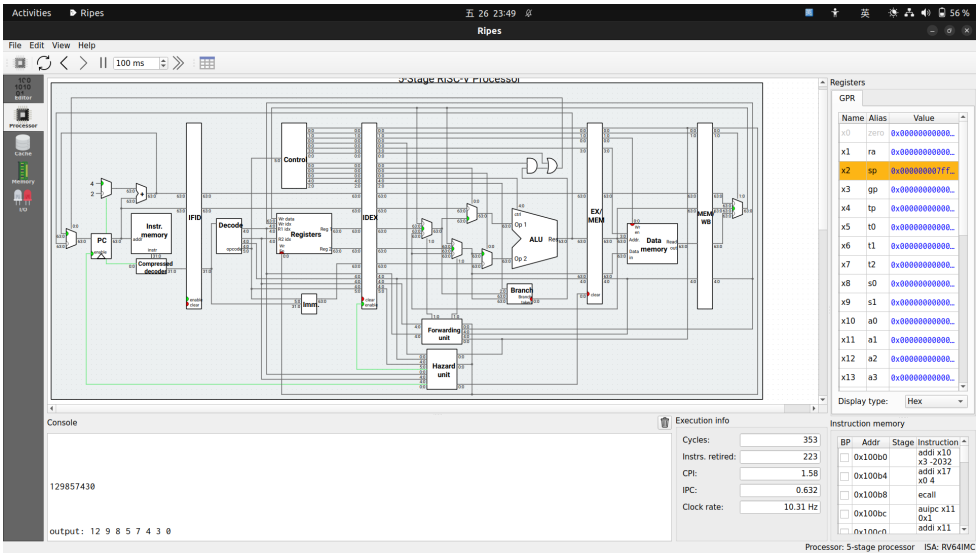
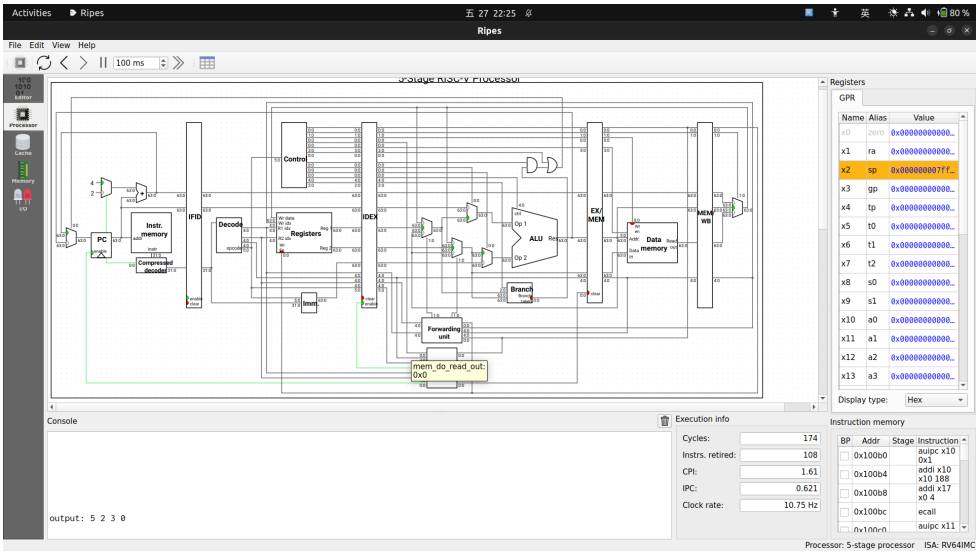
Computer Architecture HW5

