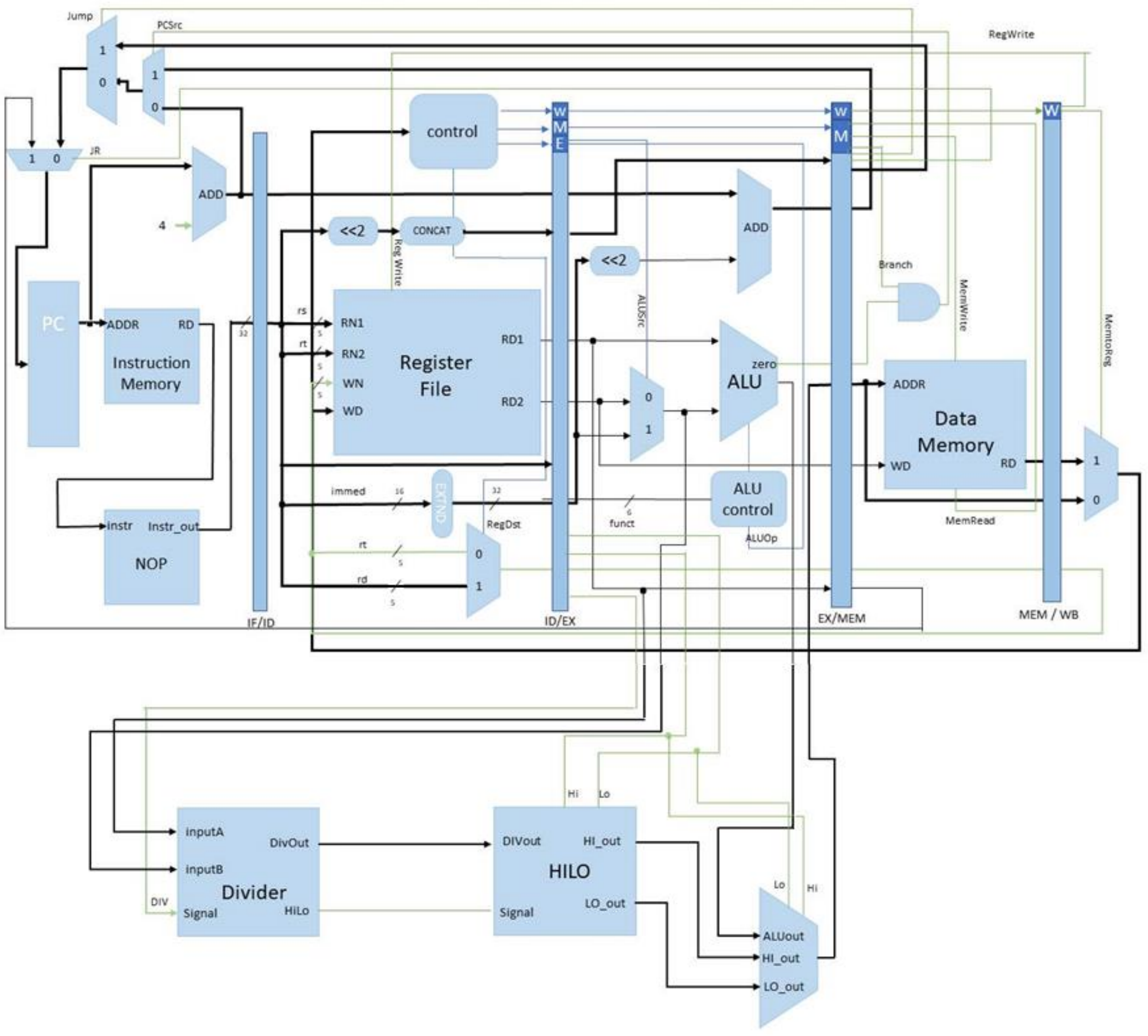
5-Stage-Pipeline-CPU 報告

負責部分 ：code、架構圖、Waveform 驗證結果

一、架構圖



二、程式說明

1. IF
   計算 PC 並抓取指令，且判斷是否需要 NOP，若需要 NOP 就將指令輸出為 32bits 的 0，若不需要就輸出抓到的指令與 PC+4 的值
2. IF_ID
   暫存從 IF 傳來的 PC+4 與指令
3. Sign_extend
   將 16 bits 的 input extend 至 32 bits
4. Reg_file
   判斷 RN1 和 RN2，把 RD1 和 RD2 放好，如果 RN1 或 RN2 為 0，就將 RD1 或 RD2 歸零，如果有東西就寫入 RD1 和 RD2。判斷從 memory 是否有寫東西回來，有的話就寫入 WN
5. Control_single
   定義每一指令的訊號，並且定義每一指令的控制訊號
6. ID
   將讀入的指令的切成一塊一塊並且 assign 給分別該有的 wire，例如 rs, rt, ,opcode 等等，先使用 Sign_extend，然後使用剛剛切好並放入的東西寫入 reg_file，最後使用 controll_single 定義訊號，並選擇 WN 的數值
7. ID_EX
   先判斷是否 reset，如果是，將所有東西歸零，接下來判斷指令，並且寫好所有的控制訊號，且暫存 ID 所傳入的值
8. EX
   負責判斷從不同地方傳來的值該如何處理的地方。EX 裡的元件有 Divider、HILO、ALU、ALU control、ADD 及兩個 MUX。Divider 計算完後將資料存入 HILO，需要時再使用 mfhi 及 mflo 指令使用。ALU control 控制 ALU 應使用的功能。MUX 決定要輸出哪個訊號。
9. EX_MEM
   先暫存從 EX 傳來的訊號，再傳送到 Memory 區域
10. Memory
    存取資料的地方，控制資料的讀入及寫入記憶體。接收 control signal 的訊號決定 MEM_WRITE、MEM_READ 的訊號。此區域也連接 Branch，不須使用記憶體時，像 jump 指令時可使用 Branch 將訊號送回 IF 區。
11. MEM_WB
    暫存從 Memory 傳來的訊號，再傳送到 WB 區域
12. WB
    接收訊號後，接收 control signal 的訊號確認 MemtoReg 的訊號來判斷是否需要將資料寫回 Register File，若需要就寫回。

