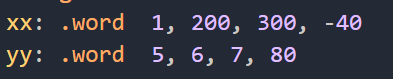
**109 fall計算機結構 PA3**

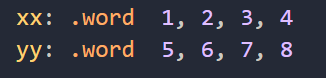
**電機23 108011235 陳昭維**

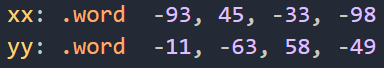
**1. Write an Assembly Program**

The code is attached with the homework on eeclass, both 5 stage with and without supporting forwarding and hazard detection.

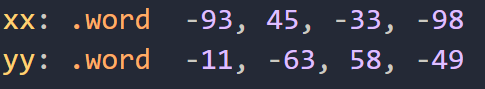
Example tested:

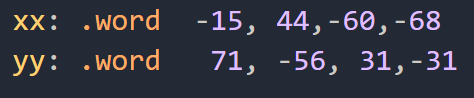


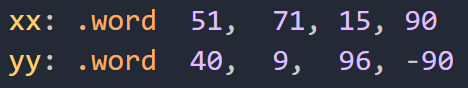




Example tested (nop):



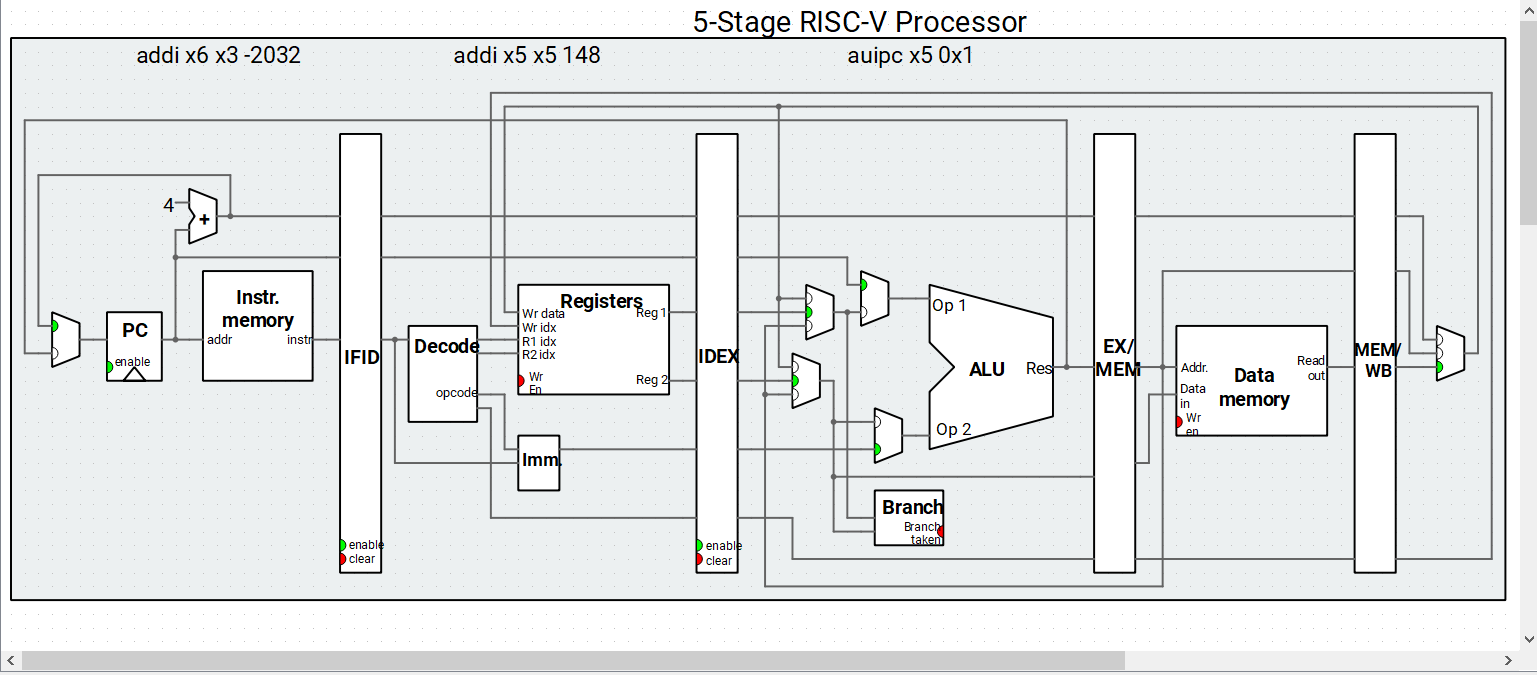


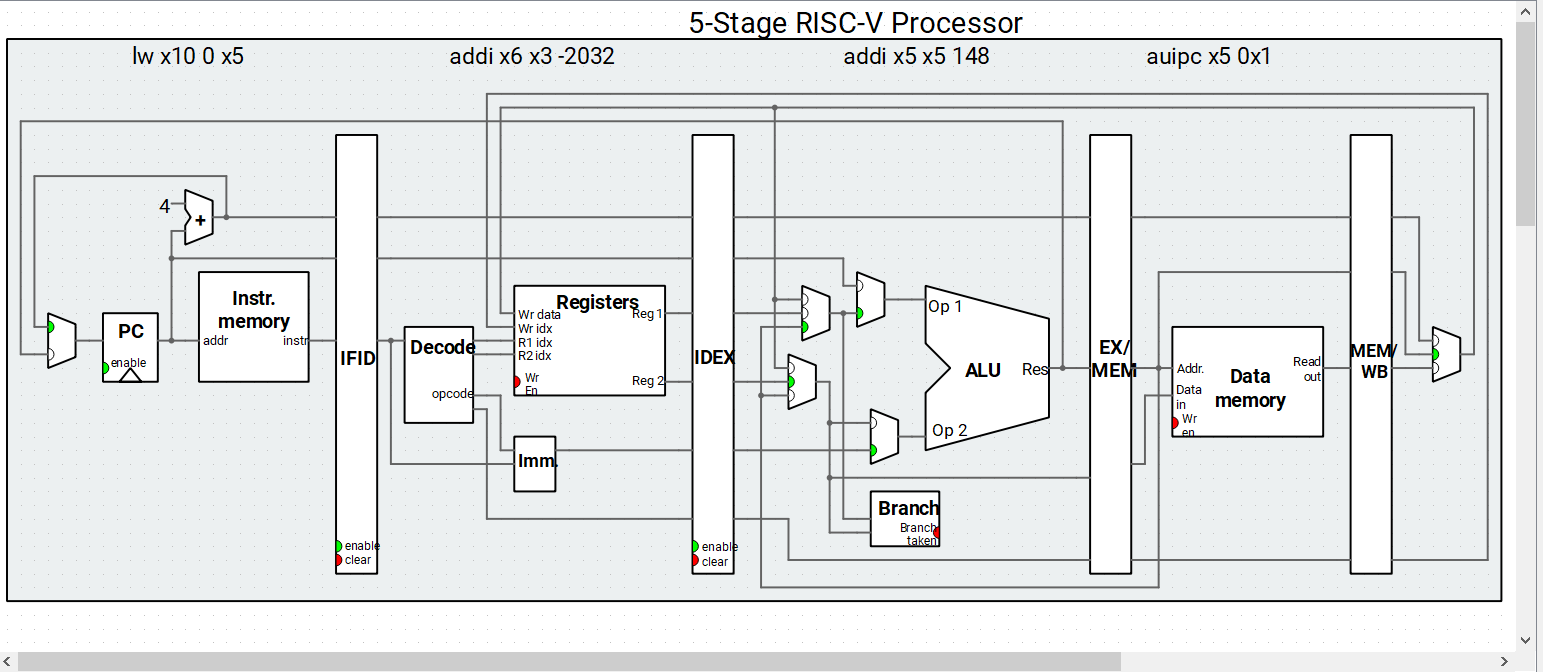