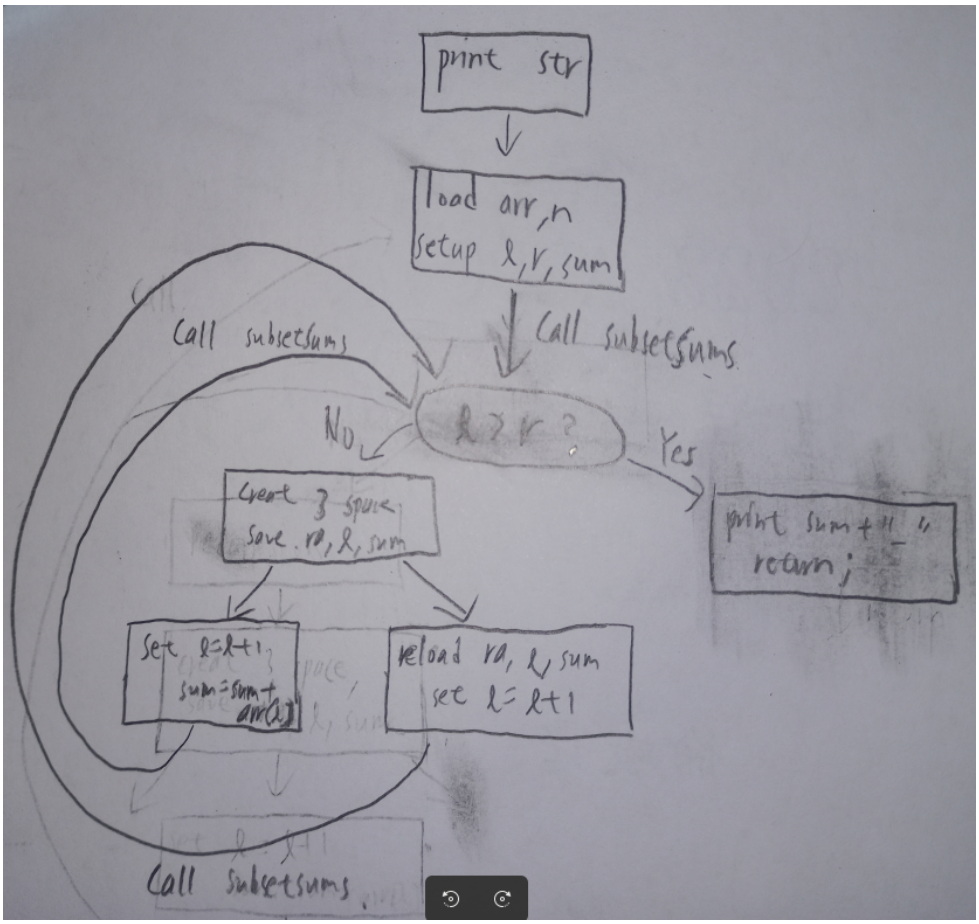
Testcase 1



Testcase 2

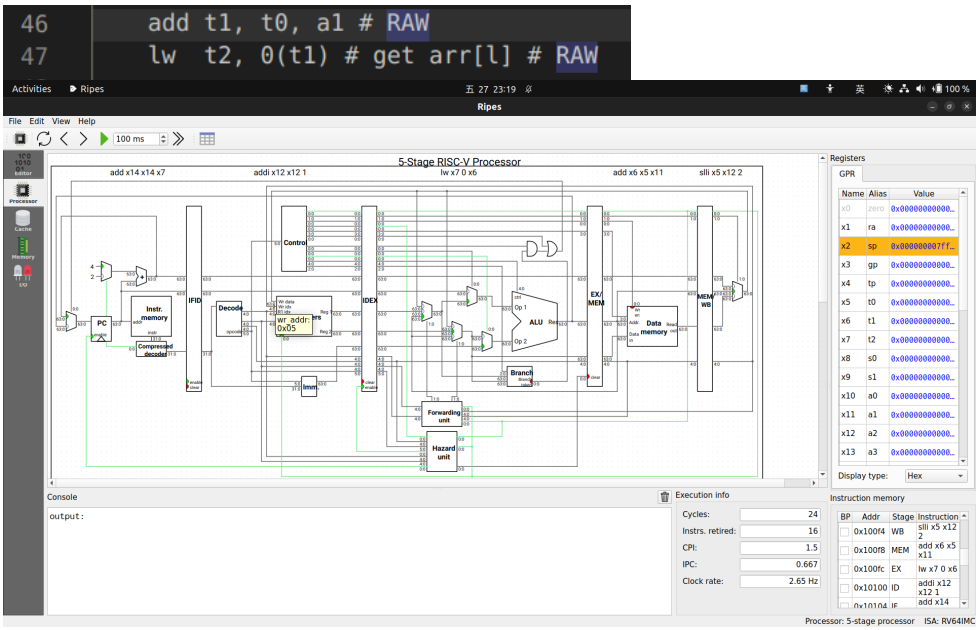


