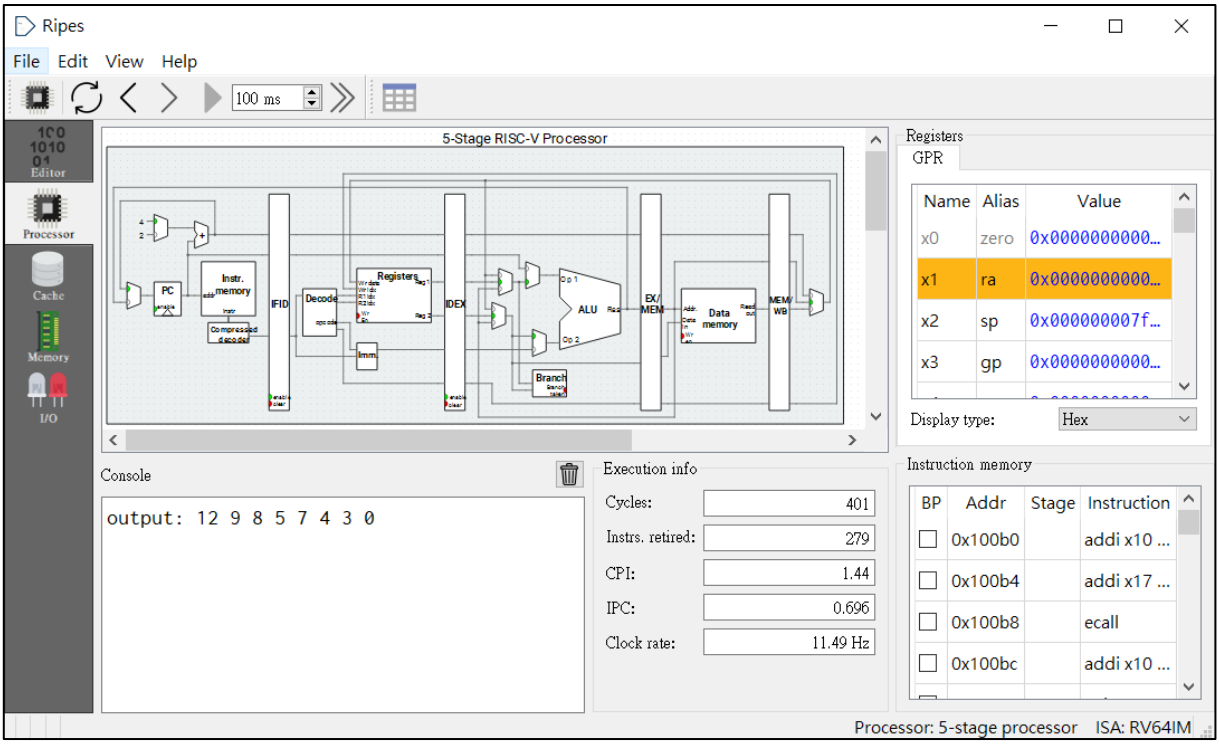
Computer Architecture: Assembly Coding and Hazard Detection Report