三、Waveform 驗證結果

輸入指令為

slt $s3, $s2, $s4

j   3

beq, $s1, $s4, 2

lw $t7, $s1, 0

beq $s1, $s1, 4

add $s2, $s0, $s2

sub $s2, $s0, $s2

add $s1, $s0, $s1

add $s1, $s0, $s1

divu, $t0, $a2

or $s2, $s0, $s2

mfhi, $s6

mflo, $s5

add $s1, $s0, $s1

sub $s2, $s0, $s2

ori $s3, $s2, 4

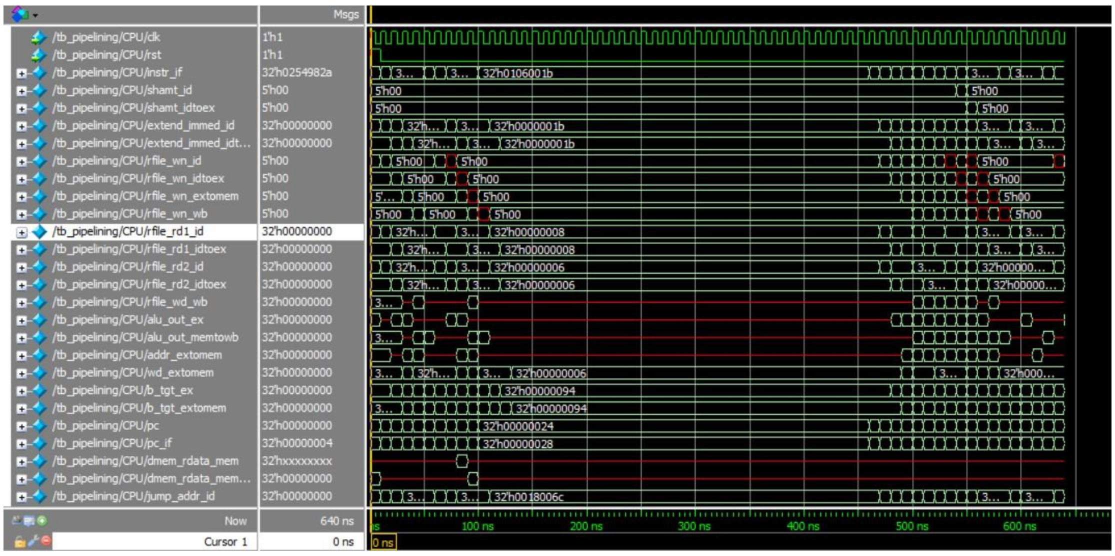
sw $zero, $s2, 24

srl $s3, $s7, 2

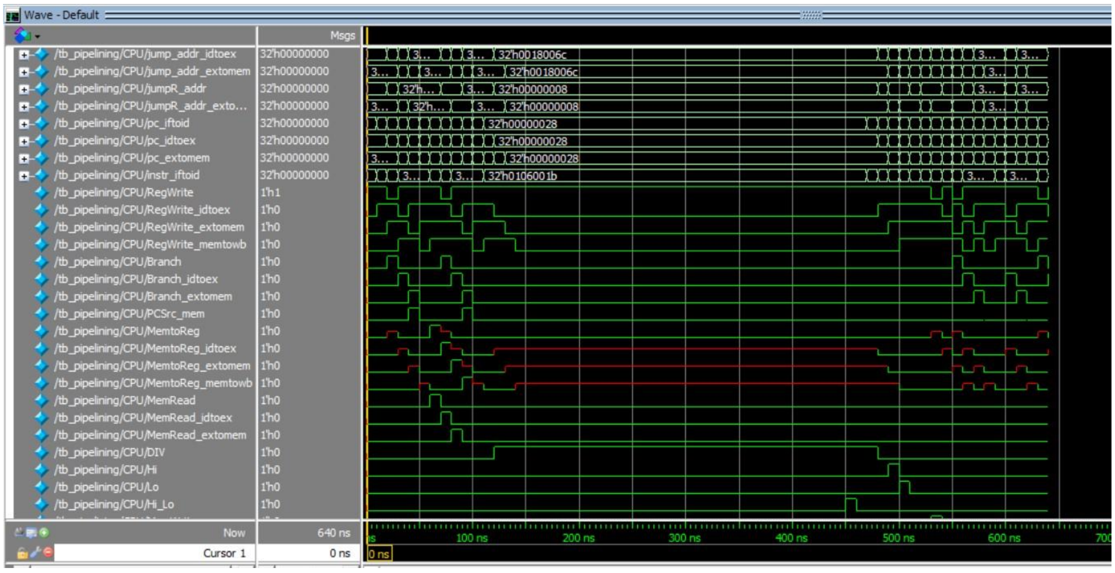
beq, $s1, $s4, 2

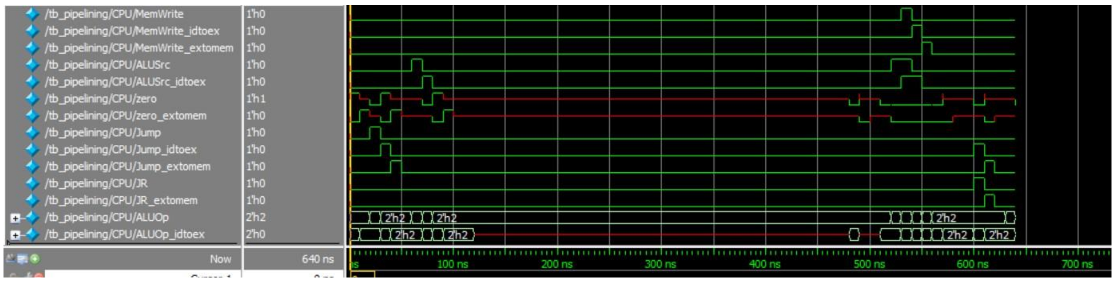
jr, $s7

Waveform：



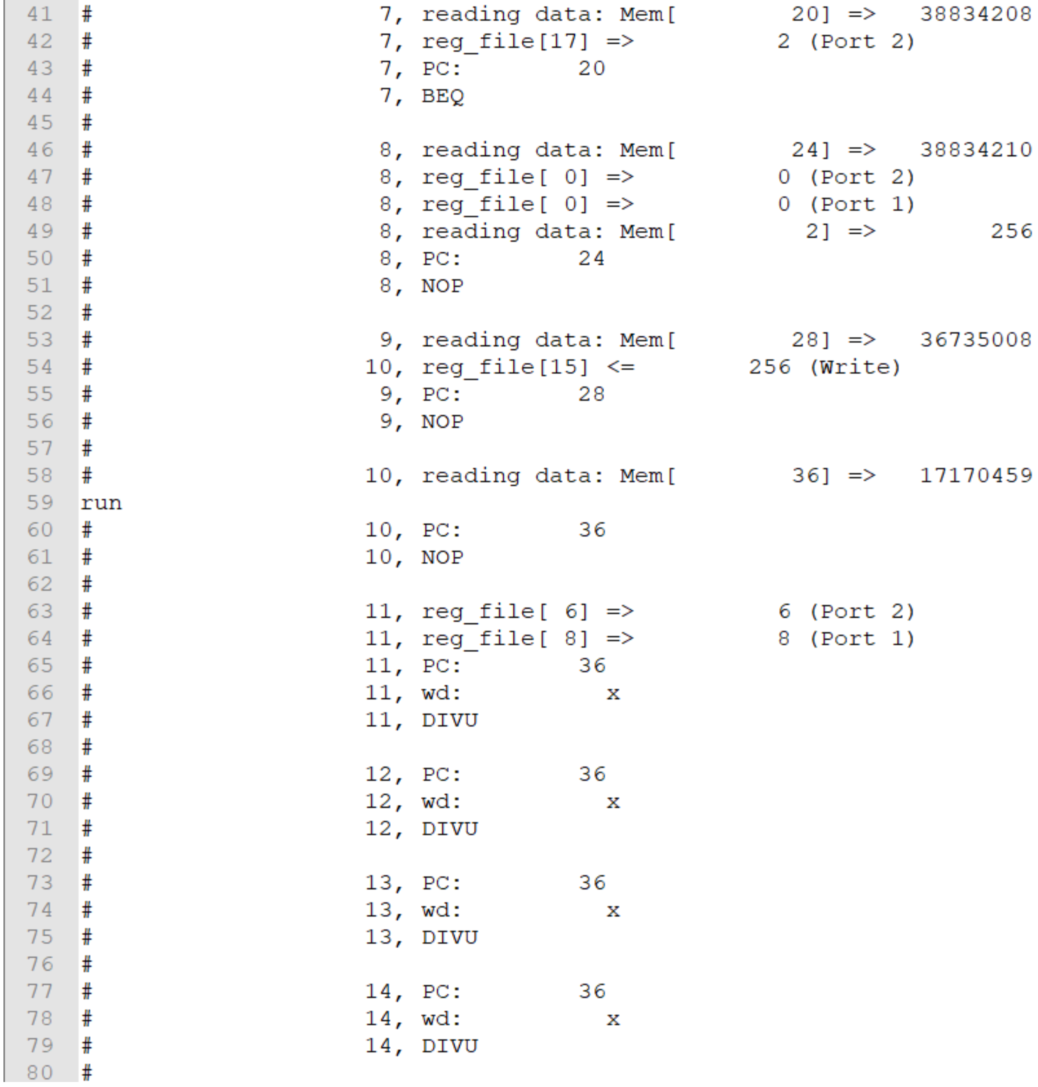
```
                                              Msgs
/tb_pipelining/CPU/clk                        1'h1
/tb_pipelining/CPU/rst                        1'h1
/tb_pipelining/CPU/instr_if                   32'h0254982a
/tb_pipelining/CPU/shamt_id                   5'h00
/tb_pipelining/CPU/shamt_idtoex               5'h00
/tb_pipelining/CPU/extend_immed_id            32'h00000000
/tb_pipelining/CPU/extend_immed_idt...        32'h00000000
/tb_pipelining/CPU/rfile_wn_id                5'h00
/tb_pipelining/CPU/rfile_wn_idtoex            5'h00
/tb_pipelining/CPU/rfile_wn_extomem           5'h00
