**Department of Computer Science**

**National Tsing Hua University**

**CS4100 Computer Architecture**

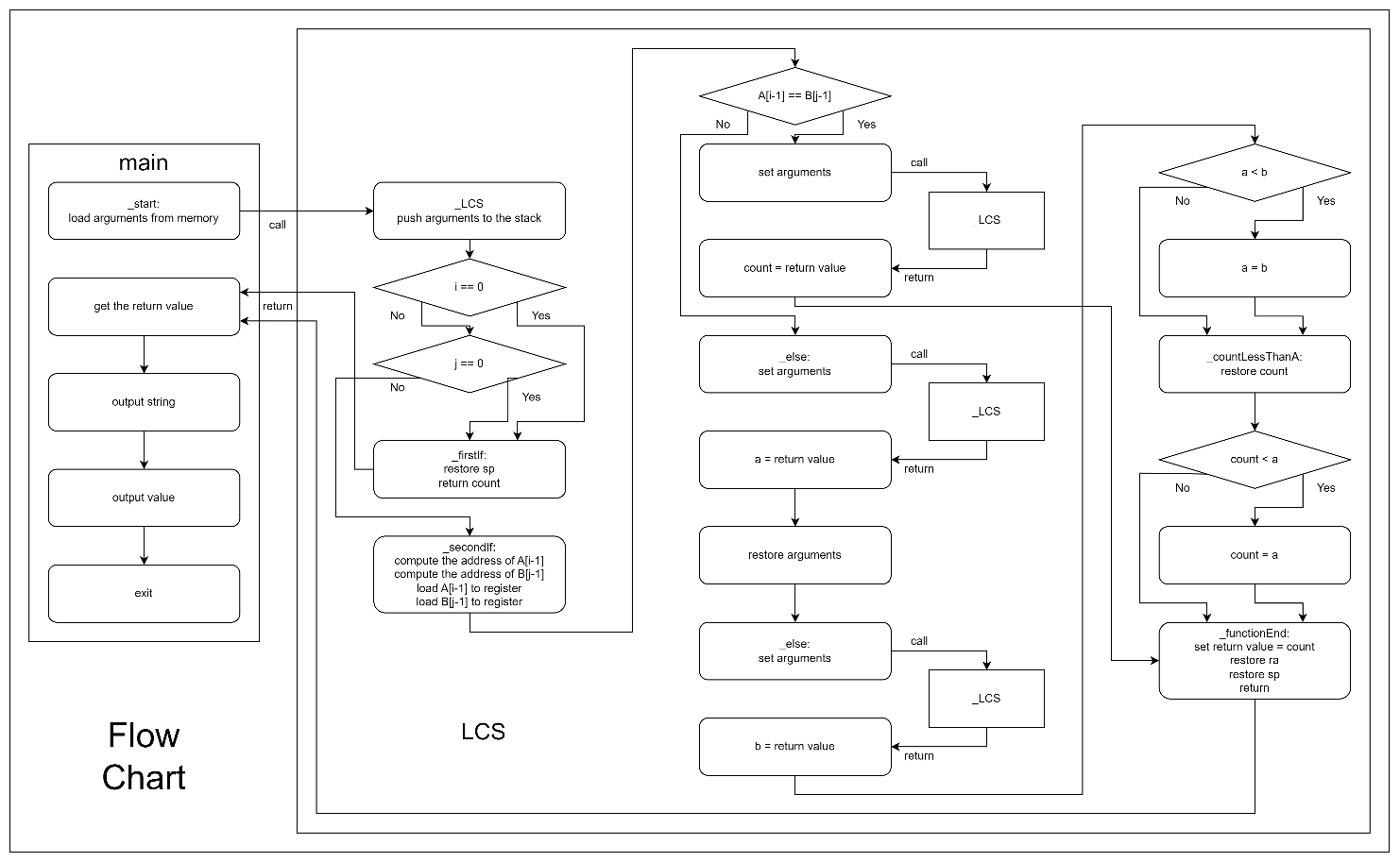
Spring, 2024, Homework 5

Due date: 5/26/2024 23:59

1. **Assembly Coding**
2. Test Result

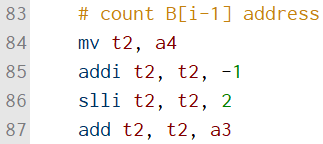
|  |  |
| --- | --- |
| Testcase 1 | Testcase 2 |
|  |  |

