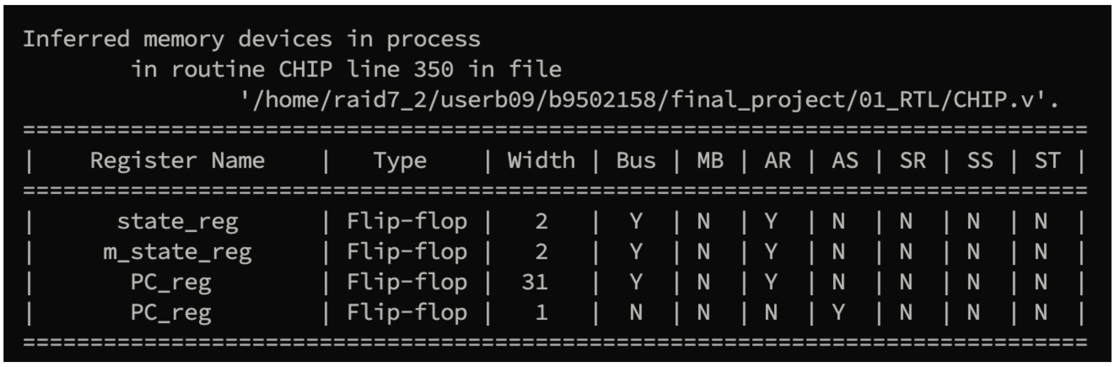
Computer Architecture Final Report

Group 45

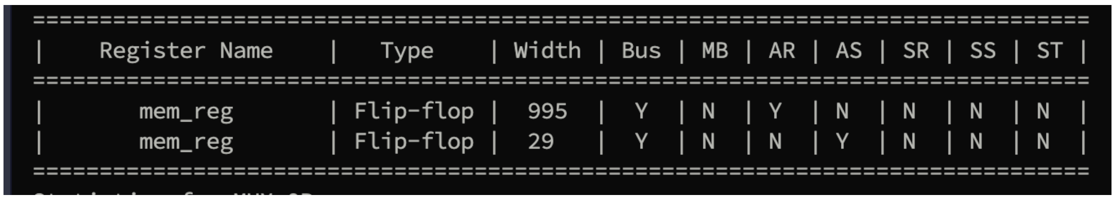
B09502158 詹宜昇

B10901180 鄭炘棠

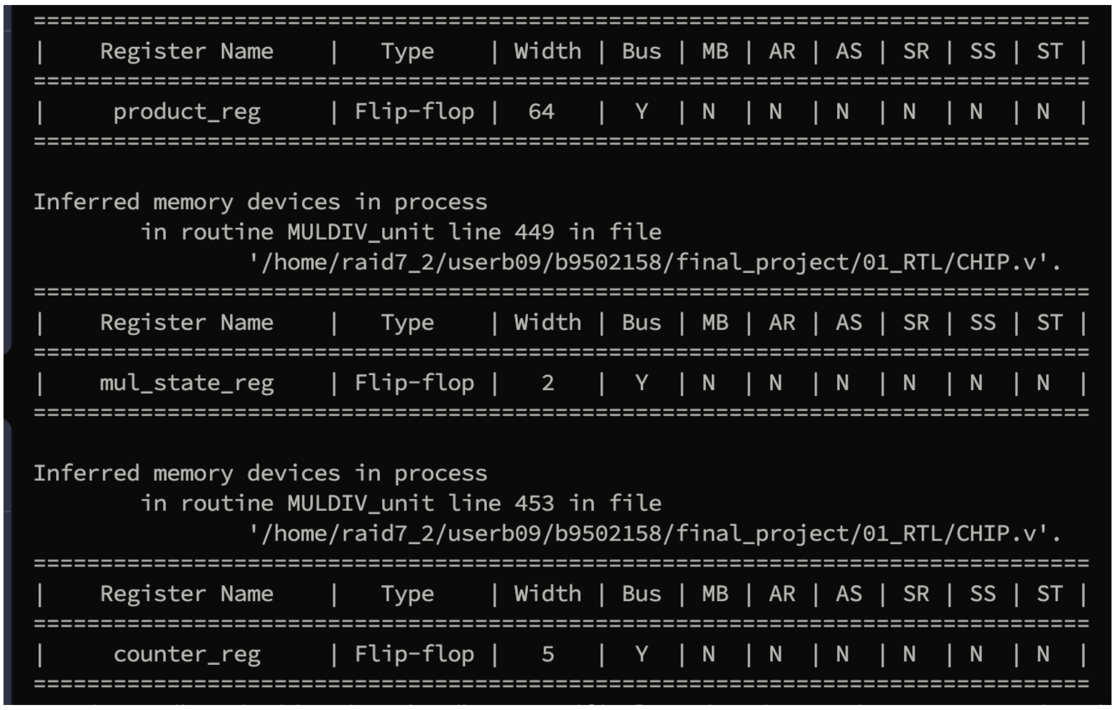
Snapshot the “Register table” in Design Compiler



```
Inferred memory devices in process
        in routine CHIP line 350 in file
                '/home/raid7_2/userb09/b9502158/final_project/01_RTL/CHIP.v'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|      state_reg      | Flip-flop |   2   |  Y  | N  | Y  | N  | N  | N  | N  |
|     m_state_reg     | Flip-flop |   2   |  Y  | N  | Y  | N  | N  | N  | N  |
|       PC_reg        | Flip-flop |  31   |  Y  | N  | Y  | N  | N  | N  | N  |
|       PC_reg        | Flip-flop |   1   |  N  | N  | N  | Y  | N  | N  | N  |
===============================================================================
```



