Intro To Processor Architecture

Project Report

Team 11: Pegasus

Tejah SS 2020112028

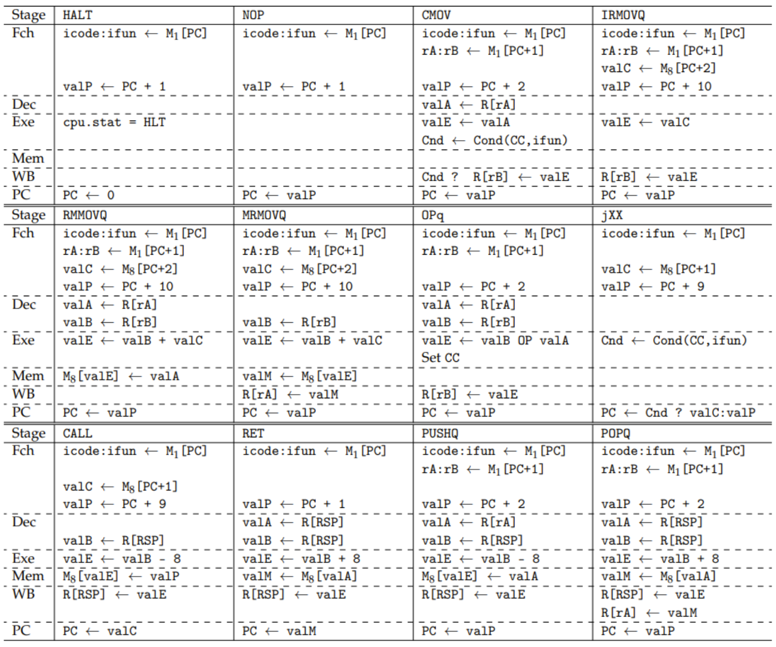
Jaishnav Yarramaneni 2020102059

Project Description:

The project involves the implementation of Y86-64 processor architecture design using Verilog. The final goal is to achieve a 5 state pipelined implementation of the processor.
This report includes the sequential and the pipelined implementation of the Y86-64 processor which contains the fetch, decode - write back , execute, memory and the PC update blocks, their testbenches and the combined testbench along with data forwarding and support for eliminating pipeline hazards along with the testing for call and return.

Sequential Implementation:

The sequential implementation of the Y86-64 processor works with fetch, decode, execute, memory, writeback and PC update according to the following table



Stage	HALT	NOP	CMOV	IRMOVQ

Fch	icode:ifun ← M_1[PC]
	
	valP ← PC + 1
		icode:ifun ← M_1[PC]
		
		valP ← PC + 1
			icode:ifun ← M_1[PC]
			rA:rB ← M_1[PC+1]
			
			valP ← PC + 2
				icode:ifun ← M_1[PC]
				rA:rB ← M_1[PC+1]
				valC ← M_8[PC+2]
				valP ← PC + 10

Dec			valA ← R[rA]	

Exe	cpu.stat = HLT		valE ← valA
			Cnd ← Cond(CC,ifun)
				valE ← valC

Mem				
WB			Cnd ? R[rB] ← valE	R[rB] ← valE
PC	PC ← 0	PC ← valP	PC ← valP	PC ← valP
Stage	RMMOVQ	MRMOVQ	OPq	jXX

Fch	icode:ifun ← M_1[PC]
	rA:rB ← M_1[PC+1]
	valC ← M_8[PC+2]
	valP ← PC + 10
		icode:ifun ← M_1[PC]
		rA:rB ← M_1[PC+1]
		valC ← M_8[PC+2]
		valP ← PC + 10
			icode:ifun ← M_1[PC]
			rA:rB ← M_1[PC+1]
			
			valP ← PC + 2
				icode:ifun ← M_1[PC]
				
				valC ← M_8[PC+1]
				valP ← PC + 9

Dec	valA ← R[rA]
	valB ← R[rB]
		
		valB ← R[rB]
			valA ← R[rA]
			valB ← R[rB]	

Exe	valE ← valB + valC	valE ← valB + valC	valE ← valB OP valA Set CC	Cnd ← Cond(CC,ifun)
Mem	M_8[valE] ← valA	valM ← M_8[valE]		
WB		R[rA] ← valM	R[rB] ← valE	
PC	PC ← valP	PC ← valP	PC ← valP	PC ← Cnd ? valC:valP
Stage	CALL	RET	PUSHQ	POPQ

Fch	icode:ifun ← M_1[PC]
	
	valC ← M_8[PC+1]
	valP ← PC + 9
		icode:ifun ← M_1[PC]
		
		valP ← PC + 1
			icode:ifun ← M_1[PC]
			rA:rB ← M_1[PC+1]
			
			valP ← PC + 2
				icode:ifun ← M_1[PC]
				rA:rB ← M_1[PC+1]
				
				valP ← PC + 2

Dec	
	valB ← R[RSP]
		valA ← R[RSP]
		valB ← R[RSP]
			valA ← R[rA]
			valB ← R[RSP]
				valA ← R[RSP]
				valB ← R[RSP]

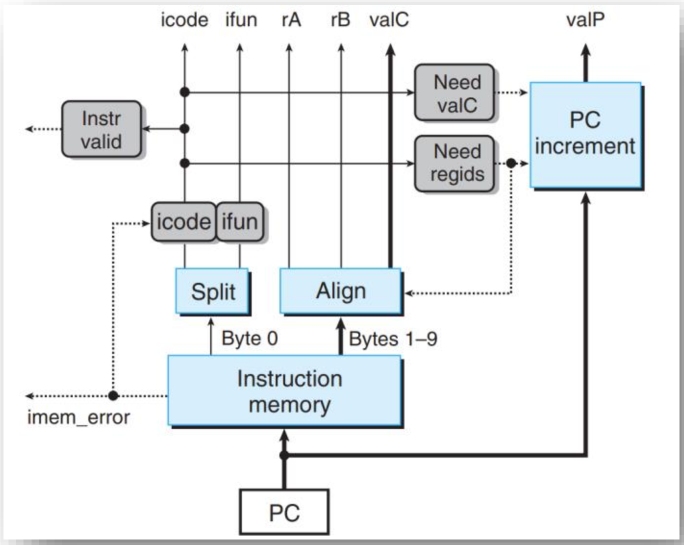
Exe	valE ← valB - 8	valE ← valB + 8	valE ← valB - 8	valE ← valB + 8
Mem	M_8[valE] ← valP	valM ← M_8[valA]	M_8[valE] ← valA	valM ← M_8[valA]

WB	R[RSP] ← valE	R[RSP] ← valE	R[RSP] ← valE	R[RSP] ← valE
				R[rA] ← valM

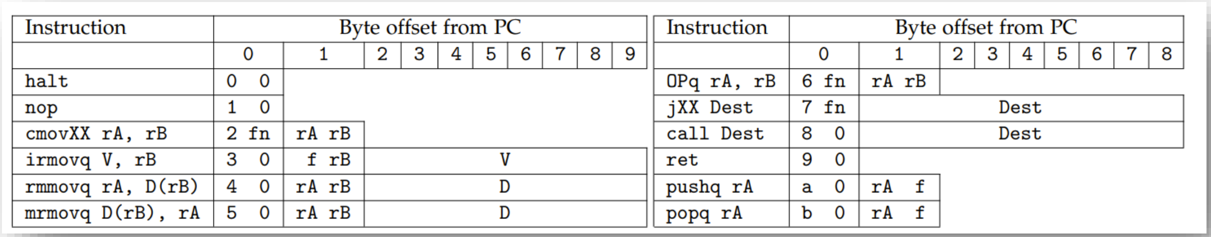
PC	PC ← valC	PC ← valM	PC ← valP	PC ← valP

The blocks are as follows:

Fetch:



The purpose of fetch block is to take PC as an input and compute the **icode**, **ifun**, **rA**, **rB**, **valC**, **valP**, **Instr valid** and the **imem_error** as the outputs. It computes this with the help of **instr** array which stores the first 10 bytes after the byte the PC is pointing to of which the first byte represents the **icode**:**ifun** , second represents the **values of registers** and rest the immediate value of destination offset depending on icode and ifun according to the following

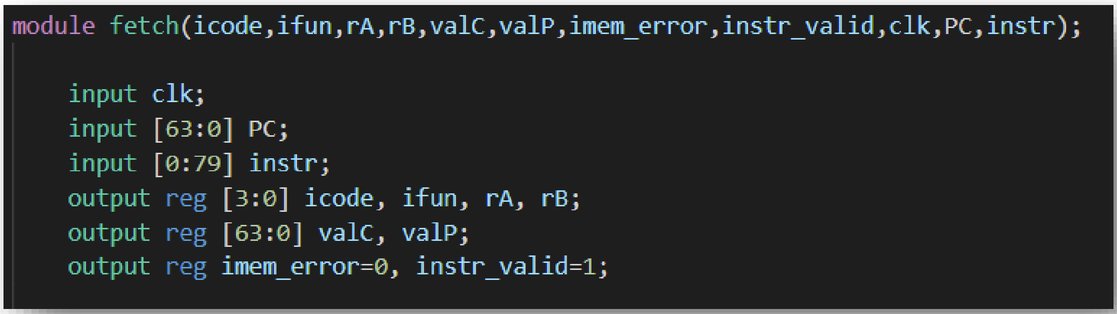


Instruction	Byte offset from PC									
	0	1	2	3	4	5	6	7	8	9
`halt`	0 0									
`nop`	1 0									
`cmovXX rA, rB`	2 fn	rA rB								
`irmovq V, rB`	3 0	f rB	V							
`rmmovq rA, D(rB)`	4 0	rA rB	D							
`mrmovq D(rB), rA`	5 0	rA rB	D							

Instruction	Byte offset from PC								
	0	1	2	3	4	5	6	7	8
`OPq rA, rB`	6 fn	rA rB							
`jXX Dest`	7 fn	Dest							
`call Dest`	8 0	Dest							
`ret`	9 0								
`pushq rA`	a 0	rA f							
`popq rA`	b 0	rA f							

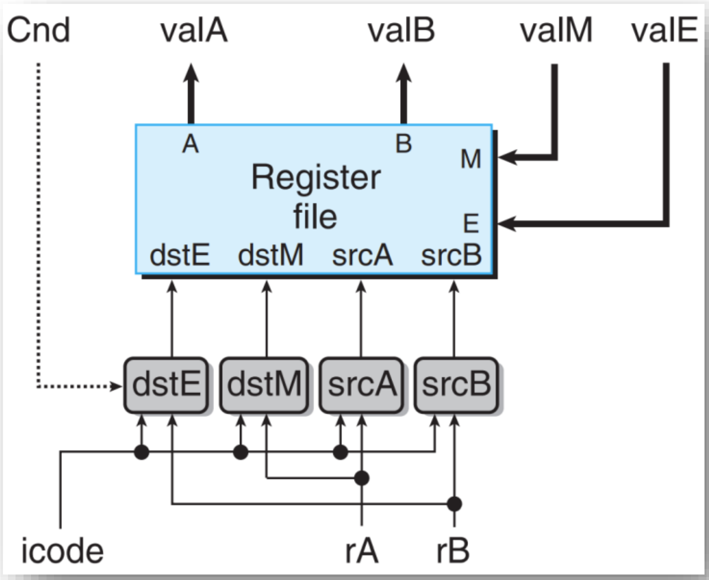
The fetch module which is defined takes the inputs as **PC** and **instr** array from which the values **icode**, **ifun**, **rA**, **rB**, **valC**, **valP**, **Instr valid** and the **imem_error** are given as outputs.

- The instructions are identified with the help of case statements which produce the required outputs for **icode**, **ifun**, **rA**, **rB**, **valC** and **valP**.
- The imem_error is set to 1 if the value of PC exceeds 20480 bytes(memory size) and sets the status code **ADR** to 1.
- If the instruction is not one from the above 12 instructions, then the instr valid is set to 0 and the status code **INS** is set to 1.
- If halt is encountered, then the status code **HLT** is set to 1.



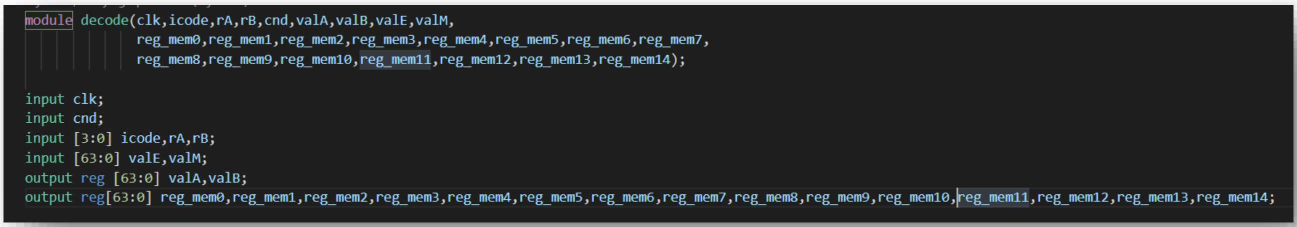
```
module fetch(icode,ifun,rA,rB,valC,valP,imem_error,instr_valid,clk,PC,instr);

    input clk;
    input [63:0] PC;
    input [0:79] instr;
    output reg [3:0] icode, ifun, rA, rB;
    output reg [63:0] valC, valP;
    output reg imem_error=0, instr_valid=1;
```

Decode and Writeback:



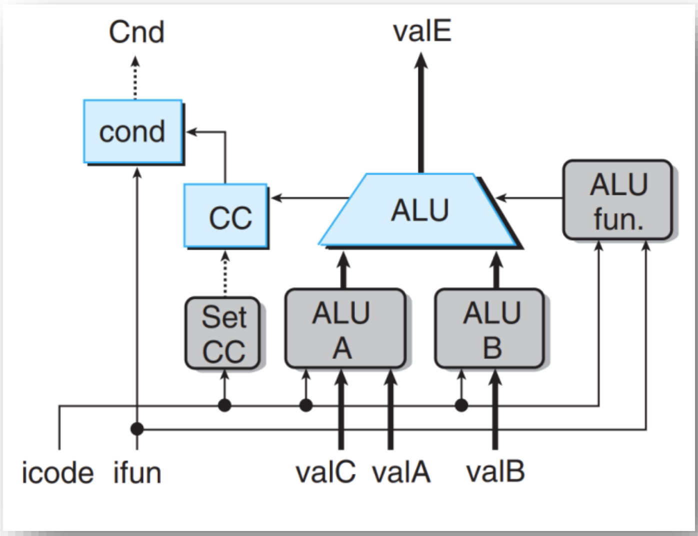
The purpose of the decode block is to read from the registers and assign the values of valA and valB and that of the write back block is to write the values back into the registers which may depend on the value of Cnd.
For this we need to have the 15 registers that are available in the processor which will only be accessed in the decode and writeback and hence they are declared here which will be present as long as the processor is running. The registers are sent as output for the testbench for the verification purpose.
The combined block of decode and write back takes **rA**, **rB**, **icode**, **Cnd**, **valM** and **valE** as inputs and give **valA** and **valB** as outputs.

