EC.204 Introduction To Processor Architecture
Project Report
Y86-64 PROCESSOR

- Himani Sharma(2022102032)
- Kripi Singla(2022102063)

OVERVIEW

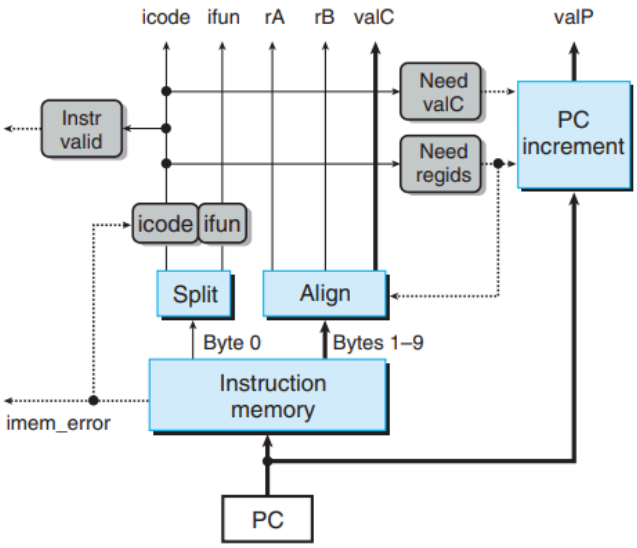
The main objective of this project is to implement a Y86-64 processor. The processor should be able to execute all the instructions in the Y86-64 Instruction set architecture. The aim of this project is to implement a sequential and 5-stage pipelined Y86-64 processor.

SEQUENTIAL IMPLEMENTATION

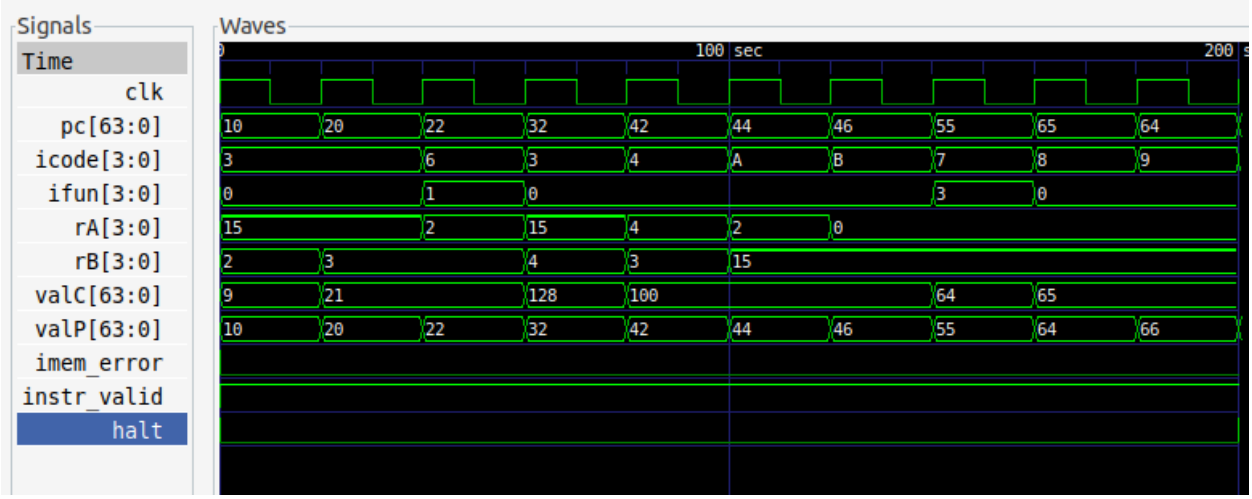
To be checked:

```
1   0x000: 30f20900000000000000 |     irmovq $9,  %rdx
2   0x00a: 30f31500000000000000 |     irmovq $21, %rbx
3   0x014: 6123                 |     subq %rdx, %rbx          # subtract
4   0x016: 30f48000000000000000 |     irmovq $128,%rsp         # Problem 4.13
5   0x020: 4043640000000000000  |     rmmovq %rsp, 100(%rbx)   # store
6   0x02a: a02f                 |     pushq %rdx               # push
7   0x02c: b00f                 |     popq  %rax               # Problem 4.14
8   0x02e: 734000000000000000   |     je done                  # Not taken
9   0x037: 804100000000000000   |     call proc                # Problem 4.18
10  0x040:                      | done:
11  0x040: 00                   |     halt
12  0x041:                      | proc:
13  0x041: 90                   |     ret                      # Return
14                              |
```

1) FETCH




```
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/SEQ/Processor_with_test_bench$ ./a.out
WARNING: ./fetch.v:22: $readmemb(1.txt): Not enough words in the file for the requested range [0:255].
VCD info: dumpfile fetch.vcd opened for output.
clk= 1 time =                   0: pc =                  10 icode = 3 ifun =  0 rA = 15 rB =  2 valC =                   9 valP =                  10 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                  10: pc =                  10 icode = 3 ifun =  0 rA = 15 rB =  2 valC =                   9 valP =                  10 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                  20: pc =                  20 icode = 3 ifun =  0 rA = 15 rB =  3 valC =                  21 valP =                  20 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                  30: pc =                  20 icode = 3 ifun =  0 rA = 15 rB =  3 valC =                  21 valP =                  20 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                  40: pc =                  22 icode = 6 ifun =  1 rA =  2 rB =  3 valC =                  21 valP =                  22 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                  50: pc =                  22 icode = 6 ifun =  1 rA =  2 rB =  3 valC =                  21 valP =                  22 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                  60: pc =                  32 icode = 3 ifun =  0 rA = 15 rB =  4 valC =                 128 valP =                  32 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                  70: pc =                  32 icode = 3 ifun =  0 rA = 15 rB =  4 valC =                 128 valP =                  32 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                  80: pc =                  42 icode = 4 ifun =  0 rA =  4 rB =  3 valC =                 100 valP =                  42 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                  90: pc =                  42 icode = 4 ifun =  0 rA =  4 rB =  3 valC =                 100 valP =                  42 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                 100: pc =                  44 icode =10 ifun =  0 rA =  2 rB = 15 valC =                 100 valP =                  44 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                 110: pc =                  44 icode =10 ifun =  0 rA =  2 rB = 15 valC =                 100 valP =                  44 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                 120: pc =                  46 icode =11 ifun =  0 rA =  0 rB = 15 valC =                 100 valP =                  46 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                 130: pc =                  46 icode =11 ifun =  0 rA =  0 rB = 15 valC =                 100 valP =                  46 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                 140: pc =                  55 icode = 7 ifun =  3 rA =  0 rB = 15 valC =                  64 valP =                  55 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                 150: pc =                  55 icode = 7 ifun =  3 rA =  0 rB = 15 valC =                  64 valP =                  55 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                 160: pc =                  65 icode = 8 ifun =  0 rA =  0 rB = 15 valC =                  65 valP =                  64 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                 170: pc =                  65 icode = 8 ifun =  0 rA =  0 rB = 15 valC =                  65 valP =                  64 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                 180: pc =                  64 icode = 9 ifun =  0 rA =  0 rB = 15 valC =                  65 valP =                  66 imem_error = 0 instr_valid = 1 halt = 0
clk= 0 time =                 190: pc =                  64 icode = 9 ifun =  0 rA =  0 rB = 15 valC =                  65 valP =                  66 imem_error = 0 instr_valid = 1 halt = 0
clk= 1 time =                 200: pc =                  65 icode = 0 ifun =  0 rA =  0 rB = 15 valC =                  65 valP =                  65 imem_error = 0 instr_valid = 1 halt = 1
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/SEQ/Processor_with_test_bench$
```



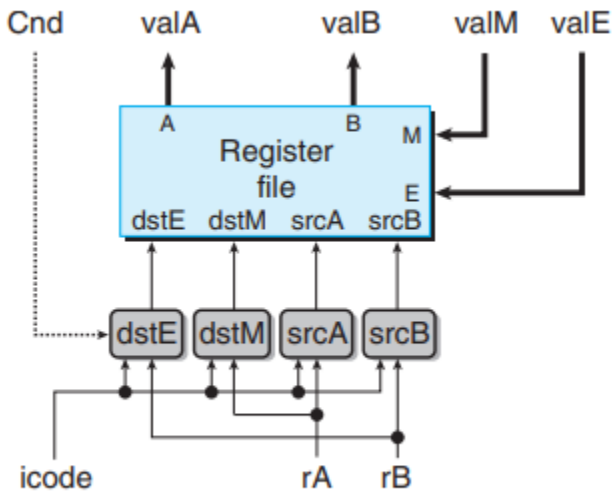
The **fetch-stage** reads the bytes of an instruction from memory, using the program counter(PC) as the memory address. From The Instruction, It extracts the two 4-bit portions of the instruction specifier byte,referred to as icode(the instruction code) and ifun(the

instruction function).It possibly fetches a register specifier byte, giving one or both of the register operand specifiers rA and rB.It also possibly fetches an 8-byte constant word valC. It computes valP to be the address of the instruction following the current one in sequential order. That is,valP equals the value of the PC plus the length of the fetched instruction.

For example, in the checked sample test case,in every cycle, the fetch stage of the sequential architecture, calculates valP with respect to a particular instruction and fetches its icode and ifun,that is:-

For irmovq, icode = 3 and ifun = 0
For subq, icode = 6 and ifun = 1
And similarly it is calculated for all other upcoming instructions.

2) DECODE

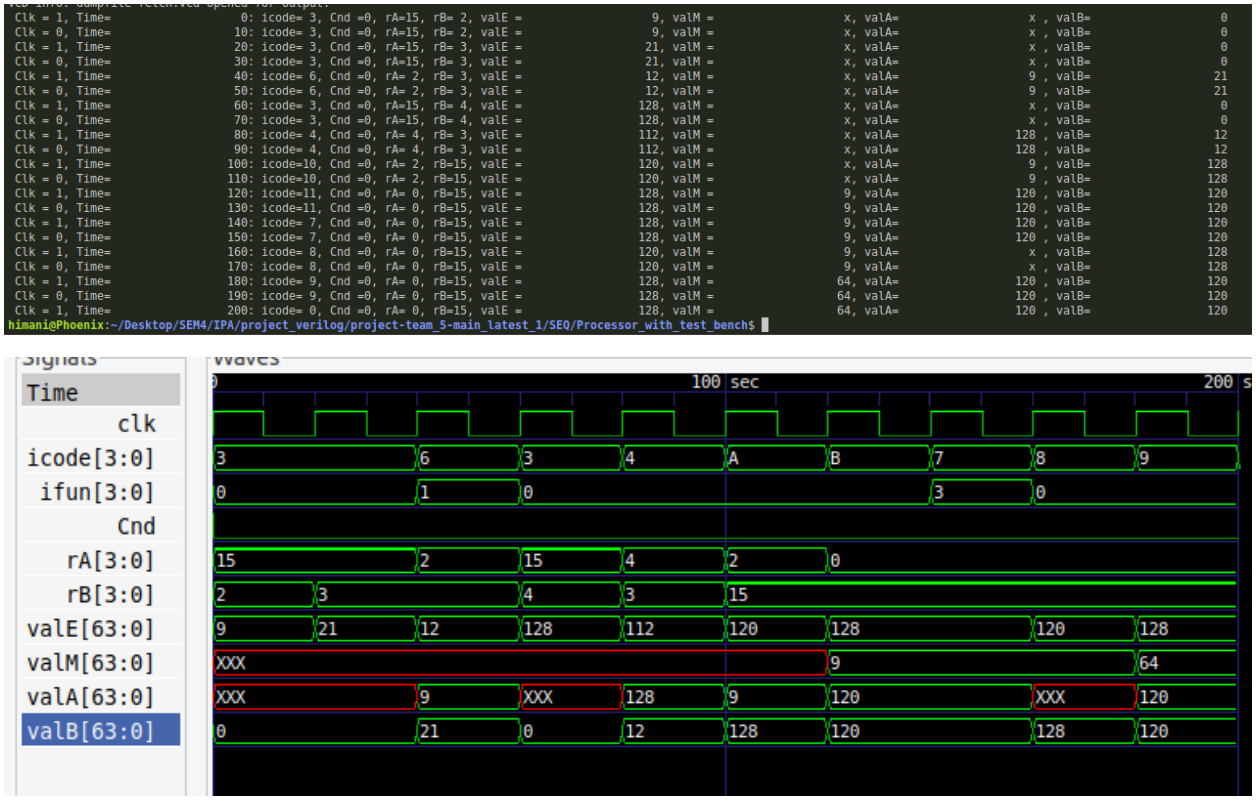


The **decode stage** reads upto two operands from the register file, giving values valA and valB. It reads the registers designated by

instruction fields rA and rB, for some instructions such as pop and return, it also reads from register %rsp.

The write-back stage writes up to two results to the register file.

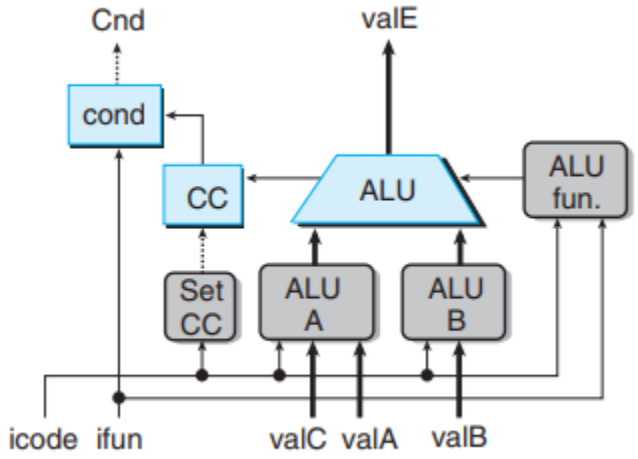
In our implementation we are combining Decode and Write Back together. Decode and write-back stages are pretty intertwined with each other in the way that decode stage cannot correctly calculate the value of valA and valB, corresponding to the registers present in the register-file, until and unless the values of these registers are updated in the write-back stage.

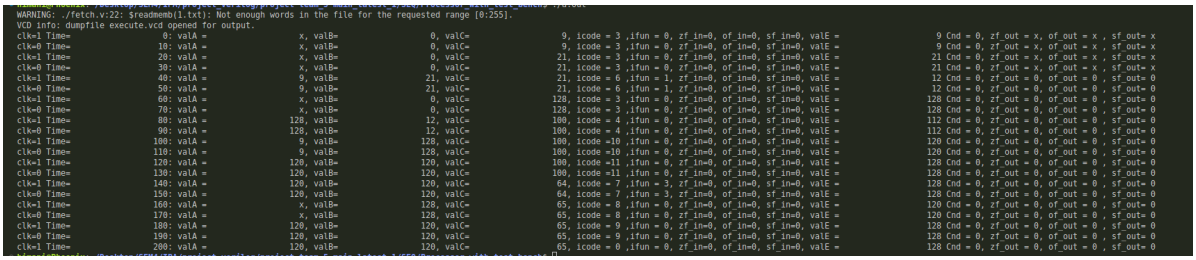


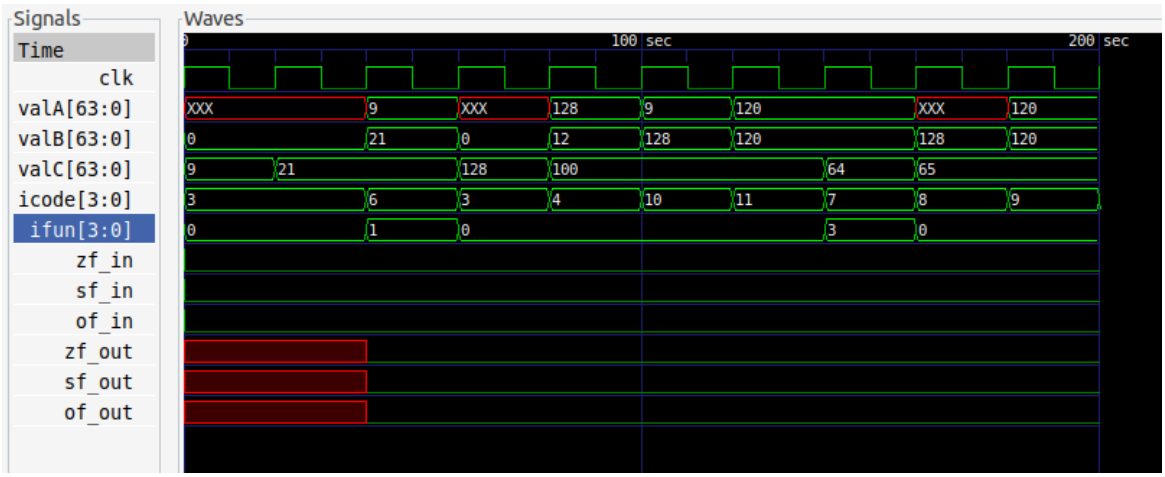
Now, as every instruction follows, for the first irmovq instructions, values, the corresponding values, in our case 9 and 21 are written-back into the registers %rdx and %rbx respectively and

then their values for valA and valB are easily computed during the decode stage of subq.
A similar pattern follows for further instructions.

