Adder-Input: 32'd 4;
Clk
reset

PC_Write_Enable , W38

Mux_Output, W16

IF_IO_Enable , W4

IF_IO_RS W5 ,5

IF_IO_RT W6 ,5

IO-EX-RT A8 5

Hazard Unit
Module

A9 IO-EX-MemRead

W34 2 ALU-OP Out

W32 Reg Dst- Out

W27, request

W24, Mem to reg

Clk reset   Reg_Write_Output,
Mem to reg , A2

W1

W12
32

Addr

Adder-Input

W12
32

PC Counter
Output

enable clk

reset

5,W5   RS- Hazard

5,W6   RT- Hazard

W17 Reg Dst

W18 Branch

W19 Mem Read

W20 Mem to Reg

W21 MemWrite

W23 Reg Write

W26 ALUOp

2 W45 ALUSrc

Control

W16

Input - Hazard

W31 Mem_Write-Out

W30 Mem_Read-Out

W27 Register-Out

W48 ALU-Src-Out

W26 Mem to Reg

Mux
Control
Module

W30, MemRead

W31, MemWrite

W32, Reg Dst

W34, ALU_OP 2

W48, ALU-Src, W49

Mem Read , A4

Mem Write   A5

Reg Dst A20

2   ALU_OP A6

ALU-Src, W49

CLK

program
counter

W4

Instruction
memory

W1
32

W1
32

W2
32

W2

W2
W2

32
W2

W32

W7 PC-Counter- Output

W3 Op- Code

W8 Read- Reg - 1

W9 Read- Reg - 2

W100
26 bit Jump- Offset

IF_IO

Addr-32-Bit

Out

W40
32

32
W41

W3

W41 32

Shift
left
two

W11 32

Jump

W104

W18

W100

CLK   require

Register
Module

Read-reg-1

W8

W4 Read-reg-2

W15 Write-Reg
5

A12 Write
reg
32

W36

Read-data-1, W46

Read-data-2, W47

32 Read-data-1, W46

32   read-data-2, w47

32

Read-data-1, A12

Read-data-2, W17

32   sign extend A 11

W2
32

mux-1-1

W1b

W102
32

a

b   out

32 W34   mux

load

control

W104

W29

W105

W105
32

16 W10 Sign extend input

5 W13 IF-IO-RS

5 W14   ↓   RT

5 W15   ↓   RD

16
W10

Sign
Extd

16

32
W11

W11

reset

Flush-Branch W16

26
W100

Flush-Jump

W104

Shift
left 2

W101

W7   32

28
W101   Concatenation-
26-32

32 W102

W100

W16
32

32
W40

W40

W2
32

32

32

W102
32

sign extend input

IF_IO - RS W13 5

↓ _RT W14 5

↓ _Rd W15 5

IO_EX

IO- EX-RS-For- A
↓ R2 For, A

IO-EX-RT_ MUX
Rd - MUX,

A1

A12, 32 — a
ℓ12 — b
ℓ2 — c out — A15
mux_module
2 ↓ A13

A15, 32 — Input-1 ALU Output — A10, 32
W44, 32 — Input-2

A10, ALU-Result, 32

A12, RT 32 Input, 32

Control Input — 4, A9

A7 — a
ℓ14 — b
ℓ5 — c out — A16
mem-1-module
2 ↓ A14

A14, 32 — a
A11, 32 — b — W44, 32

A11, 32 — Function Code — 4, A9
A6, 2 — ALU-OP Input — ALU OP output — W44

Forward-A, A13   Forward-B, A14

A7, 5 — ID-EX-RS
A6, 5 — ID-EX-RT Forwarding

A4X, 5 — a
AM, 5 — b out — 5, A21
A20

Reg Write, A1
Mem to Reg, A3
Mem Read, A4
Mem Write, A5

CLK Reset

Reg Write, ℓ5
Mem to Reg, ℓ6
Mem Read, ℓ1
Mem Write, ℓ2

ALU-Result, ℓ3

32 — RT_32_Output, ℓ17
32

EX_MEM

EX_Mem-Forwarding, ℓ9
EX_Mem-Next-Pipeline, W98

A2, ID-EX_Mux, 5

EX_MEM-RD, ℓ9
Mem-WB-RD, ℓ15
EX-Mem-Reg-Write, ℓ5
MEM-WB-Reg-Write, W36

ℓ2 ℓ1
Mem Write Reset Mem Read clk

ℓ3, 32 Address-Input data? ℓ10
output 32

ℓ17, 32 Write-data

data—
memory-output
ℓ10

ALU-Output
ℓ3

ℓ5, Reg-Write
ℓ6, Mem to Reg

CLK

Reg-Write-Output, ℓ36
Mem to Reg Output, ℓ11

MEM-WB

32 data-memory-Mux, ℓ13
ALU-Output-to-mux, ℓ13

MEM-WB-Rd-Register-Module, W37
5

W98, EX_Mem-RD5

MEM-WB-Forwarding, ℓ15

ℓ11

ℓ13 — a
ℓ14 — b out — ℓ11
MUX_Last
Control

ALU: Missing Flag output → might not matter
MeM-WB-Module1 → missing Mem to Reg-Output-Forwarding