**2.** **Simulate Pipeline**

***I type forwarding at following 1st instruction***







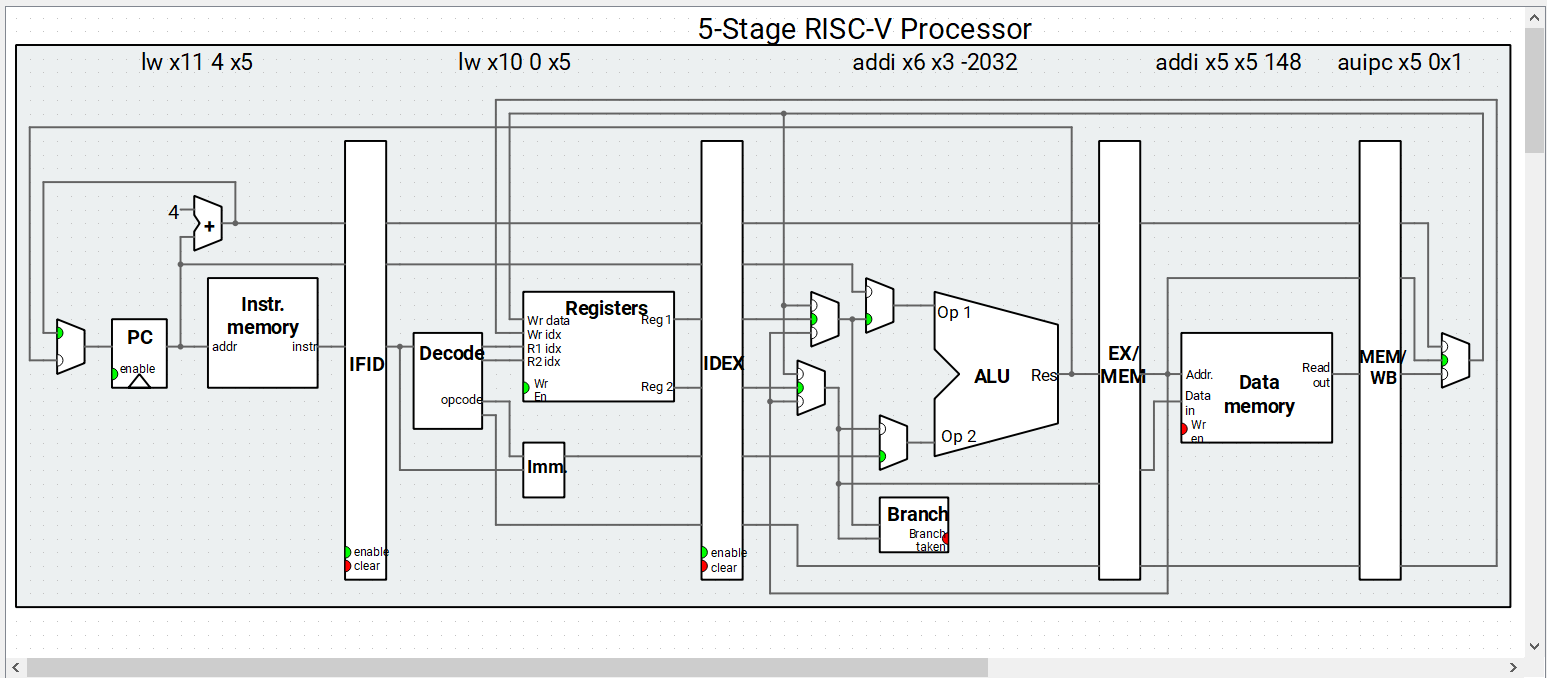
**Comment:**

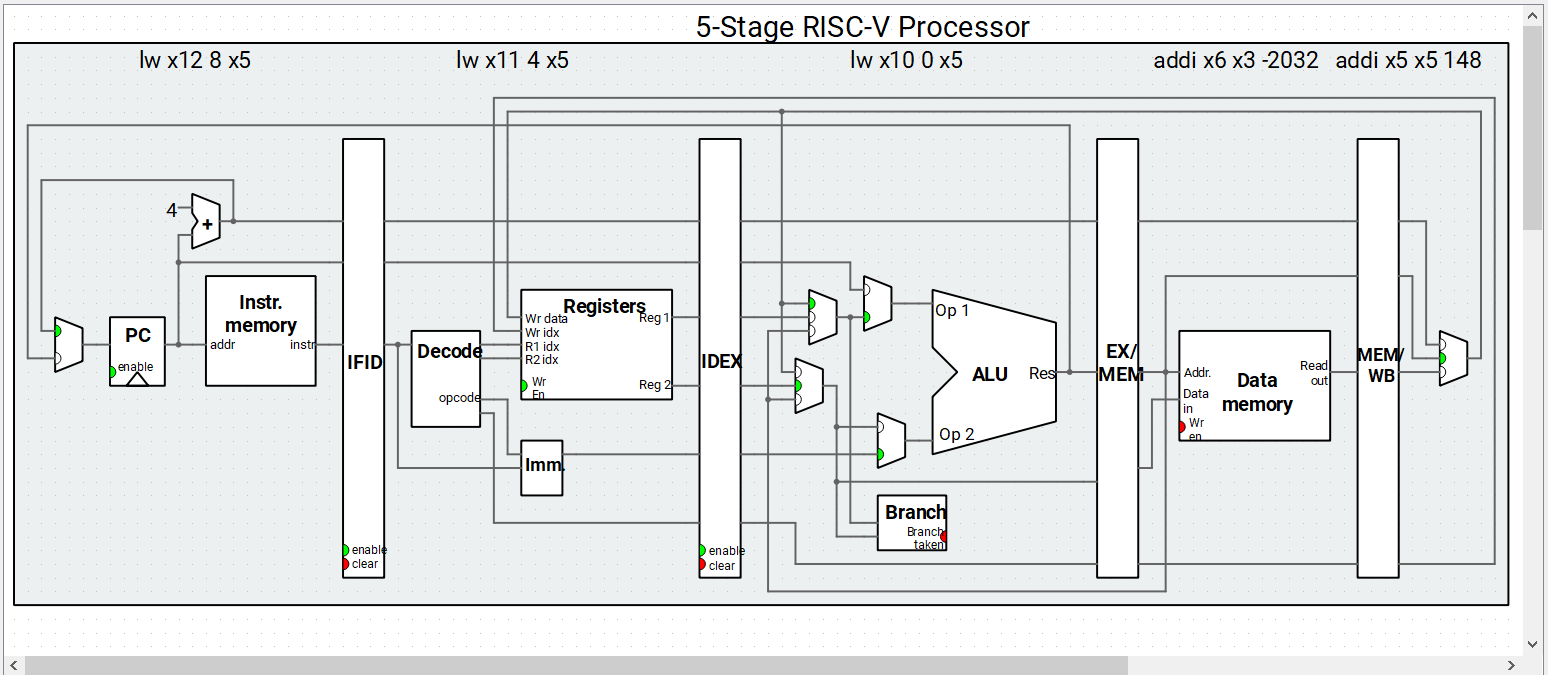
Forwarding happens between the cycle for the above stage, there is an EX/MEM forwarding for new value in register x5 to EX stage.