3) EXECUTE





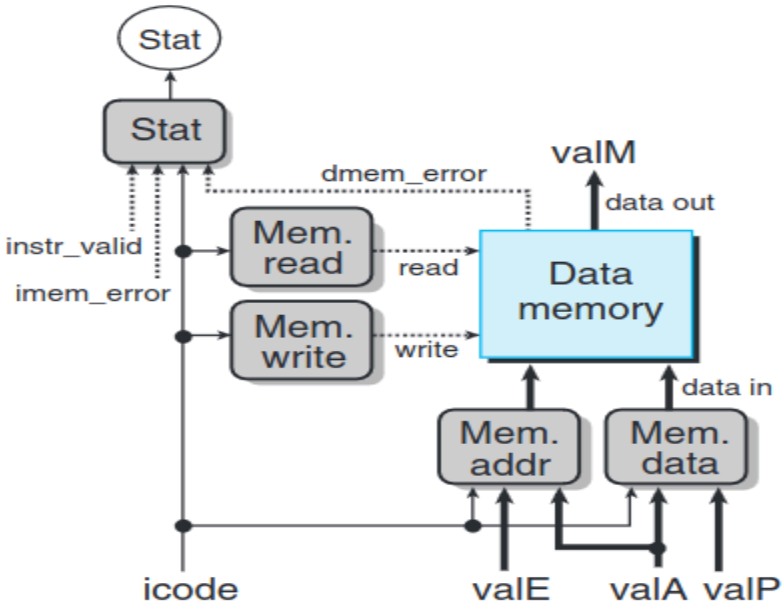


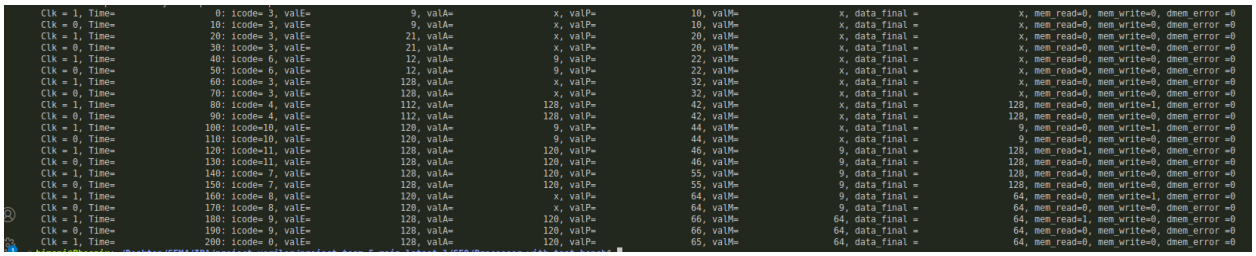
In the execute stage, the arithmetic/logic unit (ALU) either performs

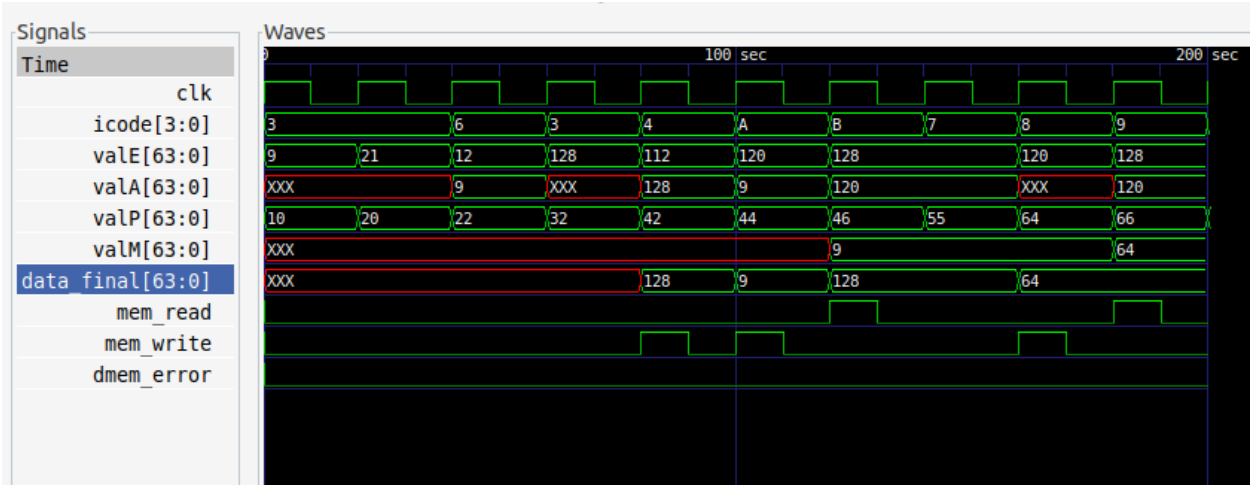
the operation specified by the instruction (according to the value of ifun), computes the effective address of a memory reference, or increments or decrements the stack pointer. We refer to the resulting value as valE.The condition codes are possibly set. For a conditional move instruction, the stage will evaluate the condition codes and move conditions and the register file is updated only if the condition holds. Similarly, for a jump instruction, it determines whether or not the branch should be taken.

The execute stage basically calculates the value of valE in most cases. For irmovq instruction, the value of valE equals the value that is being moved in the register.
For opq, the value of valE is computed using ALU functionality.
For jXX, the condition codes are computed as per zero-flag, overflow-flag and sign-flag which are set in previous Opq instructions.

4) MEMORY



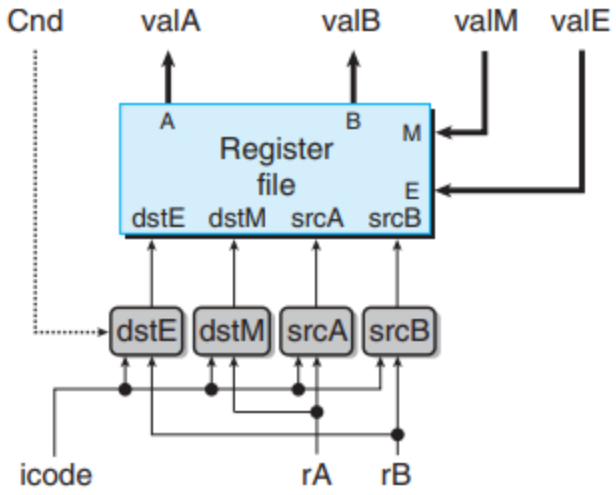




The memory stage may write data to memory, or it may read data from memory. We refer to the value read as valM.

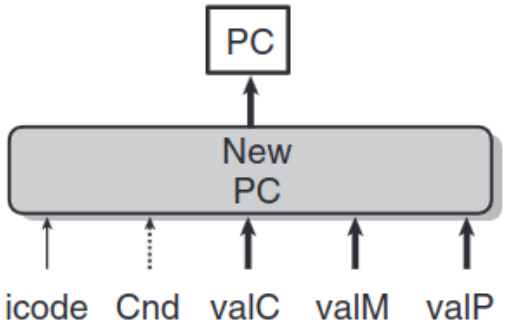
In our respective program, the memory operations are majorly done in rmmovq, pop and push instructions.

5) WRITE BACK

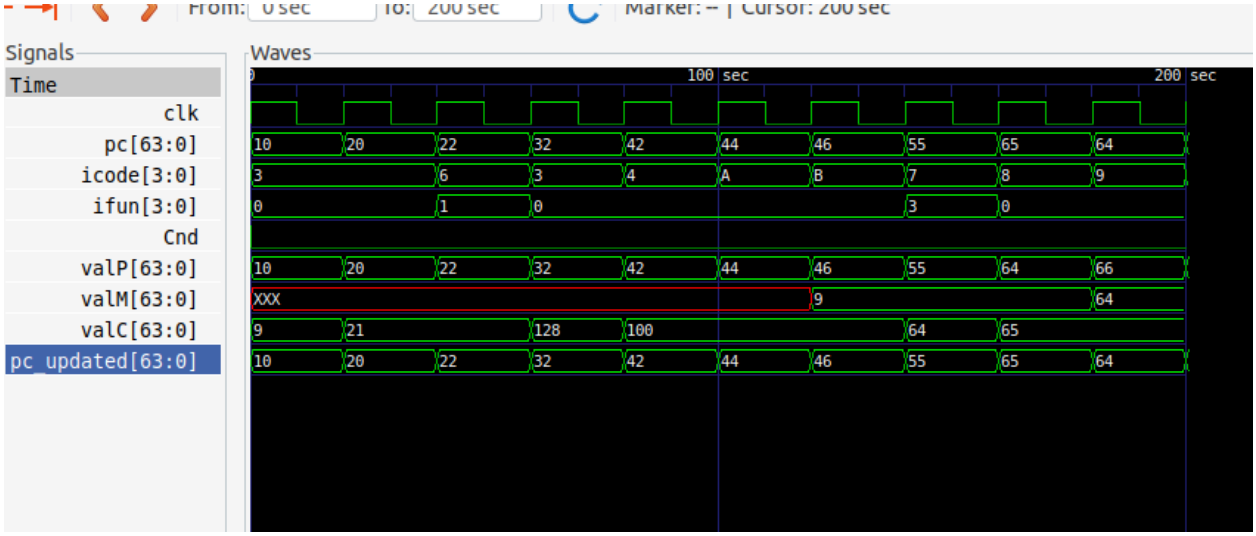


Same as Decode

6) PC UPDATE




```
WARNING: ./fetch.v:22: $readmemb(1.txt): Not enough words in the file for the requested range [0:255].
VCD info: dumpfile pc_update.vcd opened for output.
clk=1 Time=                   0: pc=                  10, icode= 3, Cnd=0, valP=                  10 , valM=                   x, valC =                   9, pc_updated =                  10
clk=0 Time=                  10: pc=                  10, icode= 3, Cnd=0, valP=                  10 , valM=                   x, valC =                   9, pc_updated =                  10
clk=1 Time=                  20: pc=                  20, icode= 3, Cnd=0, valP=                  20 , valM=                   x, valC =                  21, pc_updated =                  20
clk=0 Time=                  30: pc=                  20, icode= 3, Cnd=0, valP=                  20 , valM=                   x, valC =                  21, pc_updated =                  20
clk=1 Time=                  40: pc=                  22, icode= 6, Cnd=0, valP=                  22 , valM=                   x, valC =                  21, pc_updated =                  22
clk=0 Time=                  50: pc=                  22, icode= 6, Cnd=0, valP=                  22 , valM=                   x, valC =                  21, pc_updated =                  22
clk=1 Time=                  60: pc=                  32, icode= 3, Cnd=0, valP=                  32 , valM=                   x, valC =                 128, pc_updated =                  32
clk=0 Time=                  70: pc=                  32, icode= 3, Cnd=0, valP=                  32 , valM=                   x, valC =                 128, pc_updated =                  32
clk=1 Time=                  80: pc=                  42, icode= 4, Cnd=0, valP=                  42 , valM=                   x, valC =                 100, pc_updated =                  42
clk=0 Time=                  90: pc=                  42, icode= 4, Cnd=0, valP=                  42 , valM=                   x, valC =                 100, pc_updated =                  42
clk=1 Time=                 100: pc=                  44, icode=10, Cnd=0, valP=                  44 , valM=                   x, valC =                 100, pc_updated =                  44
clk=0 Time=                 110: pc=                  44, icode=10, Cnd=0, valP=                  44 , valM=                   x, valC =                 100, pc_updated =                  44
clk=1 Time=                 120: pc=                  46, icode=11, Cnd=0, valP=                  46 , valM=                   9, valC =                 100, pc_updated =                  46
clk=0 Time=                 130: pc=                  46, icode=11, Cnd=0, valP=                  46 , valM=                   9, valC =                 100, pc_updated =                  46
clk=1 Time=                 140: pc=                  55, icode= 7, Cnd=0, valP=                  55 , valM=                   9, valC =                  64, pc_updated =                  55
clk=0 Time=                 150: pc=                  55, icode= 7, Cnd=0, valP=                  55 , valM=                   9, valC =                  64, pc_updated =                  55
clk=1 Time=                 160: pc=                  65, icode= 8, Cnd=0, valP=                  64 , valM=                   9, valC =                  65, pc_updated =                  65
clk=0 Time=                 170: pc=                  65, icode= 8, Cnd=0, valP=                  64 , valM=                   9, valC =                  65, pc_updated =                  65
clk=1 Time=                 180: pc=                  64, icode= 9, Cnd=0, valP=                  66 , valM=                  64, valC =                  65, pc_updated =                  64
clk=0 Time=                 190: pc=                  64, icode= 9, Cnd=0, valP=                  66 , valM=                  64, valC =                  65, pc_updated =                  64
clk=1 Time=                 200: pc=                  65, icode= 0, Cnd=0, valP=                  65 , valM=                  64, valC =                  65, pc_updated =                  65
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/SEQ/Processor_with_test_bench$
```



The PC is set to the address of the next instruction.

COMPLETE SEQUENTIAL DESIGN IMPLEMENTATION

Challenges faced in Sequential Implementation:-

- We faced problems in the write-back stage regarding the update of register values; earlier, the values were not getting updated and after making certain changes, the write-back stage correctly started updating the values corresponding to respective registers present in the register file.
- Though the write-back stage started working, the values of the registers being updated were not in sync with the decode stage. To fix that, we created another module having a decode and write-back stage together.
- To further solve the problem, clock-sequencing was meticulously set for each stage , by operating the decode stage when clk = 1 and write-back on the negative edge of the clock.

Sequential Implementation

1. icode = 3, ifun = 0, rA = 15, rB = 2
   valA — x, valB = 0, valC = 9, valE = 9
   updated_PC = PC + 10 = 10

2. icode = 3, ifun = 0, rA = 15, rB = 3
   valA — x, valB = 0, valC = 21, valE = 21
   updated_PC = PC + 10 = 20

3. icode = 6, ifun = 1, rA = 2, rB = 3
   valA = 9, valB = 21
   valE = 21 − 9 = 12
   updated_PC = PC + 2 = 20 + 2 = 22

4. icode = 3, ifun = 0, rA = 5, rB = 4
   valB = 0, valC = 128, valE = 128
   updated_PC = PC + 10 = 22 + 10 = 32

5. rmmovq
   icode = 4, ifun = 0, rA = 4, rB = 3
   valA = 128, valC = 100
   valB = 12
   valE = valB + valC = 112
   data_final = M[112] = 128
   updated_PC = 32 + 10 = 42

pushq %rdx
   icode = 10, ifun = 0, rA = 2, rB = 15
   valA = 9, valB = 128, valE = valB − 8 = 120
   M[120] = 9
   data_final = 9
   valP = 44

popq %rax
   icode = 11, ifun = 0, rA = 0, rB = 15
   valA = 120, valB = 120, valE = valB + 8 = 128
   valM = M[valA] = M[120] = 9
   updated_PC = 46

je (not taken)
   icode = 7, ifun = 3
   valC = 64, cnd = 0
   PC → 46 + 9 = 55
   M[112] = 128
   M[120] = 9