/tb_pipelining/CPU/rfile_wn_wb                5'h00
/tb_pipelining/CPU/rfile_rd1_id               32'h00000000
/tb_pipelining/CPU/rfile_rd1_idtoex           32'h00000000
/tb_pipelining/CPU/rfile_rd2_id               32'h00000000
/tb_pipelining/CPU/rfile_rd2_idtoex           32'h00000000
/tb_pipelining/CPU/rfile_wd_wb                32'h00000000
/tb_pipelining/CPU/alu_out_ex                 32'h00000000
/tb_pipelining/CPU/alu_out_memtowb            32'h00000000
/tb_pipelining/CPU/addr_extomem               32'h00000000
/tb_pipelining/CPU/wd_extomem                 32'h00000000
/tb_pipelining/CPU/b_tgt_ex                   32'h00000000
/tb_pipelining/CPU/b_tgt_extomem              32'h00000000
/tb_pipelining/CPU/pc                         32'h00000000
/tb_pipelining/CPU/pc_if                      32'h00000004
/tb_pipelining/CPU/dmem_rdata_mem             32'hxxxxxxxx
/tb_pipelining/CPU/dmem_rdata_mem...          32'h00000000
/tb_pipelining/CPU/jump_addr_id               32'h00000000
Now                                           640 ns
Cursor 1                                      0 ns
```