Forwarding is needed to let addition for the following addi instruction to us the correct value in x5.

***Load forwarding at following 2nd instruction***







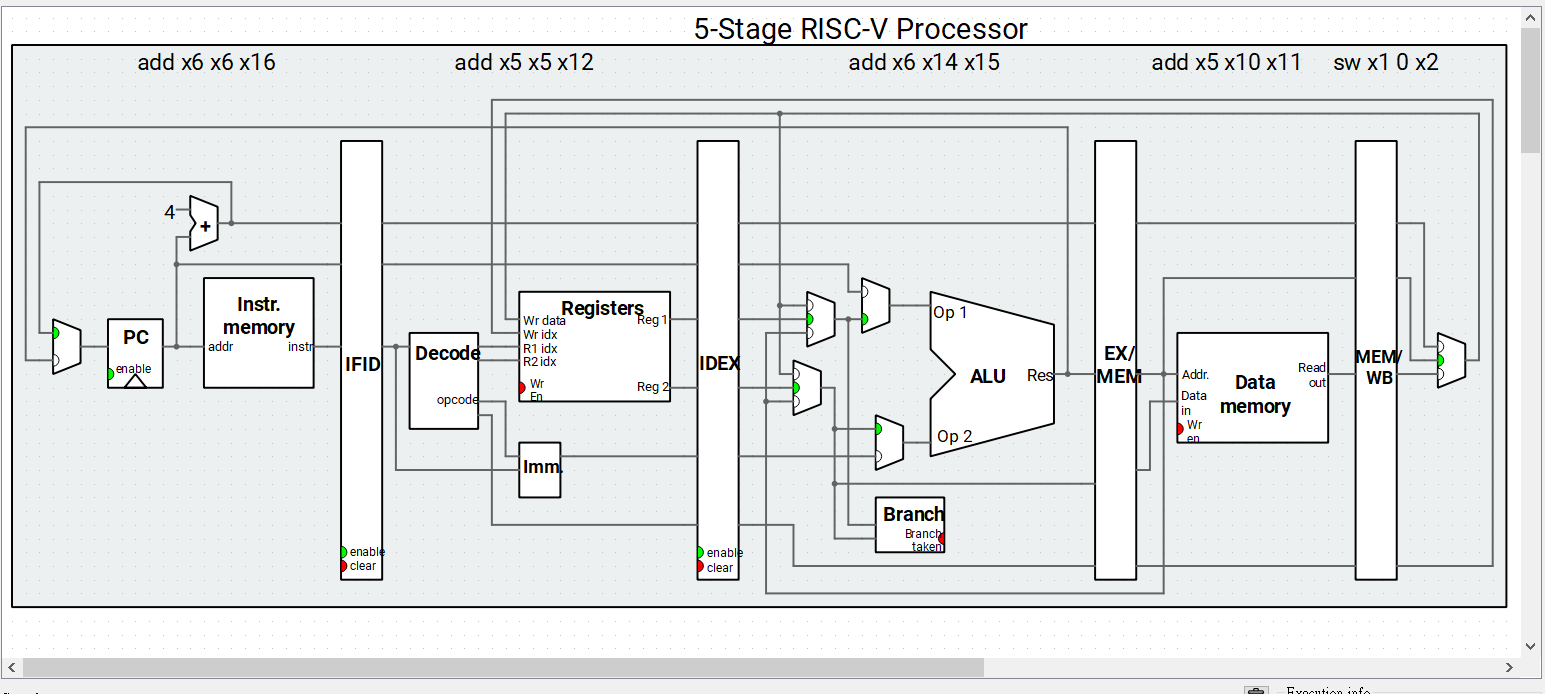
**Comment:**

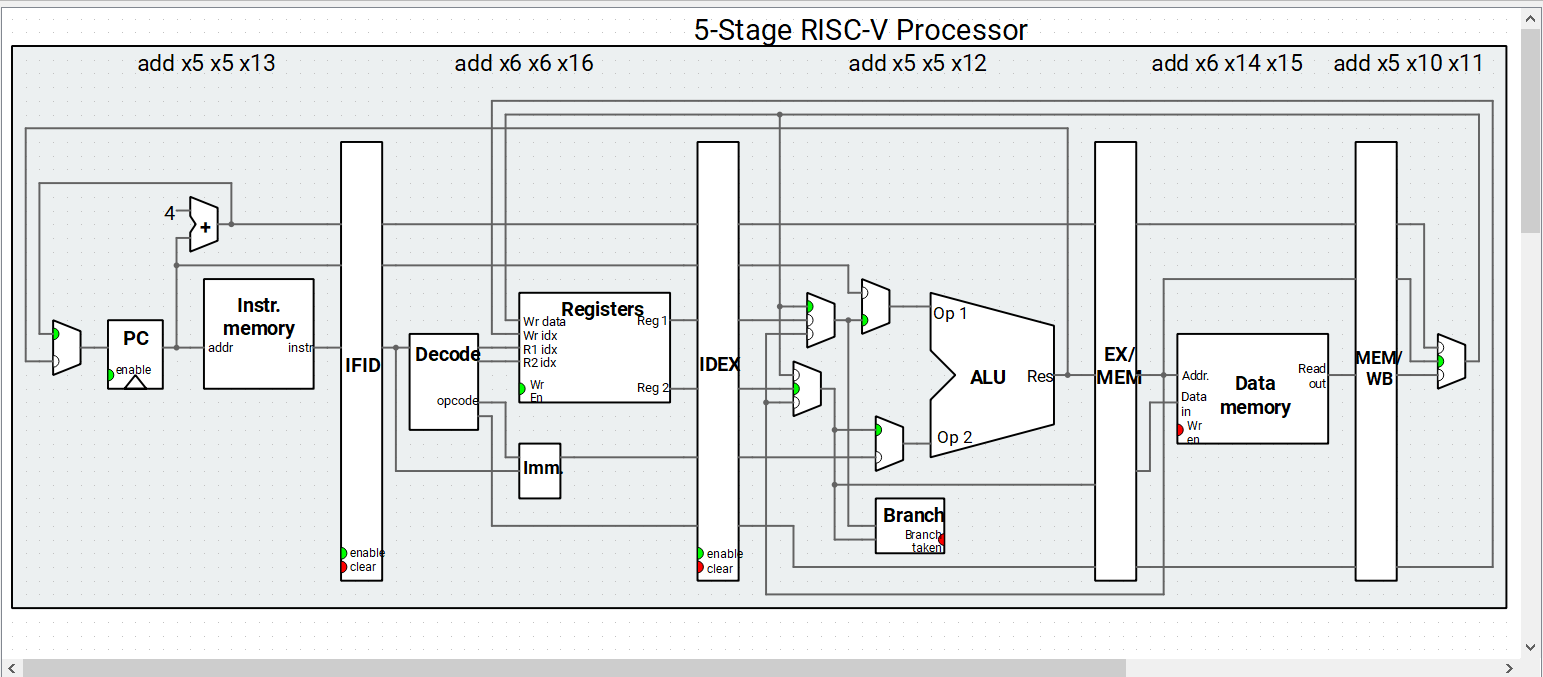
Forwarding happens between the cycle for the above stage, there is a MEM/WB forwarding of new value in register x5 to EX stage.

Forwarding is needed to let the 2nd following lw instruction to us the correct value in x5.

