- Register table

```
'/home/raid7_2/userb09/b09045/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  |
==========================================================================
Warning:  /home/raid7_2/userb09/b09045/CHIP.v:239: signed to unsigned conversion oc
R-318)
Warning:  /home/raid7_2/userb09/b09045/CHIP.v:246: signed to unsigned conversion oc
R-318)

Inferred memory devices in process
        in routine reg_file line 242 in file
                '/home/raid7_2/userb09/b09045/CHIP.v'.
==========================================================================
|    Register Name    |    Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
==========================================================================
|       mem_reg       | Flip-flop |  994  |  Y  | N  | Y  | N  | N  | N  | N  |
|       mem_reg       | Flip-flop |  30   |  Y  | N  | N  | Y  | N  | N  | N  |
==========================================================================
```

```
==========================================================================
|    Register Name    |    Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
==========================================================================
|     alu_in_reg      | Flip-flop |  32   |  Y  | N  | Y  | N  | N  | N  | N  |
|      shreg_reg      | Flip-flop |  64   |  Y  | N  | Y  | N  | N  | N  | N  |
|      state_reg      | Flip-flop |   3   |  Y  | N  | Y  | N  | N  | N  | N  |
|     counter_reg     | Flip-flop |   5   |  Y  | N  | Y  | N  | N  | N  | N  |
==========================================================================
```

- CPU architecture

I create several submodule :
Control: to handle the dataflow of each submodule
ALU_control: decode the instruction and extract the part that handle ALU
Imm_gen: to extract the immediate from the instruction
ALU: combine mulDiv of HW2 and several other fundamental operation

For instructions not referred in the lecture slides, I use mem_to_reg and pc_ctrl to directly assign the value to rd_data and PC_nxt respectively (as below)

```
always @(*) begin
    case(mem_to_reg)
        `MEM2REG_PC_PLUS_4: rd_data = PC_plus_4;
        `MEM2REG_ALU: rd_data = alu_result;
        `MEM2REG_MEM: rd_data = mem_rdata_D;
        `MEM2REG_PC_PLUS_IMM: rd_data = PC + extended_imm;
    endcase
end

always @(*) begin
    if(alu_ready)begin
        case(pc_ctrl)
            `PCCTRL_PC_PLUS_IMM: PC_nxt = (is_branch && !alu_zero) ?  PC_plus_4 :
            `PCCTRL_RS1_PLUS_IMM: PC_nxt = alu_result;
            `PCCTRL_PC_PLUS_4: PC_nxt = PC_plus_4;
            default : PC_nxt = PC ;
        endcase
    end
    else PC_nxt = PC ;
end
```

While both rd_data and PC_nxt are generated through Control submodule from opcode of instruction

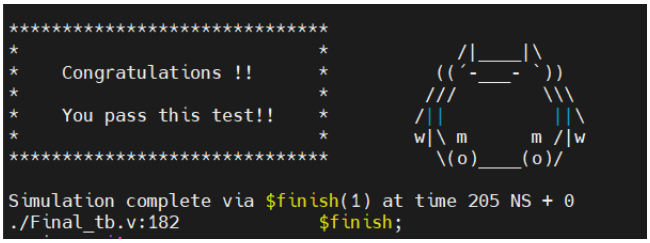
- Multicycle

```
assign alu_ready = (alu_ctrl == `MUL || alu_ctrl == `DIV) ? ready : 1;
assign out = alu_ready ? alu_result : 0;
assign out_zero = (alu_ctrl == `SUB) ? (alu_result == 0) : 0;

assign valid = (alu_ctrl == `MUL || alu_ctrl == `DIV);

assign mode = (alu_ctrl == `MUL) ? 0 : 1;
```

If alu_ctrl imply multiply or division operation then activate mulDiv by switch valid to 1, then alu_ready = 1 if mulDiv complete multicycle operation (ready =1, then assign to alu_ready)

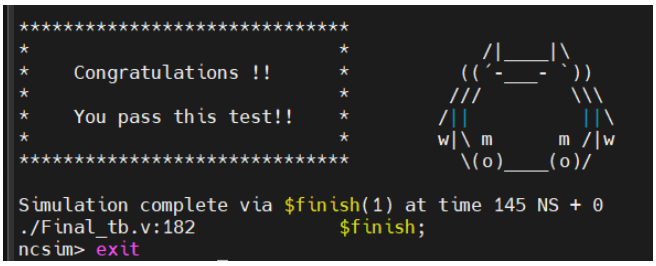
- Simulation time

Leaf: a = 3, b = 9, c = 5, d = 17



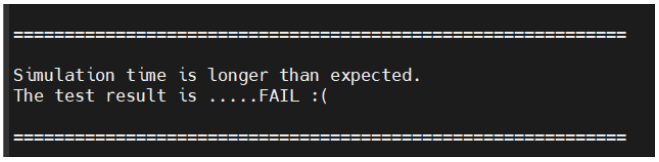
→205NS

Perm: n = 8, r = 5



→145NS

Bonus:



→QQ 我真的不知道問題出在哪拉，我覺得可能是 bonus.s 裡面 la 指令導致 decode 出不來東西?助教幫幫我，希望這樣能部分給 bonus 分(但是我 bonus.s 裡面的 todo 有寫出來，CHIP.v 裡也有延伸到 bonus 會出現的指令)

Observation: Perm take less time than Leaf

分工表: 吳名洋 100%