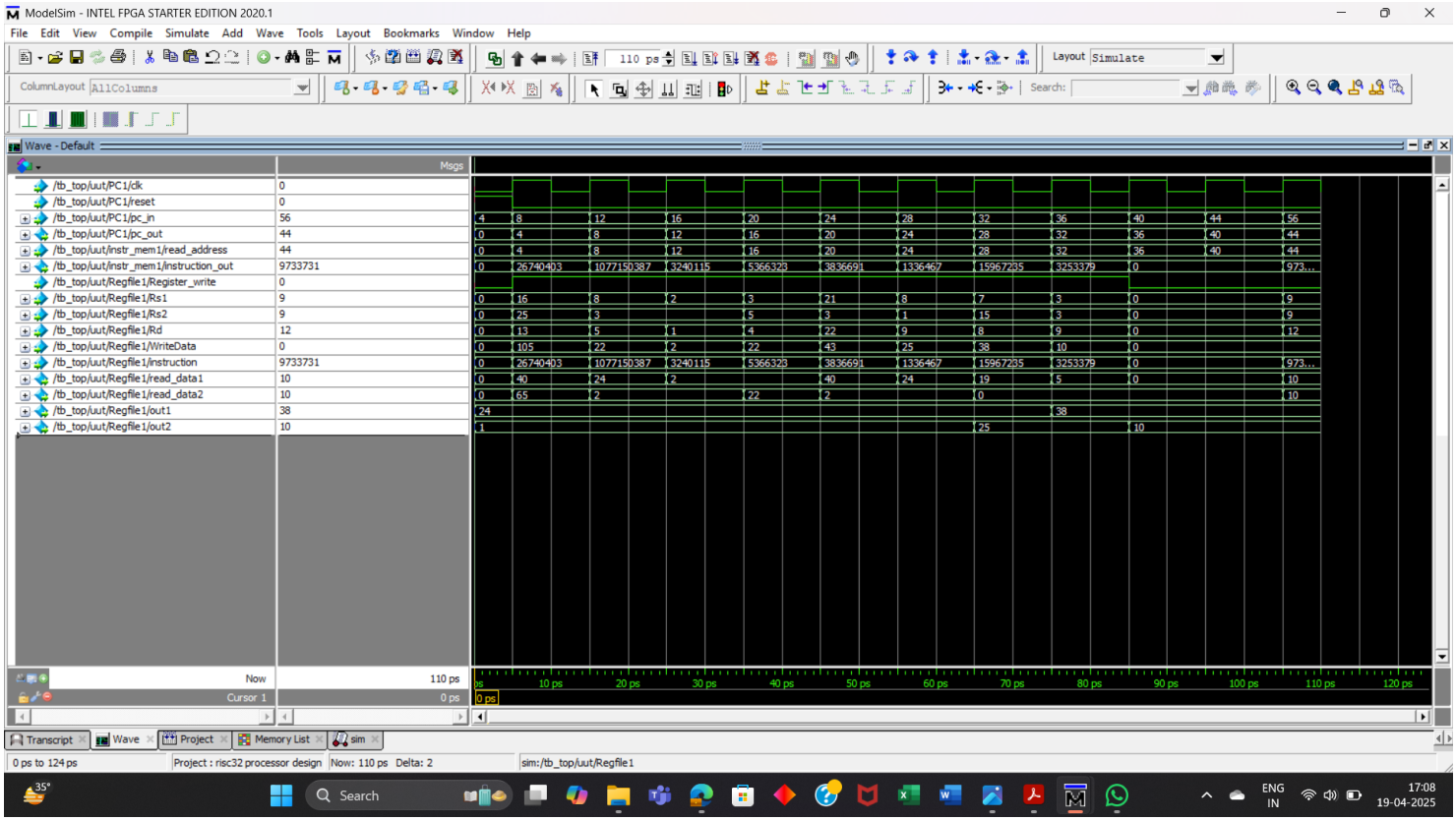
ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Wave - Default
Msgs
/tb_top/uut/PC1/clk
/tb_top/uut/PC1/reset
/tb_top/uut/PC1/pc_in
/tb_top/uut/PC1/pc_out
/tb_top/uut/instr_mem1/read_address
/tb_top/uut/instr_mem1/instruction_out
/tb_top/uut/Regfile1/Register_write
/tb_top/uut/Regfile1/Rs1
/tb_top/uut/Regfile1/Rs2
/tb_top/uut/Regfile1/Rd
/tb_top/uut/Regfile1/WriteData
/tb_top/uut/Regfile1/instruction
/tb_top/uut/Regfile1/read_data1
/tb_top/uut/Regfile1/read_data2
/tb_top/uut/Regfile1/out1
/tb_top/uut/Regfile1/out2
Now
Cursor 1
110 ps
0 ps
Transcript
Wave
Project
Memory List
sim
0 ps to 124 ps
Project : risc32 processor design
Now: 110 ps Delta: 2
sim:/tb_top/uut/Regfile1
Search
ENG
IN
17:08
19-04-2025

Explaination of the results:-

1st instruction is no operation so nothing is performed at first clock edge

2) add x13,x16,x25 content of register[16]=40(see read data1) and content of register[25]=65(see read data2) is getting added and stored in the register[13] =105(see write data)

3)sub x5,x8,x3; content of register[3] is getting subtracted from register[8] and getting stored in the register[5]so 24-2=22

4) and x1,x2,x3; content of register[3](2) is anded with content of the register[2] (2) and we get result as 2 see write data

5) or x4,x3,x5; or is done between register[3] and register[5] and result is stored in register[4] so we get (2|22=22) ;

6) addi x22,x21,3 3 is the immedate value that is being added to the register[21] (40) and result is stored in register[22](43);

7) ori x9,x8,1 register[8](24) is done bitwise or with 1 so we get register[9] as 25(see write data)

8) in load and store instructions i have not used immgen instead of that i am dir in the register file if my instruction is of type load or store it will update the read data1 and read data2 on the basis of which my result will be stored lw x8,15(x7) it means that content register[8] will be loaded with the content of memory at addres(content of reister[7](4)+15) means content of memory at address 19(38) will be stored in reister[8] so 38 is stored as you can see in next cycle out1 has become 38

9)branch instruction beq x9,x9,12 if (x9==x9)then jump to the current address +12 so my cyrrent pcin=44 in next cycle it becomes 56

Challenges i faced

initially i was using k++ in the for loop which is valid in system verilog but not but is not valid in verilog

i was not able to get the output if i did not included the default statement in the case block

somewhere i initialised the input as reg which is not valid

in the book i used to design this processor has some typo in the diagram of processor they have given so i stucked there for some time

i could also use immgen to get the output for load instructions and i tried but did not get the output i will try it again

you need to initilize the coontent of the regfile carefully if you initailize it let say by 100 and if the content of this regfile is used as address to store the data in the data memory we will not be able to store because upto only 63 we can store data