- For the decode part, **valA** and **valB** are computed according to the identified instructions using switch statements.
- For the writeback part, the values are written back into the register using the same switch statement which identify the instructions and writeback into the respective registers.



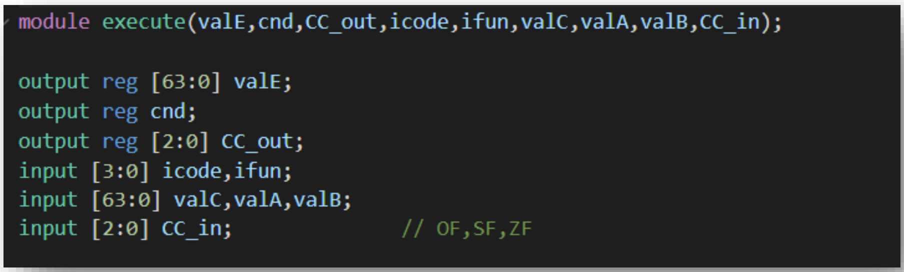
```
module decode(clk,icode,rA,rB,cnd,valA,valB,valE,valM,
              reg_mem0,reg_mem1,reg_mem2,reg_mem3,reg_mem4,reg_mem5,reg_mem6,reg_mem7,
              reg_mem8,reg_mem9,reg_mem10,reg_mem11,reg_mem12,reg_mem13,reg_mem14);

input clk;
input cnd;
input [3:0] icode,rA,rB;
input [63:0] valE,valM;
output reg [63:0] valA,valB;
output reg[63:0] reg_mem0,reg_mem1,reg_mem2,reg_mem3,reg_mem4,reg_mem5,reg_mem6,reg_mem7,reg_mem8,reg_mem9,reg_mem10,reg_mem11,reg_mem12,reg_mem13,reg_mem14;
```

Execute:



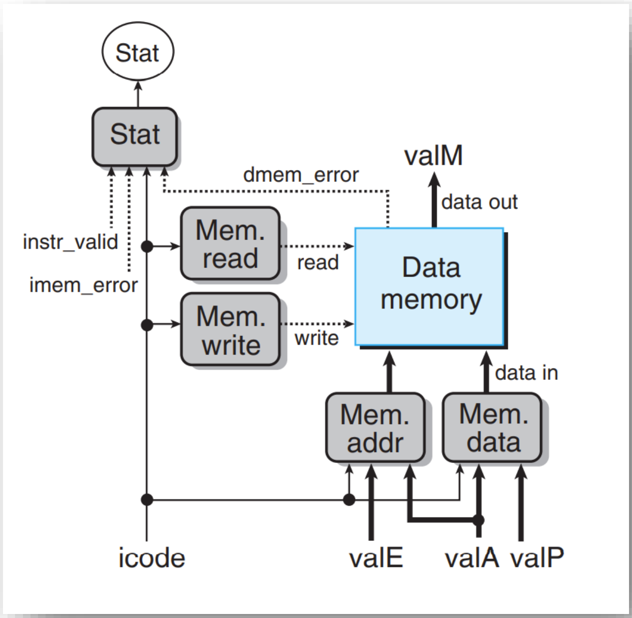
The purpose of the execute block is to implement the instruction and compute the effective address or value using the ALU and set the required condition codes(by using the prebuilt blocks that are present).
The execute block takes **icode**, **ifun**, **valC**, **valA** and **valB** as inputs and gives the outputs as **Cnd** and **valE**.

- The execute part works on switch statements which selects the instructions according to the values of **icode** and **ifun**.
- The condition codes are also set for the instructions jXX and CMOV and **valE** is computed using ALU.
- The presence of CC_in and CC_out is due to the inability of Verilog to change the input value so the value of CC_in is declared to that of CC_out for the arithmetic operations i.e. when the condition codes are generated



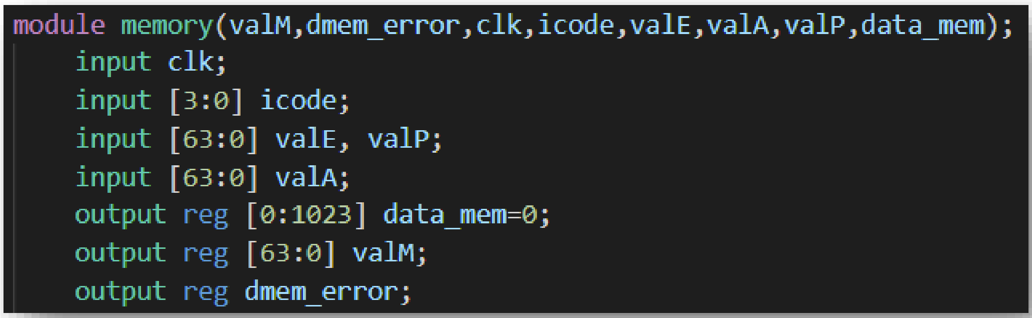
```
module execute(valE,cnd,CC_out,icode,ifun,valC,valA,valB,CC_in);

output reg [63:0] valE;
output reg cnd;
output reg [2:0] CC_out;
input [3:0] icode,ifun;
input [63:0] valC,valA,valB;
input [2:0] CC_in;             // OF,SF,ZF
```

Memory:

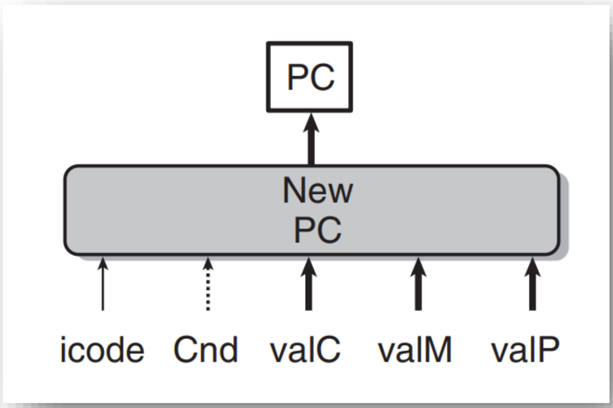


The purpose of memory block is to read and write the values from the memory. The data_mem in the processor in general is accessible everywhere but since here we set it as output for the verification purpose in testbench.
The memory block has input values as **icode**, **valE**, **valA** and **valP** and the output values as **valM**

- The **dmem_error** is set to 1 when valE exceeds 1023 which is the size of the data memory which sets the status code **ADR** to 1.
- The value of **valM** is read from the registers and the vice versa according to the value of **icode** where the instruction is selected using switch statements.

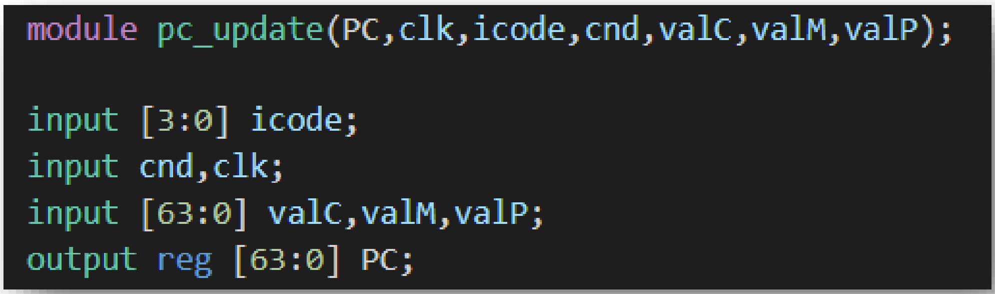

```
module memory(valM,dmem_error,clk,icode,valE,valA,valP,data_mem);
    input clk;
    input [3:0] icode;
    input [63:0] valE, valP;
    input [63:0] valA;
    output reg [0:1023] data_mem=0;
    output reg [63:0] valM;
    output reg dmem_error;
```

PC Update:



The purpose of PC update is to point the PC to the address of the next instruction.
The PC Update block has **icode**, **Cnd**, **valC**, **valM** and **valP** as inputs and the value of updated **PC** as output.

- The PC Update part works on switch statements which selects the instructions according to the values of **icode** and **Cnd** and updates the value of PC accordingly.

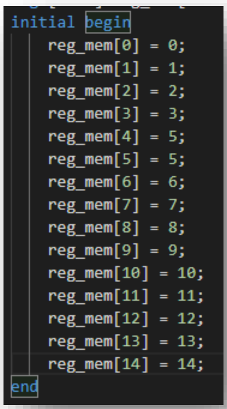

```
module pc_update(PC,clk,icode,cnd,valC,valM,valP);

input [3:0] icode;
input cnd,clk;
input [63:0] valC,valM,valP;
output reg [63:0] PC;
```

Testbench Results:

The output for the given machine level code

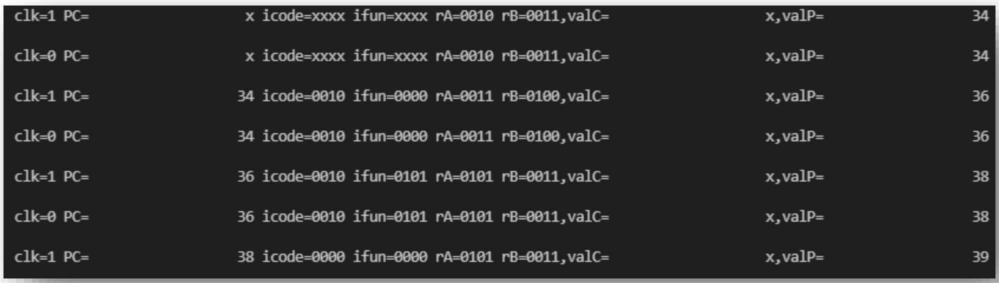
```
//OPq
instr_mem[32]=8'b01100001; //6 fn
instr_mem[33]=8'b00100011; //rA rB
//cmovxx
instr_mem[34]=8'b00100000; //2 fn
instr_mem[35]=8'b00110100; //rA rB
instr_mem[36]=8'b00100101; // 2 ge
instr_mem[37]=8'b01010011; // rA rB
//halt
instr_mem[38]=8'b00000000; // 0 0
```

for the following reg_mem



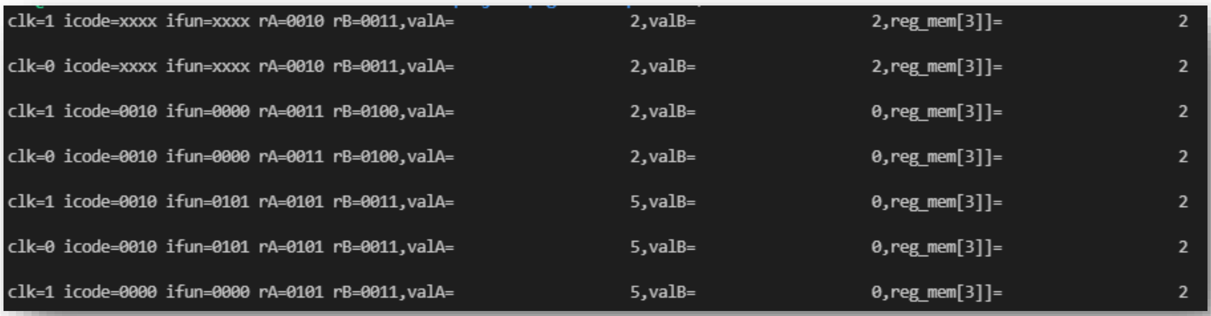
```
initial begin
    reg_mem[0] = 0;
    reg_mem[1] = 1;
    reg_mem[2] = 2;
    reg_mem[3] = 3;
    reg_mem[4] = 5;
    reg_mem[5] = 5;
    reg_mem[6] = 6;
    reg_mem[7] = 7;
    reg_mem[8] = 8;
    reg_mem[9] = 9;
    reg_mem[10] = 10;
    reg_mem[11] = 11;
    reg_mem[12] = 12;
    reg_mem[13] = 13;
    reg_mem[14] = 14;
end
```

for fetch block:



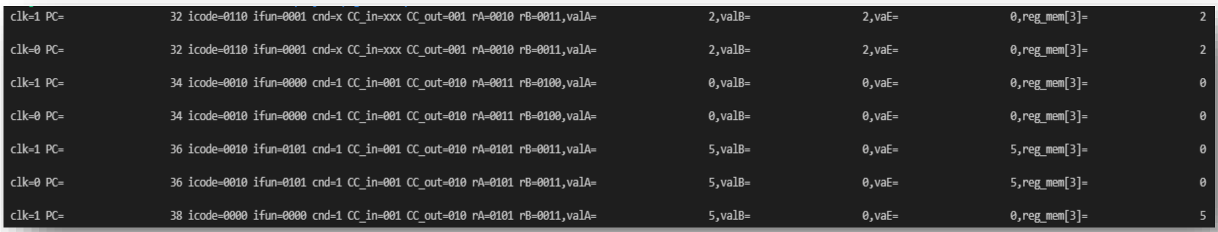
```
clk=1 PC=                    x icode=xxxx ifun=xxxx rA=0010 rB=0011,valC=                    x,valP=                    34
clk=0 PC=                    x icode=xxxx ifun=xxxx rA=0010 rB=0011,valC=                    x,valP=                    34
clk=1 PC=                   34 icode=0010 ifun=0000 rA=0011 rB=0100,valC=                    x,valP=                    36
clk=0 PC=                   34 icode=0010 ifun=0000 rA=0011 rB=0100,valC=                    x,valP=                    36
clk=1 PC=                   36 icode=0010 ifun=0101 rA=0101 rB=0011,valC=                    x,valP=                    38
clk=0 PC=                   36 icode=0010 ifun=0101 rA=0101 rB=0011,valC=                    x,valP=                    38
clk=1 PC=                   38 icode=0000 ifun=0000 rA=0101 rB=0011,valC=                    x,valP=                    39
```

for decode block:



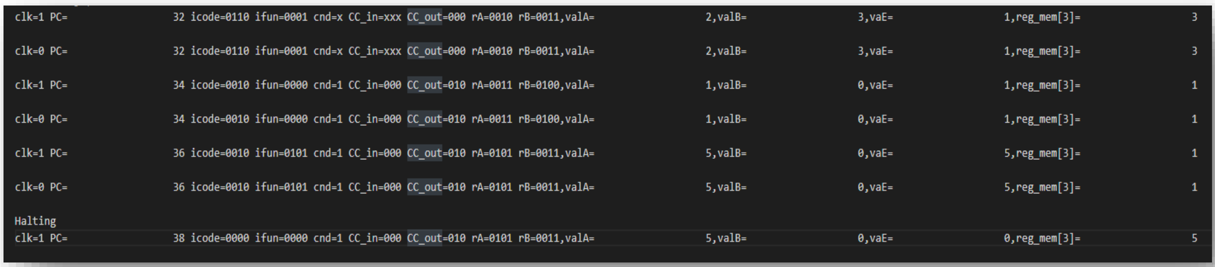
```
clk=1 icode=xxxx ifun=xxxx rA=0010 rB=0011,valA=                  2,valB=                  2,reg_mem[3]]=                  2

clk=0 icode=xxxx ifun=xxxx rA=0010 rB=0011,valA=                  2,valB=                  2,reg_mem[3]]=                  2

clk=1 icode=0010 ifun=0000 rA=0011 rB=0100,valA=                  2,valB=                  0,reg_mem[3]]=                  2

clk=0 icode=0010 ifun=0000 rA=0011 rB=0100,valA=                  2,valB=                  0,reg_mem[3]]=                  2

clk=1 icode=0010 ifun=0101 rA=0101 rB=0011,valA=                  5,valB=                  0,reg_mem[3]]=                  2

clk=0 icode=0010 ifun=0101 rA=0101 rB=0011,valA=                  5,valB=                  0,reg_mem[3]]=                  2

clk=1 icode=0000 ifun=0000 rA=0101 rB=0011,valA=                  5,valB=                  0,reg_mem[3]]=                  2
```

for execute, writeback and PC update block:



```
clk=1 PC=                  32 icode=0110 ifun=0001 cnd=x CC_in=xxx CC_out=001 rA=0010 rB=0011,valA=                  2,valB=                  2,vaE=                  0,reg_mem[3]=                  2

clk=0 PC=                  32 icode=0110 ifun=0001 cnd=x CC_in=xxx CC_out=001 rA=0010 rB=0011,valA=                  2,valB=                  2,vaE=                  0,reg_mem[3]=                  2

clk=1 PC=                  34 icode=0010 ifun=0000 cnd=1 CC_in=001 CC_out=010 rA=0011 rB=0100,valA=                  0,valB=                  0,vaE=                  0,reg_mem[3]=                  0

clk=0 PC=                  34 icode=0010 ifun=0000 cnd=1 CC_in=001 CC_out=010 rA=0011 rB=0100,valA=                  0,valB=                  0,vaE=                  0,reg_mem[3]=                  0

clk=1 PC=                  36 icode=0010 ifun=0101 cnd=1 CC_in=001 CC_out=010 rA=0101 rB=0011,valA=                  5,valB=                  0,vaE=                  5,reg_mem[3]=                  0

clk=0 PC=                  36 icode=0010 ifun=0101 cnd=1 CC_in=001 CC_out=010 rA=0101 rB=0011,valA=                  5,valB=                  0,vaE=                  5,reg_mem[3]=                  0

clk=1 PC=                  38 icode=0000 ifun=0000 cnd=1 CC_in=001 CC_out=010 rA=0101 rB=0011,valA=                  5,valB=                  0,vaE=                  0,reg_mem[3]=                  5
```

for integrated processor:

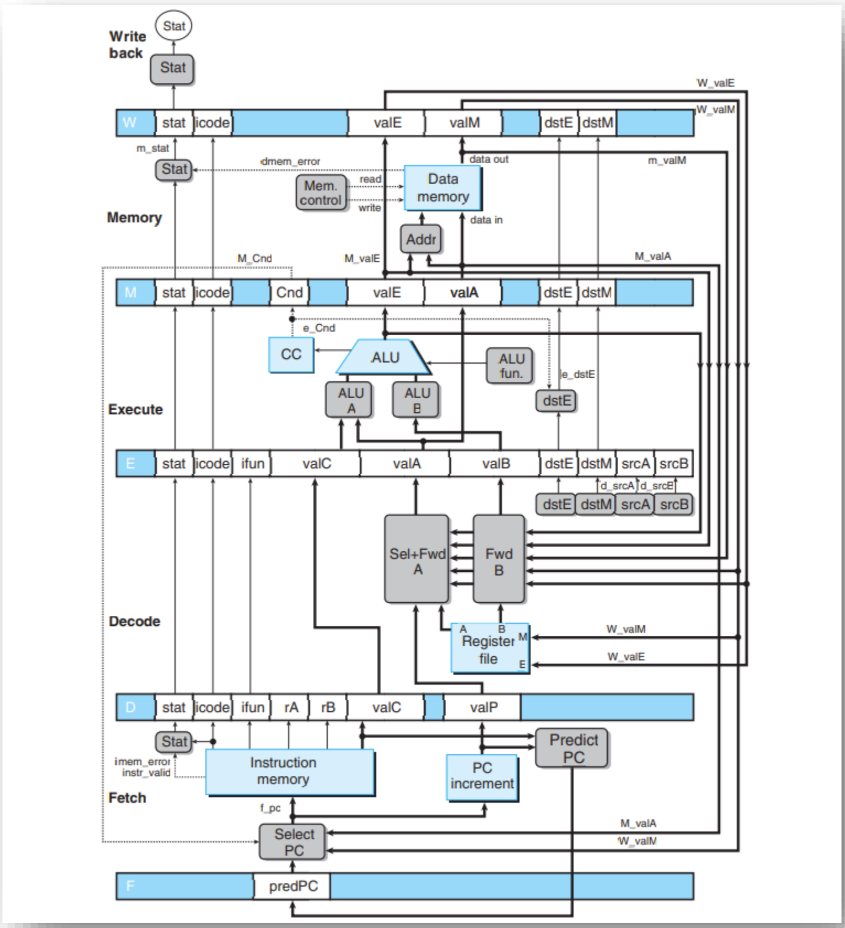


```
clk=1 PC=                  32 icode=0110 ifun=0001 cnd=x CC_in=xxx CC_out=000 rA=0010 rB=0011,valA=                  2,valB=                  3,vaE=                  1,reg_mem[3]=                  3

clk=0 PC=                  32 icode=0110 ifun=0001 cnd=x CC_in=xxx CC_out=000 rA=0010 rB=0011,valA=                  2,valB=                  3,vaE=                  1,reg_mem[3]=                  3

clk=1 PC=                  34 icode=0010 ifun=0000 cnd=1 CC_in=000 CC_out=010 rA=0011 rB=0100,valA=                  1,valB=                  0,vaE=                  1,reg_mem[3]=                  1

clk=0 PC=                  34 icode=0010 ifun=0000 cnd=1 CC_in=000 CC_out=010 rA=0011 rB=0100,valA=                  1,valB=                  0,vaE=                  1,reg_mem[3]=                  1

clk=1 PC=                  36 icode=0010 ifun=0101 cnd=1 CC_in=000 CC_out=010 rA=0101 rB=0011,valA=                  5,valB=                  0,vaE=                  5,reg_mem[3]=                  1

clk=0 PC=                  36 icode=0010 ifun=0101 cnd=1 CC_in=000 CC_out=010 rA=0101 rB=0011,valA=                  5,valB=                  0,vaE=                  5,reg_mem[3]=                  1

Halting
clk=1 PC=                  38 icode=0000 ifun=0000 cnd=1 CC_in=000 CC_out=010 rA=0101 rB=0011,valA=                  5,valB=                  0,vaE=                  0,reg_mem[3]=                  5
```

This is shown due to the constraint of space, but the code works for all the present instructions. In order to test for all the instructions, uncomment the machine level code present in processor.v and run the code.

Pipelined Implementation:

The pipelined implementation of the Y86-64 works the same way as that of the sequential implementation with the modules being same but with inclusion of the pipelined registers, slight change in the fetch and decode blocks, addition of support for data forwarding and PC prediction for improving the performance and the addition of the pipeline control logic for eliminating pipeline hazards.



We have implemented the block and the immediate register in the same block which is as shown below.

1)The fetch register is implemented in the processor.v(combined testbench).

2) fetch.v holds the fetch block, predict PC logic and the decode register.

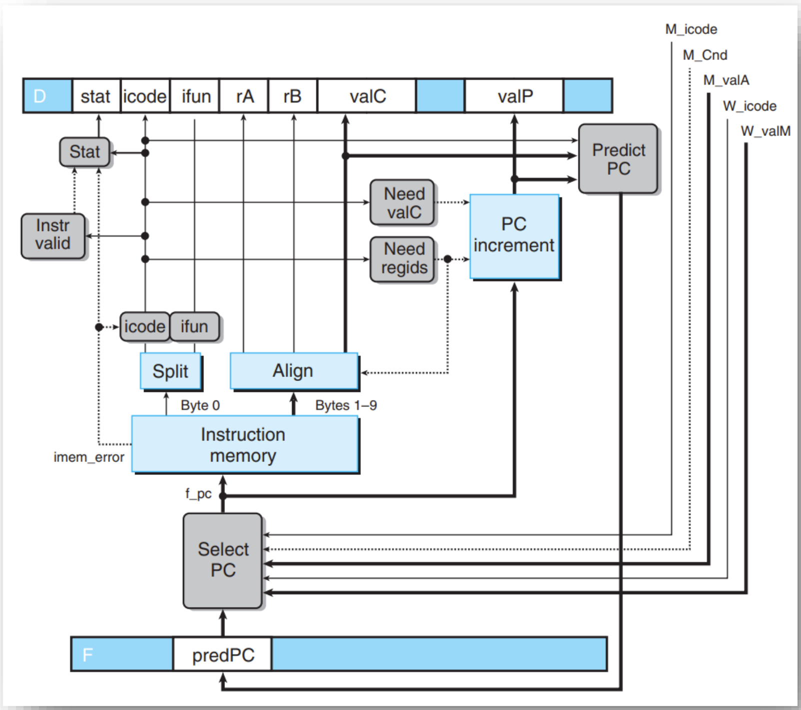
3) decode.v holds the decode block along with the execute register along with writeback.

4) execute.v holds the execute block with the memory register.

5) memory.v holds the memory block with the writeback register.

The blocks are as follows:

Fetch:

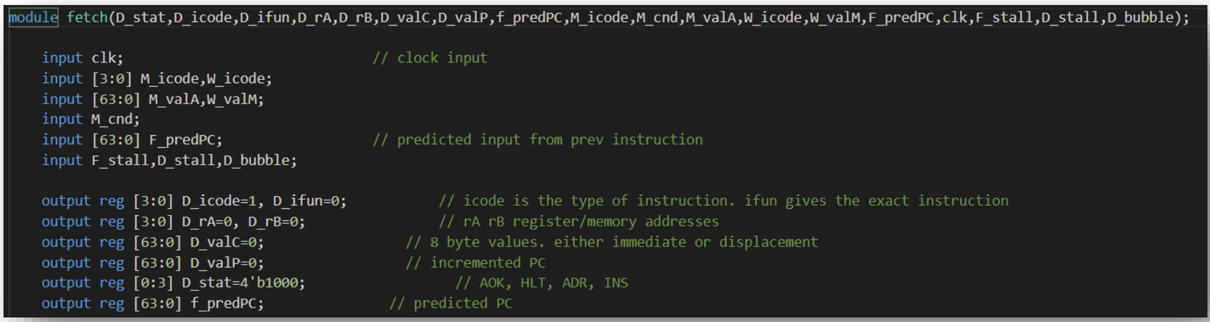


Here the **f_pc** is initialized to the initial value of PC from the processor.v block.