9. call

icode = 8 ifun = 0

valA = x valB = 128 valC = 65

valP = 55 + 9 = 64

M[120] => 64

Pc → 65

10. return

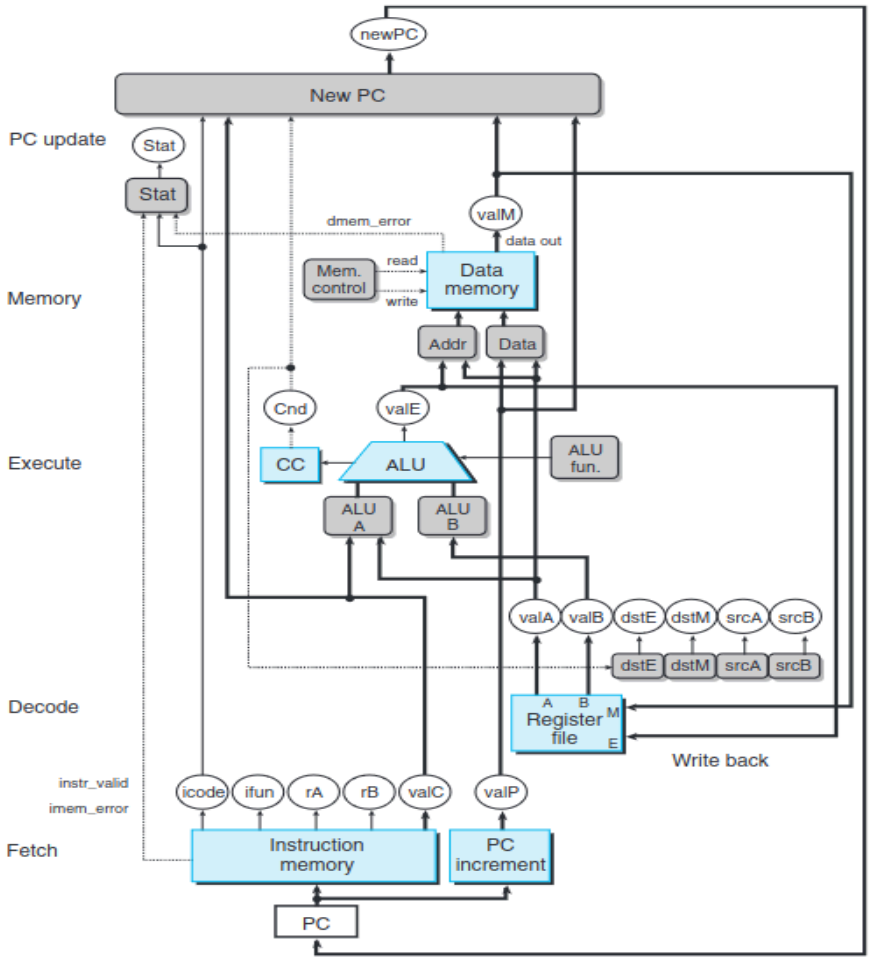
icode = 9 ifun = 0

valA = 120 valB = 120 valE = 128

valM = M[120] = 64

Pc → 65

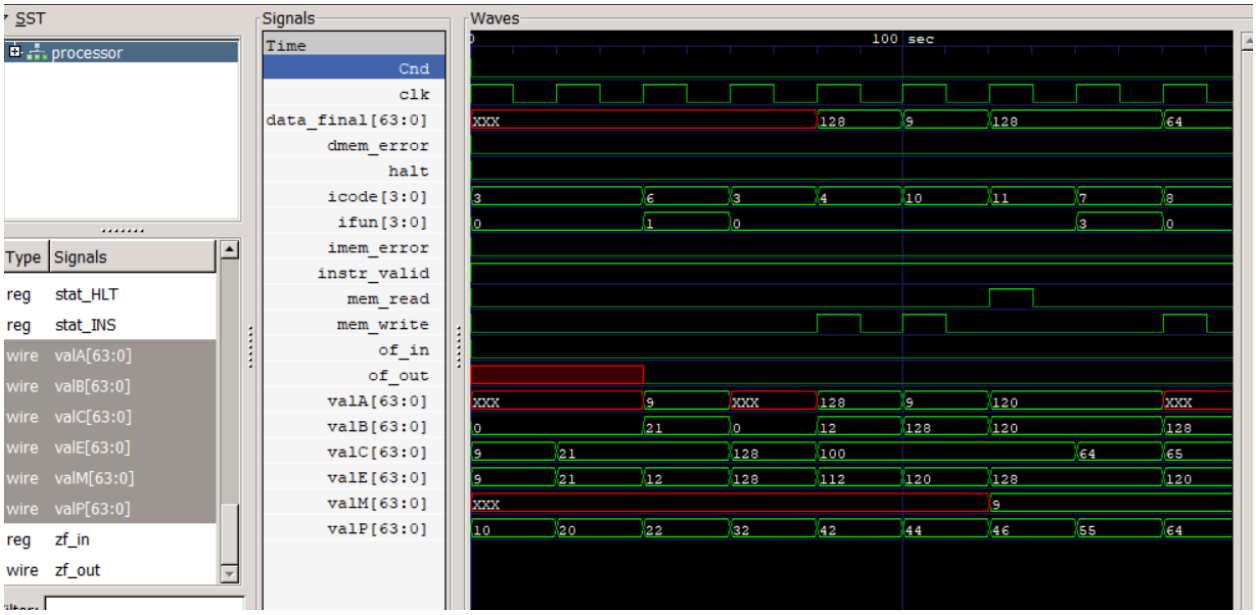
halt

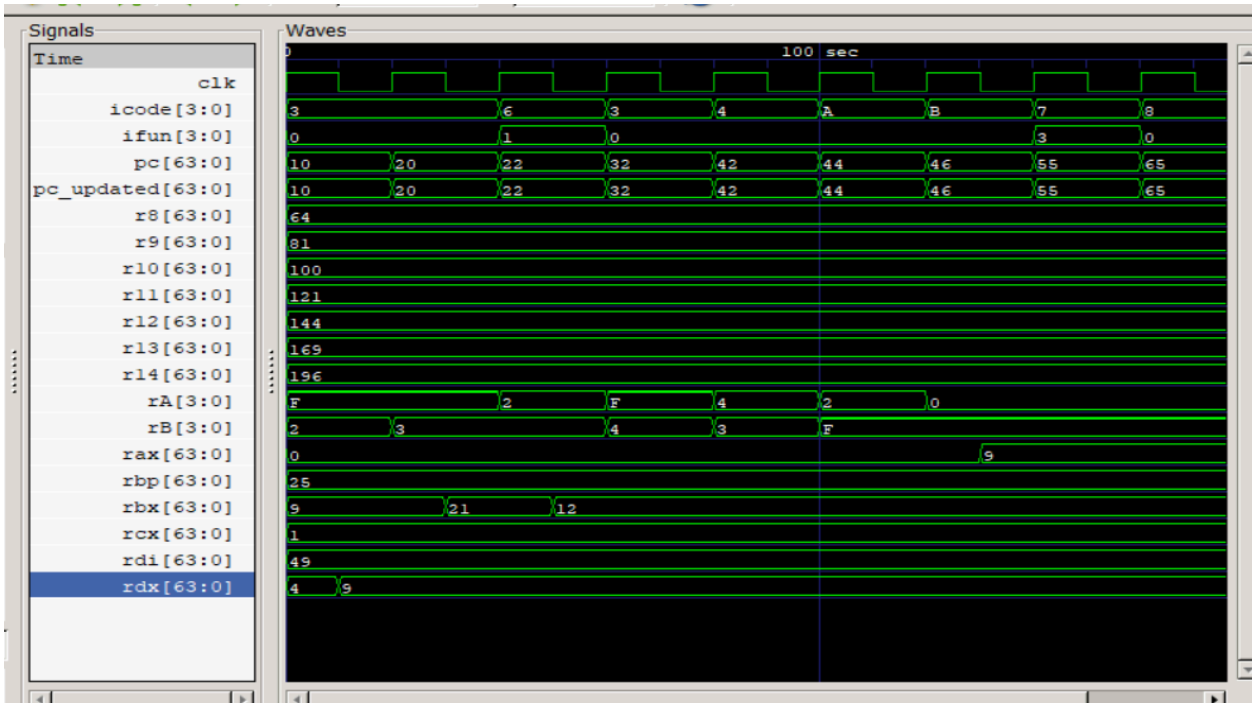


Processor output (SEQ) :

```
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/Final_Project/project-team_5-main/SEQ/Processor$ iverilog pro.v
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/Final_Project/project-team_5-main/SEQ/Processor$ ./a.out
WARNING: ./fetch.v:22: $readmemb(1.txt): Not enough words in the file for the requested range [0:255].
                   0    clk = 1 icode =  3 ifun =  0 rA = 15 rB =  2
                        valA =                   x valB =                   0 valC =                   9 valE =                   9 valM =                   x updated_pc =                  10
                        data_final =                   x dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                  10    clk = 0 icode =  3 ifun =  0 rA = 15 rB =  2
                        valA =                   x valB =                   0 valC =                   9 valE =                   9 valM =                   x updated_pc =                  10
                        data_final =                   x dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                  20    clk = 1 icode =  3 ifun =  0 rA = 15 rB =  3
                        valA =                   x valB =                   0 valC =                  21 valE =                  21 valM =                   x updated_pc =                  20
                        data_final =                   x dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                  30    clk = 0 icode =  3 ifun =  0 rA = 15 rB =  3
                        valA =                   x valB =                   0 valC =                  21 valE =                  21 valM =                   x updated_pc =                  20
                        data_final =                   x dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                  40    clk = 1 icode =  6 ifun =  1 rA =  2 rB =  3
                        valA =                   9 valB =                  21 valC =                  21 valE =                  12 valM =                   x updated_pc =                  22
                        data_final =                   x dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                  50    clk = 0 icode =  6 ifun =  1 rA =  2 rB =  3
                        valA =                   9 valB =                  21 valC =                  21 valE =                  12 valM =                   x updated_pc =                  22
                        data_final =                   x dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                  60    clk = 1 icode =  3 ifun =  0 rA = 15 rB =  4
                        valA =                   x valB =                   0 valC =                 128 valE =                 128 valM =                   x updated_pc =                  32
                        data_final =                   x dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                  70    clk = 0 icode =  3 ifun =  0 rA = 15 rB =  4
                        valA =                   x valB =                   0 valC =                 128 valE =                 128 valM =                   x updated_pc =                  32
                        data_final =                   x dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                  80    clk = 1 icode =  4 ifun =  0 rA =  4 rB =  3
                        valA =                 128 valB =                  12 valC =                 100 valE =                 112 valM =                   x updated_pc =                  42
                        data_final =                 128 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0
```

```
                  90    clk = 0 icode =  4 ifun =  0 rA =  4 rB =  3
                        valA =                 128 valB =                  12 valC =                 100 valE =                 112 valM =                   x updated_pc =                  42
                        data_final =                 128 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 100    clk = 1 icode = 10 ifun =  0 rA =  2 rB = 15
                        valA =                   9 valB =                 128 valC =                 100 valE =                 120 valM =                   x updated_pc =                  44
                        data_final =                   9 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 110    clk = 0 icode = 10 ifun =  0 rA =  2 rB = 15
                        valA =                   9 valB =                 128 valC =                 100 valE =                 120 valM =                   x updated_pc =                  44
                        data_final =                   9 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 120    clk = 1 icode = 11 ifun =  0 rA =  0 rB = 15
                        valA =                 120 valB =                 120 valC =                 100 valE =                 128 valM =                   9 updated_pc =                  46
                        data_final =                 128 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 130    clk = 0 icode = 11 ifun =  0 rA =  0 rB = 15
                        valA =                 120 valB =                 120 valC =                 100 valE =                 128 valM =                   9 updated_pc =                  46
                        data_final =                 128 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 140    clk = 1 icode =  7 ifun =  3 rA =  0 rB = 15
                        valA =                 120 valB =                 120 valC =                  64 valE =                 128 valM =                   9 updated_pc =                  55
                        data_final =                 128 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 150    clk = 0 icode =  7 ifun =  3 rA =  0 rB = 15
                        valA =                 120 valB =                 120 valC =                  64 valE =                 128 valM =                   9 updated_pc =                  55
                        data_final =                 128 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 160    clk = 1 icode =  8 ifun =  0 rA =  0 rB = 15
                        valA =                   x valB =                 128 valC =                  65 valE =                 120 valM =                   9 updated_pc =                  65
                        data_final =                  64 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 170    clk = 0 icode =  8 ifun =  0 rA =  0 rB = 15
                        valA =                   x valB =                 128 valC =                  65 valE =                 120 valM =                   9 updated_pc =                  65
                        data_final =                  64 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 180    clk = 1 icode =  9 ifun =  0 rA =  0 rB = 15
                        valA =                 120 valB =                 120 valC =                  65 valE =                 128 valM =                  64 updated_pc =                  64
                        data_final =                  64 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 190    clk = 0 icode =  9 ifun =  0 rA =  0 rB = 15
                        valA =                 120 valB =                 120 valC =                  65 valE =                 128 valM =                  64 updated_pc =                  64
                        data_final =                  64 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 0 S_AOK = 1 S_INS = 0 S_ADR = 0

                 200    clk = 1 icode =  0 ifun =  0 rA =  0 rB = 15
                        valA =                 120 valB =                 120 valC =                  65 valE =                 128 valM =                  64 updated_pc =                  65
                        data_final =                  64 dmem_error = 0 instr_valid = 1 imem_error = 0 Cnd = 0 S_HLT = 1 S_AOK = 0 S_INS = 0 S_ADR = 0
```





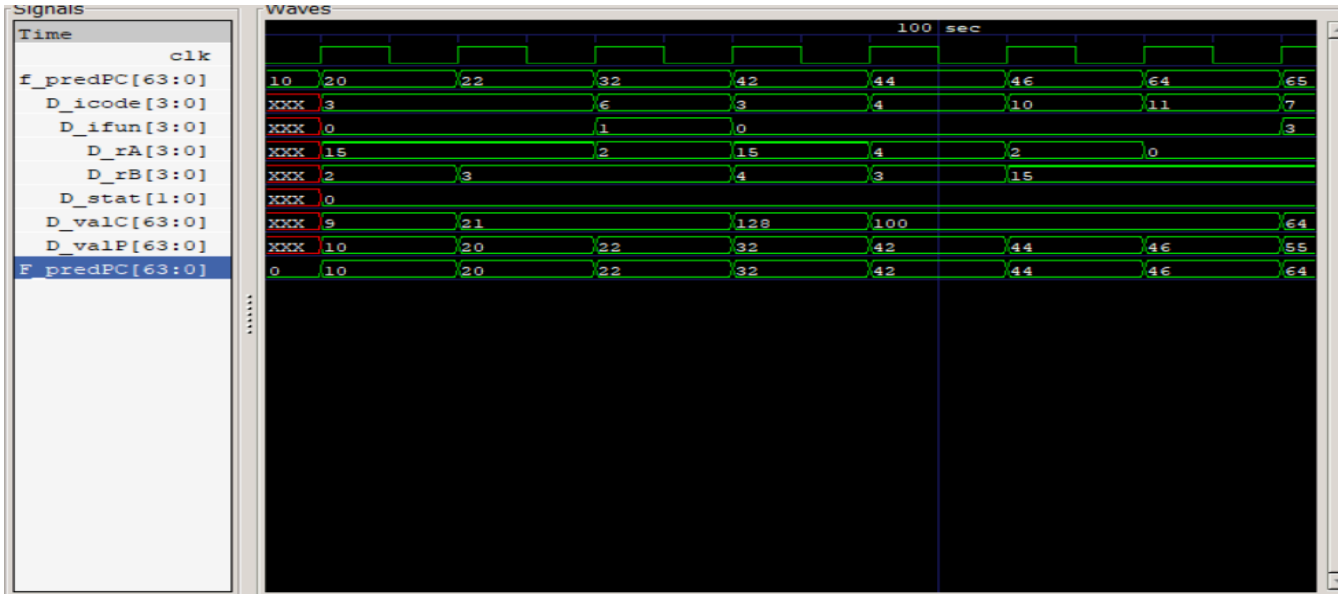
Registers had some initial values stored in them

PIPELINED IMPLEMENTATION

Pipelined Implementation consists of Pipeline Registers which store the output of every stage.

- **F** holds a **predicted** value of the program counter.
- **D** sits between fetch and decode stages. It holds information about the most recently fetched instruction.
- **E** sits between the decode and execute stages. It holds information about the most recently decoded instruction and the values read from the register file for processing by the execute stage
- **M** sits between the execute and memory stages. It holds the results of the most recently executed instruction for processing by the memory stage. It also holds information about branch conditions and branch targets for processing conditional jumps.
- **W** sits between the memory stage and the feedback paths that supply the computed results to the register file for writing and the return address to the PC selection logic when completing a ret instruction.