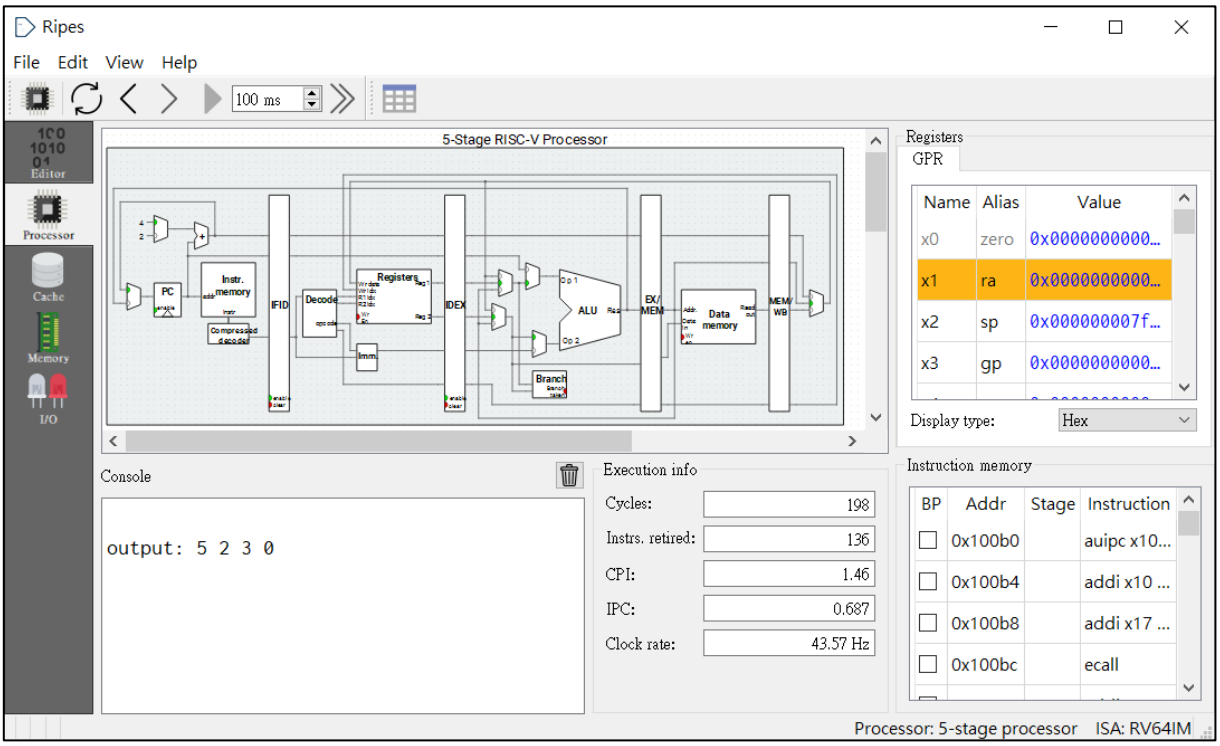
108022203 莊翔凱

1. Assembly Coding (Task: Sums of all subsets of a given set)

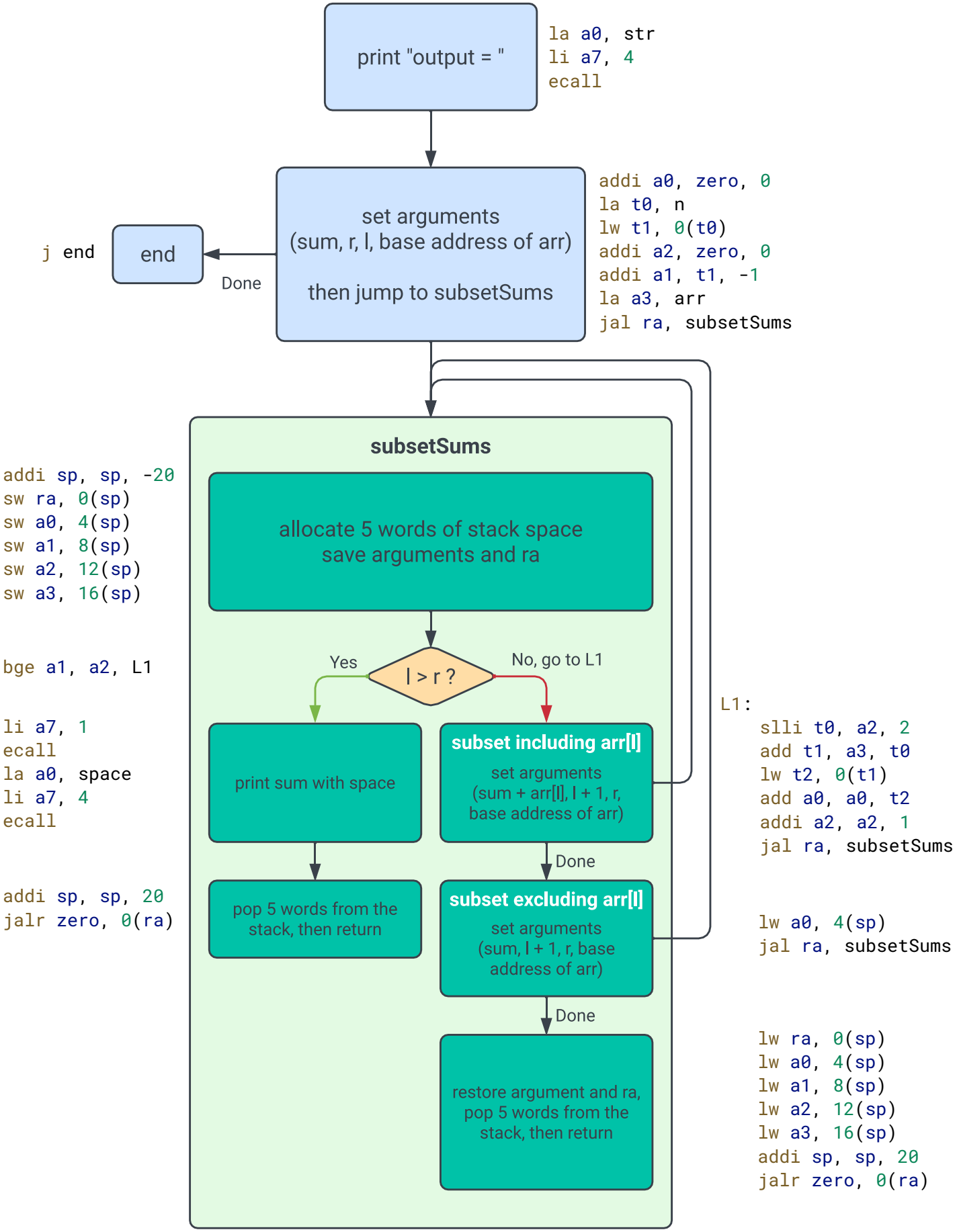
testcase 1: set = {5, 4, 3}



testcase 2: set = {2, 3}



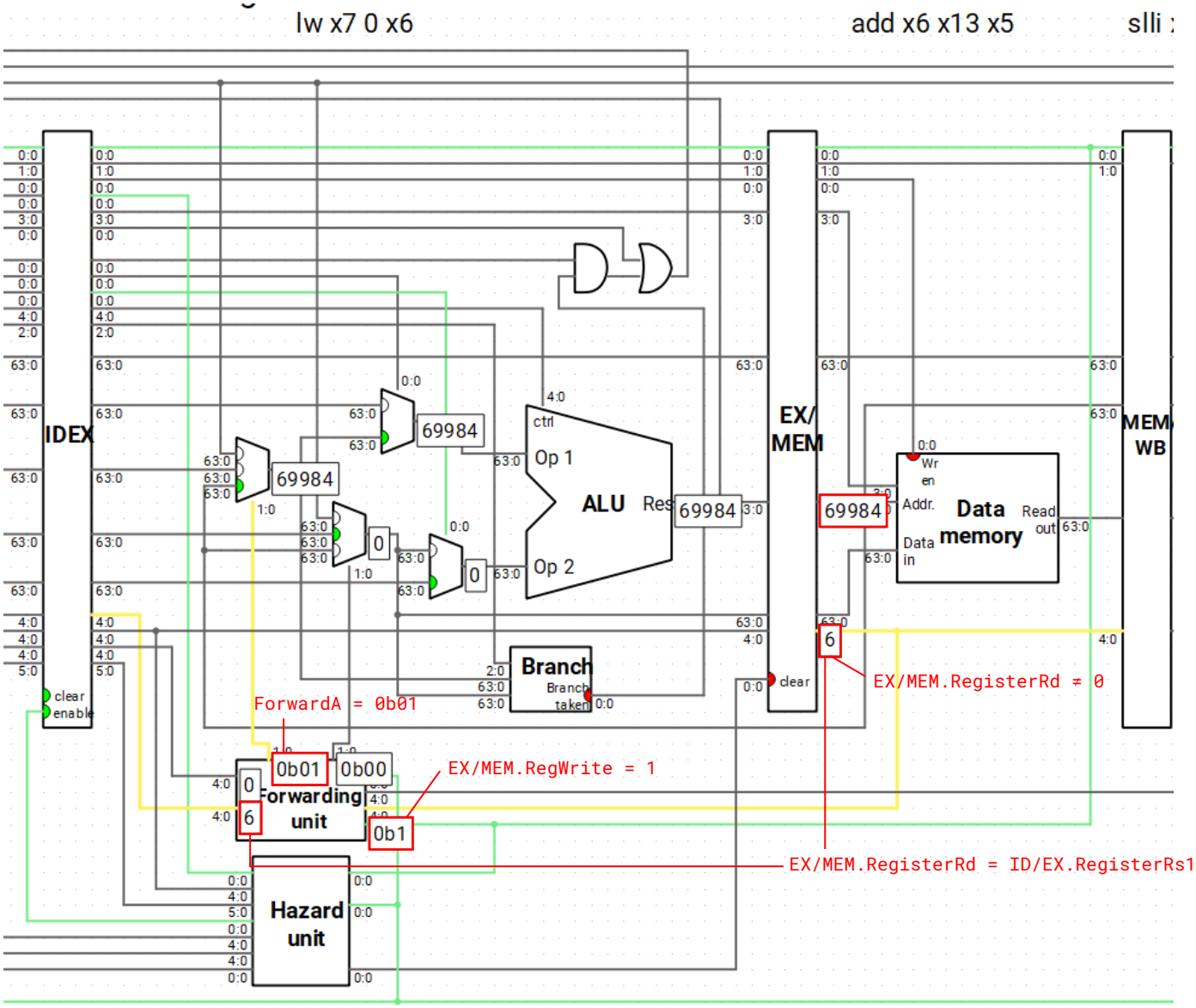
Flowchart of the assembly code



2. Hazards Analysis

Type (1): R-types RAW (read after write) at the following 1st instruction.

```
64   add t1, a3, t0
65   lw t2, 0(t1)
```



對於 Type (1) hazard，hazard detection unit 偵測 EX 與 ID stages 的連續指令，此時 EX stage 為 `lw t2, 0(t1)`，MEM stage 為 `add t1, a3, t0`，偵測 EX hazard 條件：