This takes the input as the **PC** and compute the values of **stat**, **icode**, **ifun**, **rA**, **rB**, **valC** and **valP**, and since they will be sent into the decode register, they will be named as **D_icode**, **D_ifun**, **D_rA**, **D_rB**, **D_valC** , **D_stat** and **D_valP**.

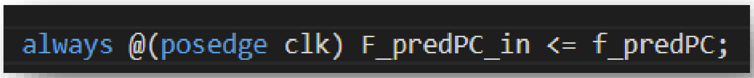
Here the fetch part functions the same as that of that of sequential, but the addition of sequential block increases the throughput of the processor.

The addition of the Predict PC block adds the functionality of the predicting PC which will be sent to the fetch register in the processor.v block.



```
module fetch(D_stat,D_icode,D_ifun,D_rA,D_rB,D_valC,D_valP,f_predPC,M_icode,M_cnd,M_valA,W_icode,W_valM,F_predPC,clk,F_stall,D_stall,D_bubble);

    input clk;                          // clock input
    input [3:0] M_icode,W_icode;
    input [63:0] M_valA,W_valM;
    input M_cnd;
    input [63:0] F_predPC;              // predicted input from prev instruction
    input F_stall,D_stall,D_bubble;

    output reg [3:0] D_icode=1, D_ifun=0;          // icode is the type of instruction. ifun gives the exact instruction
    output reg [3:0] D_rA=0, D_rB=0;               // rA rB register/memory addresses
    output reg [63:0] D_valC=0;                // 8 byte values. either immediate or displacement
    output reg [63:0] D_valP=0;                // incremented PC
    output reg [0:3] D_stat=4'b1000;                 // AOK, HLT, ADR, INS
    output reg [63:0] f_predPC;              // predicted PC
```

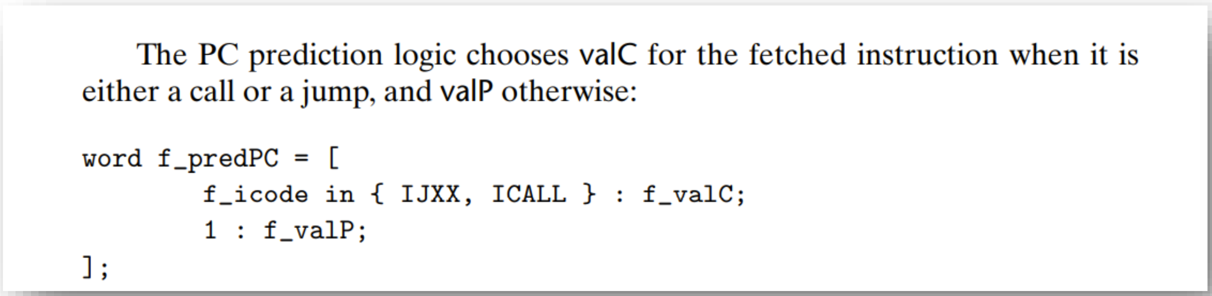
The **f_pc** is named as **f_predPC** which gets updated from the processor.v file on every clock cycle as shown below.



```
always @(posedge clk) F_predPC_in <= f_predPC;
```

The input for this block are the **clk** signal and **M_icode**, **M_cnd**, **M_valA**, **W_icode**, **W_valM** which are present for the computation of the next predicted PC i.e. **f_predPC** and **F_predPC_in** which is used to compute the values of the **D_icode**, **D_ifun**, **D_rA**, **D_rB**, **D_valC**, **D_stat** and **D_valP**.

The computation of **D_icode**, **D_ifun**, **D_rA**, **D_rB**, **D_valC**, **D_stat** and **D_valP** remains the same as that of the sequential part but the computation of **f_predPC** can be done with the help of the following

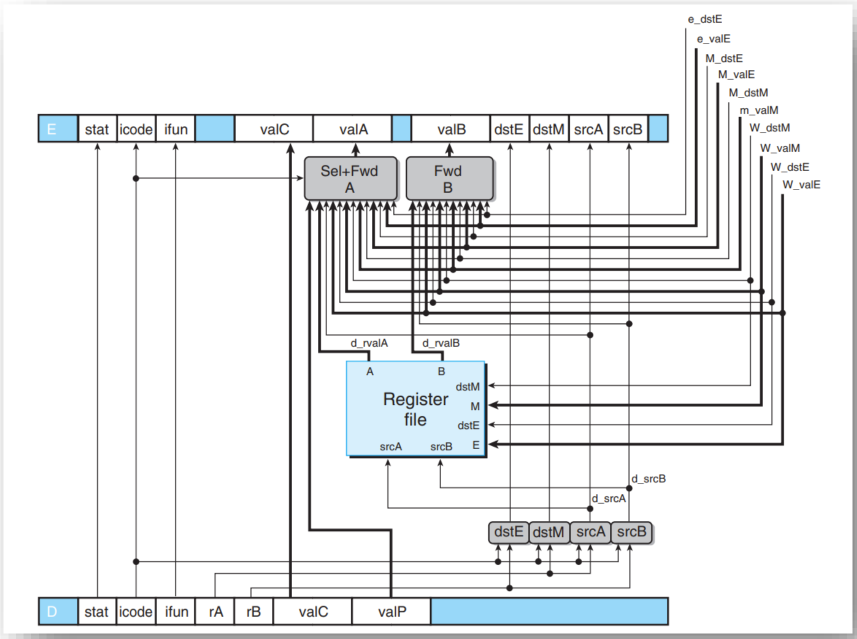


The PC prediction logic chooses valC for the fetched instruction when it is either a call or a jump, and valP otherwise:

```
word f_predPC = [
        f_icode in { IJXX, ICALL } : f_valC;
        1 : f_valP;
];
```

The register gets updated once the clk hits and the output for **D_icode**, **D_ifun**, **D_rA**, **D_rB**, **D_valC**, **D_stat** and **D_valP** is available as an output of the register.

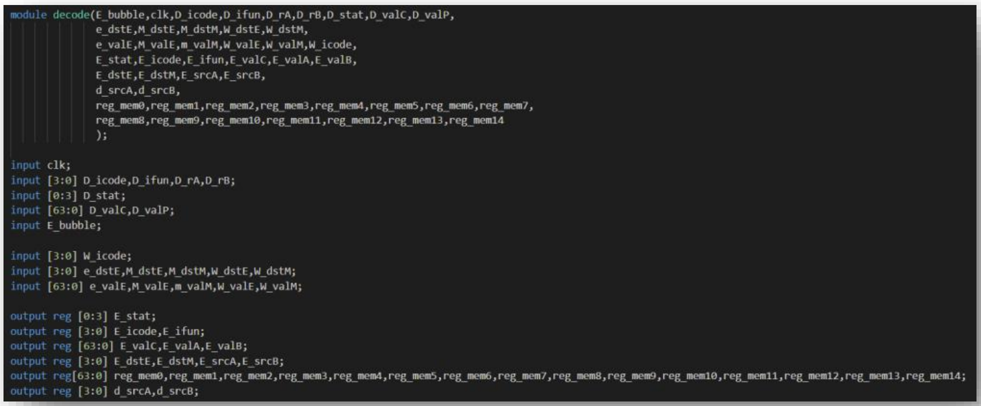
Decode and Writeback:



This takes the inputs from the decode register which are **D_icode**, **D_ifun**, **D_rA**, **D_rB**, **D_valC**, **D_stat**, **D_valP** and also the inputs required for data forwarding which include **e_dstE**, **e_valE**, **M_dstE**, **M_valE**, **M_dstM**, **m_valM**, **W_dstM**, **W_valM**, **W_dstE** and **W_valE**.

This has two additional blocks along with the blocks present earlier. These blocks help in data forwarding and are represented by the **Sel+Fwd A** and **Fwd B** which directly gives the value for valA or valB from the execute, memory or writeback stages.

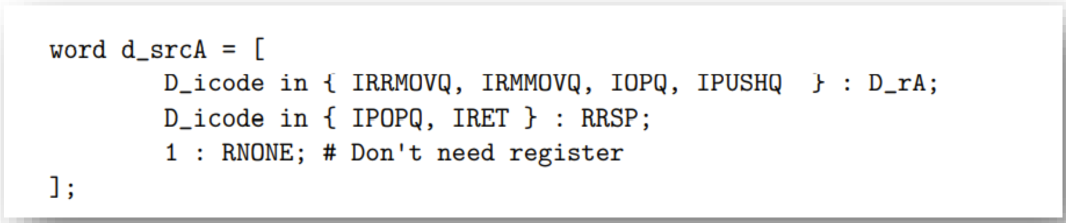
The outputs obtained from this block include **E_stat**, **E_ifun**, **E_icode**, **E_valA**, **E_valB**, **E_valC**, **dstE**, **dstM**, **srcA** and **srcB**.



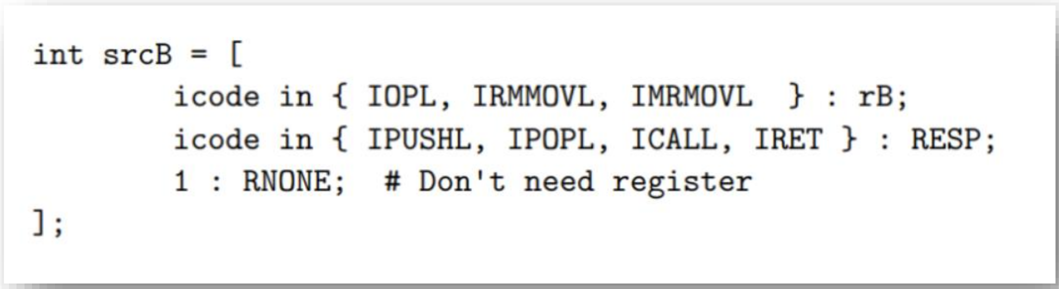
```
module decode(E_bubble,clk,D_icode,D_ifun,D_rA,D_rB,D_stat,D_valC,D_valP,
              e_dstE,M_dstE,M_dstM,W_dstE,W_dstM,
              e_valE,M_valE,m_valM,W_valE,W_valM,W_icode,
              E_stat,E_icode,E_ifun,E_valC,E_valA,E_valB,
              E_dstE,E_dstM,E_srcA,E_srcB,
              d_srcA,d_srcB,
              reg_mem0,reg_mem1,reg_mem2,reg_mem3,reg_mem4,reg_mem5,reg_mem6,reg_mem7,
              reg_mem8,reg_mem9,reg_mem10,reg_mem11,reg_mem12,reg_mem13,reg_mem14
              );

input clk;
input [3:0] D_icode,D_ifun,D_rA,D_rB;
input [0:3] D_stat;
input [63:0] D_valC,D_valP;
input E_bubble;

input [3:0] W_icode;
input [3:0] e_dstE,M_dstE,M_dstM,W_dstE,W_dstM;
input [63:0] e_valE,M_valE,m_valM,W_valE,W_valM;

output reg [0:3] E_stat;
output reg [3:0] E_icode,E_ifun;
output reg [63:0] E_valC,E_valA,E_valB;
output reg [3:0] E_dstE,E_dstM,E_srcA,E_srcB;
output reg[63:0] reg_mem0,reg_mem1,reg_mem2,reg_mem3,reg_mem4,reg_mem5,reg_mem6,reg_mem7,reg_mem8,reg_mem9,reg_mem10,reg_mem11,reg_mem12,reg_mem13,reg_mem14;
output reg [3:0] d_srcA,d_srcB;
```

Here the inputs **D_icode**, **D_ifun**, **D_rA**, **D_rB**, **D_valC**, **D_stat**, **D_valP** are taken as shown which function similar to that of the sequential computing **E_stat**, **E_ifun**, **E_icode**, **E_valA**, **E_valB** and **E_valC** and all the 14 registers which gets updated from the writeback part.