***R type forwarding at following 2nd instruction***



****

****

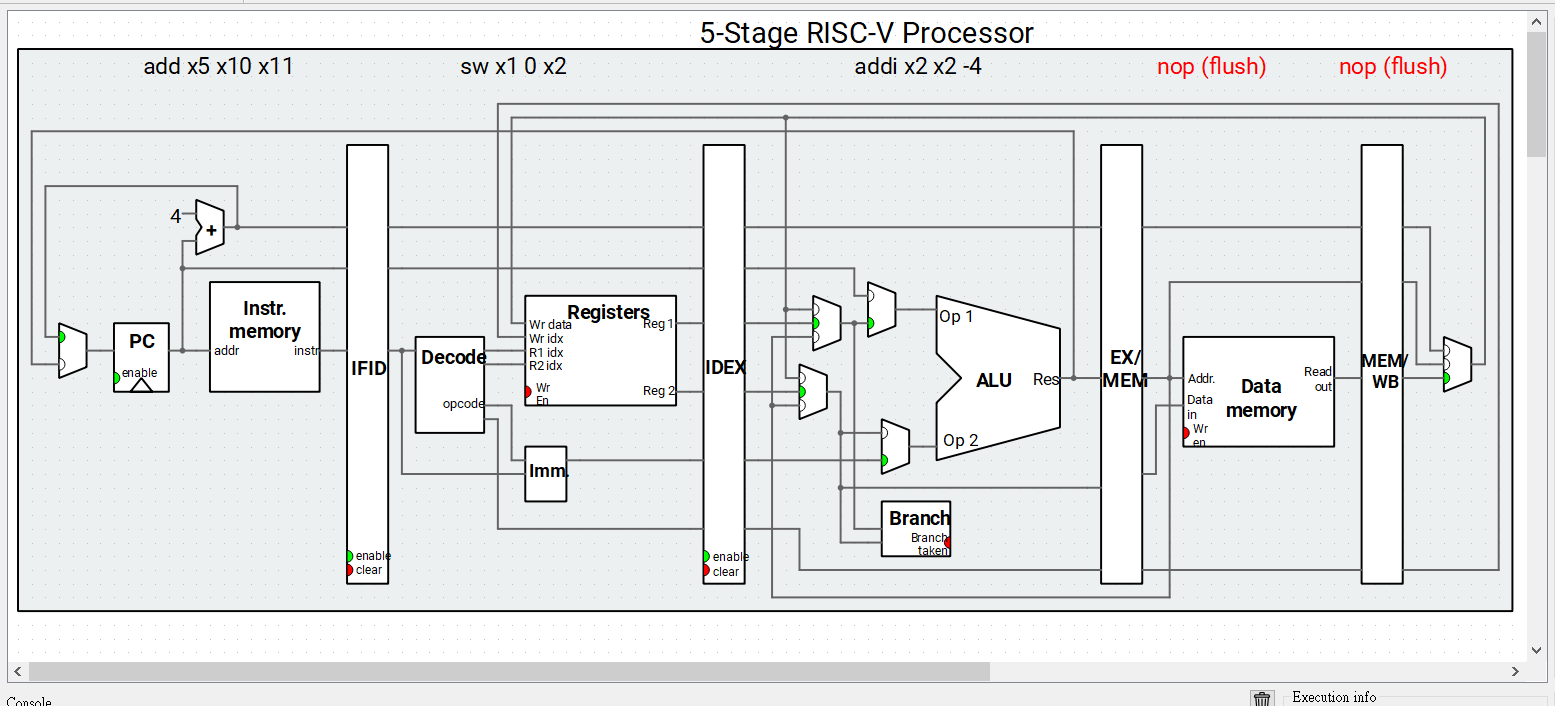
**Comment:**

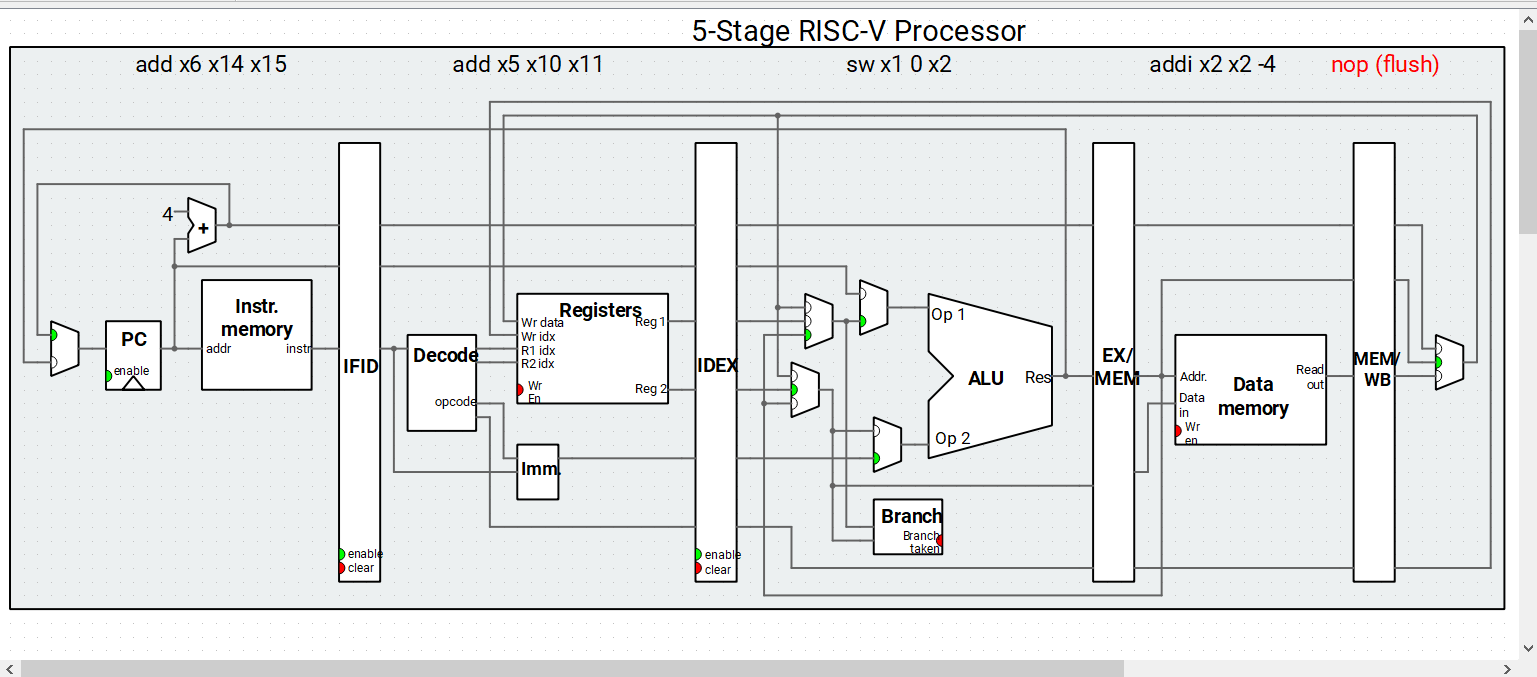
Forwarding happens between the cycle for the above stage, there is a MEM/WB forwarding of new value in register x5 to EX stage.

The forwarding is needed for the 2nd following instruction add to use the correct value in x5.