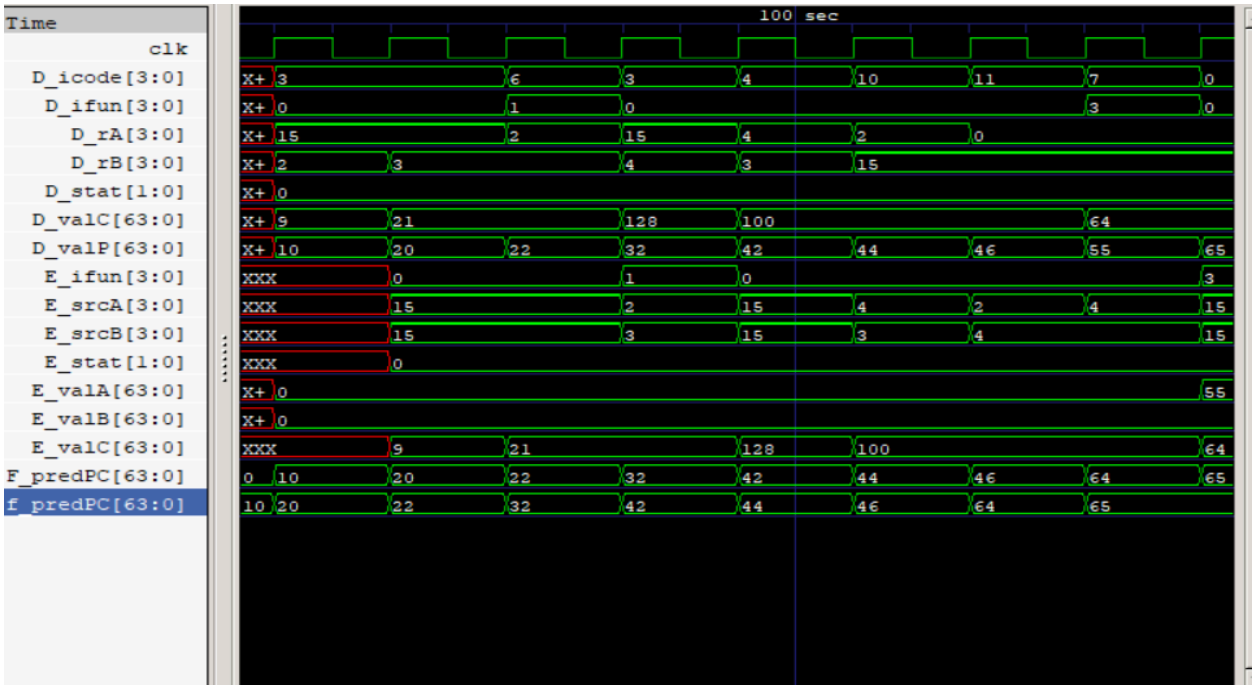
1) FETCH + PC UPDATE



```
WARNING: ./fetch_1.v:45: $readmemb(2.txt): Not enough words in the file for the requested range [0:1023].
VCD info: dumpfile fetch_1_tb.vcd opened for output.
                 0    clk=0
                      F Reg:          F_predPC =                   0
                      fetch:          f_predPC =                  10
                      D Reg:          D_icode =  x D_ifun =  x D_rA =  x D_rB =  x D_valC =                   x D_valP =                   x D_stat = x

                10    clk=1
                      F Reg:          F_predPC =                  10
                      fetch:          f_predPC =                  20
                      D Reg:          D_icode =  3 D_ifun =  0 D_rA = 15 D_rB =  2 D_valC =                   9 D_valP =                  10 D_stat = 0

                20    clk=0
                      F Reg:          F_predPC =                  10
                      fetch:          f_predPC =                  20
                      D Reg:          D_icode =  3 D_ifun =  0 D_rA = 15 D_rB =  2 D_valC =                   9 D_valP =                  10 D_stat = 0

                30    clk=1
                      F Reg:          F_predPC =                  20
                      fetch:          f_predPC =                  22
                      D Reg:          D_icode =  3 D_ifun =  0 D_rA = 15 D_rB =  3 D_valC =                  21 D_valP =                  20 D_stat = 0

                40    clk=0
                      F Reg:          F_predPC =                  20
                      fetch:          f_predPC =                  22
                      D Reg:          D_icode =  3 D_ifun =  0 D_rA = 15 D_rB =  3 D_valC =                  21 D_valP =                  20 D_stat = 0

                50    clk=1
                      F Reg:          F_predPC =                  22
                      fetch:          f_predPC =                  32
                      D Reg:          D_icode =  6 D_ifun =  1 D_rA =  2 D_rB =  3 D_valC =                  21 D_valP =                  22 D_stat = 0

                60    clk=0
                      F Reg:          F_predPC =                  22
                      fetch:          f_predPC =                  32
                      D Reg:          D_icode =  6 D_ifun =  1 D_rA =  2 D_rB =  3 D_valC =                  21 D_valP =                  22 D_stat = 0

                70    clk=1
                      F Reg:          F_predPC =                  32
                      fetch:          f_predPC =                  42
                      D Reg:          D_icode =  3 D_ifun =  0 D_rA = 15 D_rB =  4 D_valC =                 128 D_valP =                  32 D_stat = 0

                80    clk=0
                      F Reg:          F_predPC =                  32
                      fetch:          f_predPC =                  42
                      D Reg:          D_icode =  3 D_ifun =  0 D_rA = 15 D_rB =  4 D_valC =                 128 D_valP =                  32 D_stat = 0

                90    clk=1
                      F Reg:          F_predPC =                  42
                      fetch:          f_predPC =                  44
                      D Reg:          D_icode =  4 D_ifun =  0 D_rA =  4 D_rB =  3 D_valC =                 100 D_valP =                  42 D_stat = 0

               100    clk=0
                      F Reg:          F_predPC =                  42
                      fetch:          f_predPC =                  44
```

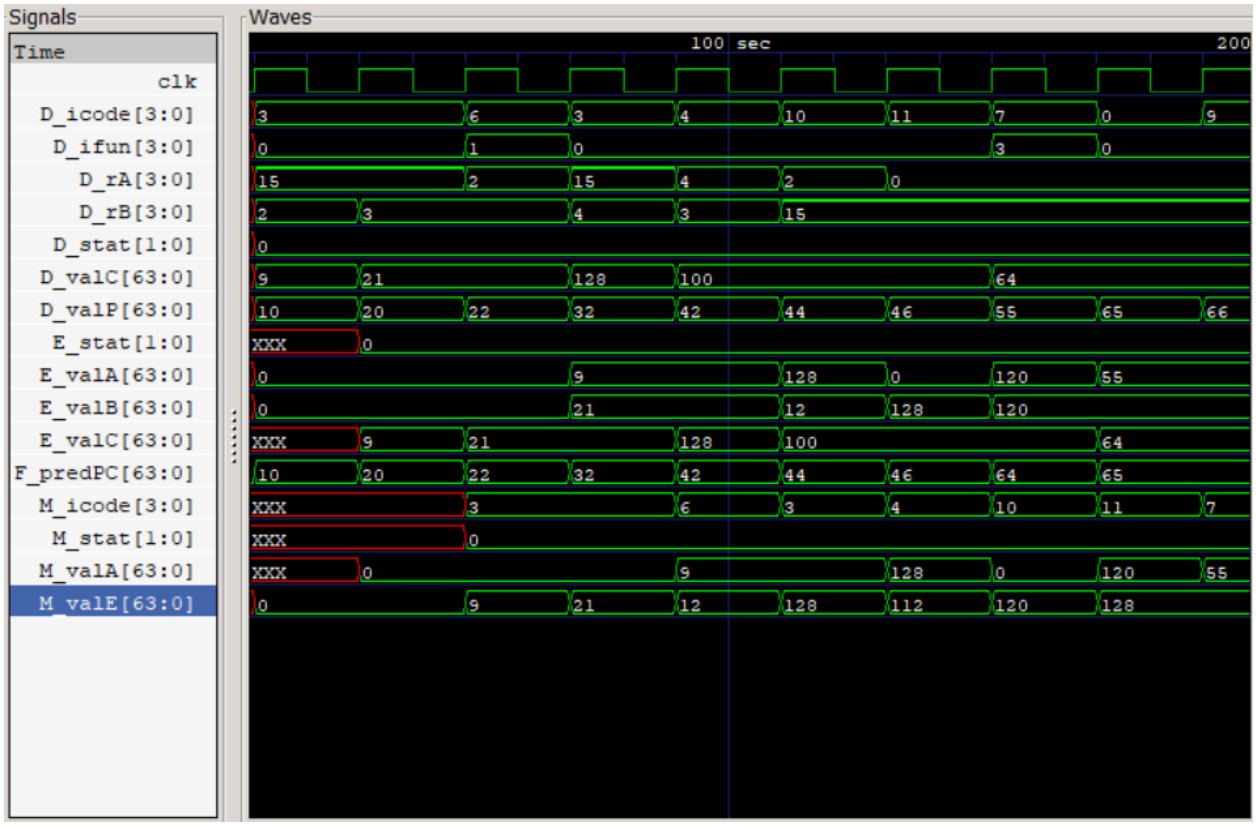
```
               100    clk=0
                      F Reg:          F_predPC =                  42
                      fetch:          f_predPC =                  44
                      D Reg:          D_icode =  4 D_ifun =  0 D_rA =  4 D_rB =  3 D_valC =                 100 D_valP =                  42 D_stat = 0

               110    clk=1
                      F Reg:          F_predPC =                  44
                      fetch:          f_predPC =                  46
                      D Reg:          D_icode = 10 D_ifun =  0 D_rA =  2 D_rB = 15 D_valC =                 100 D_valP =                  44 D_stat = 0

               120    clk=0
                      F Reg:          F_predPC =                  44
                      fetch:          f_predPC =                  46
                      D Reg:          D_icode = 10 D_ifun =  0 D_rA =  2 D_rB = 15 D_valC =                 100 D_valP =                  44 D_stat = 0

               130    clk=1
                      F Reg:          F_predPC =                  46
                      fetch:          f_predPC =                  64
                      D Reg:          D_icode = 11 D_ifun =  0 D_rA =  0 D_rB = 15 D_valC =                 100 D_valP =                  46 D_stat = 0

               140    clk=0
                      F Reg:          F_predPC =                  46
                      fetch:          f_predPC =                  64
                      D Reg:          D_icode = 11 D_ifun =  0 D_rA =  0 D_rB = 15 D_valC =                 100 D_valP =                  46 D_stat = 0

               150    clk=1
                      F Reg:          F_predPC =                  64
                      fetch:          f_predPC =                  65
                      D Reg:          D_icode =  7 D_ifun =  3 D_rA =  0 D_rB = 15 D_valC =                  64 D_valP =                  55 D_stat = 0

               160    clk=0
                      F Reg:          F_predPC =                  64
                      fetch:          f_predPC =                  65
                      D Reg:          D_icode =  7 D_ifun =  3 D_rA =  0 D_rB = 15 D_valC =                  64 D_valP =                  55 D_stat = 0

               170    clk=1
                      F Reg:          F_predPC =                  65
                      fetch:          f_predPC =                  65
                      D Reg:          D_icode =  0 D_ifun =  0 D_rA =  0 D_rB = 15 D_valC =                  64 D_valP =                  65 D_stat = 1
```

2) DECODE + WRITE BACK




```
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/Final_Project/project-team_5-main/Pipeline$ iverilog decode_up_tb.v
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/Final_Project/project-team_5-main/Pipeline$ ./a.out
WARNING: ./fetch_1.v:45: $readmemb(2.txt): Not enough words in the file for the requested range [0:1023].
VCD info: dumpfile decode_up_tb.vcd opened for output.
                   0clk = 0, F_predPC =                    0, f_pc =                    10, D_icode =  x, D_ifun =  x, D_rA =  x, D_rB =  x, D_valC =                    x, D_valP =                    x,
                                D_stat = x, E_icode =  x, E_ifun =  x, E_valA =                    x, E_valB =                    x, E_valC =                    x, E_dstE =  x, E_dstM =  x, E_srcA =  x, E_srcB =  x, E_stat = x
                  10clk = 1, F_predPC =                   10, f_pc =                    20, D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  2, D_valC =                    9, D_valP =                   10,
                                D_stat = 0, E_icode =  x, E_ifun =  x, E_valA =                    0, E_valB =                    0, E_valC =                    x, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = x
                  20clk = 0, F_predPC =                   10, f_pc =                    20, D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  2, D_valC =                    9, D_valP =                   10,
                                D_stat = 0, E_icode =  x, E_ifun =  x, E_valA =                    0, E_valB =                    0, E_valC =                    x, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = x
                  30clk = 1, F_predPC =                   20, f_pc =                    22, D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  3, D_valC =                   21, D_valP =                   20,
                                D_stat = 0, E_icode =  3, E_ifun =  0, E_valA =                    0, E_valB =                    0, E_valC =                    9, E_dstE =  2, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0
                  40clk = 0, F_predPC =                   20, f_pc =                    22, D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  3, D_valC =                   21, D_valP =                   20,
                                D_stat = 0, E_icode =  3, E_ifun =  0, E_valA =                    0, E_valB =                    0, E_valC =                    9, E_dstE =  2, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0
                  50clk = 1, F_predPC =                   22, f_pc =                    32, D_icode =  6, D_ifun =  1, D_rA =  2, D_rB =  3, D_valC =                   21, D_valP =                   22,
                                D_stat = 0, E_icode =  3, E_ifun =  0, E_valA =                    0, E_valB =                    0, E_valC =                   21, E_dstE =  3, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0
                  60clk = 0, F_predPC =                   22, f_pc =                    32, D_icode =  6, D_ifun =  1, D_rA =  2, D_rB =  3, D_valC =                   21, D_valP =                   22,
                                D_stat = 0, E_icode =  3, E_ifun =  0, E_valA =                    0, E_valB =                    0, E_valC =                   21, E_dstE =  3, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0
                  70clk = 1, F_predPC =                   32, f_pc =                    42, D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  4, D_valC =                  128, D_valP =                   32,
                                D_stat = 0, E_icode =  6, E_ifun =  1, E_valA =                    9, E_valB =                  128, E_valC =                   21, E_dstE =  3, E_dstM = 15, E_srcA =  2, E_srcB =  3, E_stat = 0
                  80clk = 0, F_predPC =                   32, f_pc =                    42, D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  4, D_valC =                  128, D_valP =                   32,
                                D_stat = 0, E_icode =  6, E_ifun =  1, E_valA =                    9, E_valB =                  128, E_valC =                   21, E_dstE =  3, E_dstM = 15, E_srcA =  2, E_srcB =  3, E_stat = 0
                  90clk = 1, F_predPC =                   42, f_pc =                    44, D_icode =  4, D_ifun =  0, D_rA =  4, D_rB =  3, D_valC =                  100, D_valP =                   42,
                                D_stat = 0, E_icode =  3, E_ifun =  0, E_valA =                    9, E_valB =                  128, E_valC =                  128, E_dstE =  4, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0
                 100clk = 0, F_predPC =                   42, f_pc =                    44, D_icode =  4, D_ifun =  0, D_rA =  4, D_rB =  3, D_valC =                  100, D_valP =                   42,
                                D_stat = 0, E_icode =  3, E_ifun =  0, E_valA =                    9, E_valB =                  128, E_valC =                  128, E_dstE =  4, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0
                 110clk = 1, F_predPC =                   44, f_pc =                    46, D_icode = 10, D_ifun =  0, D_rA =  2, D_rB = 15, D_valC =                  100, D_valP =                   44,
                                D_stat = 0, E_icode =  4, E_ifun =  0, E_valA =                  120, E_valB =                  128, E_valC =                  100, E_dstE =  3, E_dstM = 15, E_srcA =  4, E_srcB =  3, E_stat = 0
                 120clk = 0, F_predPC =                   44, f_pc =                    46, D_icode = 10, D_ifun =  0, D_rA =  2, D_rB = 15, D_valC =                  100, D_valP =                   44,
                                D_stat = 0, E_icode =  4, E_ifun =  0, E_valA =                  120, E_valB =                  128, E_valC =                  100, E_dstE =  3, E_dstM = 15, E_srcA =  4, E_srcB =  3, E_stat = 0
                 130clk = 1, F_predPC =                   46, f_pc =                    64, D_icode = 11, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                  100, D_valP =                   46,
                                D_stat = 0, E_icode = 10, E_ifun =  0, E_valA =                    9, E_valB =                  120, E_valC =                  100, E_dstE =  4, E_dstM = 15, E_srcA =  2, E_srcB =  4, E_stat = 0
                 140clk = 0, F_predPC =                   46, f_pc =                    64, D_icode = 11, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                  100, D_valP =                   46,
                                D_stat = 0, E_icode = 10, E_ifun =  0, E_valA =                    9, E_valB =                  120, E_valC =                  100, E_dstE =  4, E_dstM = 15, E_srcA =  2, E_srcB =  4, E_stat = 0
                 150clk = 1, F_predPC =                   64, f_pc =                    65, D_icode =  7, D_ifun =  3, D_rA =  0, D_rB = 15, D_valC =                   64, D_valP =                   55,
                                D_stat = 0, E_icode = 11, E_ifun =  0, E_valA =                  120, E_valB =                  120, E_valC =                  100, E_dstE =  4, E_dstM =  0, E_srcA =  4, E_srcB =  4, E_stat = 0
                 160clk = 0, F_predPC =                   64, f_pc =                    65, D_icode =  7, D_ifun =  3, D_rA =  0, D_rB = 15, D_valC =                   64, D_valP =                   55,
                                D_stat = 0, E_icode = 11, E_ifun =  0, E_valA =                  120, E_valB =                  120, E_valC =                  100, E_dstE =  4, E_dstM =  0, E_srcA =  4, E_srcB =  4, E_stat = 0
                 170clk = 1, F_predPC =                   65, f_pc =                    65, D_icode =  0, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                   64, D_valP =                   65,
                                D_stat = 1, E_icode =  7, E_ifun =  3, E_valA =                   55, E_valB =                  120, E_valC =                   64, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0
                 180clk = 0, F_predPC =                   65, f_pc =                    65, D_icode =  0, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                   64, D_valP =                   65,
                                D_stat = 1, E_icode =  7, E_ifun =  3, E_valA =                   55, E_valB =                  120, E_valC =                   64, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0
                 190clk = 1, F_predPC =                   65, f_pc =                    65, D_icode =  9, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                   64, D_valP =                   66,
                                D_stat = 1, E_icode =  0, E_ifun =  0, E_valA =                   55, E_valB =                  120, E_valC =                   64, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 1
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/Final_Project/project-team_5-main/Pipeline$
```