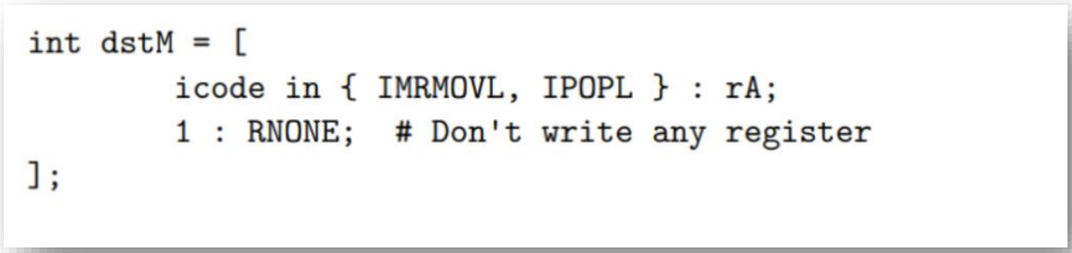
d_srcA, **d_srcB**, **d_dstE** and **d_dstM** are computed based on the following



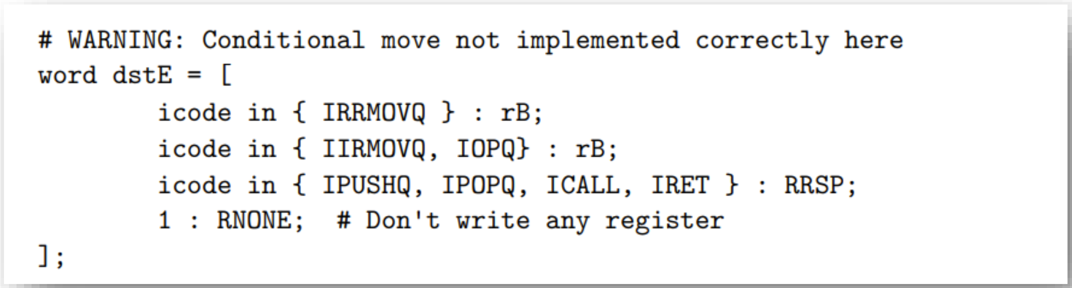
```
word d_srcA = [
        D_icode in { IRRMOVQ, IRMMOVQ, IOPQ, IPUSHQ  } : D_rA;
        D_icode in { IPOPQ, IRET } : RRSP;
        1 : RNONE; # Don't need register
];
```



```
int srcB = [
        icode in { IOPL, IRMMOVL, IMRMOVL  } : rB;
        icode in { IPUSHL, IPOPL, ICALL, IRET } : RESP;
        1 : RNONE;  # Don't need register
];
```

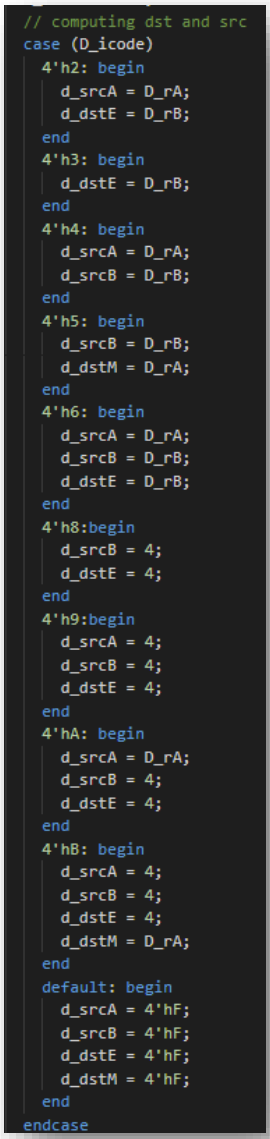


```
int dstM = [
        icode in { IMRMOVL, IPOPL } : rA;
        1 : RNONE;  # Don't write any register
];
```



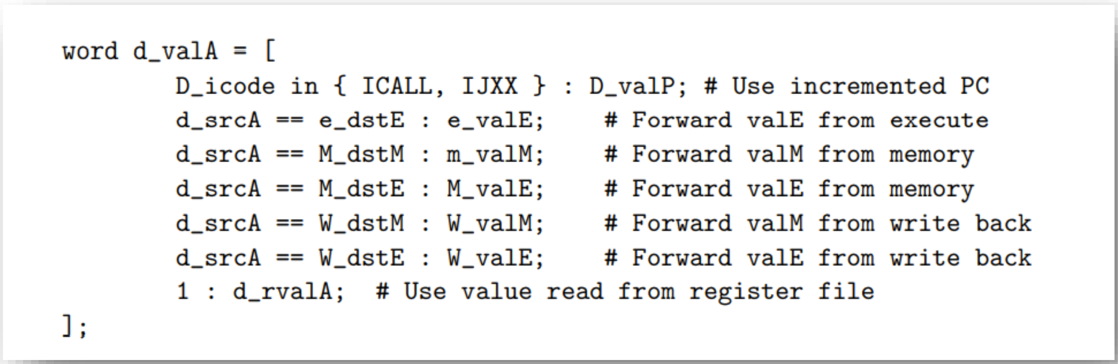
```
# WARNING: Conditional move not implemented correctly here
word dstE = [
        icode in { IRRMOVQ } : rB;
        icode in { IIRMOVQ, IOPQ} : rB;
        icode in { IPUSHQ, IPOPQ, ICALL, IRET } : RRSP;
        1 : RNONE;  # Don't write any register
];
```

And for the incomplete parts above, it is implemented as shown

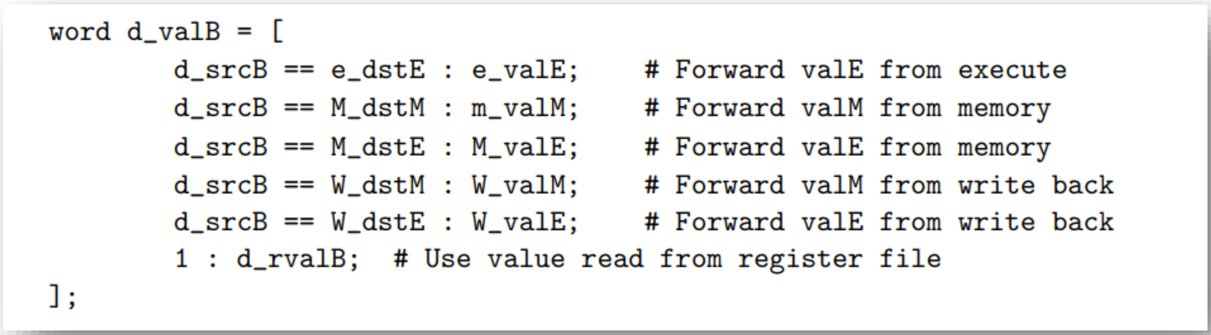


```
// computing dst and src
case (D_icode)
  4'h2: begin
    d_srcA = D_rA;
    d_dstE = D_rB;
  end
  4'h3: begin
    d_dstE = D_rB;
  end
  4'h4: begin
    d_srcA = D_rA;
    d_srcB = D_rB;
  end
  4'h5: begin
    d_srcB = D_rB;
    d_dstM = D_rA;
  end
  4'h6: begin
    d_srcA = D_rA;
    d_srcB = D_rB;
    d_dstE = D_rB;
  end
  4'h8:begin
    d_srcB = 4;
    d_dstE = 4;
  end
  4'h9:begin
    d_srcA = 4;
    d_srcB = 4;
    d_dstE = 4;
  end
  4'hA: begin
    d_srcA = D_rA;
    d_srcB = 4;
    d_dstE = 4;
  end
  4'hB: begin
    d_srcA = 4;
    d_srcB = 4;
    d_dstE = 4;
    d_dstM = D_rA;
  end
  default: begin
    d_srcA = 4'hF;
    d_srcB = 4'hF;
    d_dstE = 4'hF;
    d_dstM = 4'hF;
  end
endcase
```

For the data forwarding part the following is implemented

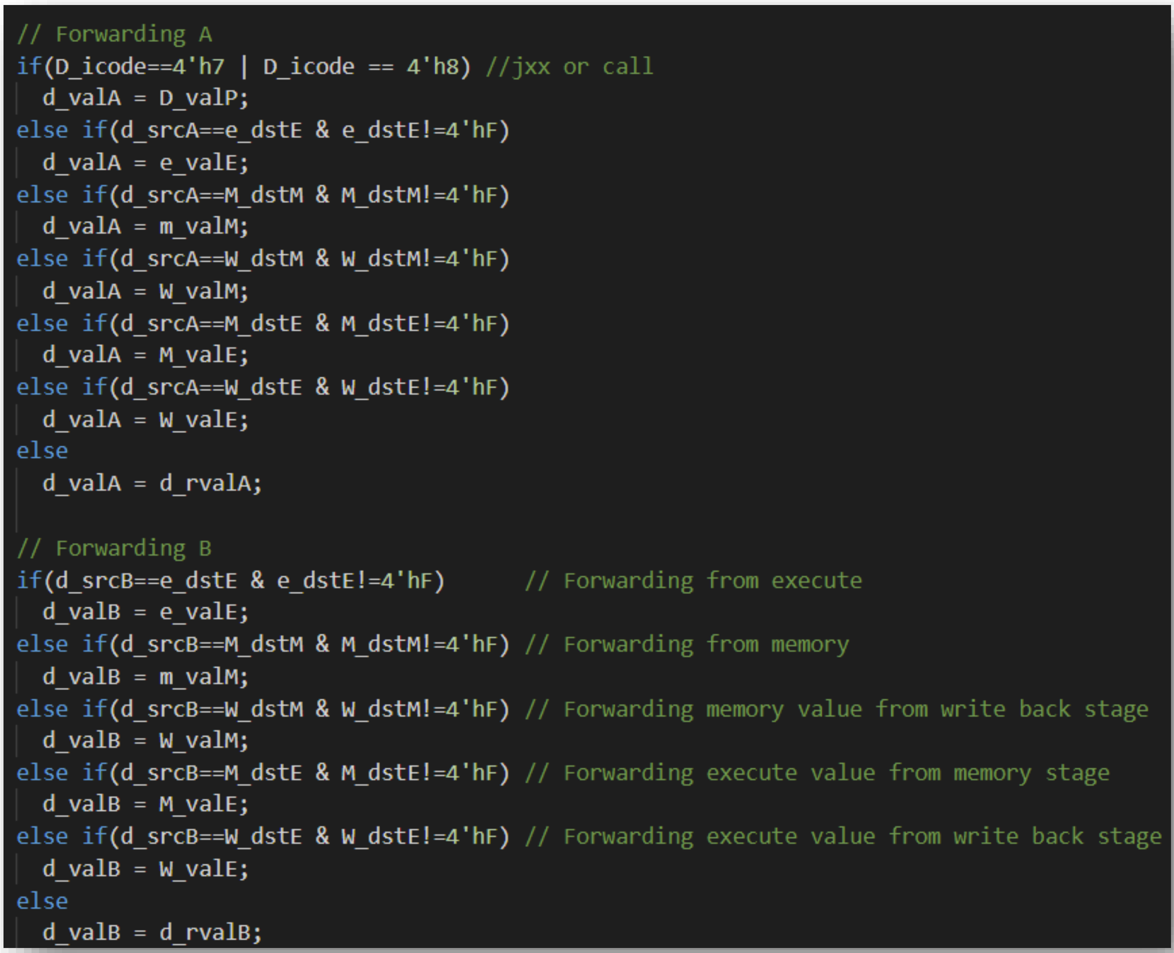


```
word d_valA = [
        D_icode in { ICALL, IJXX } : D_valP; # Use incremented PC
        d_srcA == e_dstE : e_valE;    # Forward valE from execute
        d_srcA == M_dstM : m_valM;    # Forward valM from memory
        d_srcA == M_dstE : M_valE;    # Forward valE from memory
        d_srcA == W_dstM : W_valM;    # Forward valM from write back
        d_srcA == W_dstE : W_valE;    # Forward valE from write back
        1 : d_rvalA;  # Use value read from register file
];
```



```
word d_valB = [
        d_srcB == e_dstE : e_valE;    # Forward valE from execute
        d_srcB == M_dstM : m_valM;    # Forward valM from memory
        d_srcB == M_dstE : M_valE;    # Forward valE from memory
        d_srcB == W_dstM : W_valM;    # Forward valM from write back
        d_srcB == W_dstE : W_valE;    # Forward valE from write back
        1 : d_rvalB;  # Use value read from register file
];
```

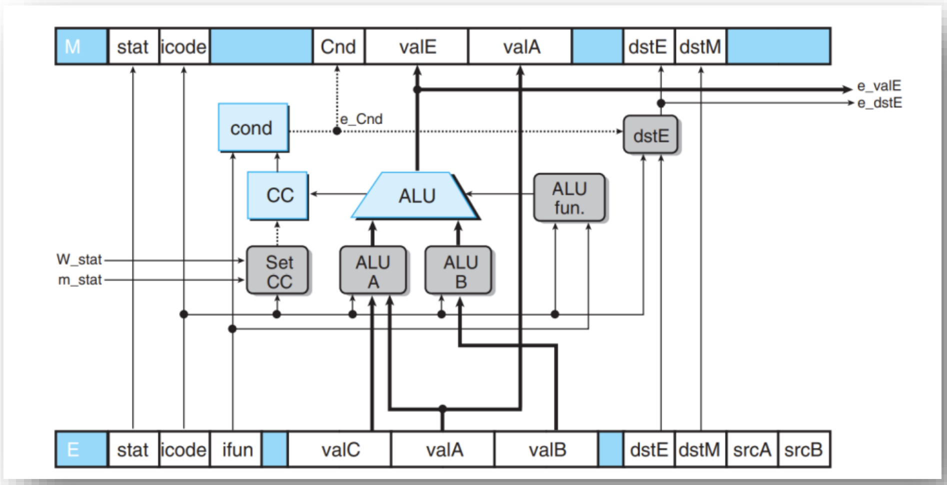
The above is implemented as follows



```
// Forwarding A
if(D_icode==4'h7 | D_icode == 4'h8) //jxx or call
  d_valA = D_valP;
else if(d_srcA==e_dstE & e_dstE!=4'hF)
  d_valA = e_valE;
else if(d_srcA==M_dstM & M_dstM!=4'hF)
  d_valA = m_valM;
else if(d_srcA==W_dstM & W_dstM!=4'hF)
  d_valA = W_valM;
else if(d_srcA==M_dstE & M_dstE!=4'hF)
  d_valA = M_valE;
else if(d_srcA==W_dstE & W_dstE!=4'hF)
  d_valA = W_valE;
else
  d_valA = d_rvalA;

// Forwarding B
if(d_srcB==e_dstE & e_dstE!=4'hF)      // Forwarding from execute
  d_valB = e_valE;
else if(d_srcB==M_dstM & M_dstM!=4'hF) // Forwarding from memory
  d_valB = m_valM;
else if(d_srcB==W_dstM & W_dstM!=4'hF) // Forwarding memory value from write back stage
  d_valB = W_valM;
else if(d_srcB==M_dstE & M_dstE!=4'hF) // Forwarding execute value from memory stage
  d_valB = M_valE;
else if(d_srcB==W_dstE & W_dstE!=4'hF) // Forwarding execute value from write back stage
  d_valB = W_valE;
else
  d_valB = d_rvalB;
```

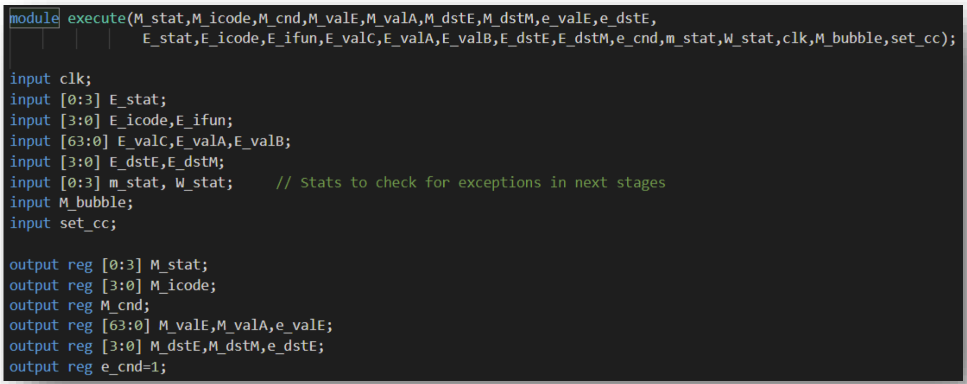
The register gets updated once the clk hits and the output for **E_stat**, **E_ifun**, **E_icode**, **E_valA**, **E_valB** and **E_valC** is available as an output of the execute register.