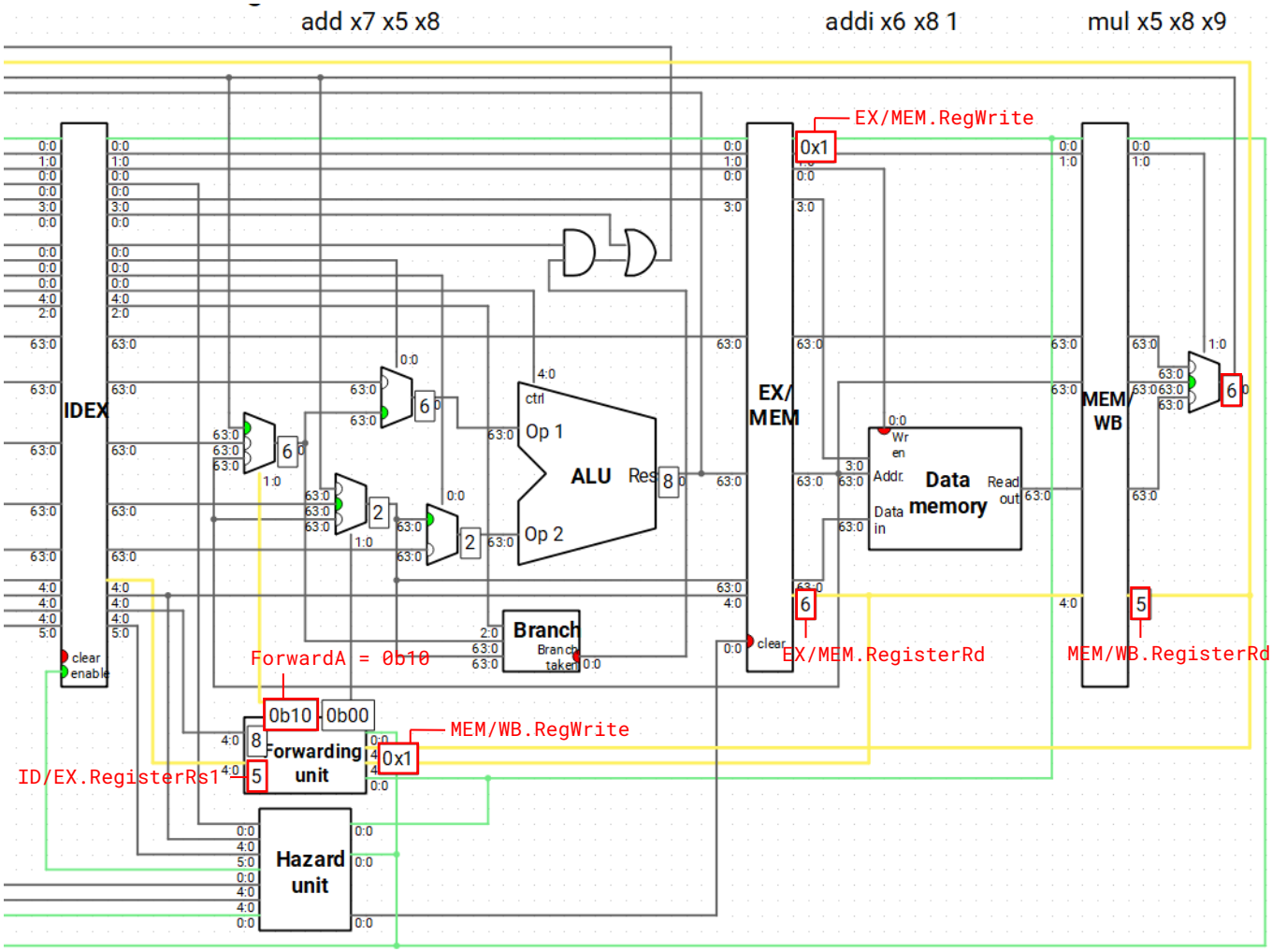
step 1: `EX/MEM.RegWrite = 0b1`
step 2: `EX/MEM.RegisterRd ≠ 0`
step 3: `EX/MEM.RegisterRd = ID/EX.RegisterRs1 = 6`

設定

`ForwardA = 0b01`。

Type (2): R-types RAW at the following 2nd instruction.

```
29   mul t0, s0, s1
30   addi t1, s0, 1
31   add t2, t0, s0
```