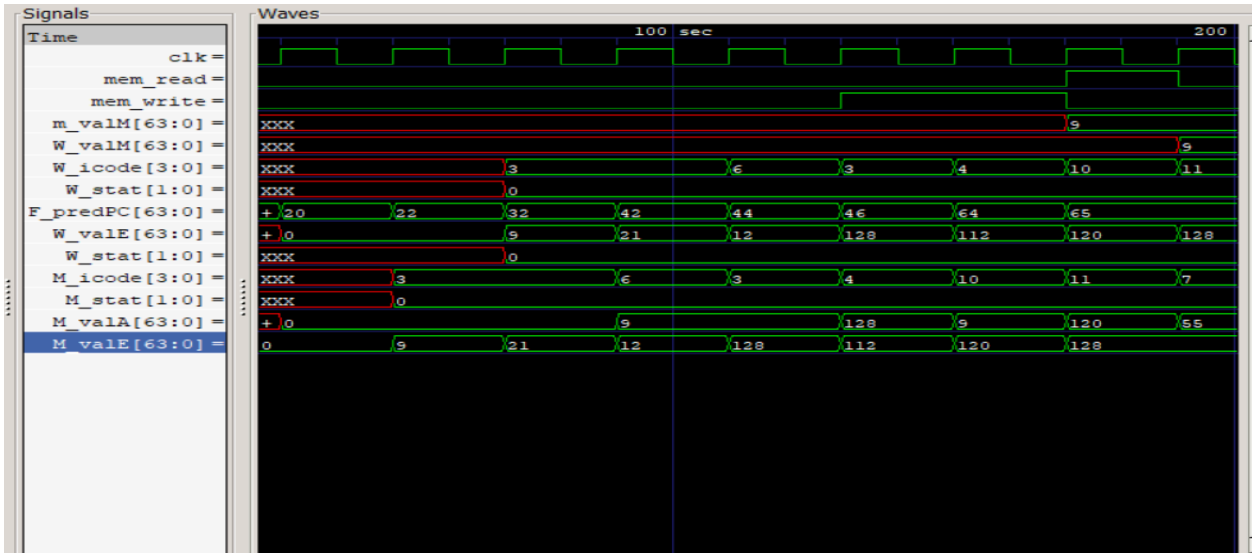
3) EXECUTE

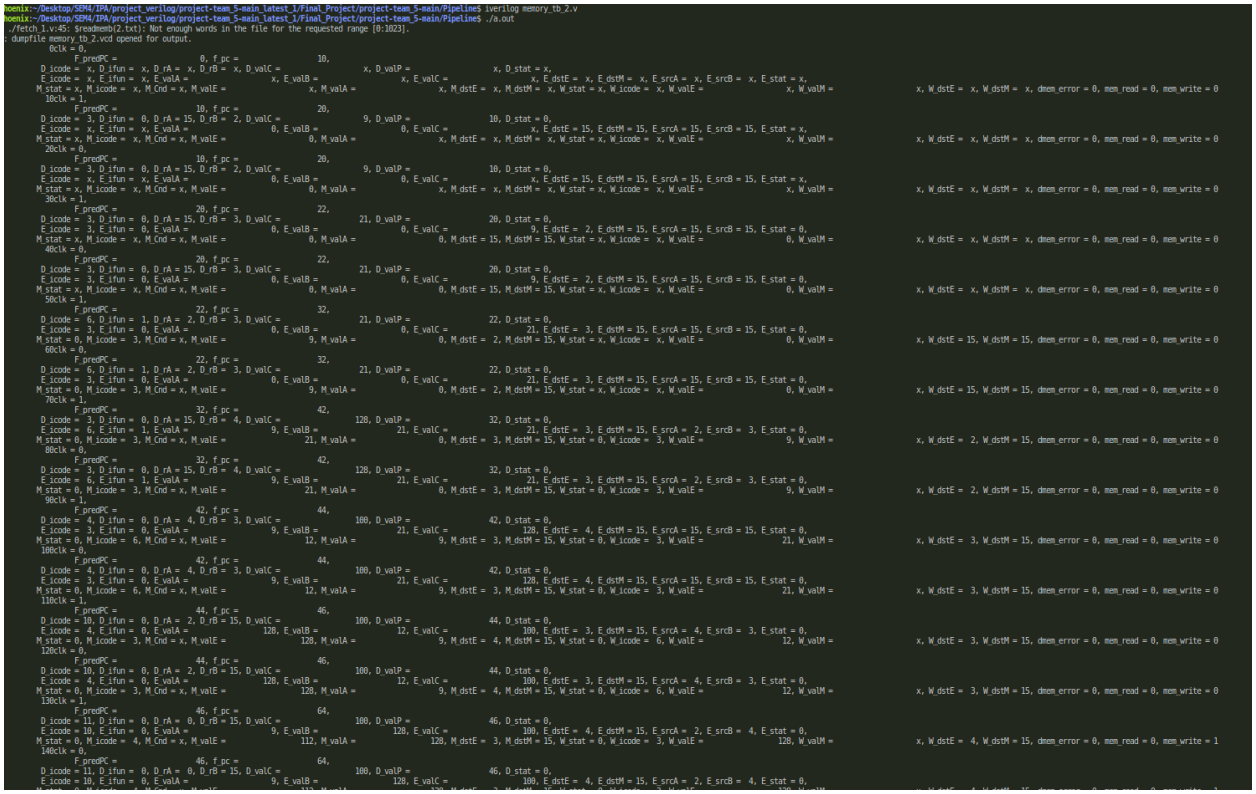


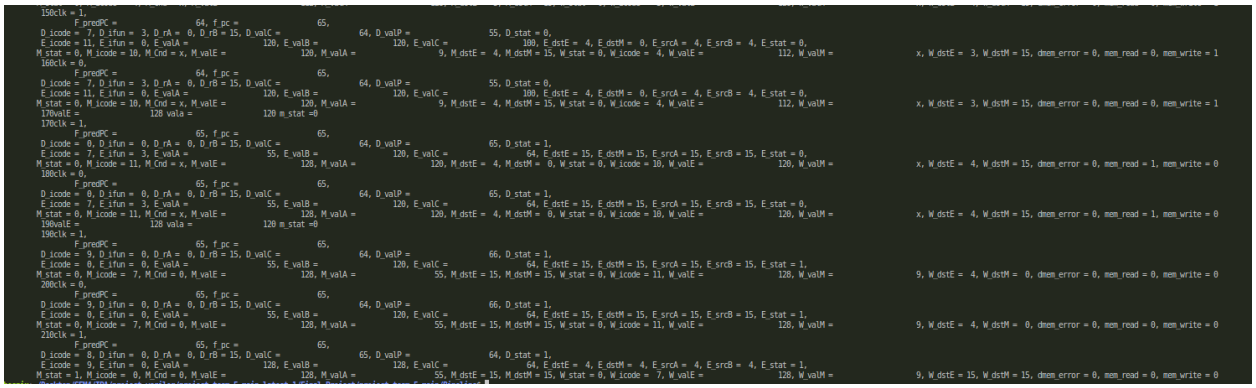

```
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/Final_Project/project-team_5-main/Pipeline$ iverilog execute_tb_2.v
himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/Final_Project/project-team_5-main/Pipeline$ ./a.out
WARNING: ./fetch_1.v:45: $readmemb(2.txt): Not enough words in the file for the requested range [0:1023].
VCD info: dumpfile decode_tb_2.vcd opened for output.
          0 clk = 0,
                  F_predPC =                    0, f_pc =                   10,
           D_icode =  x, D_ifun =  x, D_rA =  x, D_rB =  x, D_valC =                    x, D_valP =                    x, D_stat = x,
           E_icode =  x, E_ifun =  x, E_valA =                    x, E_valB =                    x, E_valC =                    x, E_dstE =  x, E_dstM =  x, E_srcA =  x, E_srcB =  x, E_stat = x,
          M_stat = x, M_icode =  x, M_Cnd = x, M_valE =                    x, M_valA =                    x, M_dstE =  x, M_dstM =  x
           10 clk = 1,
                  F_predPC =                   10, f_pc =                   20,
           D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  2, D_valC =                    9, D_valP =                   10, D_stat = 0,
           E_icode =  x, E_ifun =  x, E_valA =                    0, E_valB =                    0, E_valC =                    x, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = x,
          M_stat = x, M_icode =  x, M_Cnd = x, M_valE =                    0, M_valA =                    x, M_dstE =  x, M_dstM =  x
           20 clk = 0,
                  F_predPC =                   10, f_pc =                   20,
           D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  2, D_valC =                    9, D_valP =                   10, D_stat = 0,
           E_icode =  x, E_ifun =  x, E_valA =                    0, E_valB =                    0, E_valC =                    x, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = x,
          M_stat = x, M_icode =  x, M_Cnd = x, M_valE =                    0, M_valA =                    x, M_dstE =  x, M_dstM =  x
           30 clk = 1,
                  F_predPC =                   20, f_pc =                   22,
           D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  3, D_valC =                   21, D_valP =                   20, D_stat = 0,
           E_icode =  3, E_ifun =  0, E_valA =                    0, E_valB =                    0, E_valC =                    9, E_dstE =  2, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0,
          M_stat = x, M_icode =  x, M_Cnd = x, M_valE =                    0, M_valA =                    0, M_dstE = 15, M_dstM = 15
           40 clk = 0,
                  F_predPC =                   20, f_pc =                   22,
           D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  3, D_valC =                   21, D_valP =                   20, D_stat = 0,
           E_icode =  3, E_ifun =  0, E_valA =                    0, E_valB =                    0, E_valC =                    9, E_dstE =  2, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0,
          M_stat = x, M_icode =  x, M_Cnd = x, M_valE =                    0, M_valA =                    0, M_dstE = 15, M_dstM = 15
           50 clk = 1,
                  F_predPC =                   22, f_pc =                   32,
           D_icode =  6, D_ifun =  1, D_rA =  2, D_rB =  3, D_valC =                   21, D_valP =                   22, D_stat = 0,
           E_icode =  3, E_ifun =  0, E_valA =                    0, E_valB =                    0, E_valC =                   21, E_dstE =  3, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0,
          M_stat = 0, M_icode =  3, M_Cnd = x, M_valE =                    9, M_valA =                    0, M_dstE =  2, M_dstM = 15
           60 clk = 0,
                  F_predPC =                   22, f_pc =                   32,
           D_icode =  6, D_ifun =  1, D_rA =  2, D_rB =  3, D_valC =                   21, D_valP =                   22, D_stat = 0,
           E_icode =  3, E_ifun =  0, E_valA =                    0, E_valB =                    0, E_valC =                   21, E_dstE =  3, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0,
          M_stat = 0, M_icode =  3, M_Cnd = x, M_valE =                    9, M_valA =                    0, M_dstE =  2, M_dstM = 15
           70 clk = 1,
                  F_predPC =                   32, f_pc =                   42,
           D_icode =  3, D_ifun =  0, D_rA = 15, D_rB =  4, D_valC =                  128, D_valP =                   32, D_stat = 0,
           E_icode =  6, E_ifun =  1, E_valA =                    9, E_valB =                   21, E_valC =                   21, E_dstE =  3, E_dstM = 15, E_srcA =  2, E_srcB =  3, E_stat = 0,
          M_stat = 0, M_icode =  3, M_Cnd = x, M_valE =                   21, M_valA =                    0, M_dstE =  3, M_dstM = 15
           80 clk = 0,
```

```
100 clk = 0,
        F_predPC =                  42, f_pc =                  44,
D_icode =  4, D_ifun =  0, D_rA =  4, D_rB =  3, D_valC =                 100, D_valP =                  42, D_stat = 0,
E_icode =  3, E_ifun =  0, E_valA =                   9, E_valB =                  21, E_valC =                 128, E_dstE =  4, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0,
M_stat = 0, M_icode =  6, M_Cnd = x, M_valE =                  12, M_valA =                   9, M_dstE =  3, M_dstM = 15
 110 clk = 1,
        F_predPC =                  44, f_pc =                  46,
D_icode = 10, D_ifun =  0, D_rA =  2, D_rB = 15, D_valC =                 100, D_valP =                  44, D_stat = 0,
E_icode =  4, E_ifun =  0, E_valA =                 128, E_valB =                  12, E_valC =                 100, E_dstE =  3, E_dstM = 15, E_srcA =  4, E_srcB =  3, E_stat = 0,
M_stat = 0, M_icode =  3, M_Cnd = x, M_valE =                 128, M_valA =                   9, M_dstE =  4, M_dstM = 15
 120 clk = 0,
        F_predPC =                  44, f_pc =                  46,
D_icode = 10, D_ifun =  0, D_rA =  2, D_rB = 15, D_valC =                 100, D_valP =                  44, D_stat = 0,
E_icode =  4, E_ifun =  0, E_valA =                 128, E_valB =                  12, E_valC =                 100, E_dstE =  3, E_dstM = 15, E_srcA =  4, E_srcB =  3, E_stat = 0,
M_stat = 0, M_icode =  3, M_Cnd = x, M_valE =                 128, M_valA =                   9, M_dstE =  4, M_dstM = 15
 130 clk = 1,
        F_predPC =                  46, f_pc =                  64,
D_icode = 11, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                 100, D_valP =                  46, D_stat = 0,
E_icode = 10, E_ifun =  0, E_valA =                   9, E_valB =                 128, E_valC =                 100, E_dstE =  4, E_dstM = 15, E_srcA =  2, E_srcB =  4, E_stat = 0,
M_stat = 0, M_icode =  4, M_Cnd = x, M_valE =                 112, M_valA =                 128, M_dstE =  3, M_dstM = 15
 140 clk = 0,
        F_predPC =                  46, f_pc =                  64,
D_icode = 11, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                 100, D_valP =                  46, D_stat = 0,
E_icode = 10, E_ifun =  0, E_valA =                   9, E_valB =                 128, E_valC =                 100, E_dstE =  4, E_dstM = 15, E_srcA =  2, E_srcB =  4, E_stat = 0,
M_stat = 0, M_icode =  4, M_Cnd = x, M_valE =                 112, M_valA =                 128, M_dstE =  3, M_dstM = 15
 150 clk = 1,
        F_predPC =                  64, f_pc =                  65,
D_icode =  7, D_ifun =  3, D_rA =  0, D_rB = 15, D_valC =                  64, D_valP =                  55, D_stat = 0,
E_icode = 11, E_ifun =  0, E_valA =                 120, E_valB =                 120, E_valC =                 100, E_dstE =  4, E_dstM =  0, E_srcA =  4, E_srcB =  4, E_stat = 0,
M_stat = 0, M_icode = 10, M_Cnd = x, M_valE =                 120, M_valA =                   9, M_dstE =  4, M_dstM = 15
 160 clk = 0,
        F_predPC =                  64, f_pc =                  65,
D_icode =  7, D_ifun =  3, D_rA =  0, D_rB = 15, D_valC =                  64, D_valP =                  55, D_stat = 0,
E_icode = 11, E_ifun =  0, E_valA =                 120, E_valB =                 120, E_valC =                 100, E_dstE =  4, E_dstM =  0, E_srcA =  4, E_srcB =  4, E_stat = 0,
M_stat = 0, M_icode = 10, M_Cnd = x, M_valE =                 120, M_valA =                   9, M_dstE =  4, M_dstM = 15
 170 clk = 1,
        F_predPC =                  65, f_pc =                  65,
D_icode =  0, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                  64, D_valP =                  65, D_stat = 1,
E_icode =  7, E_ifun =  3, E_valA =                  55, E_valB =                 120, E_valC =                  64, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0,
M_stat = 0, M_icode = 11, M_Cnd = x, M_valE =                 128, M_valA =                 120, M_dstE =  4, M_dstM =  0
 180 clk = 0,
        F_predPC =                  65, f_pc =                  65,
D_icode =  0, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                  64, D_valP =                  65, D_stat = 1,
E_icode =  7, E_ifun =  3, E_valA =                  55, E_valB =                 120, E_valC =                  64, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 0,
M_stat = 0, M_icode = 11, M_Cnd = x, M_valE =                 128, M_valA =                 120, M_dstE =  4, M_dstM =  0
 190 clk = 1,
        F_predPC =                  65, f_pc =                  65,
D_icode =  9, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                  64, D_valP =                  66, D_stat = 1,
E_icode =  0, E_ifun =  0, E_valA =                  55, E_valB =                 120, E_valC =                  64, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 1,
M_stat = 0, M_icode =  7, M_Cnd = 0, M_valE =                 128, M_valA =                  55, M_dstE = 15, M_dstM = 15
 200 clk = 0,
        F_predPC =                  65, f_pc =                  65,
D_icode =  9, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                  64, D_valP =                  66, D_stat = 1,
E_icode =  0, E_ifun =  0, E_valA =                  55, E_valB =                 120, E_valC =                  64, E_dstE = 15, E_dstM = 15, E_srcA = 15, E_srcB = 15, E_stat = 1,
M_stat = 0, M_icode =  7, M_Cnd = 0, M_valE =                 128, M_valA =                  55, M_dstE = 15, M_dstM = 15
 210 clk = 1,
        F_predPC =                  65, f_pc =                  65,
D_icode =  8, D_ifun =  0, D_rA =  0, D_rB = 15, D_valC =                  65, D_valP =                  64, D_stat = 1,
E_icode =  9, E_ifun =  0, E_valA =                 120, E_valB =                 120, E_valC =                  64, E_dstE =  4, E_dstM =  4, E_srcA =  4, E_srcB =  4, E_stat = 1,
M_stat = 1, M_icode =  0, M_Cnd = 0, M_valE =                 128, M_valA =                  55, M_dstE = 15, M_dstM = 15
```

