彭亞綺

B09901144 李晴暄

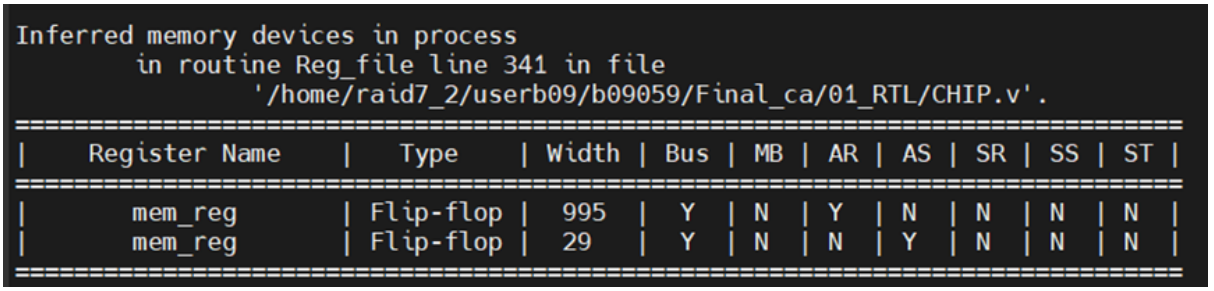
- Execution cycle

Instruction Set	Execution cycle w/o cache	Execution cycle w/ cache	Speedup
I0	75	95	0.79
I1	688	268	2.57
I2	201	181	1.11
I3	525	381	1.38

- Register table



```
Inferred memory devices in process
        in routine MUL line 575 in file
                '/home/raid7_2/userb09/b09059/Final_ca/01_RTL/CHIP.v'.
================================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
================================================================================
|        a_reg        | Flip-flop |  64   |  Y  | N  | N  | N  | N  | N  | N  |
|        b_reg        | Flip-flop |  32   |  Y  | N  | N  | N  | N  | N  | N  |
|      state_reg      | Flip-flop |   2   |  Y  | N  | Y  | N  | N  | N  | N  |
|      count_reg      | Flip-flop |   7   |  Y  | N  | N  | N  | N  | N  | N  |
|     result_reg      | Flip-flop |  64   |  Y  | N  | N  | N  | N  | N  | N  |
================================================================================
```



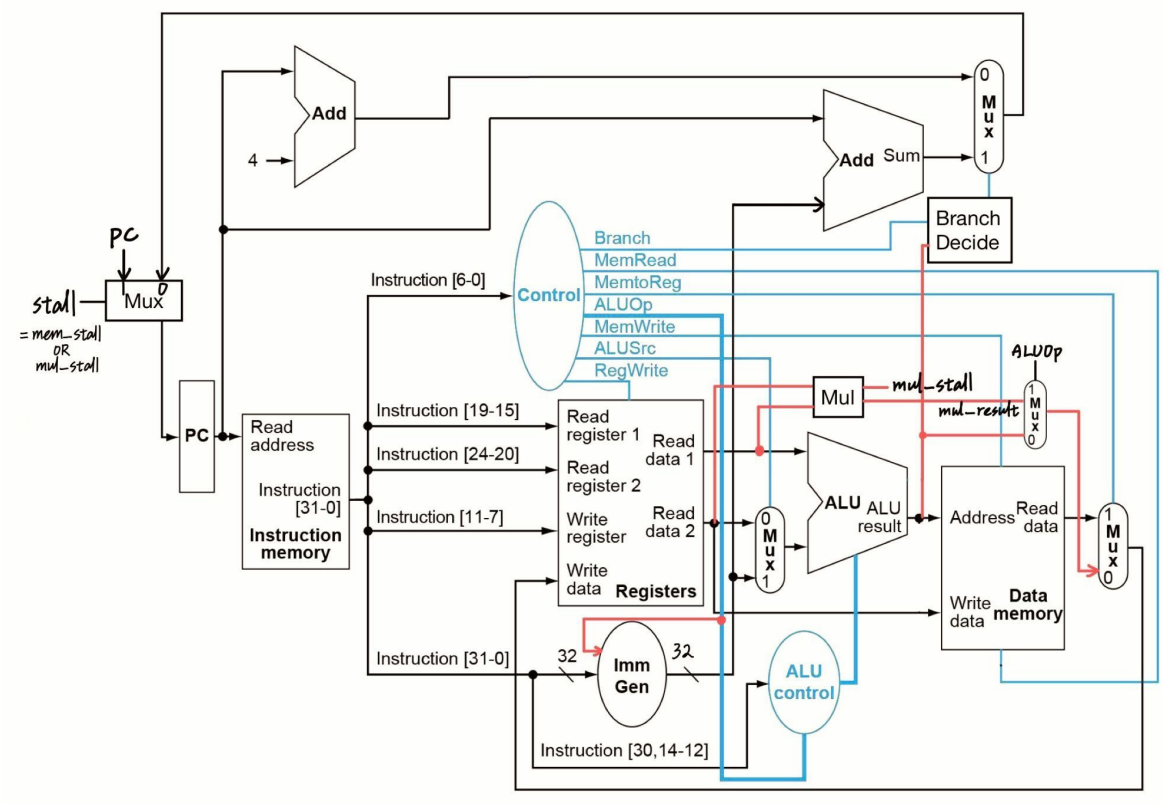
```
Inferred memory devices in process
        in routine Reg_file line 341 in file
                '/home/raid7_2/userb09/b09059/Final_ca/01_RTL/CHIP.v'.
================================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
================================================================================
|       mem_reg       | Flip-flop |  995  |  Y  | N  | Y  | N  | N  | N  | N  |
|       mem_reg       | Flip-flop |  29   |  Y  | N  | N  | Y  | N  | N  | N  |
================================================================================
```

```
Inferred memory devices in process
        in routine CHIP line 250 in file
                '/home/raid7_2/userb09/b09059/Final_ca/01_RTL/CHIP.v'.
================================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
================================================================================
|       PC_reg        | Flip-flop |   31  |  Y  | N  | Y  | N  | N  | N  | N  |
|       PC_reg        | Flip-flop |    1  |  N  | N  | N  | Y  | N  | N  | N  |
================================================================================
```