***Store type forwarding at following 2nd instruction***



****

****

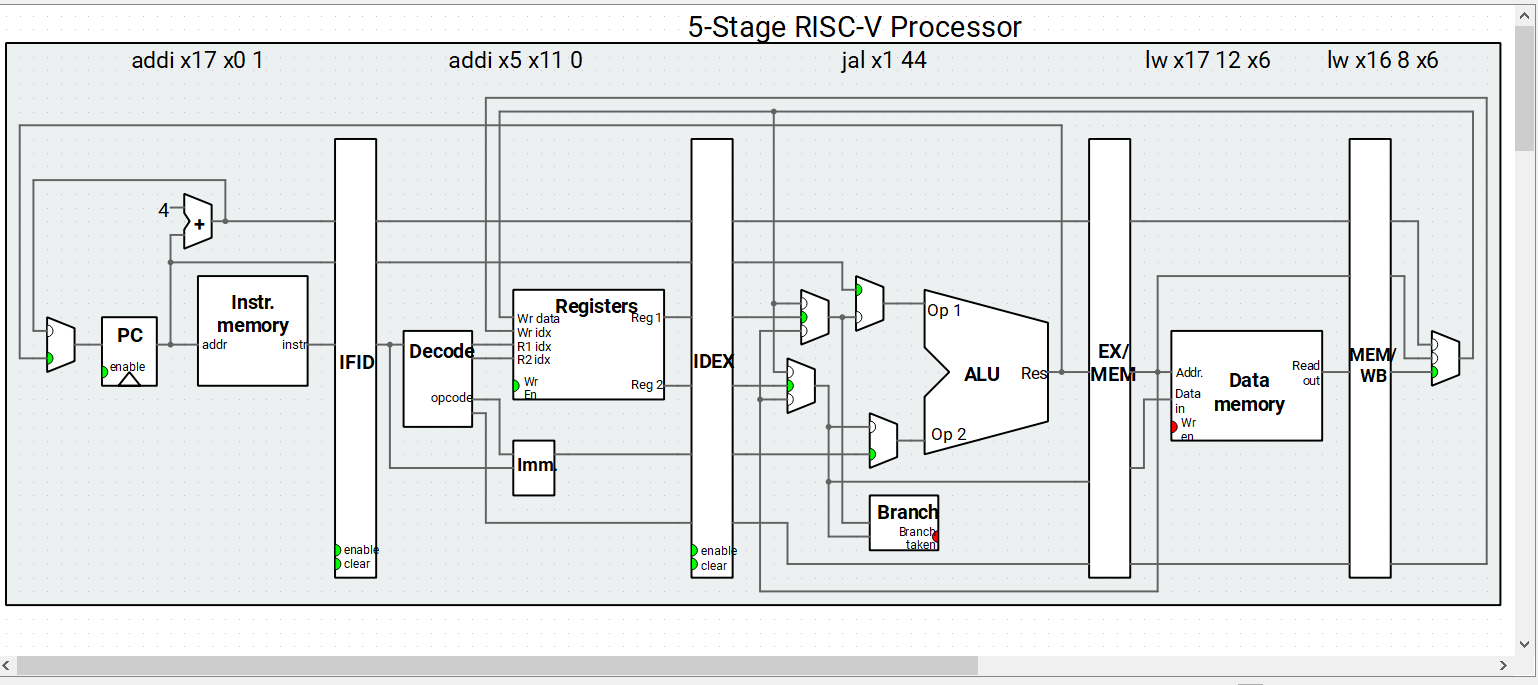
**Comment:**

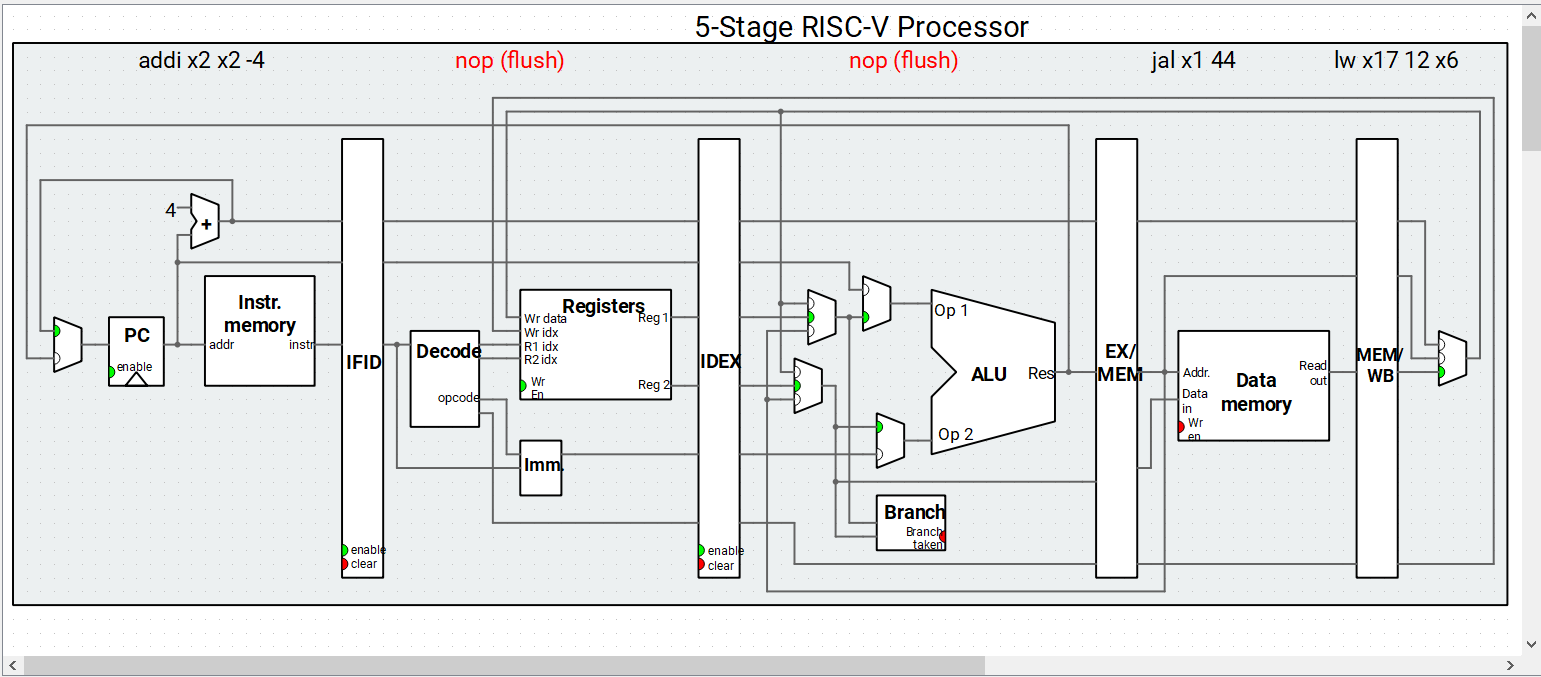
The forwarding is between the next cycle for above stage, there is an EX/MEM forwarding of new address of the stack pointer to EX stage.