Wave - Default
Msgs
/tb_pipelining/CPU/jump_addr_idtoex 32'h00000000
/tb_pipelining/CPU/jump_addr_extomem 32'h00000000
/tb_pipelining/CPU/jumpR_addr 32'h00000000
/tb_pipelining/CPU/jumpR_addr_exto... 32'h00000000
/tb_pipelining/CPU/pc_iftoid 32'h00000000
/tb_pipelining/CPU/pc_idtoex 32'h00000000
/tb_pipelining/CPU/pc_extomem 32'h00000000
/tb_pipelining/CPU/instr_iftoid 32'h00000000
/tb_pipelining/CPU/RegWrite 1'h1
/tb_pipelining/CPU/RegWrite_idtoex 1'h0
/tb_pipelining/CPU/RegWrite_extomem 1'h0
/tb_pipelining/CPU/RegWrite_memtowb 1'h0
/tb_pipelining/CPU/Branch 1'h0
/tb_pipelining/CPU/Branch_idtoex 1'h0
/tb_pipelining/CPU/Branch_extomem 1'h0
/tb_pipelining/CPU/PCSrc_mem 1'h0
/tb_pipelining/CPU/MemtoReg 1'h0
/tb_pipelining/CPU/MemtoReg_idtoex 1'h0
/tb_pipelining/CPU/MemtoReg_extomem 1'h0
/tb_pipelining/CPU/MemtoReg_memtowb 1'h0
/tb_pipelining/CPU/MemRead 1'h0
/tb_pipelining/CPU/MemRead_idtoex 1'h0
/tb_pipelining/CPU/MemRead_extomem 1'h0
/tb_pipelining/CPU/DIV 1'h0
/tb_pipelining/CPU/Hi 1'h0
/tb_pipelining/CPU/Lo 1'h0
/tb_pipelining/CPU/Hi_Lo 1'h0
32'h0018006c
32'h00000008
32'h00000028
32'h0106001b
Now 640 ns
Cursor 1 0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns

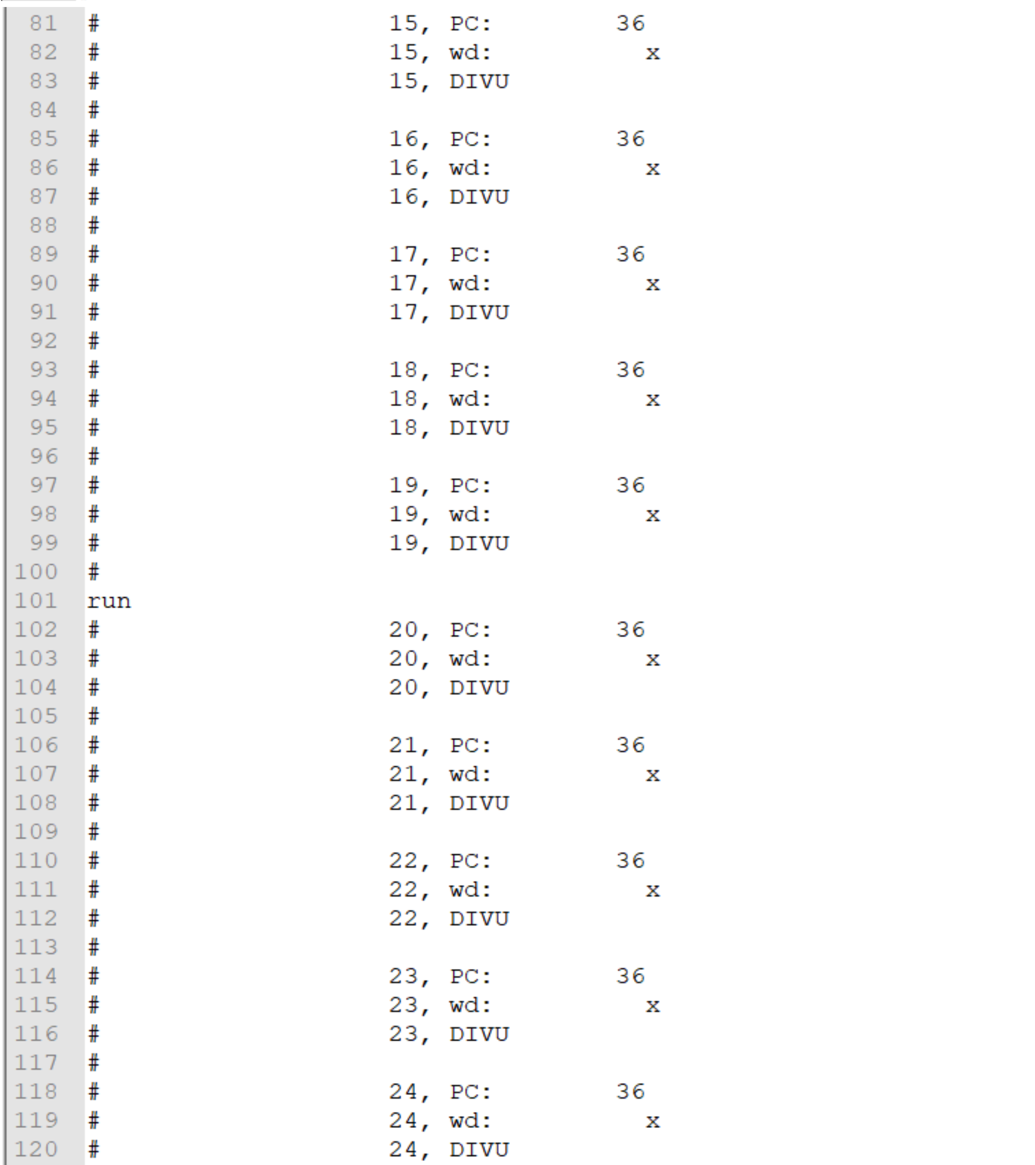

/tb_pipelining/CPU/MemWrite 1'h0
/tb_pipelining/CPU/MemWrite_idtoex 1'h0
/tb_pipelining/CPU/MemWrite_extomem 1'h0
/tb_pipelining/CPU/ALUSrc 1'h0
/tb_pipelining/CPU/ALUSrc_idtoex 1'h0
/tb_pipelining/CPU/zero 1'h1
/tb_pipelining/CPU/zero_extomem 1'h0
/tb_pipelining/CPU/Jump 1'h0
/tb_pipelining/CPU/Jump_idtoex 1'h0
/tb_pipelining/CPU/Jump_extomem 1'h0
/tb_pipelining/CPU/JR 1'h0
/tb_pipelining/CPU/JR_extomem 1'h0
/tb_pipelining/CPU/ALUOp 2'h2
/tb_pipelining/CPU/ALUOp_idtoex 2'h0
2'h2
Now 640 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns

驗證結果：

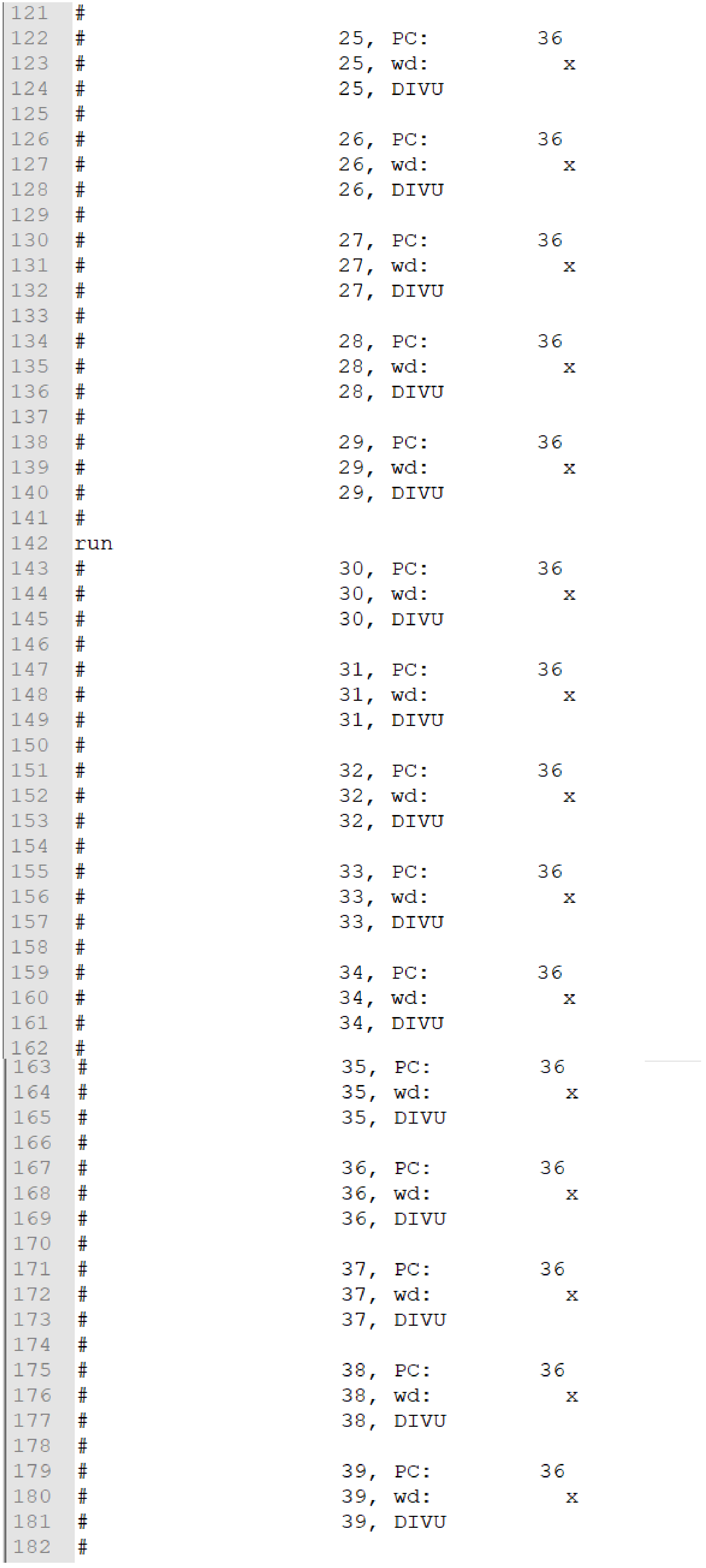
```
#                    0, reading data: Mem[         x] =>          x
# 18446744073709551615, PC:          x
#                    0, reading data: Mem[         0] =>   39098410
#                    0, reg_file[ 0] =>          0 (Port 2)
#                    0, reg_file[ 0] =>          0 (Port 1)
#                    0, PC:          0
#                    0, NOP
#
#                    1, reading data: Mem[         4] =>  134217731
#                    1, reg_file[20] =>          5 (Port 2)
#                    1, reg_file[18] =>          3 (Port 1)
#                    1, PC:          4
#                    1, wd:           0
#                    1, SLT
#
#                    2, reading data: Mem[         8] =>  305397762
#                    2, reg_file[ 0] =>          0 (Port 2)
#                    2, reg_file[ 0] =>          0 (Port 1)
#                    2, PC:          8
#                    2, J
#
#                    3, reading data: Mem[        12] => 2385444864
#                    3, PC:         12
#                    3, NOP
#
#                    4, reading data: Mem[        16] =>  305201156
#                    5, reg_file[19] <=          1 (Write)
#                    4, PC:         16
#                    4, NOP
#
#                    5, reading data: Mem[        12] => 2385444864
#                    5, PC:         12
#                    5, NOP
#
#                    6, reading data: Mem[        16] =>  305201156
#                    6, reg_file[15] =>         21 (Port 2)
#                    6, reg_file[17] =>          2 (Port 1)
#                    6, PC:         16
#                    6, LW
#
```