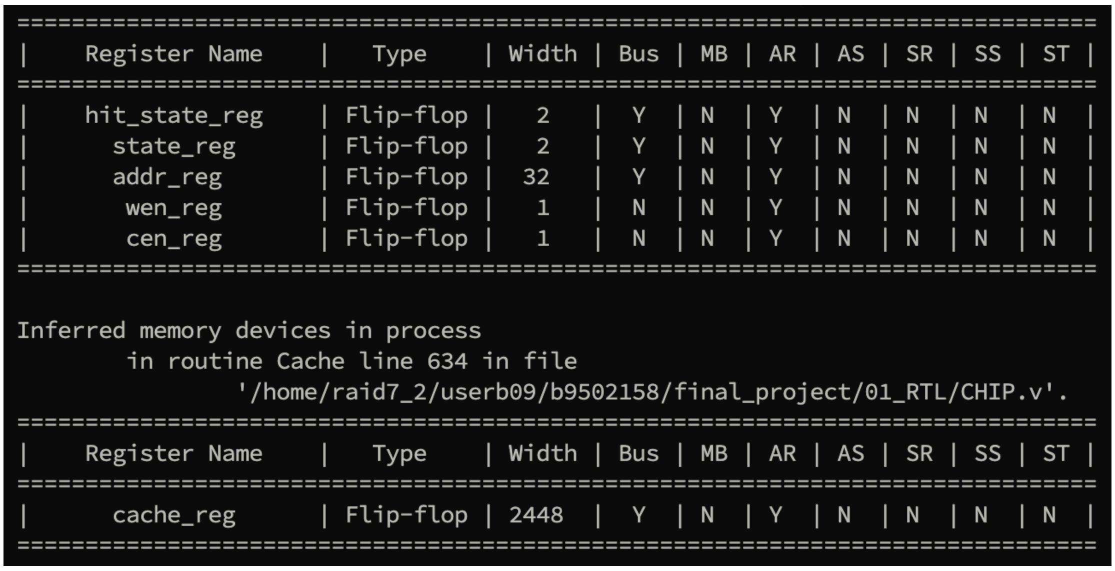
```
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|       mem_reg       | Flip-flop |  995  |  Y  | N  | Y  | N  | N  | N  | N  |
|       mem_reg       | Flip-flop |  29   |  Y  | N  | N  | Y  | N  | N  | N  |
===============================================================================
```



```
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|     product_reg     | Flip-flop |  64   |  Y  | N  | N  | N  | N  | N  | N  |
===============================================================================

Inferred memory devices in process
        in routine MULDIV_unit line 449 in file
                '/home/raid7_2/userb09/b9502158/final_project/01_RTL/CHIP.v'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|    mul_state_reg    | Flip-flop |   2   |  Y  | N  | N  | N  | N  | N  | N  |
===============================================================================

Inferred memory devices in process
        in routine MULDIV_unit line 453 in file
                '/home/raid7_2/userb09/b9502158/final_project/01_RTL/CHIP.v'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|     counter_reg     | Flip-flop |   5   |  Y  | N  | N  | N  | N  | N  | N  |
===============================================================================
```