Flowchart



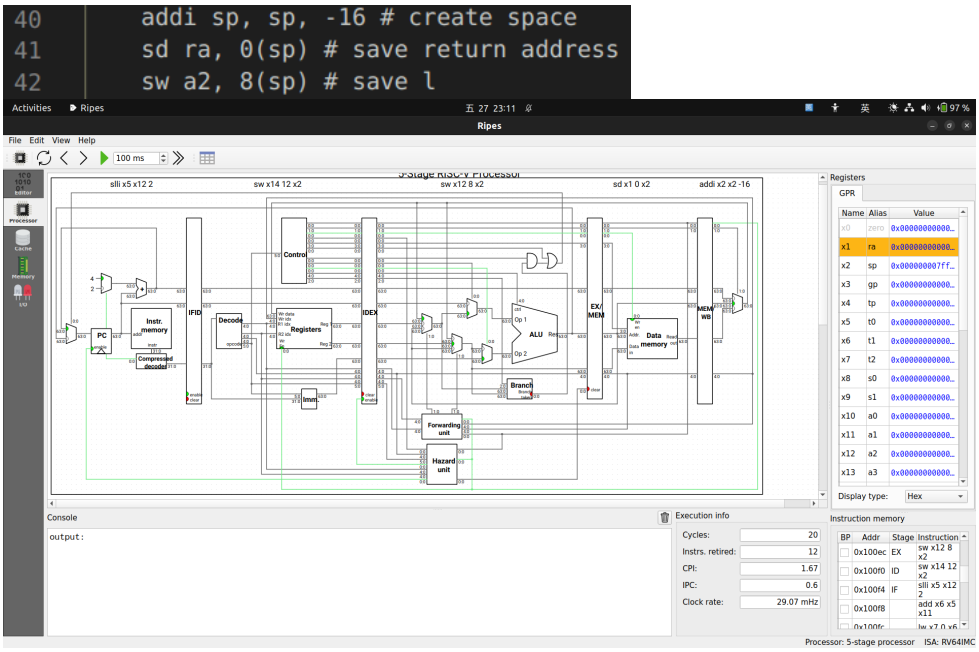
Hazard in my code:

Type (1):