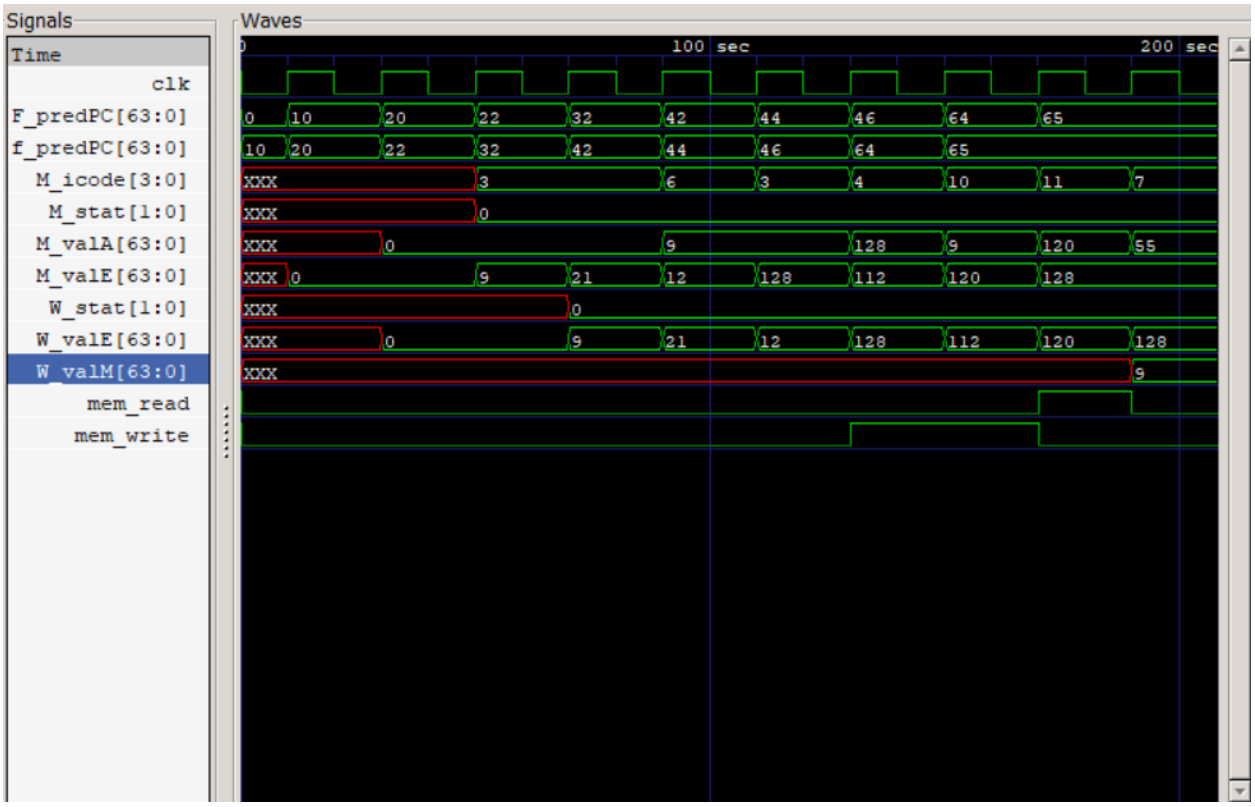
4) MEMORY







5) WRITE-BACK



reg_file.txt

```
1   // 0x00000000
2   0000000000000009
3   xxxxxxxxxxxxxxxx
4   0000000000000009
5   0000000000000080
6   0000000000000078
7   xxxxxxxxxxxxxxxx
8   xxxxxxxxxxxxxxxx
9   xxxxxxxxxxxxxxxx
10  xxxxxxxxxxxxxxxx
11  xxxxxxxxxxxxxxxx
12  xxxxxxxxxxxxxxxx
13  xxxxxxxxxxxxxxxx
14  xxxxxxxxxxxxxxxx
15  xxxxxxxxxxxxxxxx
16  xxxxxxxxxxxxxxxx
17  0000000000000009
18
19
20
21
```

In our implementation we are separately working on Decode and Write Back stages.

The function performed by each block remains almost the same as sequential architecture, with a few fundamental changes.
PC update is shifted to the beginning of the cycle and is directly computed along with the fetch stage.
For every instruction, valP is set to the predicted value of PC.

Handling Hazards in Pipeline

Introducing pipelining into a system with feedback can lead to problems when there are dependencies between successive instructions. These dependencies can take two forms: (1) data dependencies and (2) control hazards.

Avoiding Data Pipeline Hazards:

In our implementation, data pipeline hazards are handled using forwarding logic.
The technique of passing a result value directly from one pipeline stage to an earlier one is commonly known as data forwarding.
Data forwarding requires adding additional data connections and control logic to the basic hardware structure.

- It can also use the ALU output (signal e_valE) for operand valA or valB.
- It can use the value that has just been read from the data memory (signal m_valM) for operand valA or valB.
- It can use the value in the memory stage (signal M_valE) for operand valA or valB.
- It can use the value in the write-back stage (signal W_valE or signal W_valM) for operand valA or valB.

Avoiding Load/Use Data Hazards:

One class of data hazards cannot be handled purely by forwarding, because memory reads occur late in the pipeline. These are called load/use hazards and they occur when one instruction reads a value from memory for register while the next instruction needs this value as a source operand. We can avoid a load/use data hazard with a combination of stalling and forwarding. This requires modifications of the control logic.

Avoiding Control Hazards:

Control hazards arise when the processor cannot reliably determine the address of the next instruction based on the current instruction in the fetch stage. Control hazards can only occur in our pipelined processor for ret and jump instructions. In case of ret, the processor is stalled for 3 clock cycles after the ret instruction. While in case of jump misprediction, the pipeline can simply cancel the two mis-fetched instructions by injecting bubbles into the decode and execute stages on the following cycle while also fetching the instruction following the jump instruction.

Pipe-Control Logic

This block is used for handling data and control hazards. In cases of return, mispredicted branches and load/use hazards, we can observe misbehavior and incorrect results. To handle such cases, we might need to stall some stages and insert a bubble in some.

Condition	F	D	E	M	W
Processing `ret`	stall	bubble	normal	normal	normal
Load/Use Hazard	stall	stall	bubble	normal	normal
Mispredicted Branch	normal	bubble	bubble	normal	normal

COMPLETE PIPELINE DESIGN IMPLEMENTATION

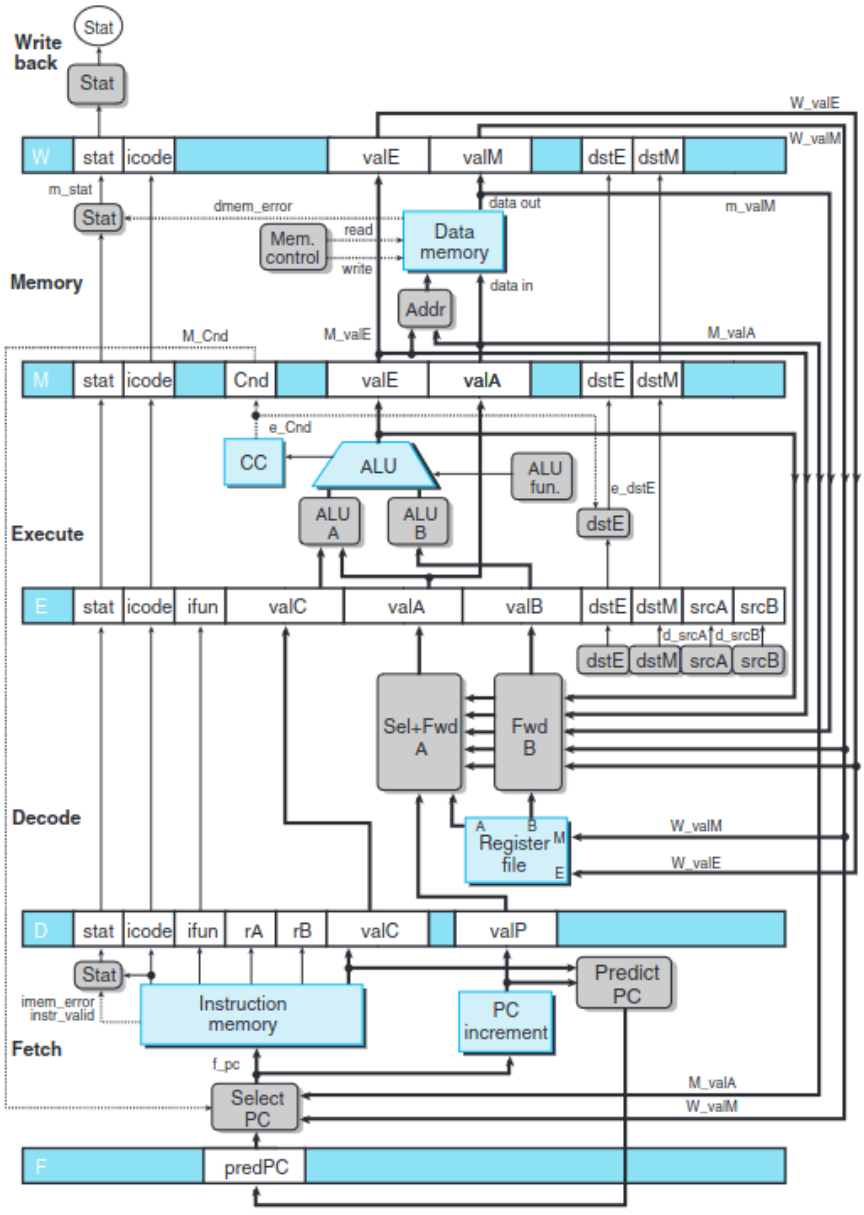


Figure 4.52 Hardware structure of PIPE, our final pipelined implementation. The additional bypassing paths enable forwarding the results from the three preceding instructions. This allows us to handle most forms of data hazards without stalling the pipeline.

```
: dumpfile pipe.vcd opened for output.
          0    clk=0
               F Reg:          F_predPC = 0
               fetch:          f_predPC = 10
               D Reg:          D_icode = xxxx D_ifun = xxxx D_rA = xxxx D_rB = xxxx D_valC = 0 D_valP = 0 D_stat = 0
                E Reg:         E_icode = xxxx E_ifun = xxxx E_valA = 0 E_valB = 0 E_valC = 0 E_dstE = xxxx E_dstM = xxxx E_srcA = 0 E_srcB = 0 E_stat = 0
               execute:        e_cnd = x e_valE = 0
               M Reg:          M_icode = xxxx M_cnd = x M_valA = 0 M_valE = 0 M_dstE = xxxx, M_dstM = xxxx M_stat = 0
               memory:         m_valM = 0
               W Reg:          W_icode = xxxx W_valE = 0 W_valM = 0 W_dstE = xxxx W_dstM = xxxx W_stat = 0
               F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

         10    clk=1
               F Reg:          F_predPC = 10
               fetch:          f_predPC = 20
               D Reg:          D_icode = 0011 D_ifun = 0000 D_rA = 1111 D_rB = 0010 D_valC = 9 D_valP = 10 D_stat = 0
                E Reg:         E_icode = xxxx E_ifun = xxxx E_valA = 0 E_valB = 0 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
               execute:        e_cnd = x e_valE = 0
               M Reg:          M_icode = xxxx M_cnd = x M_valA = 0 M_valE = 0 M_dstE = xxxx, M_dstM = xxxx M_stat = 0
               memory:         m_valM = 0
               W Reg:          W_icode = xxxx W_valE = 0 W_valM = 0 W_dstE = xxxx W_dstM = xxxx W_stat = 0
               F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

         20    clk=0
               F Reg:          F_predPC = 10
               fetch:          f_predPC = 20
               D Reg:          D_icode = 0011 D_ifun = 0000 D_rA = 1111 D_rB = 0010 D_valC = 9 D_valP = 10 D_stat = 0
                E Reg:         E_icode = xxxx E_ifun = xxxx E_valA = 0 E_valB = 0 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
               execute:        e_cnd = x e_valE = 0
               M Reg:          M_icode = xxxx M_cnd = x M_valA = 0 M_valE = 0 M_dstE = xxxx, M_dstM = xxxx M_stat = 0
               memory:         m_valM = 0
               W Reg:          W_icode = xxxx W_valE = 0 W_valM = 0 W_dstE = xxxx W_dstM = xxxx W_stat = 0
               F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

         30    clk=1
               F Reg:          F_predPC = 20
               fetch:          f_predPC = 22
               D Reg:          D_icode = 0011 D_ifun = 0000 D_rA = 1111 D_rB = 0011 D_valC = 21 D_valP = 20 D_stat = 0
                E Reg:         E_icode = 0011 E_ifun = 0000 E_valA = 0 E_valB = 0 E_valC = 9 E_dstE = 0010 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
               execute:        e_cnd = x e_valE = 9
               M Reg:          M_icode = xxxx M_cnd = x M_valA = 0 M_valE = 0 M_dstE = 1111, M_dstM = 1111 M_stat = 0
               memory:         m_valM = 0
               W Reg:          W_icode = xxxx W_valE = 0 W_valM = 0 W_dstE = xxxx W_dstM = xxxx W_stat = 0
               F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

         40    clk=0
               F Reg:          F_predPC = 20
               fetch:          f_predPC = 22
               D Reg:          D_icode = 0011 D_ifun = 0000 D_rA = 1111 D_rB = 0011 D_valC = 21 D_valP = 20 D_stat = 0
                E Reg:         E_icode = 0011 E_ifun = 0000 E_valA = 0 E_valB = 0 E_valC = 9 E_dstE = 0010 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
               execute:        e_cnd = x e_valE = 9
               M Reg:          M_icode = xxxx M_cnd = x M_valA = 0 M_valE = 0 M_dstE = 1111, M_dstM = 1111 M_stat = 0
               memory:         m_valM = 0
               W Reg:          W_icode = xxxx W_valE = 0 W_valM = 0 W_dstE = xxxx W_dstM = xxxx W_stat = 0
               F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

         50    clk=1
               F Reg:          F_predPC = 22
               fetch:          f_predPC = 32
               D Reg:          D_icode = 0110 D_ifun = 0001 D_rA = 0010 D_rB = 0011 D_valC = 21 D_valP = 22 D_stat = 0
                E Reg:         E_icode = 0011 E_ifun = 0000 E_valA = 0 E_valB = 0 E_valC = 21 E_dstE = 0011 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
               execute:        e_cnd = x e_valE = 21
               M Reg:          M_icode = 0011 M_cnd = x M_valA = 0 M_valE = 9 M_dstE = 0010, M_dstM = 1111 M_stat = 0
               memory:         m_valM = 0
               W Reg:          W_icode = xxxx W_valE = 0 W_valM = 0 W_dstE = 1111 W_dstM = 1111 W_stat = 0
               F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0
```