The forwarding is needed for the following instruction sw to use the correct stack pointer address.

***Jump and link Flush***





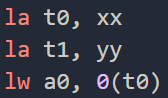
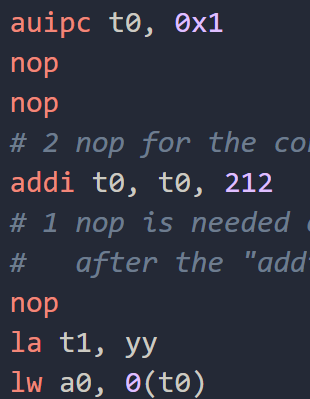


**Comment:**

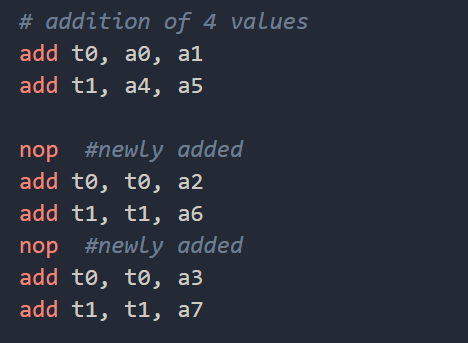
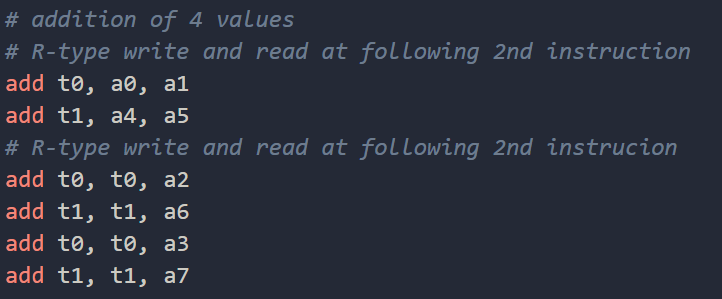
The flush is needed because the jump and link jumps to the new function, thus the loaded instruction before entering the new function needs flush.

**3. 5-stage pipeline without forwarding and or hazard detection**

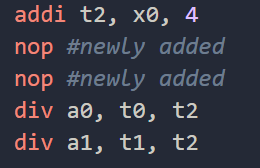
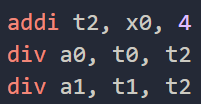
*The left code is the original code and the right code is the modified code*

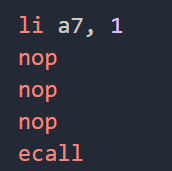
The load address instruction is break downed into auipc and addi instruction since there is a load instruction at following 2nd instruction, two nop needs to be put in between. And after that the instruction lw a0, 0(t0) needs the new value of t0 after the instruction t0, t0, 212, so one more nop need to be put in between.



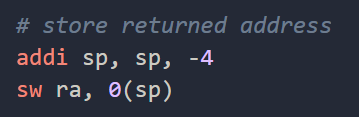
Since there is data dependency of add instruction after at 2nd following of the first/second and the third/forth line. So nop first is inserted to the third line and fifth line, then for add t1, a4, a5 and add t1, t1, a6 nop doesn’t need to be inserted because the previous nop inserted makes it able to catch up write back then read.



Since there is data dependency of the div instruction after the first line addi instruction, so 2 nop has to be inserted to catch up the write back then read.



Also, for my code, ecall needs 3 nop after the instruction li, a7, 1 for it to work without hazard.



The sw after the addi instruction doesn’t need nop but still can works fine.