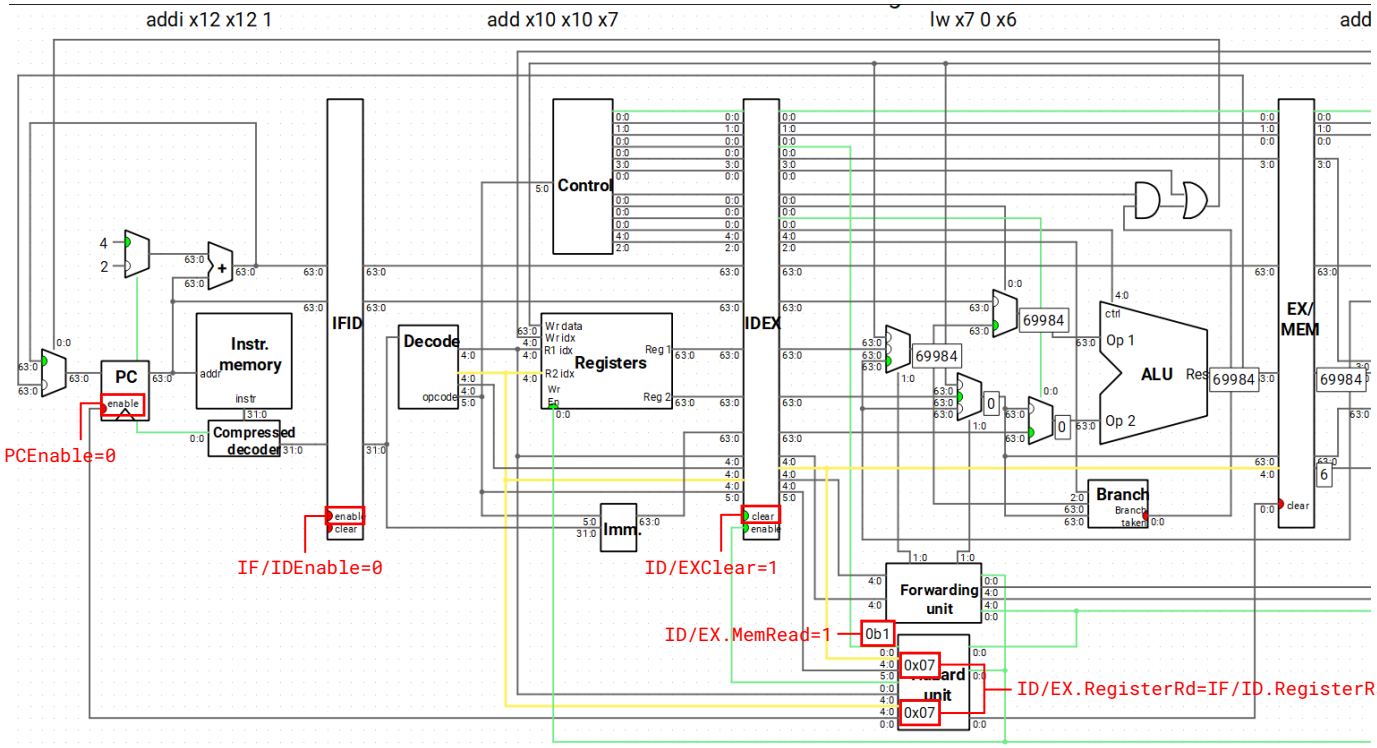
對於 Type (2) hazard，由於作業 code 中並未出現此種 hazard，故以另外一段 code 做解釋。此時 WB stage 為 `mul t0, s0, s1`，MEM stage 為 `addi t1, s0, 1`，EX stage 為 `add t2, t0, s0`，偵測 MEM hazard 條件：

step 1: `MEM/WB.RegWrite = 1`
step 2: `MEM/WB.RegisterRd ≠ 0`
step 3: `not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)`
`and(EX/MEM.RegisterRd = ID/EX.RegisterRs1))`
step 4: `MEM/WB.RegisterRd = ID/EX.RegisterRs1 = 5`

設定 `ForwardA = 0b10`。

Type (3): Load RAW at the following 1st instruction

```
65   lw t2, 0(t1)
66   add a0, a0, t2
```

