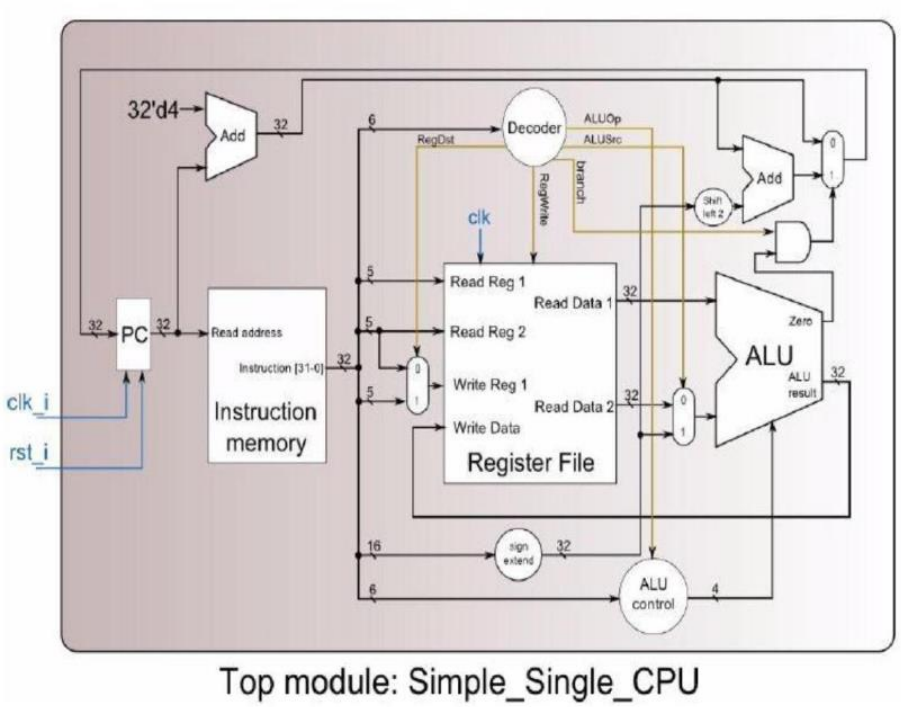
Computer Organization Lab2

Name: 梁詠晴

ID:109550134

Architecture diagrams:



Top module: Simple_Single_CPU

結構連線:

Program Counter PC:

Input : clk_i, rst_i, mux_pc_o

Output : pc_out_o

Adder Adder1:

Input : 32'd4, pc_out_o

Ouput : adder1_o

Instr_Memory IM :

Input : pc_out_o

Ouput : instr_o

MUX_2to1 Mux_Write_Reg

Input : instr_o[20:16], instr_o[15:11, RegDst_o

Ouput : mux_write_o

Reg_File RF :

Input : clk_i, rst_i, instr_o[25:21], instr_o[20:16], mux_write_o, alu_result, RegWrite_o

Ouput : rs_data, rd_data

Decoder Decoder :

Input : instr_o[31:26]

Ouput : RegWrite_o, ALU_op_o, ALUSrc_o, RegDst_o, Branch_o

ALU_Ctrl AC :

Input : instr_o[5:0], ALU_op_o

Ouput : ALUCtrl_o

Sign_Extend SE :

Input : instr_o[15:0]

Ouput : se_o

MUX_2to1 Mux_ALUSrc :

Input : rd_data, se_o, ALUSrc_o

Ouput : mux_alu_o

ALU ALU :

Input : rs_data, mux_alu_o, ALUCtrl_o

Ouput : alu_result, zero

Adder Adder2 :

Input : adder1_o, sl2

Ouput : adder2_o

Shift_Left_Two_32 Shifter :

Input : se_o

Ouput : sl2

MUX_2to1 Mux_PC_Source :

Input : adder1_o, adder2_o, (zero & Branch_o)

Ouput : mux_pc_o

Hardware module analysis:

PC： 表示程式執行的位置，每回合加 4

Adder： 將 2 個 32bit 的 source 相加

IM： 讀 pc 的位置，並依照 I-format、R-rormat 不同輸出相對應 address 的指令

MUX_2to1： 根據輸入條件決定要輸出哪一個 input

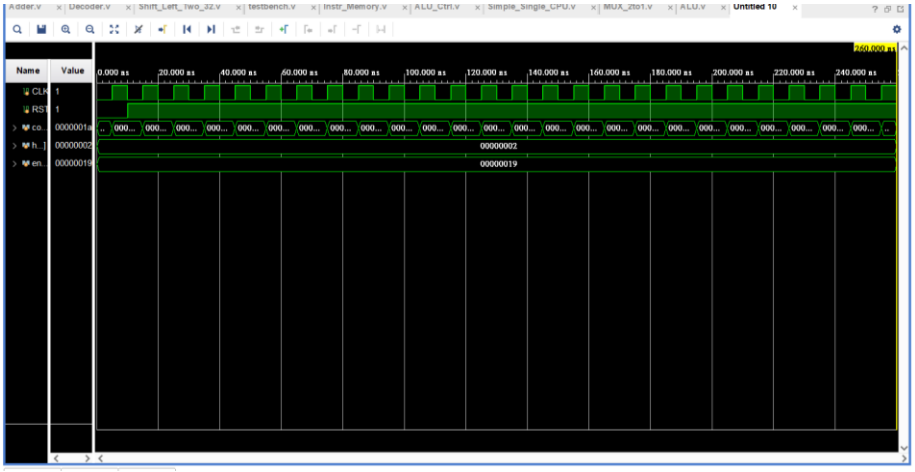
Decoder： 根據 opcode 決定指令需不需要 RegWrite_o, ALU_op_o, ALUSrc_o, RegDst_o, Branch_o

ALU_Ctrl： 根據 decoder 輸出決定 ALU 要做甚麼運算

ALU： 根據 ALU ctrl 對兩個輸入運算

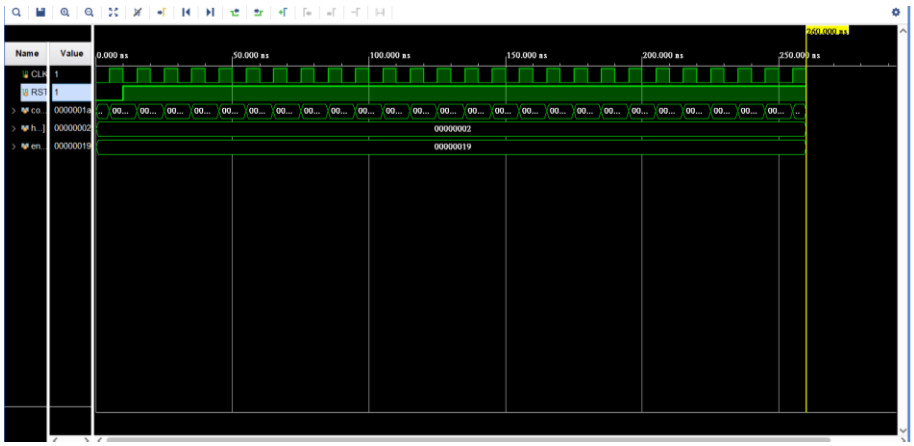
Finished part:

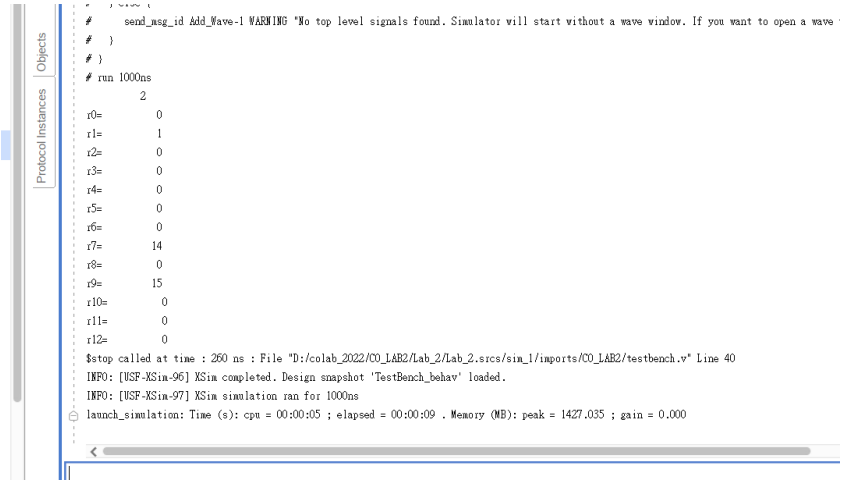
Test data 1:




```
#    }
# }
# run 1000ns
           2
r0=            0
r1=           10
r2=            4
r3=            0
r4=            0
r5=            6
r6=            0
r7=            0
r8=            0
r9=            0
r10=           0
r11=           0
r12=           0
$stop called at time : 260 ns : File "D:/colab_2022/CO_LAB2/Lab_2/Lab_2.srcs/sim_1/imports/CO_LAB2/testbench.v" Line 40
INFO: [USF-XSim-96] XSim completed. Design snapshot 'TestBench_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:04 ; elapsed = 00:00:20 . Memory (MB): peak = 1427.035 ; gain = 0.000
```

Test data 2:





Problems you met and solutions:

在運算時答案數值錯誤，發現是因為 shift left 2 bit 在進行切割的時候寫錯，變成保留左 30bit+00 而非 右 30bit+00。更正為正確答案後就得到正確結果了:
assign data_o = {data_i[29:0] , 2'b00};

Summary:

這次的 lab 中大量使用到了 module 之間的連接，透過這樣的做法可以更明確的觀察各個部分之間訊息傳遞的關係，清楚的分層也可以讓程式更好閱讀。