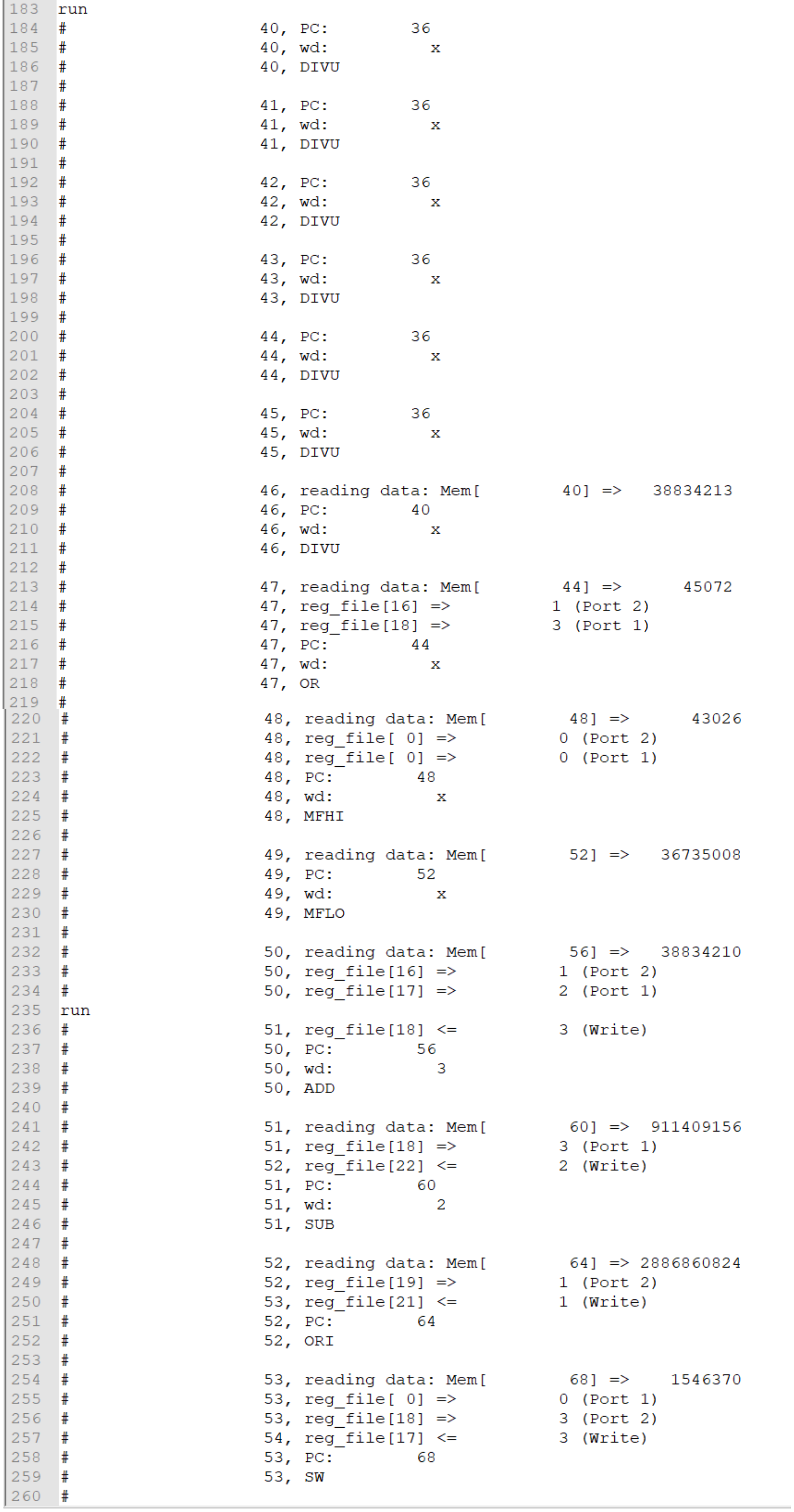
```
#                     7, reading data: Mem[        20] =>   38834208
#                     7, reg_file[17] =>          2 (Port 2)
#                     7, PC:         20
#                     7, BEQ
#
#                     8, reading data: Mem[        24] =>   38834210
#                     8, reg_file[ 0] =>          0 (Port 2)
#                     8, reg_file[ 0] =>          0 (Port 1)
#                     8, reading data: Mem[         2] =>        256
#                     8, PC:         24
#                     8, NOP
#
#                     9, reading data: Mem[        28] =>   36735008
#                    10, reg_file[15] <=        256 (Write)
#                     9, PC:         28
#                     9, NOP
#
#                    10, reading data: Mem[        36] =>   17170459
run
#                    10, PC:         36
#                    10, NOP
#
#                    11, reg_file[ 6] =>          6 (Port 2)
#                    11, reg_file[ 8] =>          8 (Port 1)
#                    11, PC:         36
#                    11, wd:          x
#                    11, DIVU
#
#                    12, PC:         36
#                    12, wd:          x
#                    12, DIVU
#
#                    13, PC:         36
#                    13, wd:          x
#                    13, DIVU
#
#                    14, PC:         36
#                    14, wd:          x
#                    14, DIVU
#
```