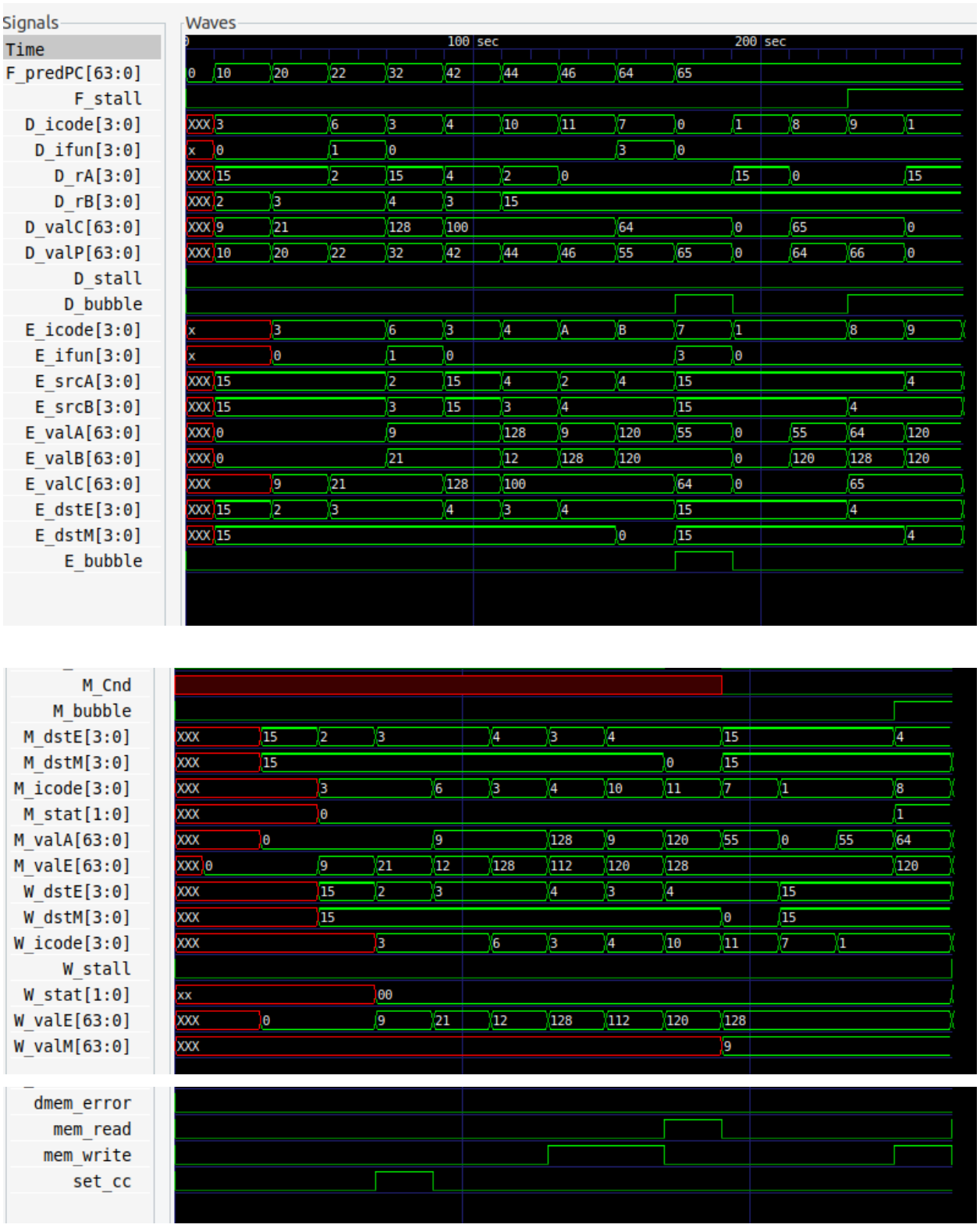
```
 60    clk=0
       F Reg:          F_predPC = 22
       fetch:          f_predPC = 32
       D Reg:          D_icode = 0110 D_ifun = 0001 D_rA = 0010 D_rB = 0011 D_valC = 21 D_valP = 22 D_stat = 0
        E Reg:         E_icode = 0011 E_ifun = 0000 E_valA = 0 E_valB = 0 E_valC = 21 E_dstE = 0011 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = x e_valE = 21
       M Reg:          M_icode = 0011 M_cnd = x M_valA = 0 M_valE = 9 M_dstE = 0010, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = xxxx W_valE = 0 W_valM = 0 W_dstE = 1111 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

 70    clk=1
       F Reg:          F_predPC = 32
       fetch:          f_predPC = 42
       D Reg:          D_icode = 0011 D_ifun = 0000 D_rA = 1111 D_rB = 0100 D_valC = 128 D_valP = 32 D_stat = 0
        E Reg:         E_icode = 0110 E_ifun = 0001 E_valA = 9 E_valB = 21 E_valC = 21 E_dstE = 0011 E_dstM = 1111 E_srcA = 2 E_srcB = 3 E_stat = 0
       execute:        e_cnd = x e_valE = 12
       M Reg:          M_icode = 0011 M_cnd = x M_valA = 0 M_valE = 21 M_dstE = 0011, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0011 W_valE = 9 W_valM = 0 W_dstE = 0010 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 1 m_stat =0 w_stat =0 W_stall =0

 80    clk=0
       F Reg:          F_predPC = 32
       fetch:          f_predPC = 42
       D Reg:          D_icode = 0011 D_ifun = 0000 D_rA = 1111 D_rB = 0100 D_valC = 128 D_valP = 32 D_stat = 0
        E Reg:         E_icode = 0110 E_ifun = 0001 E_valA = 9 E_valB = 21 E_valC = 21 E_dstE = 0011 E_dstM = 1111 E_srcA = 2 E_srcB = 3 E_stat = 0
       execute:        e_cnd = x e_valE = 12
       M Reg:          M_icode = 0011 M_cnd = x M_valA = 0 M_valE = 21 M_dstE = 0011, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0011 W_valE = 9 W_valM = 0 W_dstE = 0010 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 1 m_stat =0 w_stat =0 W_stall =0

 90    clk=1
       F Reg:          F_predPC = 42
       fetch:          f_predPC = 44
       D Reg:          D_icode = 0100 D_ifun = 0000 D_rA = 0100 D_rB = 0011 D_valC = 100 D_valP = 42 D_stat = 0
        E Reg:         E_icode = 0011 E_ifun = 0000 E_valA = 9 E_valB = 21 E_valC = 128 E_dstE = 0100 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = x e_valE = 128
       M Reg:          M_icode = 0110 M_cnd = x M_valA = 9 M_valE = 12 M_dstE = 0011, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0011 W_valE = 21 W_valM = 0 W_dstE = 0011 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

100    clk=0
       F Reg:          F_predPC = 42
       fetch:          f_predPC = 44
       D Reg:          D_icode = 0100 D_ifun = 0000 D_rA = 0100 D_rB = 0011 D_valC = 100 D_valP = 42 D_stat = 0
        E Reg:         E_icode = 0011 E_ifun = 0000 E_valA = 9 E_valB = 21 E_valC = 128 E_dstE = 0100 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = x e_valE = 128
       M Reg:          M_icode = 0110 M_cnd = x M_valA = 9 M_valE = 12 M_dstE = 0011, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0011 W_valE = 21 W_valM = 0 W_dstE = 0011 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

110    clk=1
       F Reg:          F_predPC = 44
       fetch:          f_predPC = 46
       D Reg:          D_icode = 1010 D_ifun = 0000 D_rA = 0010 D_rB = 1111 D_valC = 100 D_valP = 44 D_stat = 0
        E Reg:         E_icode = 0100 E_ifun = 0000 E_valA = 128 E_valB = 12 E_valC = 100 E_dstE = 0011 E_dstM = 1111 E_srcA = 4 E_srcB = 3 E_stat = 0
       execute:        e_cnd = x e_valE = 112
       M Reg:          M_icode = 0011 M_cnd = x M_valA = 9 M_valE = 128 M_dstE = 0100, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0110 W_valE = 12 W_valM = 0 W_dstE = 0011 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

120    clk=0
       F Reg:          F_predPC = 44
       fetch:          f_predPC = 46
       D Reg:          D_icode = 1010 D_ifun = 0000 D_rA = 0010 D_rB = 1111 D_valC = 100 D_valP = 44 D_stat = 0
        E Reg:         E_icode = 0100 E_ifun = 0000 E_valA = 128 E_valB = 12 E_valC = 100 E_dstE = 0011 E_dstM = 1111 E_srcA = 4 E_srcB = 3 E_stat = 0
       execute:        e_cnd = x e_valE = 112
       M Reg:          M_icode = 0011 M_cnd = x M_valA = 9 M_valE = 128 M_dstE = 0100, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0110 W_valE = 12 W_valM = 0 W_dstE = 0011 W_dstM = 1111 W_stat = 0
```

```
130    clk=1
       F Reg:          F_predPC = 46
       fetch:          f_predPC = 64
       D Reg:          D_icode = 1011 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 100 D_valP = 46 D_stat = 0
        E Reg:         E_icode = 1010 E_ifun = 0000 E_valA = 9 E_valB = 128 E_valC = 100 E_dstE = 0100 E_dstM = 1111 E_srcA = 2 E_srcB = 4 E_stat = 0
       execute:        e_cnd = x e_valE = 120
       M Reg:          M_icode = 0100 M_cnd = x M_valA = 128 M_valE = 112 M_dstE = 0011, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0011 W_valE = 128 W_valM = 0 W_dstE = 0100 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

140    clk=0
       F Reg:          F_predPC = 46
       fetch:          f_predPC = 64
       D Reg:          D_icode = 1011 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 100 D_valP = 46 D_stat = 0
        E Reg:         E_icode = 1010 E_ifun = 0000 E_valA = 9 E_valB = 128 E_valC = 100 E_dstE = 0100 E_dstM = 1111 E_srcA = 2 E_srcB = 4 E_stat = 0
       execute:        e_cnd = x e_valE = 120
       M Reg:          M_icode = 0100 M_cnd = x M_valA = 128 M_valE = 112 M_dstE = 0011, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0011 W_valE = 128 W_valM = 0 W_dstE = 0100 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

150    clk=1
       F Reg:          F_predPC = 64
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0111 D_ifun = 0011 D_rA = 0000 D_rB = 1111 D_valC = 64 D_valP = 55 D_stat = 0
        E Reg:         E_icode = 1011 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 100 E_dstE = 0100 E_dstM = 0000 E_srcA = 4 E_srcB = 4 E_stat = 0
       execute:        e_cnd = x e_valE = 128
       M Reg:          M_icode = 1010 M_cnd = x M_valA = 9 M_valE = 120 M_dstE = 0100, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0100 W_valE = 112 W_valM = 0 W_dstE = 0011 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

160    clk=0
       F Reg:          F_predPC = 64
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0111 D_ifun = 0011 D_rA = 0000 D_rB = 1111 D_valC = 64 D_valP = 55 D_stat = 0
        E Reg:         E_icode = 1011 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 100 E_dstE = 0100 E_dstM = 0000 E_srcA = 4 E_srcB = 4 E_stat = 0
       execute:        e_cnd = x e_valE = 128
       M Reg:          M_icode = 1010 M_cnd = x M_valA = 9 M_valE = 120 M_dstE = 0100, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0100 W_valE = 112 W_valM = 0 W_dstE = 0011 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

170valE =               128 vala =              120 m_stat =0
170    clk=1
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0000 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 64 D_valP = 65 D_stat = 1
        E Reg:         E_icode = 0111 E_ifun = 0011 E_valA = 55 E_valB = 120 E_valC = 64 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 1011 M_cnd = x M_valA = 120 M_valE = 128 M_dstE = 0100, M_dstM = 0000 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 1010 W_valE = 120 W_valM = 0 W_dstE = 0100 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 1 E_bubble = 1 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

180    clk=0
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0000 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 64 D_valP = 65 D_stat = 1
        E Reg:         E_icode = 0111 E_ifun = 0011 E_valA = 55 E_valB = 120 E_valC = 64 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 1011 M_cnd = x M_valA = 120 M_valE = 128 M_dstE = 0100, M_dstM = 0000 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 1010 W_valE = 120 W_valM = 0 W_dstE = 0100 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 1 E_bubble = 1 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

190valE =               128 vala =              120 m_stat =0
190    clk=1
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
        E Reg:         E_icode = 0001 E_ifun = 0000 E_valA = 0 E_valB = 0 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 0111 M_cnd = 0 M_valA = 55 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 1011 W_valE = 128 W_valM = 9 W_dstE = 0100 W_dstM = 0000 W_stat = 0
```

```
200    clk=0
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
        E Reg:         E_icode = 0001 E_ifun = 0000 E_valA = 0 E_valB = 0 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 0111 M_cnd = 0 M_valA = 55 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 1011 W_valE = 128 W_valM = 9 W_dstE = 0100 W_dstM = 0000 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

210    clk=1
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 1000 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 65 D_valP = 64 D_stat = 1
        E Reg:         E_icode = 0001 E_ifun = 0000 E_valA = 55 E_valB = 120 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 0001 M_cnd = 0 M_valA = 0 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 0111 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

220    clk=0
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 1000 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 65 D_valP = 64 D_stat = 1
        E Reg:         E_icode = 0001 E_ifun = 0000 E_valA = 55 E_valB = 120 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 0001 M_cnd = 0 M_valA = 0 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 0111 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

230    clk=1
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 1001 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 65 D_valP = 66 D_stat = 1
        E Reg:         E_icode = 1000 E_ifun = 0000 E_valA = 64 E_valB = 128 E_valC = 65 E_dstE = 0100 E_dstM = 1111 E_srcA = 15 E_srcB = 4 E_stat = 1
       execute:        e_cnd = 0 e_valE = 120
       M Reg:          M_icode = 0001 M_cnd = 0 M_valA = 55 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 0001 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
       F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

240    clk=0
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 1001 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 65 D_valP = 66 D_stat = 1
        E Reg:         E_icode = 1000 E_ifun = 0000 E_valA = 64 E_valB = 128 E_valC = 65 E_dstE = 0100 E_dstM = 1111 E_srcA = 15 E_srcB = 4 E_stat = 1
       execute:        e_cnd = 0 e_valE = 120
       M Reg:          M_icode = 0001 M_cnd = 0 M_valA = 55 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 0001 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
       F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

250    clk=1
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
        E Reg:         E_icode = 1001 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 65 E_dstE = 0100 E_dstM = 0100 E_srcA = 4 E_srcB = 4 E_stat = 1
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 1000 M_cnd = 0 M_valA = 64 M_valE = 120 M_dstE = 0100, M_dstM = 1111 M_stat = 1
       memory:         m_valM = 9
       W Reg:          W_icode = 0001 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
       F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =1 set_cc = 0 m_stat =1 w_stat =0 W_stall =0

260    clk=0
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
        E Reg:         E_icode = 1001 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 65 E_dstE = 0100 E_dstM = 0100 E_srcA = 4 E_srcB = 4 E_stat = 1
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 1000 M_cnd = 0 M_valA = 64 M_valE = 120 M_dstE = 0100, M_dstM = 1111 M_stat = 1
       memory:         m_valM = 9
       W Reg:          W_icode = 0001 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
       F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =1 set_cc = 0 m_stat =1 w_stat =0 W_stall =0
```

```
HALT
       270    clk=1
              F Reg:          F_predPC = 65
              fetch:          f_predPC = 65
              D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
               E Reg:         E_icode = 0001 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
              execute:        e_cnd = 0 e_valE = 128
              M Reg:          M_icode = 1001 M_cnd = 0 M_valA = 120 M_valE = 128 M_dstE = 0100, M_dstM = 0100 M_stat = 1
              memory:         m_valM = 64
              W Reg:          W_icode = 1000 W_valE = 120 W_valM = 9 W_dstE = 0100 W_dstM = 1111 W_stat = 1
              F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =1 set_cc = 0 m_stat =1 w_stat =1 W_stall =1

○ himani@Phoenix:~/Desktop/SEM4/IPA/project_verilog/project-team_5-main_latest_1/Final_Project/project-team_5-main/Pipeline$
```

Since the last instruction is Halt, we are finishing the program with HALT in the write back stage.