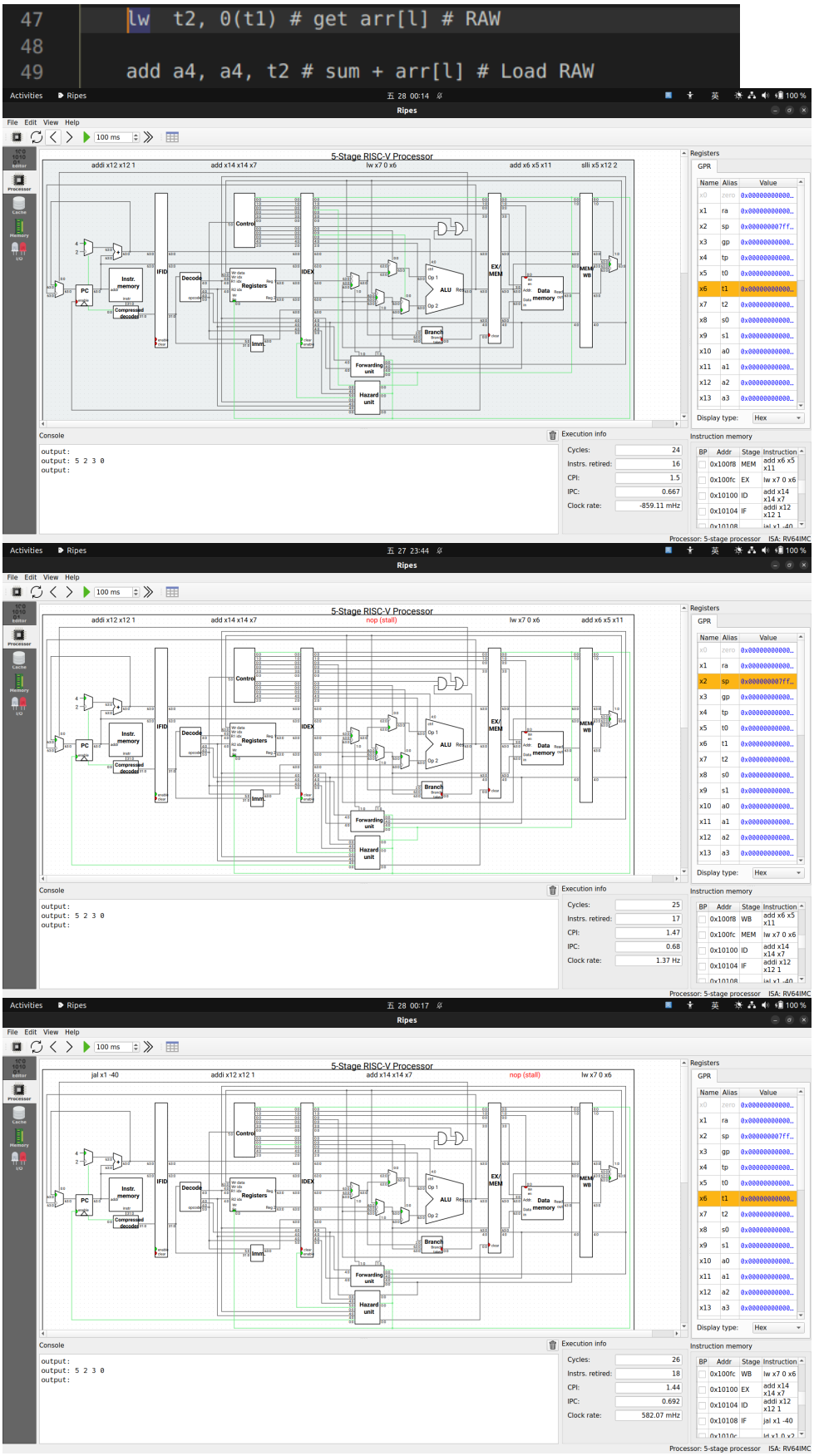
The processor detects dependency on MEM and EX stages for Type (1). Then, the
Forwarding unit sets the control signal of MUX before rs1 to select the t1 value
forwarded from the MEM stage to the EX stage

Type (2):



The processor detects dependency on WB and EX stages for Type (2). Then, the
Forwarding unit sets the control signal of MUX before rs1 to select the sp value
forwarded from the WB stage to the EX stage.