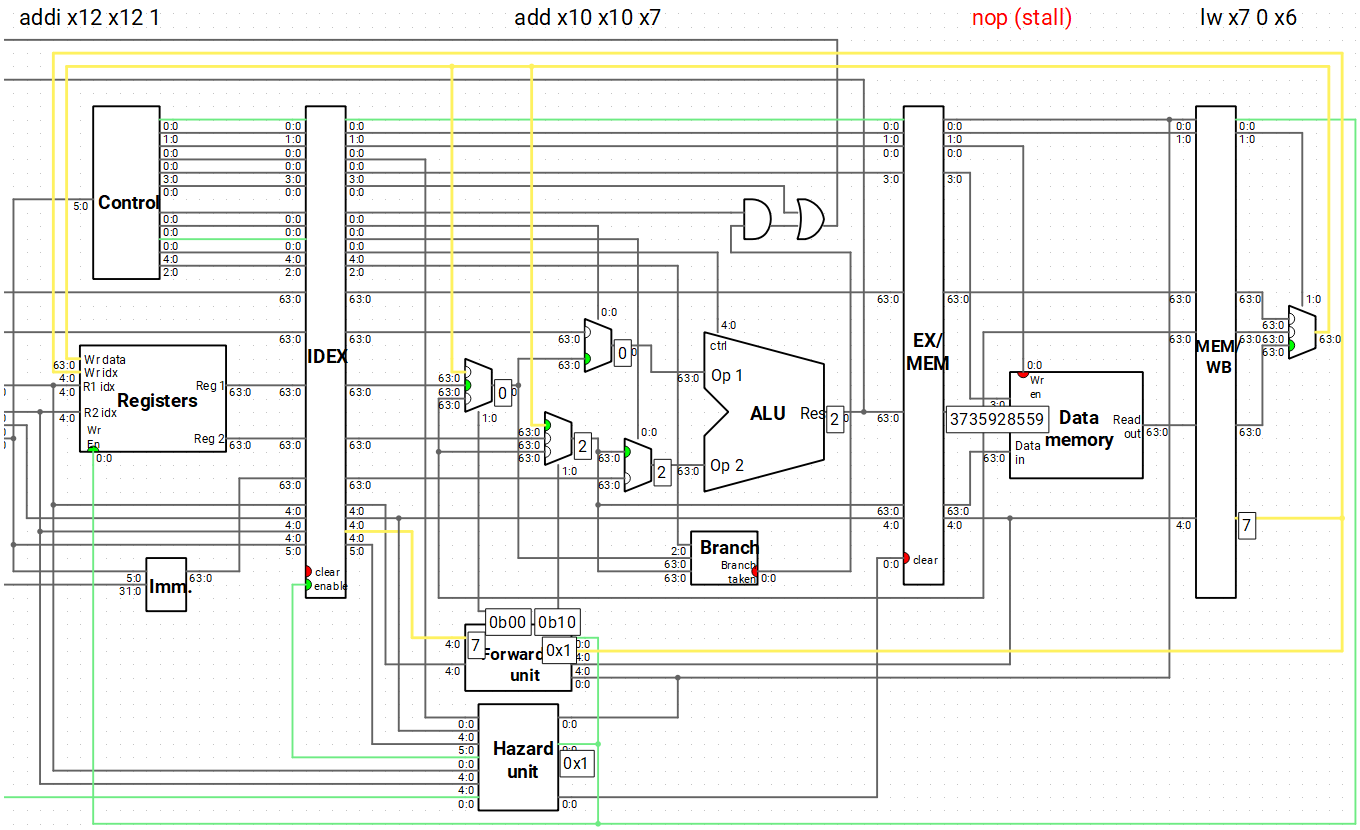


對於 Type (3) hazard，hazard detection unit 偵測 EX 與 ID stages 的連續指令，此時 EX stage 為 `lw t2,0(t1)`，ID stage 為 `add a0, a0, t2`，偵測條件：

step 1: `ID/EX.MemRead = 0b01`
step 2: `ID/EX.RegisterRd = IF/ID.RegisterRs2 = 0x07`

條件成立，則設定

`ID/EXClear = 1` (Turn EX into NOP in next cycle)
`IF/IDEnable = 0, PCEnable = 0` (Prevent update of PC and IF/ID)。



經過兩個 clock cycle 後，WB stage 為 `lw t2,0(t1)`，MEM stage 為 nop，EX stage 為 `add a0, a0, t2`，利用 MEM hazard Forwarding 的條件：

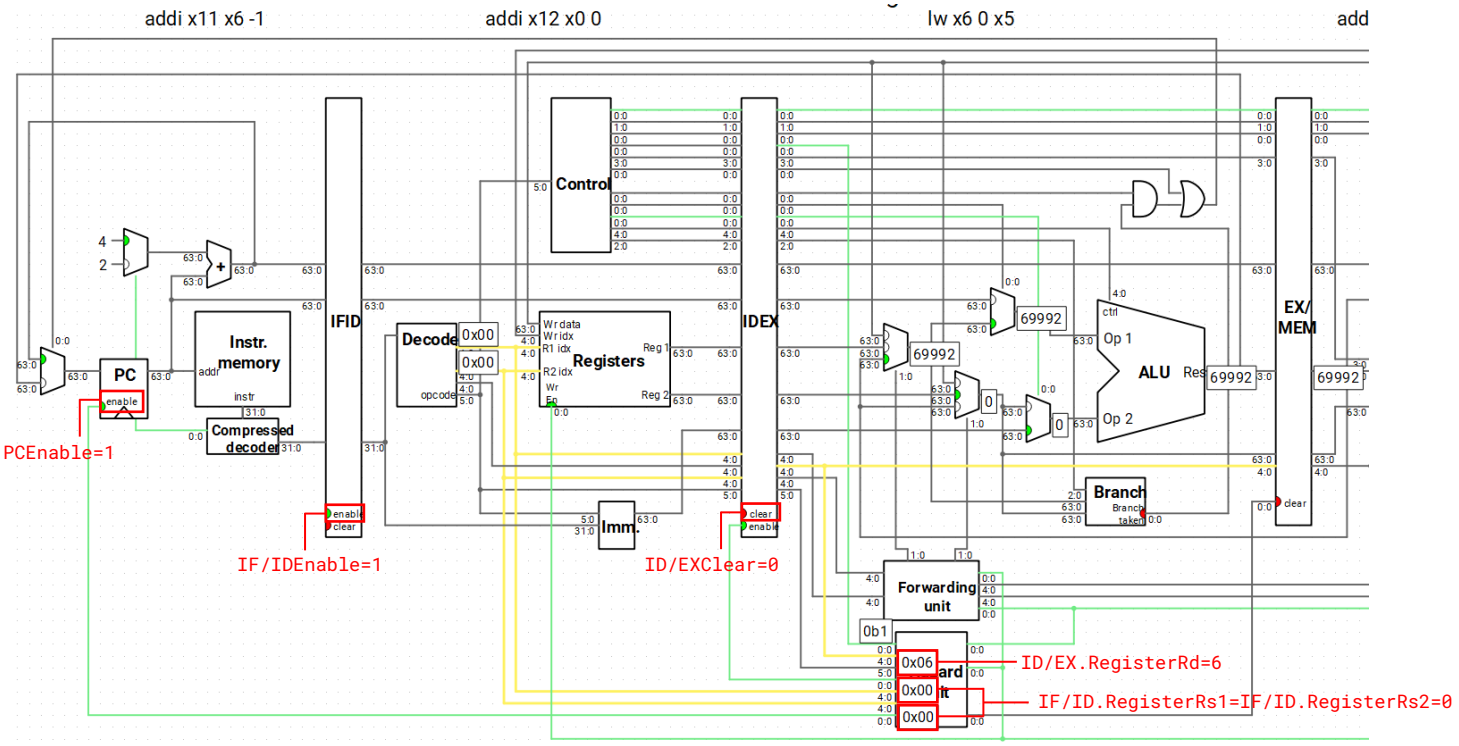
step 1: `MEM/WB.RegWrite = 1`
step 2: `MEM/WB.RegisterRd ≠ 0`
step 3: `not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)`
`and(EX/MEM.RegisterRd = ID/EX.RegisterRs2))`
step 4: `MEM/WB.RegisterRd = ID/EX.RegisterRs2 = 7`