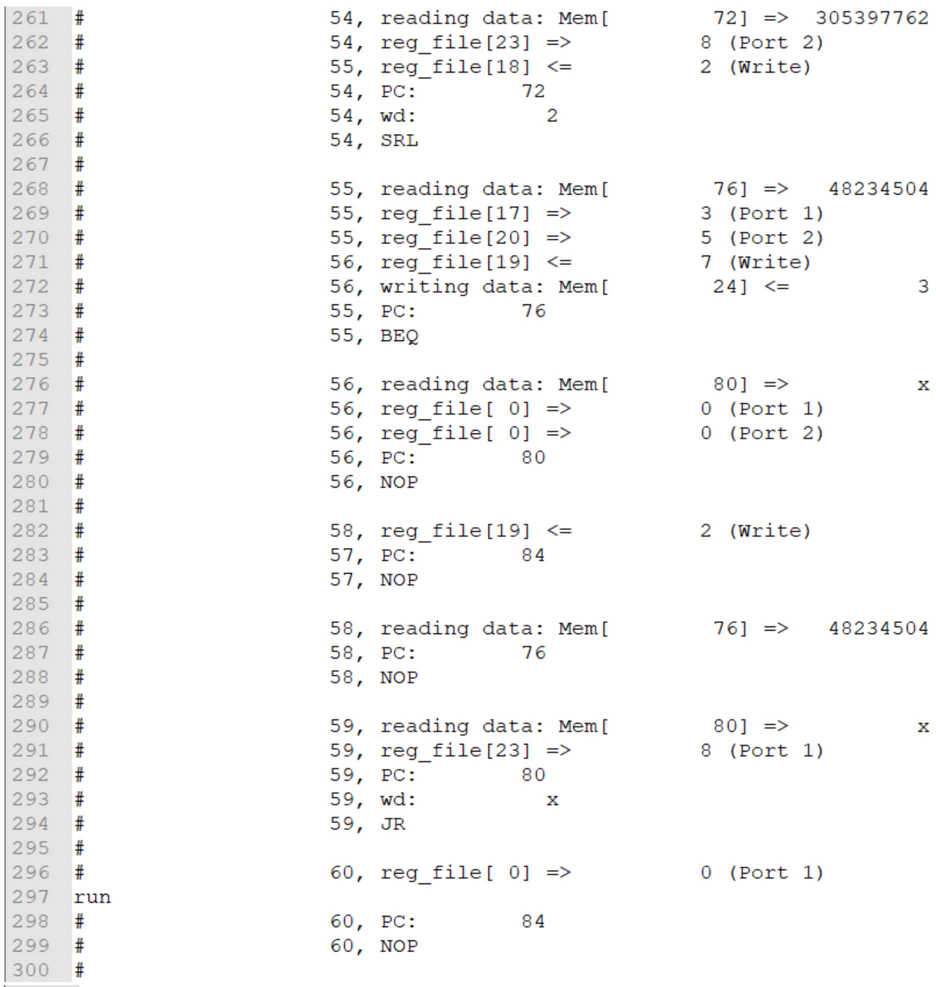
```
#                    15, PC:         36
#                    15, wd:          x
#                    15, DIVU
#
#                    16, PC:         36
#                    16, wd:          x
#                    16, DIVU
#
#                    17, PC:         36
#                    17, wd:          x
#                    17, DIVU
#
#                    18, PC:         36
#                    18, wd:          x
#                    18, DIVU
#
#                    19, PC:         36
#                    19, wd:          x
#                    19, DIVU
#
run
#                    20, PC:         36
#                    20, wd:          x
#                    20, DIVU
#
#                    21, PC:         36
#                    21, wd:          x
#                    21, DIVU
#
#                    22, PC:         36
#                    22, wd:          x
#                    22, DIVU
#
#                    23, PC:         36
#                    23, wd:          x
#                    23, DIVU
#
#                    24, PC:         36
#                    24, wd:          x
#                    24, DIVU
```