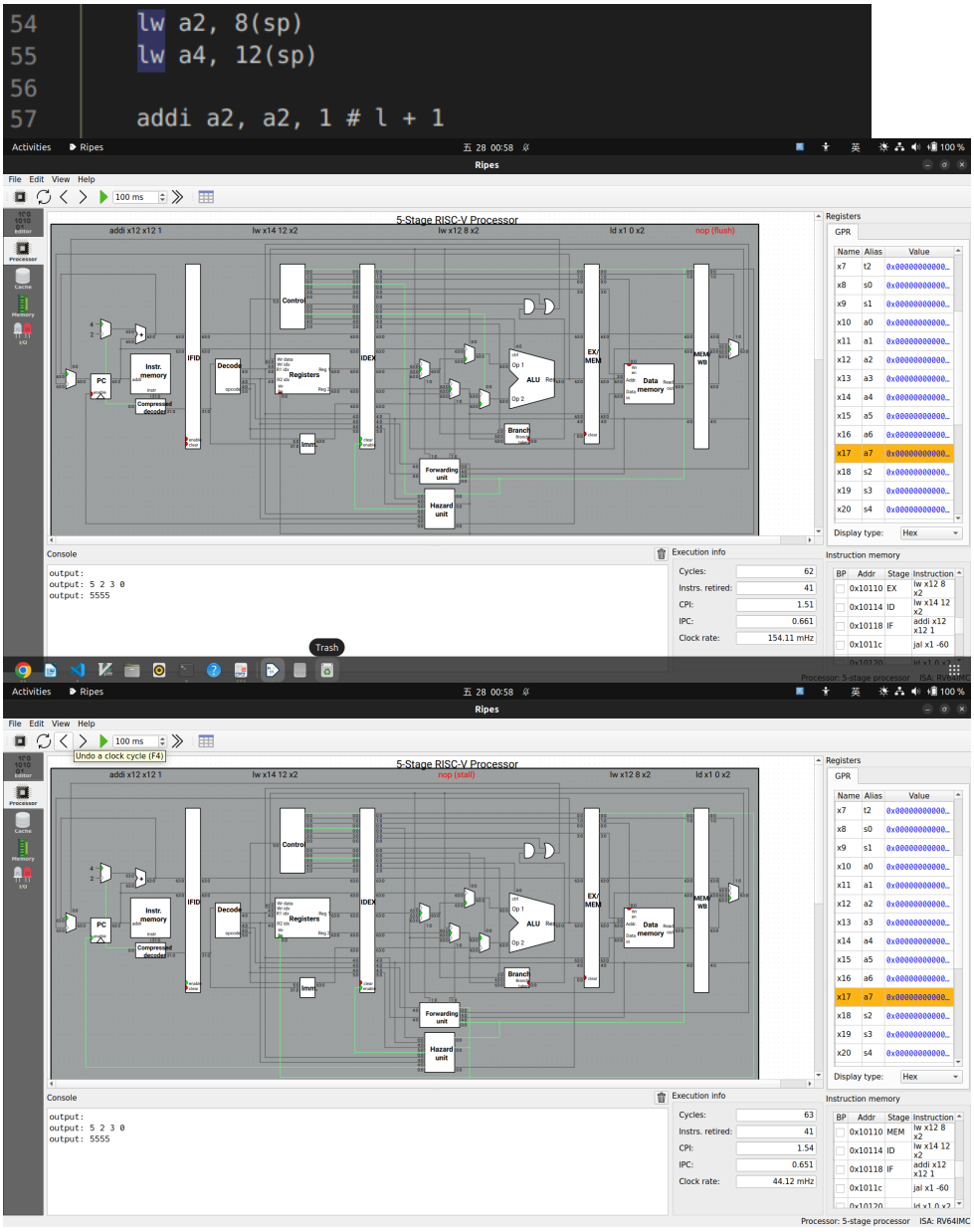
Type (3):



The processor detects dependency on EX(lw) and ID(add) stages for Type (3). Then the Hazard unit force the control signals in ID/EX register to 0. Turn EX into NOP, while the instructions in IF and ID stall for one cycle. Then, the

Forwarding unit sets the control signal of MUX before rs2 to select the t2 value forwarded from the WB stage to the EX stage.

Type (4):