設定

`ForwardB = 0b10`。

Type (4): Load RAW at the following 2nd instruction.

```
33    lw t1, 0(t0)
34    addi a2, zero, 0
35    addi a1, t1, -1
```

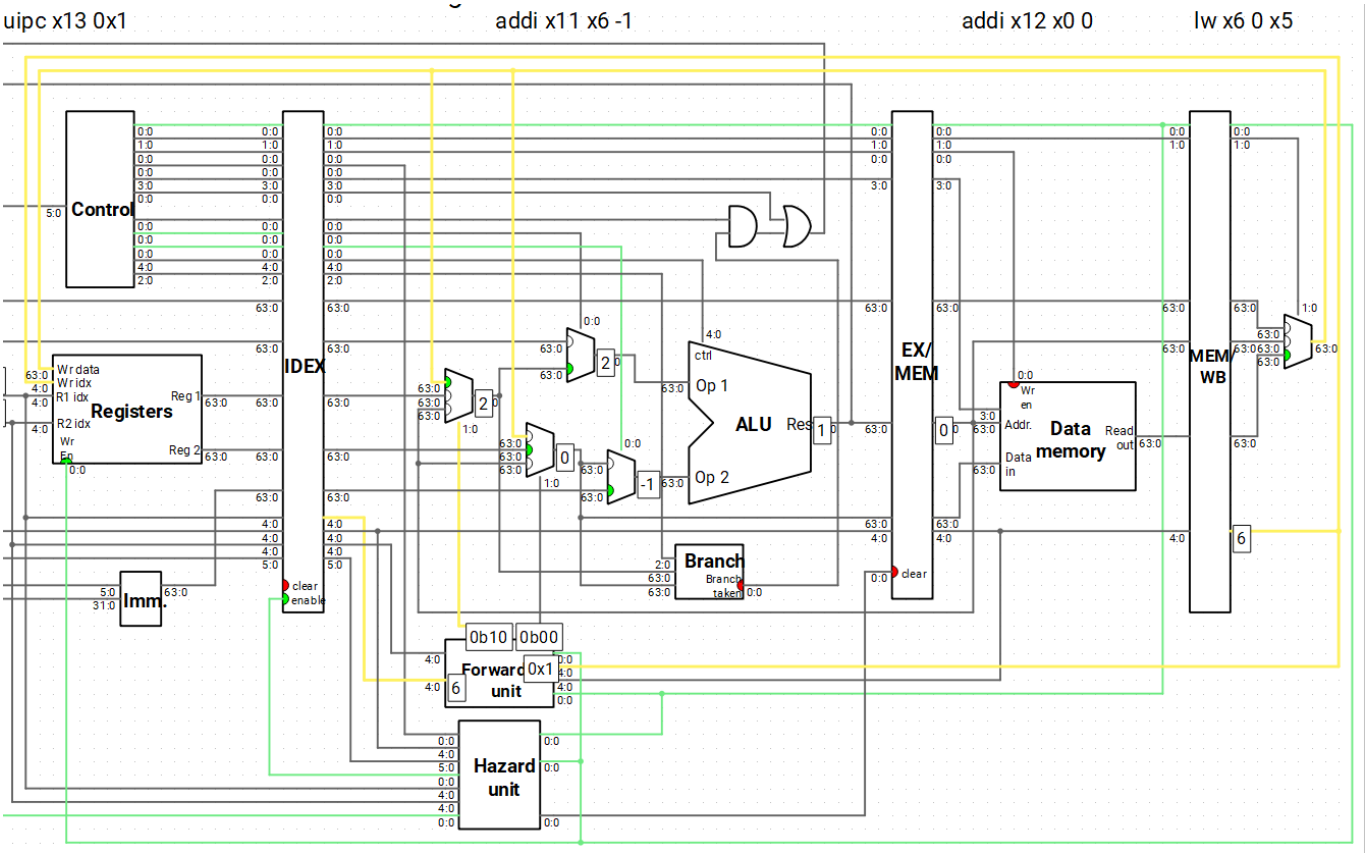


對於 Type (4) hazard，hazard detection unit 偵測 EX 與 ID stages 的連續指令，此時 EX stage 為 `lw t1,0(t0)`，ID stage 為 `addi a2, zero, 0`，IF stage 為 `addi a1, t1, -1`，偵測條件：

step 1: `ID/EX.MemRead = 0b01`
step 2: `IF/ID.RegisterRs1 = IF/ID.RegisterRs2 = 0`
`≠ ID/EX.RegisterRd`

load-use hazard 條件不成立，則設定

`ID/EXClear = 0` (No `stall` in next cycle)
`IF/IDEnable = 1, PCEnable = 1` (will update PC and IF/ID)。



