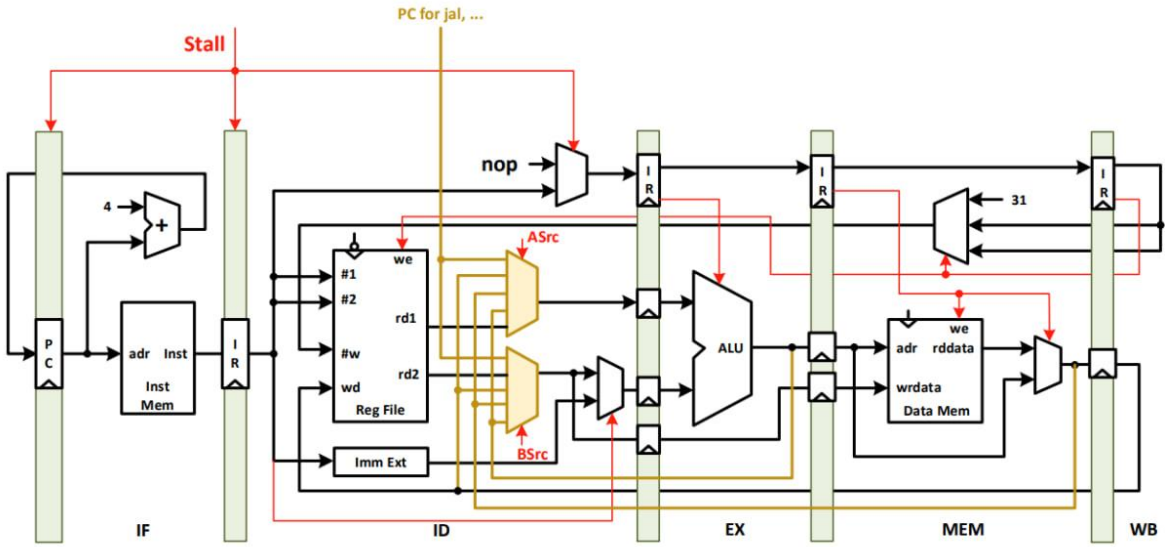
In the name of God

CA2

Advance Computer Architecture

Mohammad Moein Joneidi Jafaari 810100113

Mohammad Yahyapour 810100234



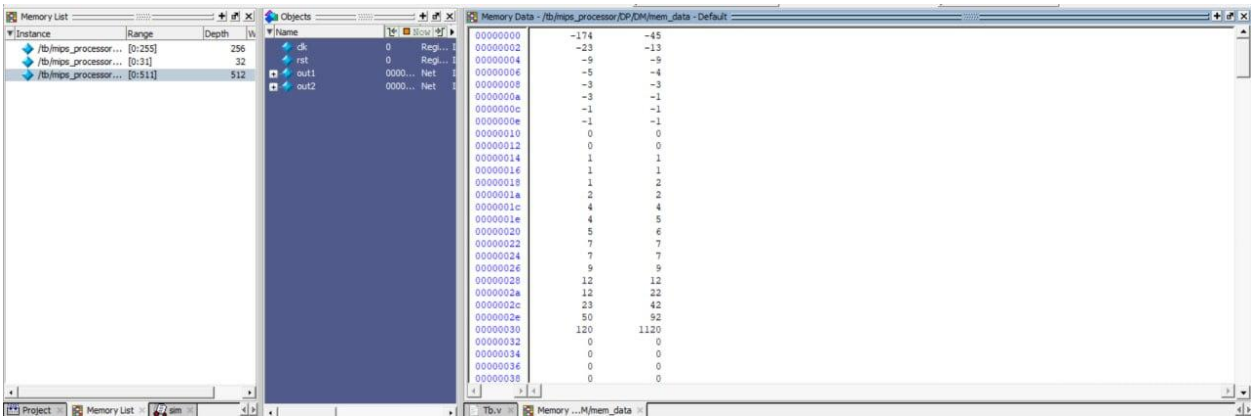
datapah

```
21  always @(*) begin
22      ASrc = 3'b000;
23      BSrc = 3'b000;
24
25      if (RegWrite_E && (write_reg_E != 5'd0) && (write_reg_E == rs_D) && (MemtoReg_E == 1'b0)) ASrc = 3'b001;
26      if (RegWrite_E && (write_reg_E != 5'd0) && (write_reg_E == rt_D) && (MemtoReg_E == 1'b0)) BSrc = 3'b001;
27
28      if ((ASrc == 3'b000) && RegWrite_M && (write_reg_M != 5'd0) && (write_reg_M == rs_D) && (MemtoReg_M == 1'b1)) ASrc = 3'b010;
29      if ((BSrc == 3'b000) && RegWrite_M && (write_reg_M != 5'd0) && (write_reg_M == rt_D) && (MemtoReg_M == 1'b1)) BSrc = 3'b010;
30
31      if ((ASrc == 3'b000) && RegWrite_M && (write_reg_M != 5'd0) && (write_reg_M == rs_D) && (MemtoReg_M == 1'b0)) ASrc = 3'b011;
32      if ((BSrc == 3'b000) && RegWrite_M && (write_reg_M != 5'd0) && (write_reg_M == rt_D) && (MemtoReg_M == 1'b0)) BSrc = 3'b011;
33
34      if ((ASrc == 3'b000) && RegWrite_W && (write_reg_W != 5'd0) && (write_reg_W == rs_D)) ASrc = 3'b100;
35      if ((BSrc == 3'b000) && RegWrite_W && (write_reg_W != 5'd0) && (write_reg_W == rt_D)) BSrc = 3'b100;
36
37      if ((ASrc == 3'b000) && DataC_W && (rs_D == 5'd31)) ASrc = 3'b101;
38      if ((BSrc == 3'b000) && DataC_W && (rt_D == 5'd31)) BSrc = 3'b101;
39  end
```

Forward Logic

```
37      assign re1_D =  (opcode == `RT)|
38                      (opcode == `addi)|
39                      (opcode == `addiu)|
40                      (opcode == `slti)|
41                      (opcode == `sltiu)|
42                      (opcode == `andi)|
43                      (opcode == `ori)|
44                      (opcode == `xori)|
45                      (opcode == `lui)|
46                      (opcode == `lw)|
47                      (opcode == `sw)|
48                      (opcode == `beq)|
49                      (opcode == `bne)|
50                      (opcode == `j)|
51                      (opcode == `jal);
52
53      assign re2_D = (opcode == `RT)|(opcode == `sw);
54
55      assign stall = ((rs == ws_E) & (MemRead_E == 1'b1) & (ws_E != 1'b0) & re1_D) |
56                     ((rt == ws_E) & (MemRead_E == 1'b1) & (ws_E != 1'b0) & re2_D);
```

Hazard Logic



Result