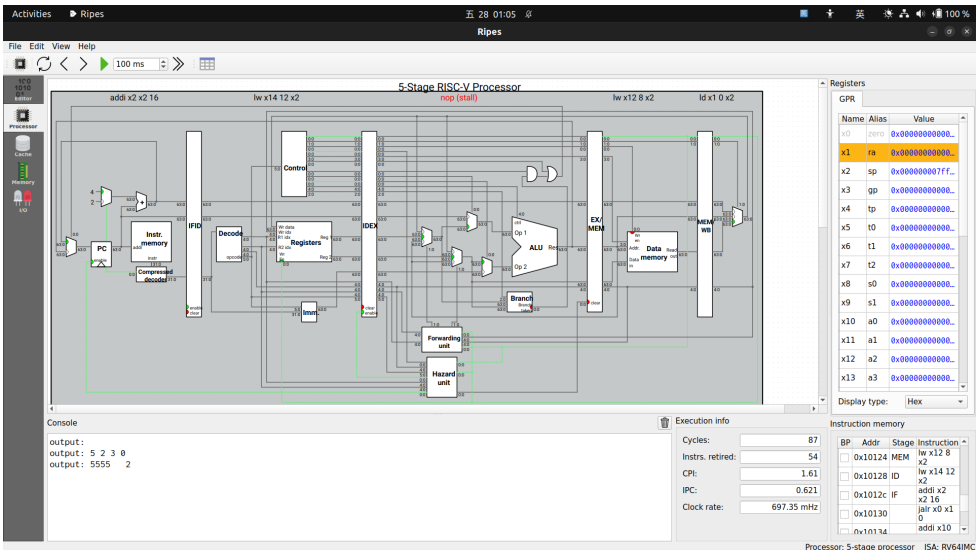


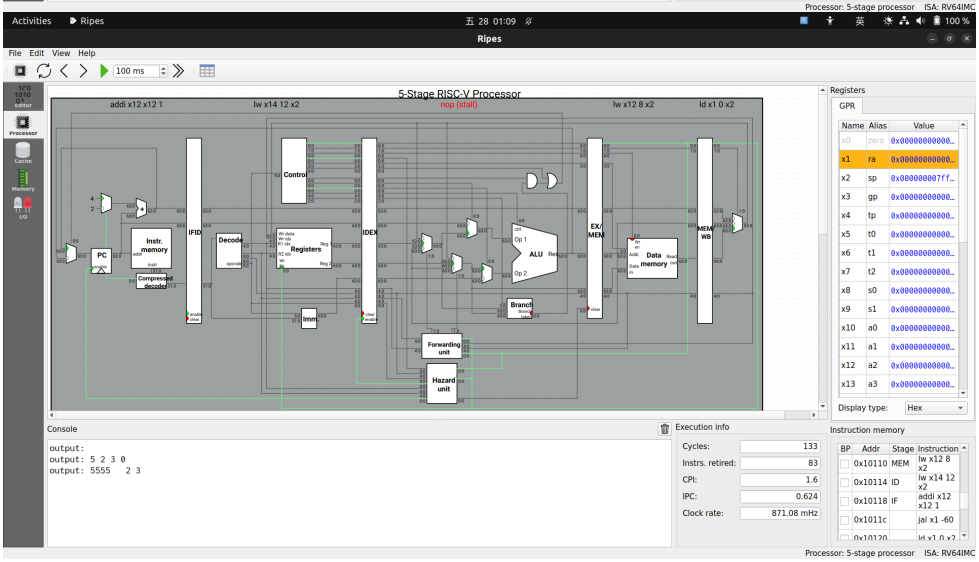
(guessing)The processor detects dependency on EX(lw) and ID(lw) stages for Type (4). Then the Hazard unit force the control signals in ID/EX register to 0. Turn EX into NOP, while the instructions in IF and ID stall for one cycle. However, this is a RAR, it shouldn't happen in RISC-V processor, so this is something I feel strange. And the possibility dependency checking on EX(lw) and IF(addi) also makes no sense to me. Since in IF stage I wouldn't know the rs1,rs2.

Type (5):

The processor determines whether to branch on EX stages for Type (5). After the processor decide to branch, it will flush the instructions in IF and ID to NOP, making sure the next instruction fetched at IF will be correct.

Observation





These are the strange phenomenon that I saw, the processor has insert a NOP, while there is no hazard(theorically), and there is no NOP in the following case.

I guess the reason might be three consecutive load instruction on memory(stack), it might take times to load memory to register, so the processor insert an NOP to

have more time to do this task.(also it might take time to slide through stack).