Execute:



This takes the inputs as the outputs from the execute pipelined register which include **E_stat**, **E_ifun**, **E_icode**, **E_valA**, **E_valB**, **E_valC**, **E_dstE**, **E_dstM**, **W_stat** and **m_stat.**

The outputs obtained from this block include **M_stat**, **M_icode**, **Cnd**, **M_valE**, **M_valA**, **M_dstE**, **M_dstM**, **e_valE** and **e_dstE**.



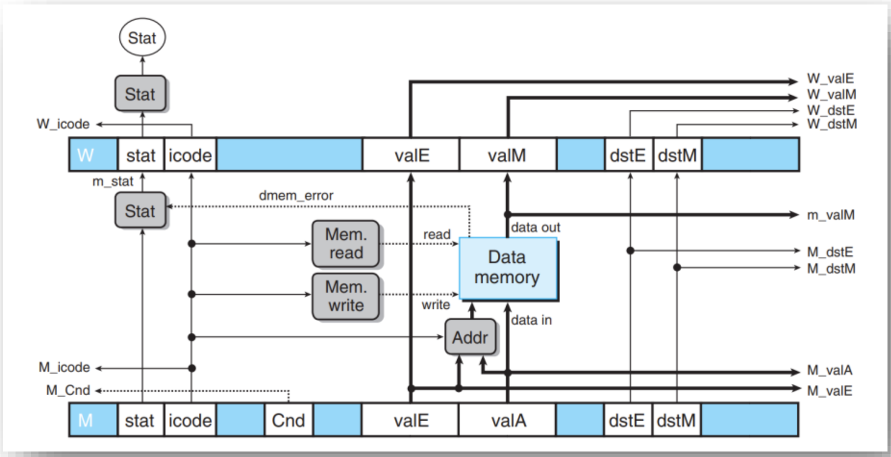
```
module execute(M_stat,M_icode,M_cnd,M_valE,M_valA,M_dstE,M_dstM,e_valE,e_dstE,
                E_stat,E_icode,E_ifun,E_valC,E_valA,E_valB,E_dstE,E_dstM,e_cnd,m_stat,W_stat,clk,M_bubble,set_cc);

input clk;
input [0:3] E_stat;
input [3:0] E_icode,E_ifun;
input [63:0] E_valC,E_valA,E_valB;
input [3:0] E_dstE,E_dstM;
input [0:3] m_stat, W_stat;     // Stats to check for exceptions in next stages
input M_bubble;
input set_cc;

output reg [0:3] M_stat;
output reg [3:0] M_icode;
output reg M_cnd;
output reg [63:0] M_valE,M_valA,e_valE;
output reg [3:0] M_dstE,M_dstM,e_dstE;
output reg e_cnd=1;
```

Here **E_stat**, **E_ifun**, **E_icode**, **E_valA**, **E_valB**, **E_valC**, **E_dstE** and **E_dstM** which are the inputs when passed through this block compute **M_stat**, **M_icode**, **Cnd**, **M_valE**, **M_valA**, **M_dstE**, **M_dstM** and **e_valE** as outputs in similar way to that of sequential.

The value of **e_dstE** is computed based on the **e_**Cnd which will make it either E_dstE or an empty register. **W_stat** and **m_stat** takes care to not change the conditional codes when an instruction such as halt occurs.

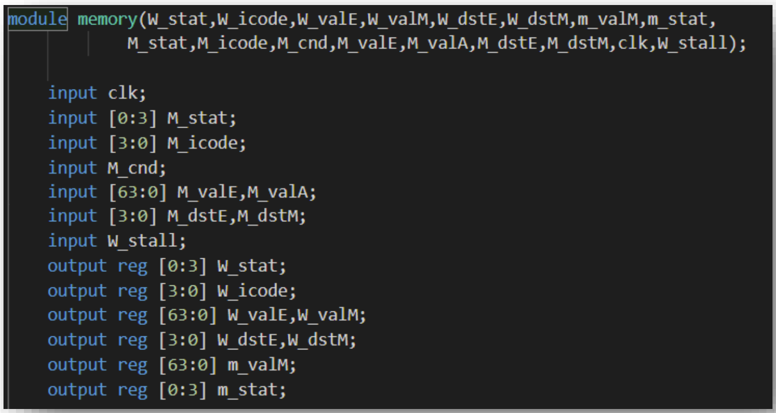
The register gets updated once the clk hits and the output for **M_stat**, **M_icode**, **M_valA**, **M_valE**, **M_dstE** and **M_dstM** is available as an output of the memory register.

Memory:



This takes the inputs as the outputs from the memory pipelined register which include **M_stat**, **M_icode**, **M_Cnd**, **M_valE, M_valA, M_dstE and M_dstM**

The outputs obtained from this block include **W_stat, W_icode, W_valE, W_valM, W_dstE, W_dstM and m_valM**.



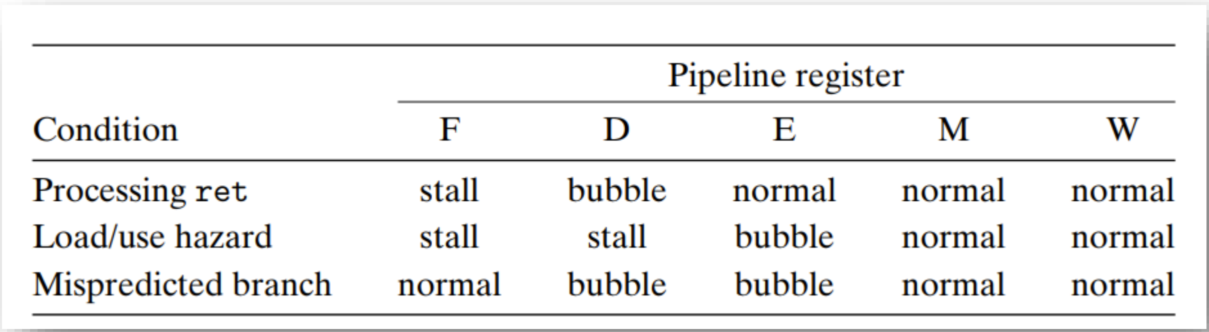
```
module memory(W_stat,W_icode,W_valE,W_valM,W_dstE,W_dstM,m_valM,m_stat,
              M_stat,M_icode,M_cnd,M_valE,M_valA,M_dstE,M_dstM,clk,W_stall);

    input clk;
    input [0:3] M_stat;
    input [3:0] M_icode;
    input M_cnd;
    input [63:0] M_valE,M_valA;
    input [3:0] M_dstE,M_dstM;
    input W_stall;
    output reg [0:3] W_stat;
    output reg [3:0] W_icode;
    output reg [63:0] W_valE,W_valM;
    output reg [3:0] W_dstE,W_dstM;
    output reg [63:0] m_valM;
    output reg [0:3] m_stat;
```

This block functions the same way as that of the sequential memory block which takes **M_icode**, **M_Cnd**, **M_valE, M_valA, M_dstE and M_dstM** as inputs and gives **W_icode, W_valE, W_valM, W_dstE, W_dstM and m_valM** as outputs.

This also updates the **m_stat** to **M_stat** if dmem_error has not occurred.

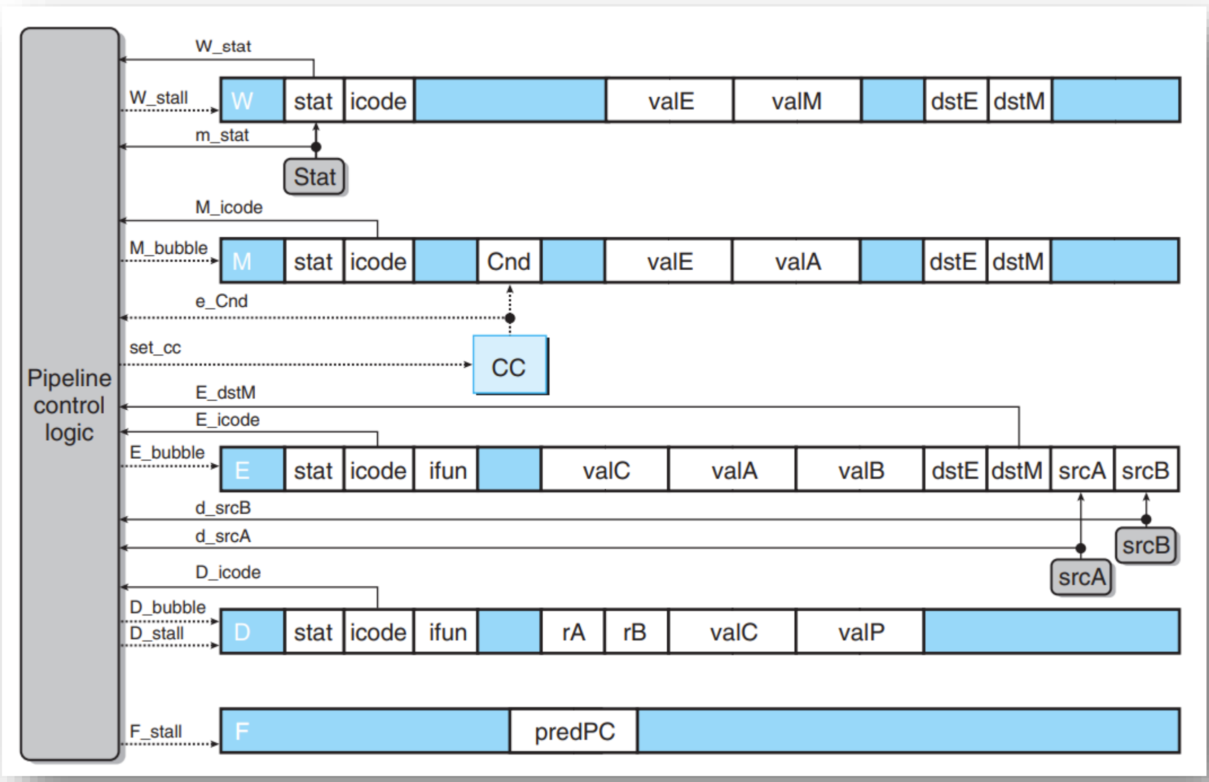
The register gets updated once the clk hits and the output for **W_stat**, **W_icode**, **W_valE**, **W_valM**, **W_dstE** and **W_dstM** is available as an output of the memory register.