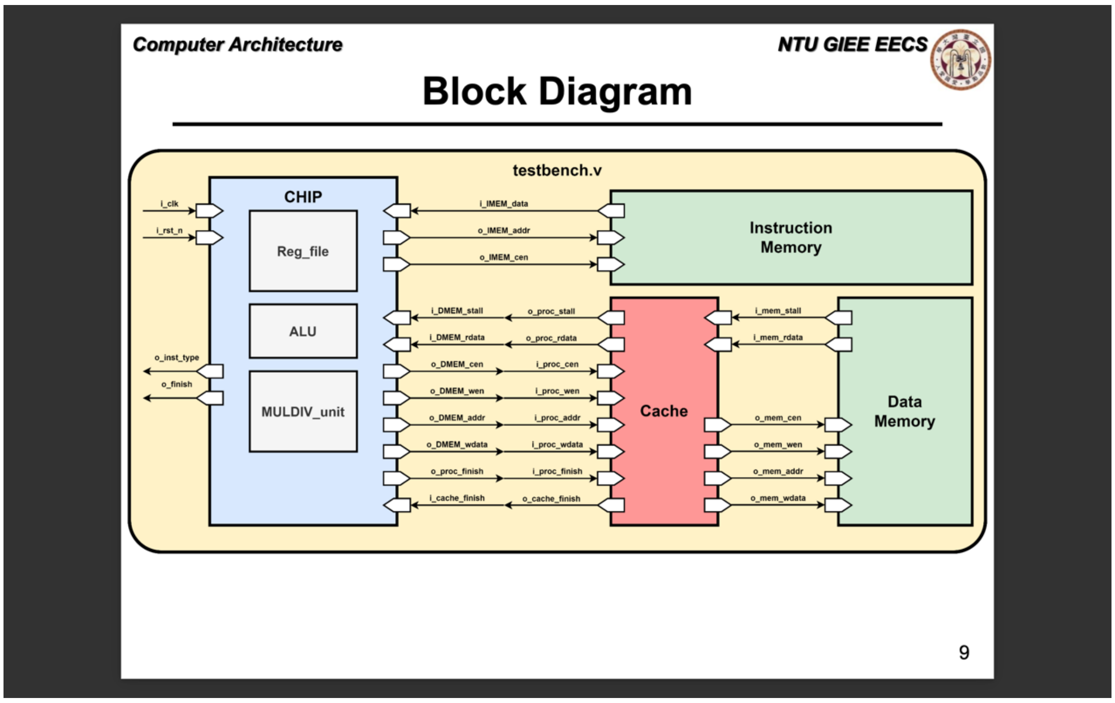


```
==================================================================================
|    Register Name     |    Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
==================================================================================
|    hit_state_reg     | Flip-flop |   2   |  Y  | N  | Y  | N  | N  | N  | N  |
|      state_reg       | Flip-flop |   2   |  Y  | N  | Y  | N  | N  | N  | N  |
|     addr_reg         | Flip-flop |  32   |  Y  | N  | Y  | N  | N  | N  | N  |
|      wen_reg         | Flip-flop |   1   |  N  | N  | Y  | N  | N  | N  | N  |
|      cen_reg         | Flip-flop |   1   |  N  | N  | Y  | N  | N  | N  | N  |
==================================================================================

Inferred memory devices in process
        in routine Cache line 634 in file
                '/home/raid7_2/userb09/b9502158/final_project/01_RTL/CHIP.v'.
==================================================================================
|    Register Name     |    Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
==================================================================================
|      cache_reg       | Flip-flop | 2448  |  Y  | N  | Y  | N  | N  | N  | N  |
==================================================================================
```

Work description

1. Draw the block diagram of your CPU architecture



Our CPU implementation is directly inspired by the block diagram given in the slides. However, we do not implement ALU submodule. We first read the instructions from instruction memory then decode it as shown in Instruction Set List. Later we perform different types of arithmetic operation on data and address read from Reg_file given by the instruction's opcode, function 3 and function 7. We also define finite state machine for load, store instructions to determine when to stall in order to fetch from memory.

2. Describe how you design the data path of instructions not referred in the lecture

slides (jal, jalr, auipc, ...)
由於我們的實作方式為 decode instruction 後從 Reg_file 讀出要進行操作的資料直接去進行處理因此我們只需要算出 rd, rs1, rs2, next_PC 以及這些 address 要存取什麼即可
jal: rd 存原本的 PC+4, next_PC 為 PC + imm[20:0]
jalr: jalr 與 jal 的處理方式類似，差別在於 next_PC 為 rs1_data 和 imm[11:0] 的 signed 相加
auipc: load unsigned imm 的 31:20 bits 再加到 PC 上作為 rd。

3. Describe how you handle multi-cycle instructions (mul, div ...)
  在 mul-unit 裡，當接收到 valid 為 1 的訊號時，藉由將 counter 從 0 數到 31，做 32-cycles 的運算，運算完之後，輸出乘法完的結果，並把 ready 的輸出設為 1。以上接收 valid、運算乘法、輸出 ready 都使用 FSM 來控制。
  在 CPU 裡，當接收到 MUL 指令後，使用 FSM，先將 valid 設為 1，接著將 valid 一直設為 0，等到接收到 ready 為 1 時，將乘法完的結果存到 rd。

4. Describe your observation:
  一開始實作 LW、SW 時，沒有考慮到要 stall 導致沒辦法從 data memory 取得資料。Stall 的操作是模擬現實中，從 data memory 拿資料是需要時間的。所以我們在 CPU 設定，當 i_DMEM_stall 為 0 時，我們才把拿到的資料存到 rd，並更新 next_pc。

Cache Design

Performance:

Instruction Set	Without Cache	With cache	Speedup
I0	77	49	1.57
I1	450	364	1.24
I2	410	360	1.14
I3	1324	692	1.91

Address:

tag	index	Word offset	Byte offset
24	4	2	2

cache architecture:

valid	tag	data
1 bit	24 bits	128 bits

Number of blocks = 2^4 = 16

Total size = (1+24+128)*16 = 2448 bits

Work Distribution

詹宜昇：instructions except mul

鄭炘棠：mul, cache