經過兩個 clock cycle 後，此時 WB stage 為 `lw t1,0(t0)`，MEM stage 為 `addi a2, zero, 0`，EX stage 為 `addi a1, t1, -1`，利用 MEM hazard Forwarding 的條件：

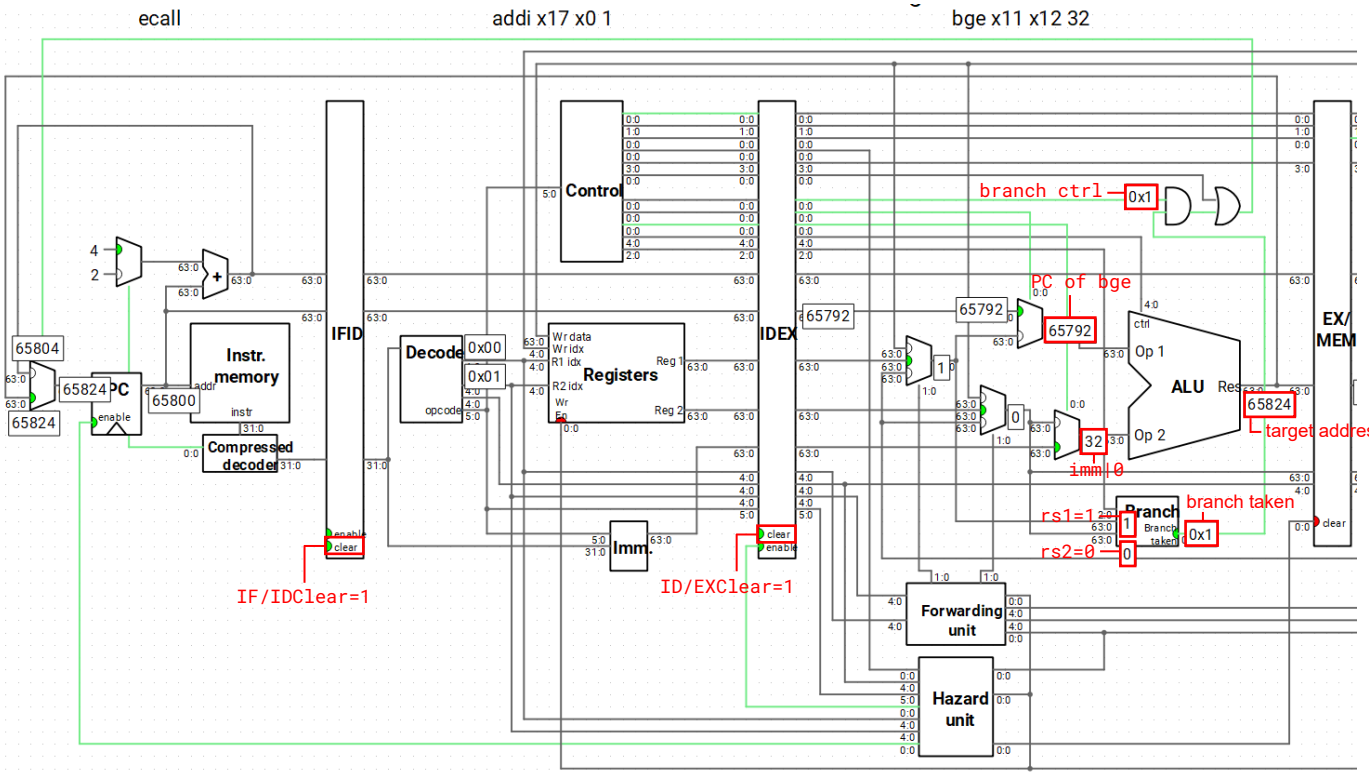
step 1: `MEM/WB.RegWrite = 1`
step 2: `MEM/WB.RegisterRd ≠ 0`
step 3: `MEM/WB.RegisterRd = ID/EX.RegisterRs1 = 6`
step 4: `not(EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)`
`and(EX/MEM.RegisterRd = ID/EX.RegisterRs2))`

設定

`ForwardA = 0b10`。

Type (5): Branch instruction (control hazard)

```
49    bge a1, a2, L1
50
51    #Print sum (in a0) by setting ecall argument to 1
52    li a7, 1
53    ecall
54    #Print delimiter string by setting ecall argument to 4
55    la a0, space
56    li a7, 4
57    ecall
58
59    addi sp, sp, 20
60    jalr zero, 0(ra)
61
62  L1: # Subset including arr[l]
```



對於 Type (5) hazard，Ripes pipeline simulator 在 EX stage 偵測 control hazard，此時 EX stage 為 `bge a1, a2, L1`，ID stage 為 `addi a7, zero, 1`，IF stage 為 `ecall`。Branch unit 進行 `bge a1, a2, L1` 的比較，確認為 branch taken 後 (`rs1 ≥ rs2`)，則設定

target address: `65792 + 32 = 65824`
`IF/IDClear = 1, ID/EXClear = 1`
(Turn ID and EX into NOPs in next clock cycle)。