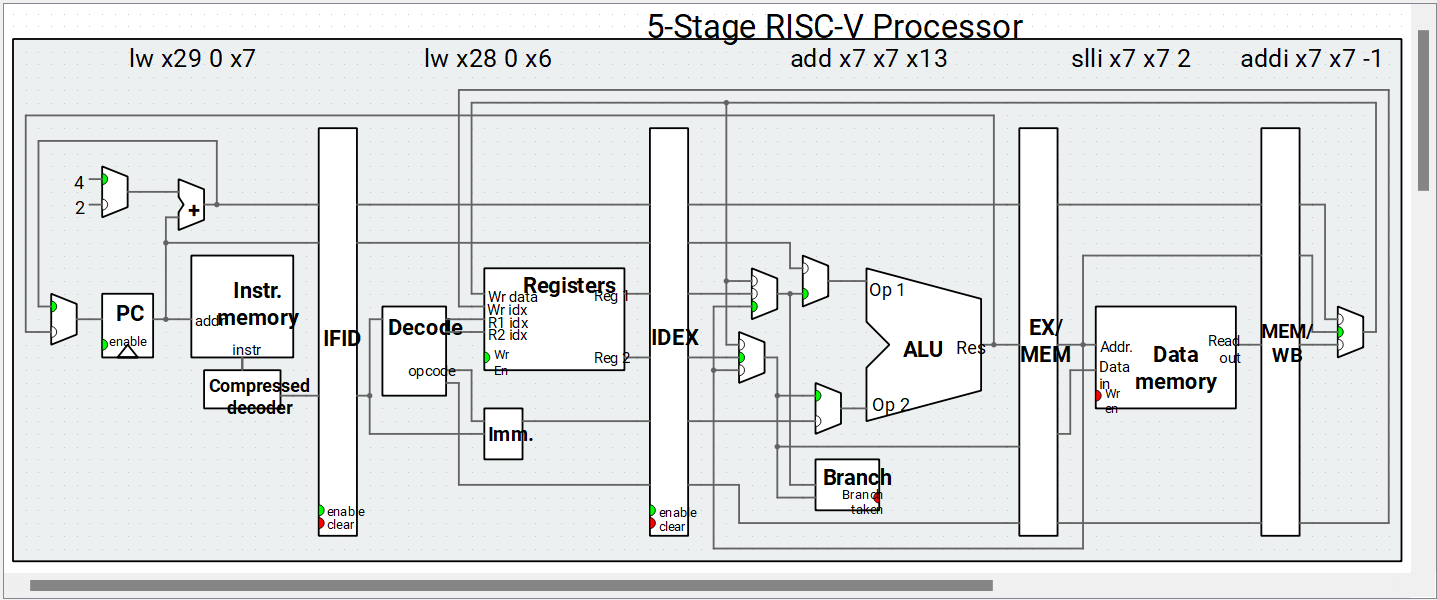
1. Flow Chart



1. **Hazard in Your Code**

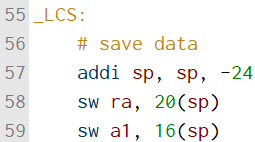
* Type 1:

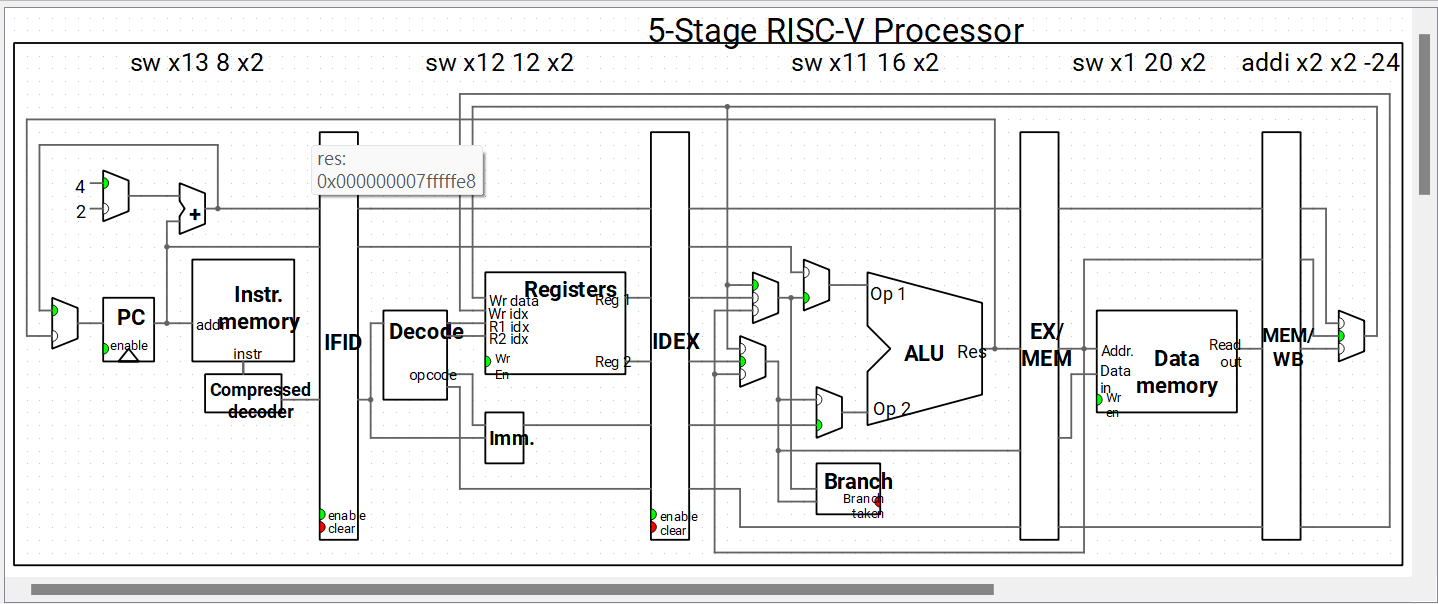




t2是slli t2, t2, 2的rd，又被add t2, t2, a3當作rs1，因此會產生Type 1 hazard，紅色框框圈起來的mux顯示了為了要解決這個hazard，把EX/MEM存的值(上一個ALU結果) forward給ALU，這樣就可以拿到正確的值做計算。

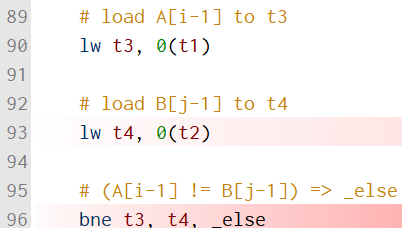
* Type 2:

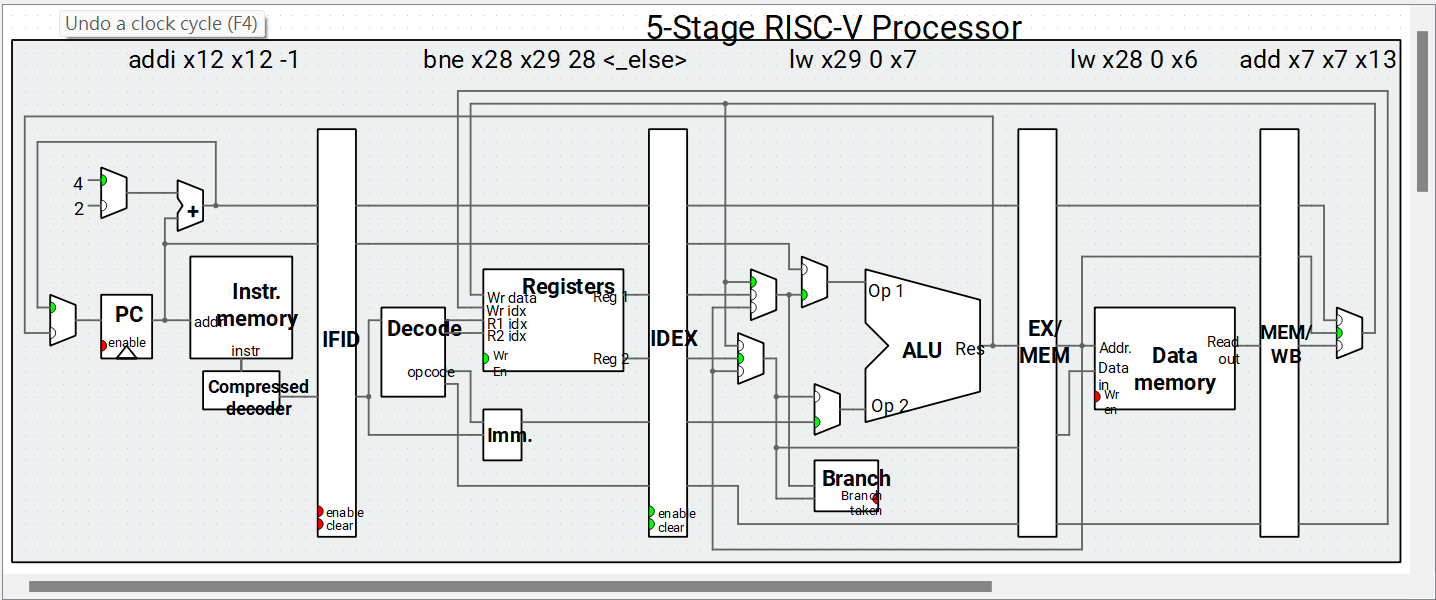


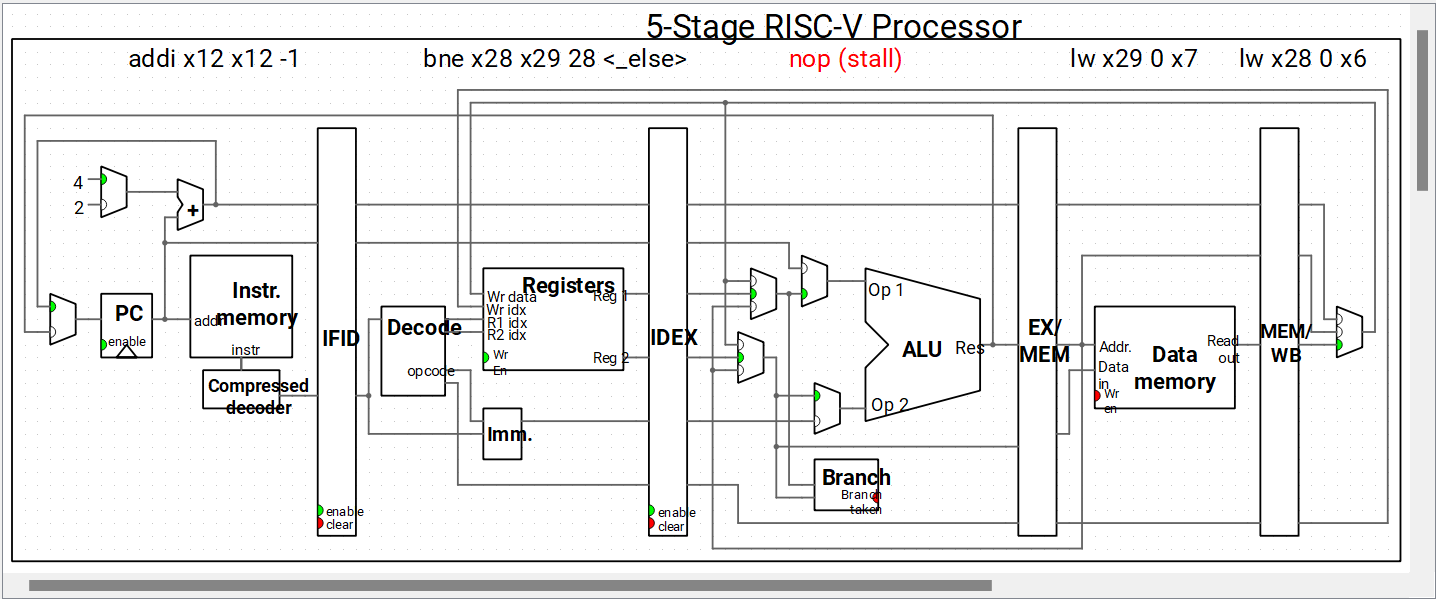


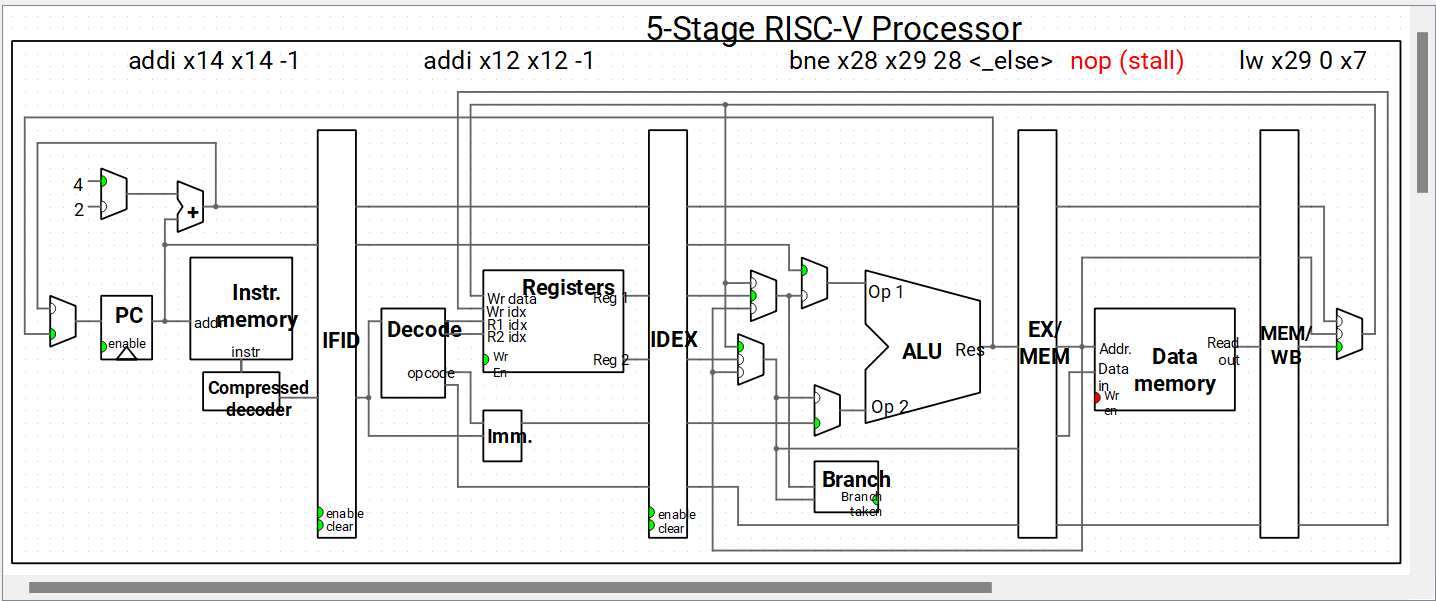
sp是addi sp, sp, -24的rd，也是sw a1, 16(sp)的rs1，因此會有Type 2 hazard，但因為結果早就算出來了，所以只需要從MEM/WB將正確的值forward給ALU就可以解決這個hazard了。

* Type 3:



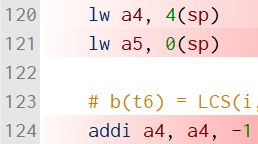


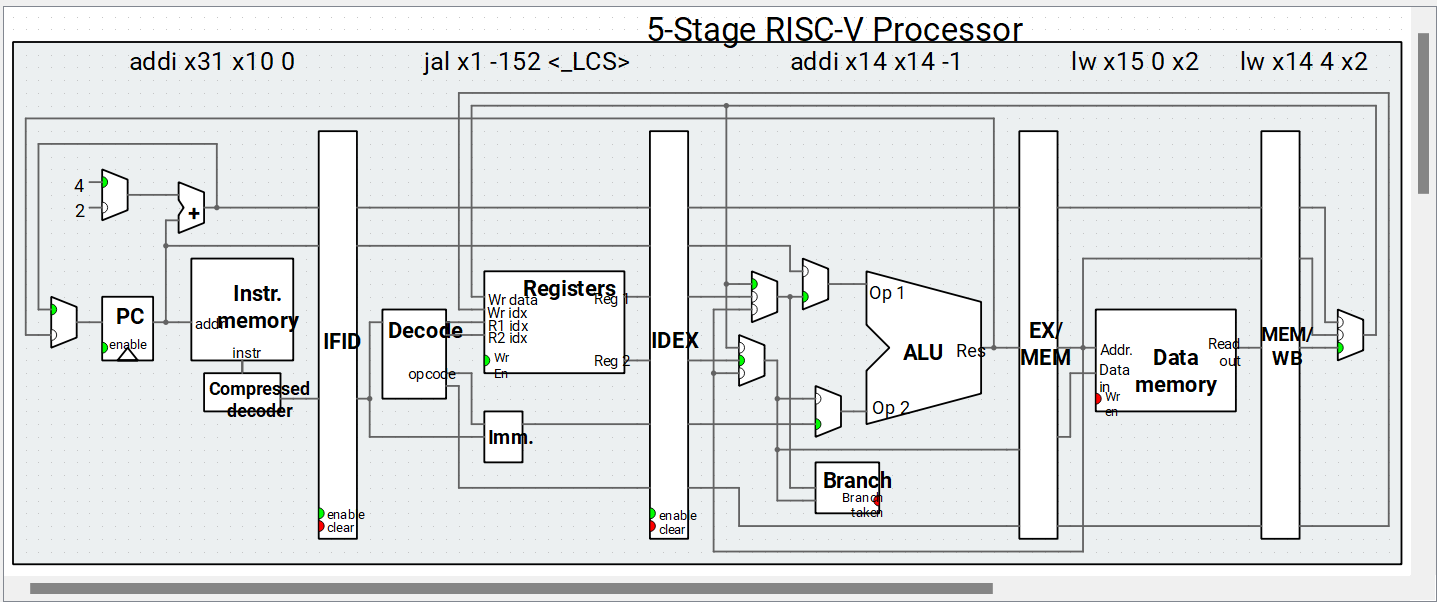




lw t4, 0(t2)與bne t3, t4, \_else會有Type 3 hazard，ID/EX的clear訊號被設成1，IF/ID的enable訊號被設成0，PC的enable訊號被設成0，使處理器stall一個cycle，產生一個bubble等待load指令從memory拿到正確的值，然後透過foward的方式(紅色框框)，將正確的值forward給ALU的input，以解決hazard的問題。

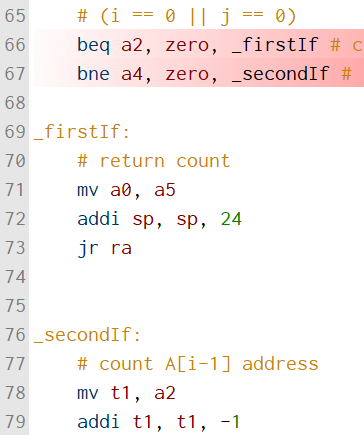
* Type 4:

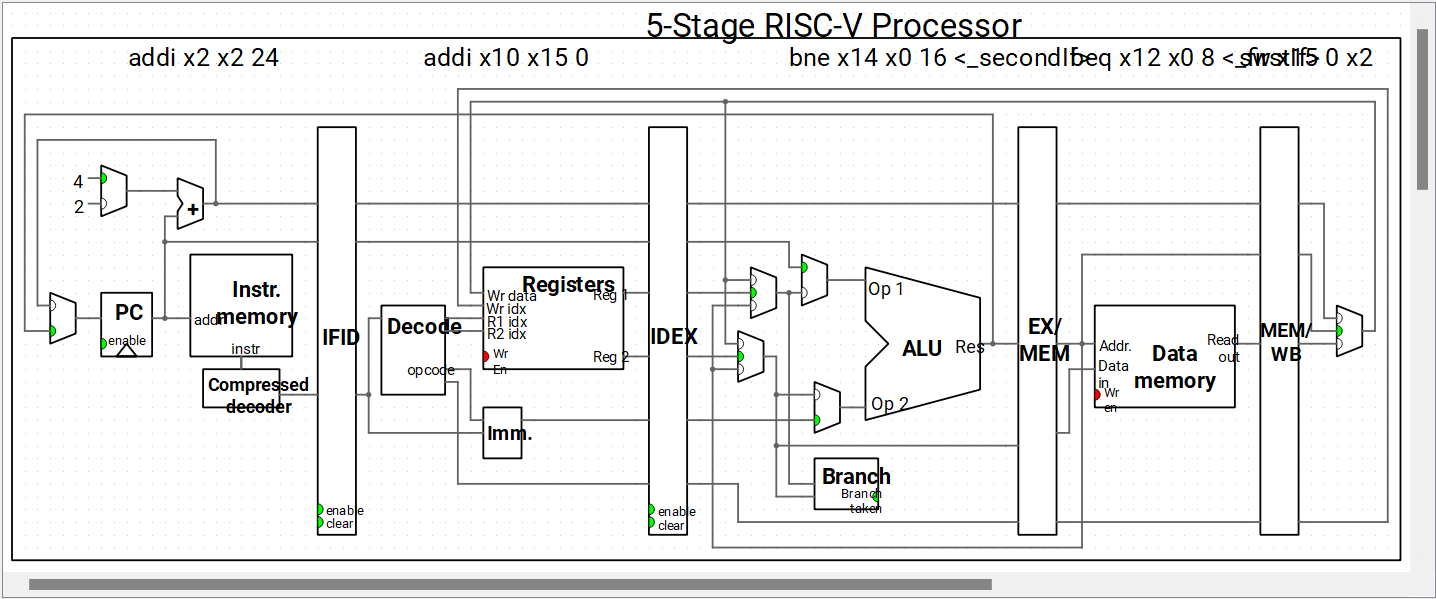


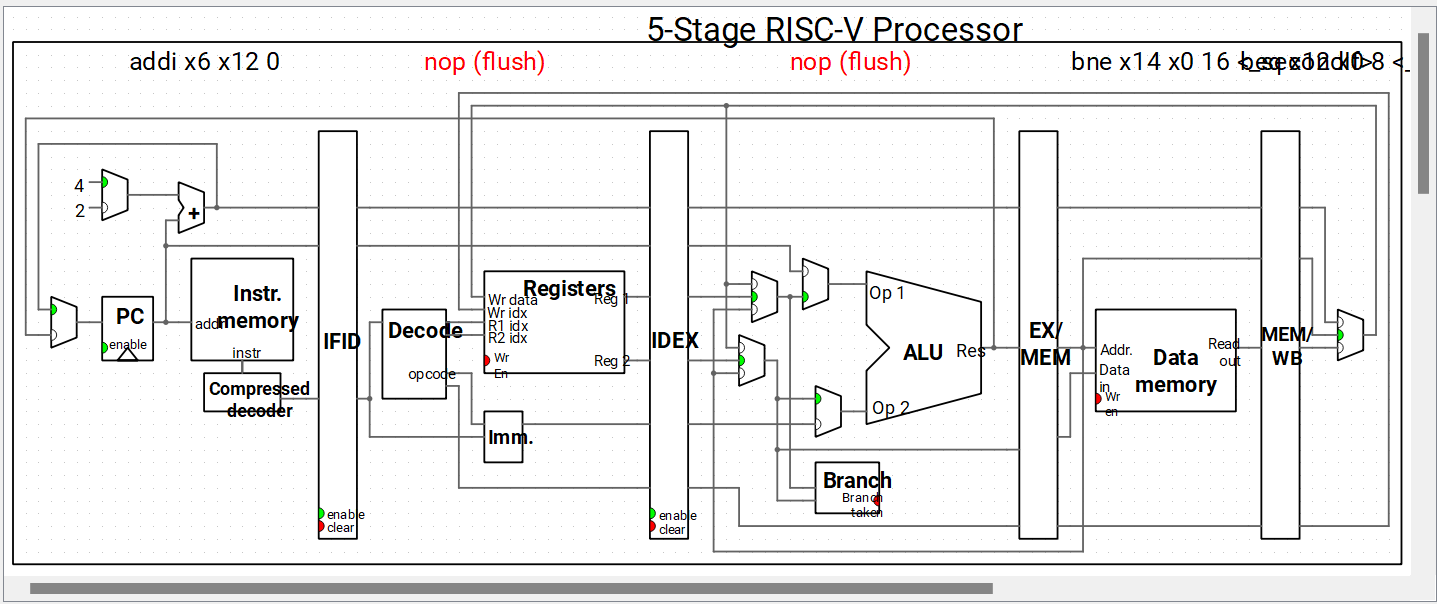


lw a4, 0(sp)與addi a4, a4, -1會產生Type 4 hazard，與Type 3不同的是，由於正確的值已經被算出來了，因此不用stall，只需要將MEM/WB的值forward給ALU，就可以解決這個hazard。

* Type 5:







branch taken後若branch指令後面還有指令執行的話需要flush掉，當偵測到Type 5 hazard後，IF/IF和ID/EX的clear訊號都被設成1，以flush掉不需要執行的指令，並從已經算好的新PC位址拿到正確的指令，以解決這個hazard。