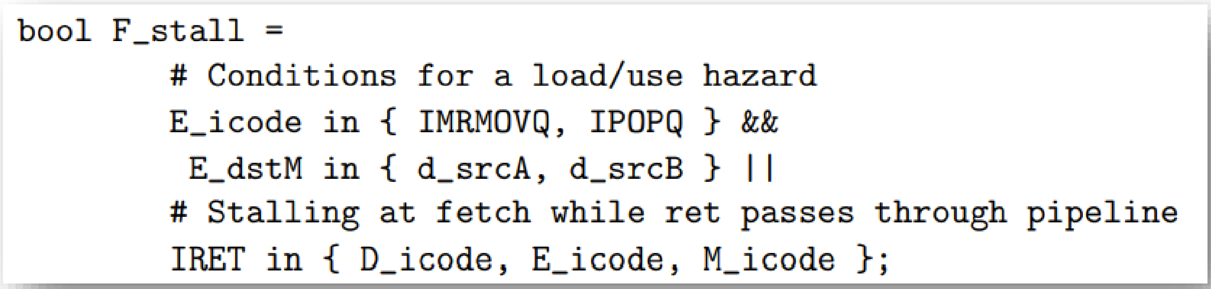
Note: As we can see, the functions used are modified for also eliminating the data and control hazards which requires a pipeline control logic which is implemented in a separate module which works according to the following logic



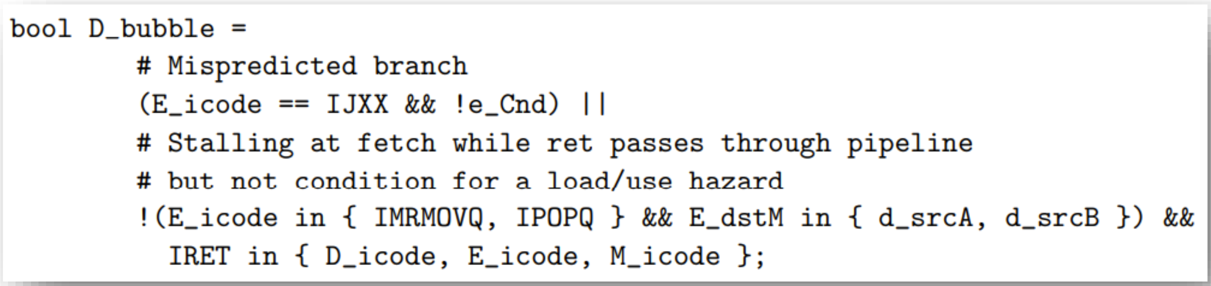
	Pipeline register				
Condition	F	D	E	M	W
Processing `ret`	stall	bubble	normal	normal	normal
Load/use hazard	stall	stall	bubble	normal	normal
Mispredicted branch	normal	bubble	bubble	normal	normal

This when implemented along with the registers will be as the following

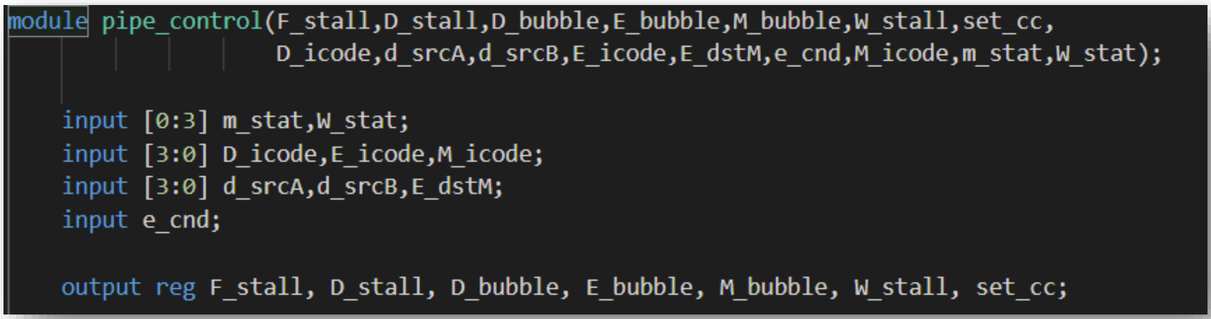




```
bool F_stall =
        # Conditions for a load/use hazard
        E_icode in { IMRMOVQ, IPOPQ } &&
         E_dstM in { d_srcA, d_srcB } ||
        # Stalling at fetch while ret passes through pipeline
        IRET in { D_icode, E_icode, M_icode };
```



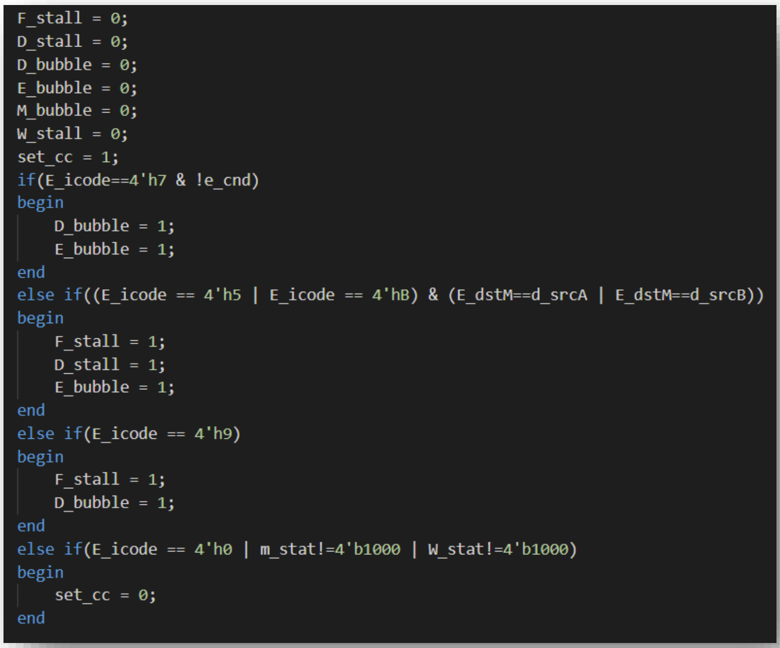
```
bool D_bubble =
        # Mispredicted branch
        (E_icode == IJXX && !e_Cnd) ||
        # Stalling at fetch while ret passes through pipeline
        # but not condition for a load/use hazard
        !(E_icode in { IMRMOVQ, IPOPQ } && E_dstM in { d_srcA, d_srcB }) &&
          IRET in { D_icode, E_icode, M_icode };
```



```
module pipe_control(F_stall,D_stall,D_bubble,E_bubble,M_bubble,W_stall,set_cc,
                    D_icode,d_srcA,d_srcB,E_icode,E_dstM,e_cnd,M_icode,m_stat,W_stat);

    input [0:3] m_stat,W_stat;
    input [3:0] D_icode,E_icode,M_icode;
    input [3:0] d_srcA,d_srcB,E_dstM;
    input e_cnd;

    output reg F_stall, D_stall, D_bubble, E_bubble, M_bubble, W_stall, set_cc;
```

This takes the inputs as **m_stat**, **W_stat**, **D_icode**, **E_icode**, **M_icode**, **d_srcA**, **d_srcB**, **d_dstM** and **e_cnd** and the outputs obtained are as **F_stall**, **D_stall**, **D_bubble**, **E_bubble**, **M_bubble**, **W_stall** and **set_cc**.

These work according to the logic mentioned above and is implemented as follows.



```
F_stall = 0;
D_stall = 0;
D_bubble = 0;
E_bubble = 0;
M_bubble = 0;
W_stall = 0;
set_cc = 1;
if(E_icode==4'h7 & !e_cnd)
begin
    D_bubble = 1;
    E_bubble = 1;
end
else if((E_icode == 4'h5 | E_icode == 4'hB) & (E_dstM==d_srcA | E_dstM==d_srcB))
begin
    F_stall = 1;
    D_stall = 1;
    E_bubble = 1;
end
else if(E_icode == 4'h9)
begin
    F_stall = 1;
    D_bubble = 1;
end
else if(E_icode == 4'h0 | m_stat!=4'b1000 | W_stat!=4'b1000)
begin
    set_cc = 0;
end
```

Simulation Results for Pipelined Block including Data Forwarding and Elimination of Data and Control Hazards :

The assembly code that is used for testing is as follows which is the code for GCD:

```
main:

irmovq $0x0, %rax

irmovq $0x10, %rdx

irmovq $0xc, %rbx

jmp check

check:

addq %rax, %rbx

je rbxres

addq %rax, %rdx

je rdxres

jmp loop2

loop2:

rrmovq %rdx, %rsi

rrmovq %rbx, %rdi

subq %rbx, %rsi

jge ab1

subq %rdx, %rdi

jge ab2

ab1:

rrmovq %rbx, %rdx

rrmovq %rsi, %rbx

jmp check

ab2:

rrmovq %rbx, %rdx

rrmovq %rdi, %rbx

jmp check

rbxres:

rrmovq %rdx, %rcx
```

```
halt

rdxres:

rrmovq %rbx, %rcx

halt
```

The equivalent Machine level code is as follows

```
//main:
    //irmovq $0x0, %rax
    instr_mem[0]=8'b00110000; //3 0
    instr_mem[1]=8'b00000000; //F rB=0
    instr_mem[2]=8'b00000000;
    instr_mem[3]=8'b00000000;
    instr_mem[4]=8'b00000000;
    instr_mem[5]=8'b00000000;
    instr_mem[6]=8'b00000000;
    instr_mem[7]=8'b00000000;
    instr_mem[8]=8'b00000000;
    instr_mem[9]=8'b00000000; //V=0
    //irmovq $0x10, %rdx
    instr_mem[10]=8'b00110000; //3 0
    instr_mem[11]=8'b00000010; //F rB=2
    instr_mem[12]=8'b00000000;
    instr_mem[13]=8'b00000000;
    instr_mem[14]=8'b00000000;
    instr_mem[15]=8'b00000000;
    instr_mem[16]=8'b00000000;
    instr_mem[17]=8'b00000000;
    instr_mem[18]=8'b00000000;
    instr_mem[19]=8'b00100000; //V=32
    //irmovq $0xc, %rbx
    instr_mem[20]=8'b00110000; //3 0
    instr_mem[21]=8'b00000011; //F rB=3
    instr_mem[22]=8'b00000000;
    instr_mem[23]=8'b00000000;
    instr_mem[24]=8'b00000000;
    instr_mem[25]=8'b00000000;
    instr_mem[26]=8'b00000000;
    instr_mem[27]=8'b00000000;
    instr_mem[28]=8'b00000000;
    instr_mem[29]=8'b00010000; //V=12
    //jmp check
    instr_mem[30]=8'b01110000; //7 fn
    instr_mem[31]=8'b00000000; //Dest
    instr_mem[32]=8'b00000000; //Dest
    instr_mem[33]=8'b00000000; //Dest
    instr_mem[34]=8'b00000000; //Dest
    instr_mem[35]=8'b00000000; //Dest
    instr_mem[36]=8'b00000000; //Dest
    instr_mem[37]=8'b00000000; //Dest
    instr_mem[38]=8'b00100111; //Dest=39
    // check:
    // addq %rax, %rbx
    instr_mem[39]=8'b01100000; //5 fn
    instr_mem[40]=8'b00000011; //rA=0 rB=3
    // je rbxres
    instr_mem[41]=8'b01110011; //7 fn=3
    instr_mem[42]=8'b00000000; //Dest
    instr_mem[43]=8'b00000000; //Dest
    instr_mem[44]=8'b00000000; //Dest
    instr_mem[45]=8'b00000000; //Dest
    instr_mem[46]=8'b00000000; //Dest
    instr_mem[47]=8'b00000000; //Dest
    instr_mem[48]=8'b00000000; //Dest
    instr_mem[49]=8'b01111010; //Dest=122
    // addq %rax, %rdx
    instr_mem[50]=8'b01100000; //5 fn
    instr_mem[51]=8'b00000010; //rA=0 rB=2
    // je rdxres
    instr_mem[52]=8'b01110011; //7 fn=3
    instr_mem[53]=8'b00000000; //Dest
    instr_mem[54]=8'b00000000; //Dest
    instr_mem[55]=8'b00000000; //Dest
    instr_mem[56]=8'b00000000; //Dest
    instr_mem[57]=8'b00000000; //Dest
    instr_mem[58]=8'b00000000; //Dest
    instr_mem[59]=8'b00000000; //Dest
    instr_mem[60]=8'b01111101; //Dest=125
    // jmp loop2
    instr_mem[61]=8'b01110000; //7 fn=0
    instr_mem[62]=8'b00000000; //Dest
    instr_mem[63]=8'b00000000; //Dest
    instr_mem[64]=8'b00000000; //Dest
    instr_mem[65]=8'b00000000; //Dest
    instr_mem[66]=8'b00000000; //Dest
    instr_mem[67]=8'b00000000; //Dest
    instr_mem[68]=8'b00000000; //Dest
    instr_mem[69]=8'b01000110; //Dest
```

```
    // loop2:
    // rrmovq %rdx, %rsi
    instr_mem[70]=8'b00100000; //2 fn=0
    instr_mem[71]=8'b00100110; //rA=2 rB=6
    // rrmovq %rbx, %rdi
    instr_mem[72]=8'b00100000; //2 fn=0
    instr_mem[73]=8'b00110111; //rA=3 rB=7
    // subq %rbx, %rsi
    instr_mem[74]=8'b01100001; //5 fn=1
    instr_mem[75]=8'b00110110; //rA=3 rB=6
    // jge ab1
    instr_mem[76]=8'b01110101; //7 fn=5
    instr_mem[77]=8'b00000000; //Dest
    instr_mem[78]=8'b00000000; //Dest
    instr_mem[79]=8'b00000000; //Dest
    instr_mem[80]=8'b00000000; //Dest
    instr_mem[81]=8'b00000000; //Dest
    instr_mem[82]=8'b00000000; //Dest
    instr_mem[83]=8'b00000000; //Dest
    instr_mem[84]=8'b01100000; //Dest=96
    // subq %rdx, %rdi
    instr_mem[85]=8'b01100001; //5 fn
    instr_mem[86]=8'b00100111; //rA=2 rB=7
    // jge ab2
    instr_mem[87]=8'b01110101; //7 fn=5
    instr_mem[88]=8'b00000000; //Dest
    instr_mem[89]=8'b00000000; //Dest
    instr_mem[90]=8'b00000000; //Dest
    instr_mem[91]=8'b00000000; //Dest
    instr_mem[92]=8'b00000000; //Dest
    instr_mem[93]=8'b00000000; //Dest
    instr_mem[94]=8'b00000000; //Dest
    instr_mem[95]=8'b01101101; //Dest=109
```

```
    // ab1:
    // rrmovq %rbx, %rdx
    instr_mem[96]=8'b00100000; //2 fn=0
    instr_mem[97]=8'b00110010; //rA=3 rB=2
    // rrmovq %rsi, %rbx
    instr_mem[98]=8'b00100000; //2 fn=0
    instr_mem[99]=8'b01100011; //rA=6 rB=3
    // jmp check
    instr_mem[100]=8'b01110000; //7 fn=0
    instr_mem[101]=8'b00000000; //Dest
    instr_mem[102]=8'b00000000; //Dest
    instr_mem[103]=8'b00000000; //Dest
    instr_mem[104]=8'b00000000; //Dest
    instr_mem[105]=8'b00000000; //Dest
    instr_mem[106]=8'b00000000; //Dest
    instr_mem[107]=8'b00000000; //Dest
    instr_mem[108]=8'b00100111; //Dest=39
```

```
    // ab2:
    // rrmovq %rbx, %rdx
    instr_mem[109]=8'b00100000; //2 fn=0
    instr_mem[110]=8'b00110010; //rA=3 rB=2
    // rrmovq %rdi, %rbx
    instr_mem[111]=8'b00100000; //2 fn=0
    instr_mem[112]=8'b01110011; //rA=7 rB=3
    // jmp check
    instr_mem[113]=8'b01110000; //7 fn=0
    instr_mem[114]=8'b00000000; //Dest
    instr_mem[115]=8'b00000000; //Dest
    instr_mem[116]=8'b00000000; //Dest
    instr_mem[117]=8'b00000000; //Dest
    instr_mem[118]=8'b00000000; //Dest
    instr_mem[119]=8'b00000000; //Dest
    instr_mem[120]=8'b00000000; //Dest
    instr_mem[121]=8'b00100111; //Dest=39
```

```
    // rbxres:
    // rrmovq %rdx, %rcx
    instr_mem[122]=8'b00100000; //2 fn=0
    instr_mem[123]=8'b00100001; //rA=2 rB=1
    // halt
    instr_mem[124]=8'b00000000;

    // rdxres:
    // rrmovq %rbx, %rcx
    instr_mem[125]=8'b00100000; //2 fn=0
    instr_mem[126]=8'b00110001; //rA=3 rB=1
    // halt
    instr_mem[127]=8'b00000000;
```