Cache:

```
Inferred memory devices in process
        in routine Cache line 479 in file
                '/home/raid7_2/userb09/b09059/Final_ca/01_RTL/CHIP.v'.
================================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
================================================================================
| cache_mem_addr_reg  | Flip-flop |  512  |  Y  | N  | Y  | N  | N  | N  | N  |
| cache_mem_data_reg  | Flip-flop |  512  |  Y  | N  | Y  | N  | N  | N  | N  |
|      state_reg      | Flip-flop |    3  |  Y  | N  | Y  | N  | N  | N  | N  |
|      valid_reg      | Flip-flop |   16  |  Y  | N  | Y  | N  | N  | N  | N  |
================================================================================
```

- CPU architecture



- Design of jal , jalr , auipc

  auipc : 將PC+imm 寫入rd, 繼續讀取下一個instruction

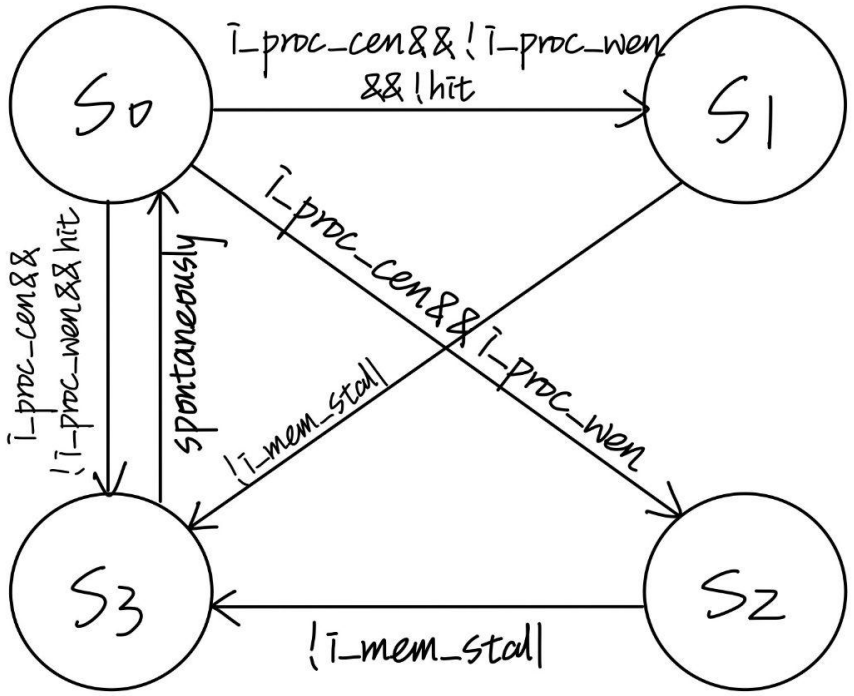
  jal:PC跳到PC+imm, 將下一個PC寫入rd

  jalr:下一個PC設為rs1+imm

```
5'd0:begin//auipc
    next_PC = PC + 4;
    r_wdata = PC + imm;
end
5'd1:begin //jal
    r_wdata = PC + 4;
    next_PC = PC +imm;
end
5'd2:begin //jalr
    next_PC = rdata1 +imm;
end
```

- How you handle multi-cycle instructions (mul)
  使用HW2寫的來修改,
  MUL module
  inputs: clock signal,reset signal,two input data,i_vaild
  outputs: mul_result,mul_stall
  為了handle multi-cycle instructions需要mul_stall，當運算還未做完時，將mul_stall訊號拉至1，使PC停止增加，直到mul運算完成再將mul_result傳出,mul_stall拉回0，PC繼續增加。

- Briefly describe your cache architecture
  implement method: direct mapped, 16 blocks
  1 word/block



- Describe how your cache improves time performance
  I0因為沒有重複access同一個memory address, 但是做了cache後每次access memory會多需要一個cycle來判斷是否hit, 所以I0的execution cycle上升。但其他的測資的execution cycle都有降低。

- Distribution table

  彭亞綺:

  Control module

  ALU control module

  ALU module

  MUL module

  李晴暄

  ImmGen

  BranchDecide