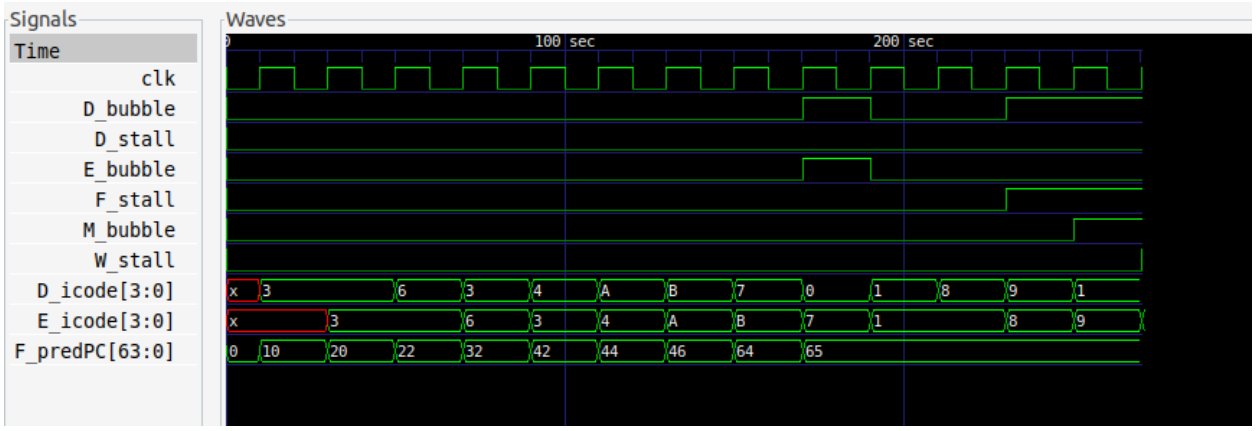
HAZARDS:

1) RETURN

Program used:

```
1    0x000: 30f20900000000000000 |     irmovq $9,  %rdx
2    0x00a: 30f31500000000000000 |     irmovq $21, %rbx
3    0x014: 6123                 |     subq %rdx, %rbx          # subtract
4    0x016: 30f480000000000000 |     irmovq $128,%rsp         # Problem 4.13
5    0x020: 4043640000000000000 |     rmmovq %rsp, 100(%rbx)   # store
6    0x02a: a02f                 |     pushq %rdx               # push
7    0x02c: b00f                 |     popq  %rax               # Problem 4.14
8    0x02e: 7340000000000000    |     je done                  # Not taken
9    0x037: 8041000000000000    |     call proc                # Problem 4.18
10   0x040:                      | done:
11   0x040: 00                   |     halt
12   0x041:                      | proc:
13   0x041: 90                   |     ret                      # Return
14                               |
```

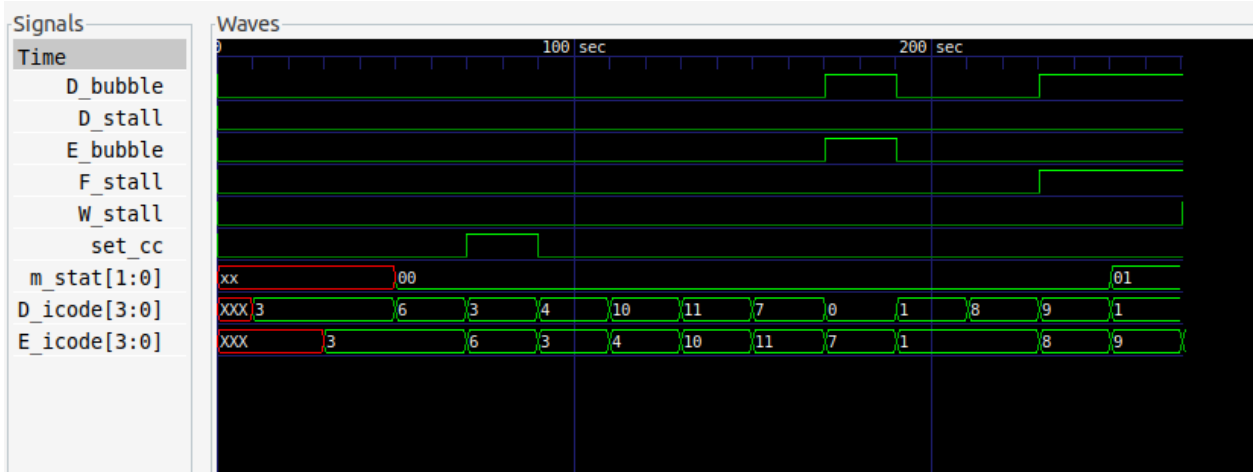
```
            230   clk=1
                  F Reg:          F_predPC = 65
                  fetch:          f_predPC = 65
                  D Reg:          D_icode = 1001 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 65 D_valP = 66 D_stat = 1
                   E Reg:         E_icode = 1000 E_ifun = 0000 E_valA = 64 E_valB = 128 E_valC = 65 E_dstE = 0100 E_dstM = 1111 E_srcA = 15 E_srcB = 4 E_stat = 1
                  execute:        e_cnd = 0 e_valE = 120
                  M Reg:          M_icode = 0001 M_cnd = 0 M_valA = 55 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
                  memory:         m_valM = 9
                  W Reg:          W_icode = 0001 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
                  F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

            240   clk=0
                  F Reg:          F_predPC = 65
                  fetch:          f_predPC = 65
                  D Reg:          D_icode = 1001 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 65 D_valP = 66 D_stat = 1
                   E Reg:         E_icode = 1000 E_ifun = 0000 E_valA = 64 E_valB = 128 E_valC = 65 E_dstE = 0100 E_dstM = 1111 E_srcA = 15 E_srcB = 4 E_stat = 1
                  execute:        e_cnd = 0 e_valE = 120
                  M Reg:          M_icode = 0001 M_cnd = 0 M_valA = 55 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
                  memory:         m_valM = 9
                  W Reg:          W_icode = 0001 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
                  F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

            250   clk=1
                  F Reg:          F_predPC = 65
                  fetch:          f_predPC = 65
                  D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
                   E Reg:         E_icode = 1001 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 65 E_dstE = 0100 E_dstM = 0100 E_srcA = 4 E_srcB = 4 E_stat = 1
                  execute:        e_cnd = 0 e_valE = 128
                  M Reg:          M_icode = 1000 M_cnd = 0 M_valA = 64 M_valE = 120 M_dstE = 0100, M_dstM = 1111 M_stat = 1
                  memory:         m_valM = 9
                  W Reg:          W_icode = 0001 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
                  F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =1 set_cc = 0 m_stat =1 w_stat =0 W_stall =0

            260   clk=0
                  F Reg:          F_predPC = 65
                  fetch:          f_predPC = 65
                  D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
                   E Reg:         E_icode = 1001 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 65 E_dstE = 0100 E_dstM = 0100 E_srcA = 4 E_srcB = 4 E_stat = 1
                  execute:        e_cnd = 0 e_valE = 128
                  M Reg:          M_icode = 1000 M_cnd = 0 M_valA = 64 M_valE = 120 M_dstE = 0100, M_dstM = 1111 M_stat = 1
                  memory:         m_valM = 9
                  W Reg:          W_icode = 0001 W_valE = 128 W_valM = 9 W_dstE = 1111 W_dstM = 1111 W_stat = 0
                  F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =1 set_cc = 0 m_stat =1 w_stat =0 W_stall =0

HALT
            270   clk=1
                  F Reg:          F_predPC = 65
                  fetch:          f_predPC = 65
                  D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
                   E Reg:         E_icode = 0001 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
                  execute:        e_cnd = 0 e_valE = 128
                  M Reg:          M_icode = 1001 M_cnd = 0 M_valA = 120 M_valE = 128 M_dstE = 0100, M_dstM = 0100 M_stat = 1
                  memory:         m_valM = 64
                  W Reg:          W_icode = 1000 W_valE = 120 W_valM = 9 W_dstE = 0100 W_dstM = 1111 W_stat = 1
                  F_stall = 1 D_stall = 0 D_bubble = 1 E_bubble = 0 M_bubble =1 set_cc = 0 m_stat =1 w_stat =1 W_stall =1
```



2) LOAD/USE - MRMOVQ

Program used:

```
0x000: irmovq $128,%rdx
0x00a: irmovq  $3,%rcx
0x014: rmmovq %rcx, 0(%rdx)
0x01e: irmovq $10,%rbx
0x028: mrmovq 0(%rdx),%rax #
     bubble
0x032: addq %rbx,%rax # Use
0x034: halt
```



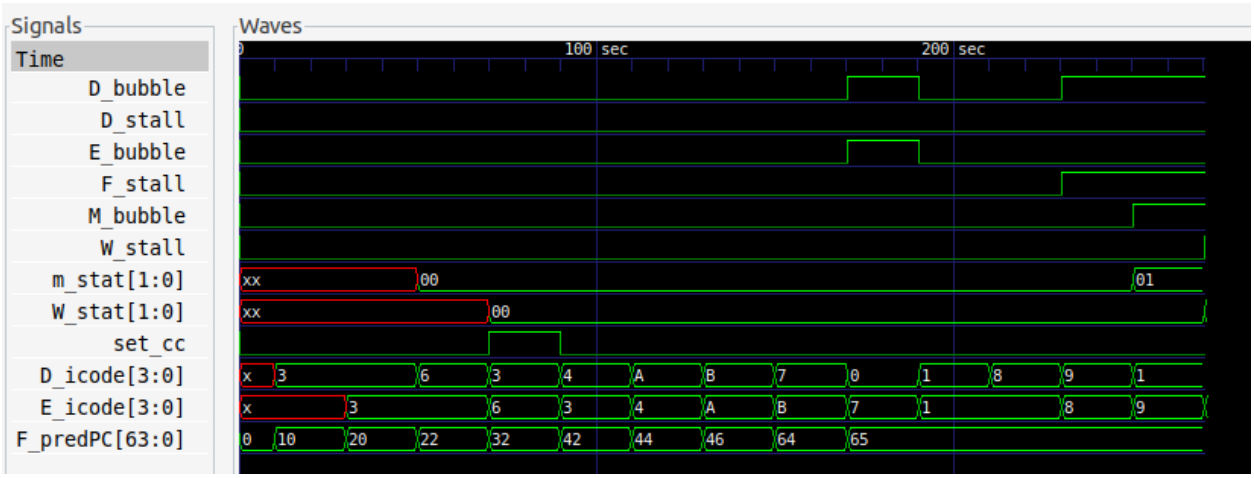
3) MISPREDICTED BRANCH

For the above example taken, jump is not taken yet the predicted PC value always assumes that the jump is taken and hence updates valP considering the same. It's only when jump reaches the execute stage and we find that e_Cnd = 0, the pipeline control logic sets the value of D_bubble and E_bubble as 1 and hence introduces a bubble in the decode and execute stage so that the following instruction can be implemented correctly.

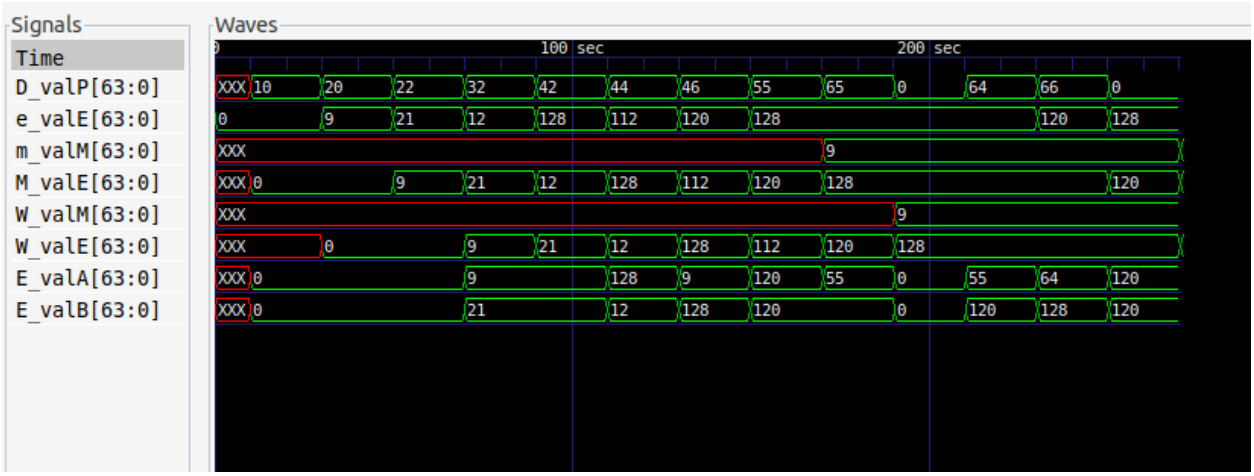
Code used:

```
1   0x000: 30f20900000000000000 |     irmovq $9,  %rdx
2   0x00a: 30f31500000000000000 |     irmovq $21, %rbx
3   0x014: 6123                 |     subq %rdx, %rbx          # subtract
4   0x016: 30f48000000000000000 |     irmovq $128,%rsp         # Problem 4.13
5   0x020: 4043640000000000000 |     rmmovq %rsp, 100(%rbx)   # store
6   0x02a: a02f                 |     pushq %rdx               # push
7   0x02c: b00f                 |     popq  %rax               # Problem 4.14
8   0x02e: 734000000000000000   |     je done                  # Not taken
9   0x037: 804100000000000000   |     call proc                # Problem 4.18
10  0x040:                      | done:
11  0x040: 00                   |     halt
12  0x041:                      | proc:
13  0x041: 90                   |     ret                      # Return
14                              |
```

```
150    clk=1
       F Reg:          F_predPC = 64
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0111 D_ifun = 0011 D_rA = 0000 D_rB = 1111 D_valC = 64 D_valP = 55 D_stat = 0
        E Reg:         E_icode = 1011 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 100 E_dstE = 0100 E_dstM = 0000 E_srcA = 4 E_srcB = 4 E_stat = 0
       execute:        e_cnd = x e_valE = 128
       M Reg:          M_icode = 1010 M_cnd = x M_valA = 9 M_valE = 120 M_dstE = 0100, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0100 W_valE = 112 W_valM = 0 W_dstE = 0011 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

160    clk=0
       F Reg:          F_predPC = 64
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0111 D_ifun = 0011 D_rA = 0000 D_rB = 1111 D_valC = 64 D_valP = 55 D_stat = 0
        E Reg:         E_icode = 1011 E_ifun = 0000 E_valA = 120 E_valB = 120 E_valC = 100 E_dstE = 0100 E_dstM = 0000 E_srcA = 4 E_srcB = 4 E_stat = 0
       execute:        e_cnd = x e_valE = 128
       M Reg:          M_icode = 1010 M_cnd = x M_valA = 9 M_valE = 120 M_dstE = 0100, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 0
       W Reg:          W_icode = 0100 W_valE = 112 W_valM = 0 W_dstE = 0011 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

170    clk=1
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0000 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 64 D_valP = 65 D_stat = 1
        E Reg:         E_icode = 0111 E_ifun = 0011 E_valA = 55 E_valB = 120 E_valC = 64 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 1011 M_cnd = x M_valA = 120 M_valE = 128 M_dstE = 0100, M_dstM = 0000 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 1010 W_valE = 120 W_valM = 0 W_dstE = 0100 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 1 E_bubble = 1 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

180    clk=0
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0000 D_ifun = 0000 D_rA = 0000 D_rB = 1111 D_valC = 64 D_valP = 65 D_stat = 1
        E Reg:         E_icode = 0111 E_ifun = 0011 E_valA = 55 E_valB = 120 E_valC = 64 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 1011 M_cnd = x M_valA = 120 M_valE = 128 M_dstE = 0100, M_dstM = 0000 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 1010 W_valE = 120 W_valM = 0 W_dstE = 0100 W_dstM = 1111 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 1 E_bubble = 1 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0

190    clk=1
       F Reg:          F_predPC = 65
       fetch:          f_predPC = 65
       D Reg:          D_icode = 0001 D_ifun = 0000 D_rA = 1111 D_rB = 1111 D_valC = 0 D_valP = 0 D_stat = 0
        E Reg:         E_icode = 0001 E_ifun = 0000 E_valA = 0 E_valB = 0 E_valC = 0 E_dstE = 1111 E_dstM = 1111 E_srcA = 15 E_srcB = 15 E_stat = 0
       execute:        e_cnd = 0 e_valE = 128
       M Reg:          M_icode = 0111 M_cnd = 0 M_valA = 55 M_valE = 128 M_dstE = 1111, M_dstM = 1111 M_stat = 0
       memory:         m_valM = 9
       W Reg:          W_icode = 1011 W_valE = 128 W_valM = 9 W_dstE = 0100 W_dstM = 0000 W_stat = 0
       F_stall = 0 D_stall = 0 D_bubble = 0 E_bubble = 0 M_bubble =0 set_cc = 0 m_stat =0 w_stat =0 W_stall =0
```



4) DATA FORWARDING



Implementation and Working:

- The forwarding logic is as follows:
- d_valA = e_valE if d_srcA = e_dstE (Forward valE from execute)
- d_valA = m_valM if d_srcA = M_dstM (Forward valM from memory)
- d_valA = M_valE if d_srcA = M_dstE (Forward valE from memory)
- d_valA = W_valE if d_srcA = W_dstE (Forward valM from write back)
- d_valA = W_valE if d_srcA = W_dstE (Forward valE from write back)
- At positive edge all the values are stored into the pipelined register variables.

CHALLENGES FACED

1) Understanding the processor implementation using pipeline control logic took us a good amount of time.
2) Dealing with minor bugs in the code was quite challenging as they were very difficult to spot.
3) Initially, after combining all the blocks, the program was not showing any output.
   We later resolved this by combining each block one by one and checking for outputs.
4) It was hence discovered that combining with pipeline control logic was causing issues, as the value of D_stall, F_stall, etc was not being initialized to zero, and hence there was no output.

It was later resolved by putting $display() debugging statements in the fetch module which finally helped us solve the major issues with our pipeline control logic.

5) Next issue involved fixing the ret instruction. After putting several debugging statements and closely observing the outputs, we realized how the issue was with the way F_predPC was updated and fixed that.
6) Another issue was regarding the Load-Use Hazard. The value of m_valM was not getting updated at the right time and hence there was an issue with forwarding. This was fixed by updating the value of m_valM correctly in the memory module.

ACKNOWLEDGEMENT

Engaging in this project has been stimulating and has offered us valuable learning opportunities. Throughout the past month, we've gained significant insights into Processor Architecture. We would like to thank our Prof. Deepak Gangadharan and the TAs for their invaluable support in seeing this project through to completion.