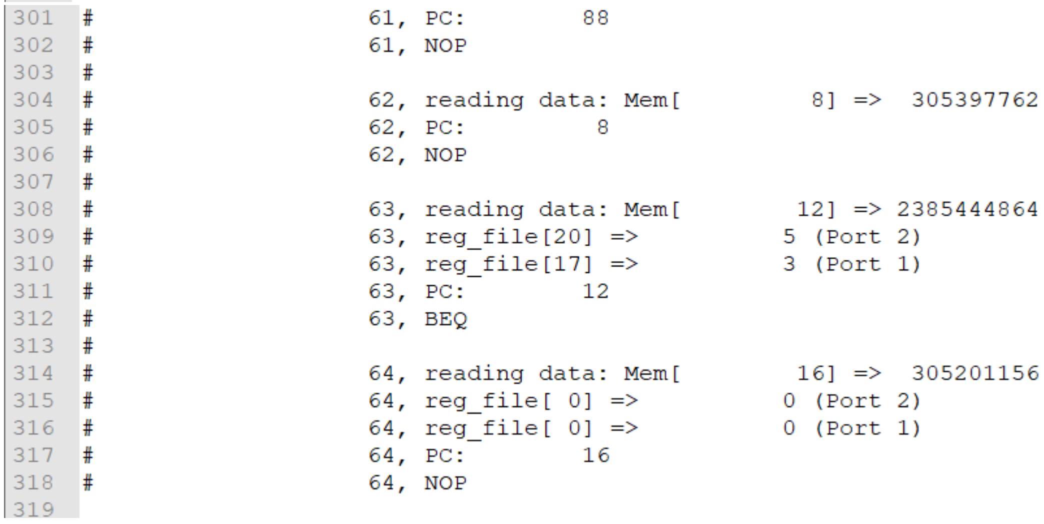
```
#
#                    25, PC:       36
#                    25, wd:         x
#                    25, DIVU
#
#                    26, PC:       36
#                    26, wd:         x
#                    26, DIVU
#
#                    27, PC:       36
#                    27, wd:         x
#                    27, DIVU
#
#                    28, PC:       36
#                    28, wd:         x
#                    28, DIVU
#
#                    29, PC:       36
#                    29, wd:         x
#                    29, DIVU
#
run
#                    30, PC:       36
#                    30, wd:         x
#                    30, DIVU
#
#                    31, PC:       36
#                    31, wd:         x
#                    31, DIVU
#
#                    32, PC:       36
#                    32, wd:         x
#                    32, DIVU
#
#                    33, PC:       36
#                    33, wd:         x
#                    33, DIVU
#
#                    34, PC:       36
#                    34, wd:         x
#                    34, DIVU
#
#                    35, PC:       36
#                    35, wd:         x
#                    35, DIVU
#
#                    36, PC:       36
#                    36, wd:         x
#                    36, DIVU
#
#                    37, PC:       36
#                    37, wd:         x
#                    37, DIVU
#
#                    38, PC:       36
#                    38, wd:         x
#                    38, DIVU
#
#                    39, PC:       36
#                    39, wd:         x
#                    39, DIVU
#
```



```
run
#                   40, PC:          36
#                   40, wd:            x
#                   40, DIVU
#
#                   41, PC:          36
#                   41, wd:            x
#                   41, DIVU
#
#                   42, PC:          36
#                   42, wd:            x
#                   42, DIVU
#
#                   43, PC:          36
#                   43, wd:            x
#                   43, DIVU
#
#                   44, PC:          36
#                   44, wd:            x
#                   44, DIVU
#
#                   45, PC:          36
#                   45, wd:            x
#                   45, DIVU
#
#                   46, reading data: Mem[         40] =>   38834213
#                   46, PC:          40
#                   46, wd:            x
#                   46, DIVU
#
#                   47, reading data: Mem[         44] =>      45072
#                   47, reg_file[16] =>          1 (Port 2)
#                   47, reg_file[18] =>          3 (Port 1)
#                   47, PC:          44
#                   47, wd:            x
#                   47, OR
#
#                   48, reading data: Mem[         48] =>      43026
#                   48, reg_file[ 0] =>          0 (Port 2)
#                   48, reg_file[ 0] =>          0 (Port 1)
#                   48, PC:          48
#                   48, wd:            x
#                   48, MFHI
#
#                   49, reading data: Mem[         52] =>   36735008
#                   49, PC:          52
#                   49, wd:            x
#                   49, MFLO
#
#                   50, reading data: Mem[         56] =>   38834210
#                   50, reg_file[16] =>          1 (Port 2)
#                   50, reg_file[17] =>          2 (Port 1)
run
#                   51, reg_file[18] <=          3 (Write)
#                   50, PC:          56
#                   50, wd:            3
#                   50, ADD
#
#                   51, reading data: Mem[         60] =>  911409156
#                   51, reg_file[18] =>          3 (Port 1)
#                   52, reg_file[22] <=          2 (Write)
#                   51, PC:          60
#                   51, wd:            2
#                   51, SUB
#
#                   52, reading data: Mem[         64] => 2886860824
#                   52, reg_file[19] =>          1 (Port 2)
#                   53, reg_file[21] <=          1 (Write)
#                   52, PC:          64
#                   52, ORI
#
#                   53, reading data: Mem[         68] =>    1546370
#                   53, reg_file[ 0] =>          0 (Port 1)
#                   53, reg_file[18] =>          3 (Port 2)
#                   54, reg_file[17] <=          3 (Write)
#                   53, PC:          68
#                   53, SW
#
```



```
#                     54, reading data: Mem[        72] =>  305397762
#                     54, reg_file[23] =>          8 (Port 2)
#                     55, reg_file[18] <=          2 (Write)
#                     54, PC:        72
#                     54, wd:          2
#                     54, SRL
#
#                     55, reading data: Mem[        76] =>   48234504
#                     55, reg_file[17] =>          3 (Port 1)
#                     55, reg_file[20] =>          5 (Port 2)
#                     56, reg_file[19] <=          7 (Write)
#                     56, writing data: Mem[        24] <=          3
#                     55, PC:        76
#                     55, BEQ
#
#                     56, reading data: Mem[        80] =>          x
#                     56, reg_file[ 0] =>          0 (Port 1)
#                     56, reg_file[ 0] =>          0 (Port 2)
#                     56, PC:        80
#                     56, NOP
#
#                     58, reg_file[19] <=          2 (Write)
#                     57, PC:        84
#                     57, NOP
#
#                     58, reading data: Mem[        76] =>   48234504
#                     58, PC:        76
#                     58, NOP
#
#                     59, reading data: Mem[        80] =>          x
#                     59, reg_file[23] =>          8 (Port 1)
#                     59, PC:        80
#                     59, wd:          x
#                     59, JR
#
#                     60, reg_file[ 0] =>          0 (Port 1)
run
#                     60, PC:        84
#                     60, NOP
#
```



```
#                     61, PC:        88
#                     61, NOP
#
#                     62, reading data: Mem[         8] =>  305397762
#                     62, PC:         8
#                     62, NOP
#
#                     63, reading data: Mem[        12] => 2385444864
#                     63, reg_file[20] =>          5 (Port 2)
#                     63, reg_file[17] =>          3 (Port 1)
#                     63, PC:        12
#                     63, BEQ
#
#                     64, reading data: Mem[        16] =>  305201156
#                     64, reg_file[ 0] =>          0 (Port 2)
#                     64, reg_file[ 0] =>          0 (Port 1)
#                     64, PC:        16
#                     64, NOP
```