The output obtained is as follows

```
clk=0 f_predPC=                  10 F_predPC=                   0 D_icode= 1,E_icode= x,
M_icode= x, ifun= 0,rax=                    x,rdx=                   x,rbx=
x,rcx=                   x

clk=1 f_predPC=                  20 F_predPC=                  10 D_icode= 3,E_icode= 1,
M_icode= x, ifun= 0,rax=                    2,rdx=                   2,rbx=
5,rcx=                   1

clk=0 f_predPC=                  20 F_predPC=                  10 D_icode= 3,E_icode= 1,
M_icode= x, ifun= 0,rax=                    2,rdx=                   2,rbx=
5,rcx=                   1

clk=1 f_predPC=                  30 F_predPC=                  20 D_icode= 3,E_icode= 3,
M_icode= 1, ifun= 0,rax=                    2,rdx=                   2,rbx=
5,rcx=                   1

clk=0 f_predPC=                  30 F_predPC=                  20 D_icode= 3,E_icode= 3,
M_icode= 1, ifun= 0,rax=                    2,rdx=                   2,rbx=
5,rcx=                   1

clk=1 f_predPC=                  39 F_predPC=                  30 D_icode= 3,E_icode= 3,
M_icode= 3, ifun= 0,rax=                    2,rdx=                   2,rbx=
5,rcx=                   1

clk=0 f_predPC=                  39 F_predPC=                  30 D_icode= 3,E_icode= 3,
M_icode= 3, ifun= 0,rax=                    2,rdx=                   2,rbx=
5,rcx=                   1

clk=1 f_predPC=                  41 F_predPC=                  39 D_icode= 7,E_icode= 3,
M_icode= 3, ifun= 0,rax=                    2,rdx=                   2,rbx=
5,rcx=                   1

clk=0 f_predPC=                  41 F_predPC=                  39 D_icode= 7,E_icode= 3,
M_icode= 3, ifun= 0,rax=                    2,rdx=                   2,rbx=
5,rcx=                   1

clk=1 f_predPC=                 122 F_predPC=                  41 D_icode= 6,E_icode= 7,
M_icode= 3, ifun= 0,rax=                    0,rdx=                   2,rbx=
5,rcx=                   1

clk=0 f_predPC=                 122 F_predPC=                  41 D_icode= 6,E_icode= 7,
M_icode= 3, ifun= 0,rax=                    0,rdx=                   2,rbx=
5,rcx=                   1

clk=1 f_predPC=                 124 F_predPC=                 122 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                    0,rdx=                  32,rbx=
5,rcx=                   1
```

```
clk=0 f_predPC=                 124 F_predPC=                 122 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                  32,rbx=
5,rcx=                   1

clk=1 f_predPC=                 124 F_predPC=                 124 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                 124 F_predPC=                 124 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  52 F_predPC=                 124 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                  52 F_predPC=                 124 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                 125 F_predPC=                  52 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                 125 F_predPC=                  52 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                 127 F_predPC=                 125 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                 127 F_predPC=                 125 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                 127 F_predPC=                 127 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                 127 F_predPC=                 127 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  70 F_predPC=                 127 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                  70 F_predPC=                 127 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  72 F_predPC=                  70 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                  72 F_predPC=                  70 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  74 F_predPC=                  72 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1
```

```
clk=0 f_predPC=                  74 F_predPC=                  72 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  76 F_predPC=                  74 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                  76 F_predPC=                  74 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  96 F_predPC=                  76 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                  96 F_predPC=                  76 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  98 F_predPC=                  96 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                  98 F_predPC=                  96 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                 100 F_predPC=                  98 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                 100 F_predPC=                  98 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  39 F_predPC=                 100 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                  39 F_predPC=                 100 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                  41 F_predPC=                  39 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                  41 F_predPC=                  39 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                 122 F_predPC=                  41 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=0 f_predPC=                 122 F_predPC=                  41 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                  32,rbx=
16,rcx=                   1

clk=1 f_predPC=                 124 F_predPC=                 122 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                  16,rbx=
16,rcx=                   1
```

```
clk=0 f_predPC=                 124 F_predPC=                  122 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=1 f_predPC=                 124 F_predPC=                  124 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=0 f_predPC=                 124 F_predPC=                  124 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=1 f_predPC=                  52 F_predPC=                  124 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=0 f_predPC=                  52 F_predPC=                  124 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=1 f_predPC=                 125 F_predPC=                   52 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=0 f_predPC=                 125 F_predPC=                   52 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=1 f_predPC=                 127 F_predPC=                  125 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=0 f_predPC=                 127 F_predPC=                  125 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=1 f_predPC=                 127 F_predPC=                  127 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=0 f_predPC=                 127 F_predPC=                  127 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=1 f_predPC=                  70 F_predPC=                  127 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=0 f_predPC=                  70 F_predPC=                  127 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=1 f_predPC=                  72 F_predPC=                   70 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=0 f_predPC=                  72 F_predPC=                   70 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1

clk=1 f_predPC=                  74 F_predPC=                   72 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   16,rbx=
16,rcx=                   1
```

```
clk=0 f_predPC=                  74 F_predPC=                  72 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=1 f_predPC=                  76 F_predPC=                  74 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=0 f_predPC=                  76 F_predPC=                  74 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=1 f_predPC=                  96 F_predPC=                  76 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=0 f_predPC=                  96 F_predPC=                  76 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=1 f_predPC=                  98 F_predPC=                  96 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=0 f_predPC=                  98 F_predPC=                  96 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=1 f_predPC=                 100 F_predPC=                  98 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=0 f_predPC=                 100 F_predPC=                  98 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=1 f_predPC=                  39 F_predPC=                 100 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=0 f_predPC=                  39 F_predPC=                 100 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=1 f_predPC=                  41 F_predPC=                  39 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=0 f_predPC=                  41 F_predPC=                  39 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=1 f_predPC=                 122 F_predPC=                  41 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=0 f_predPC=                 122 F_predPC=                  41 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1

clk=1 f_predPC=                 124 F_predPC=                 122 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                  16,rbx=
16,rcx=                    1
```

```
clk=0 f_predPC=                124 F_predPC=                122 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                  0,rdx=                    16,rbx=
16,rcx=                   1

clk=1 f_predPC=                124 F_predPC=                124 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                   1

clk=0 f_predPC=                124 F_predPC=                124 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                   1

clk=1 f_predPC=                124 F_predPC=                124 D_icode= 0,E_icode= 2,
M_icode= 7, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                   1

clk=0 f_predPC=                124 F_predPC=                124 D_icode= 0,E_icode= 2,
M_icode= 7, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                   1

clk=1 f_predPC=                124 F_predPC=                124 D_icode= 0,E_icode= 0,
M_icode= 2, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                   1

clk=0 f_predPC=                124 F_predPC=                124 D_icode= 0,E_icode= 0,
M_icode= 2, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                   1

clk=1 f_predPC=                124 F_predPC=                124 D_icode= 0,E_icode= 0,
M_icode= 0, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                   1

clk=0 f_predPC=                124 F_predPC=                124 D_icode= 0,E_icode= 0,
M_icode= 0, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                   1

Halting
clk=1 f_predPC=                124 F_predPC=                124 D_icode= 0,E_icode= 0,
M_icode= 0, ifun= 0,rax=                  0,rdx=                    16,rbx=
0,rcx=                  16
```

which is same as that of the expected output for the pipelined implementation.

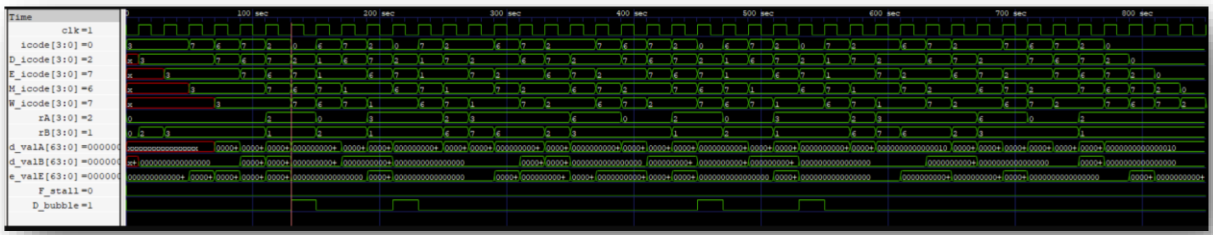
Here the following are being tested:

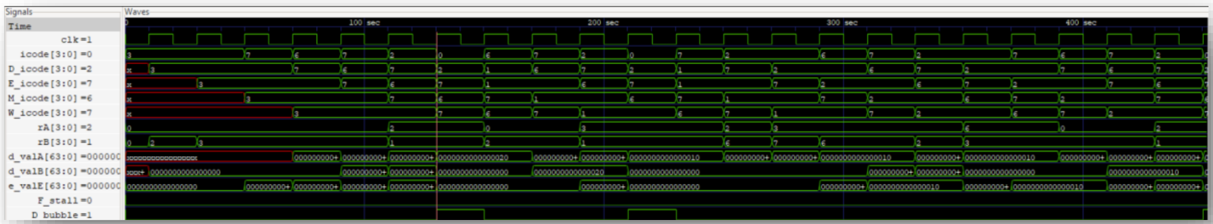
`irmovq`, `jXX`, `addq`, `rrmovq`, `subq`, `jge`, `halt.`

The data dependencies for this is implemented as follows

When the irmovq is implemented, the %rbx register is updated to the value of 0Xc which when the jump is taken in the next clock cycle and the clock cycle after, the register is needed which is available due to data forwarding which is seen in the above code.

The waveforms for the codes are as follows:


Time
clk =1
icode[3:0] =0
D_icode[3:0] =2
E_icode[3:0] =7
M_icode[3:0] =6
W_icode[3:0] =7
rA[3:0] =2
rB[3:0] =1
d_valA[63:0] =000000
d_valB[63:0] =000000
e_valE[63:0] =000000
F_stall =0
D_bubble =1
100 sec
200 sec
300 sec
400 sec
500 sec
600 sec
700 sec
800 sec


Signals
Waves
Time
clk =1
icode[3:0] =0
D_icode[3:0] =2
E_icode[3:0] =7
M_icode[3:0] =6
W_icode[3:0] =7
rA[3:0] =2
rB[3:0] =1
d_valA[63:0] =000000
d_valB[63:0] =000000
e_valE[63:0] =000000
F_stall =0
D_bubble =1
100 sec
200 sec
300 sec
400 sec

The assembly code for testing of pipeline with call and return is given by

```
# Execution begins at address 0
    .pos 0
    irmovq stack, %rsp      # Set up stack pointer
    call main       # Execute main program
    halt            # Terminate program


main:   irmovq $0x10,%rdx
    irmovq $0xc,%rbx
    call gcd        # gcd(a,b)
    ret

# gcd(quad a, quad b)
# a in %rdx, b in %rbx
gcd:
  irmovq $0x0, %rax
  jmp check
check:
  addq %rax, %rbx
  je rbxres
  addq %rax, %rdx
  je rdxres
  jmp loop2
loop2:
  rrmovq %rdx, %rsi
  rrmovq %rbx, %rdi

  subq %rbx, %rsi
  jge ab1
  subq %rdx, %rdi
  jge ab2
ab1:
  rrmovq %rbx, %rdx
  rrmovq %rsi, %rbx
  jmp check
ab2:
  rrmovq %rbx, %rdx
  rrmovq %rdi, %rbx
  jmp check
rbxres:
  rrmovq %rdx, %rcx
  ret
rdxres:
  rrmovq %rbx, %rcx
  ret                   # Return

# Stack starts here and grows to lower addresses
```

```
    .pos 1023
stack:
```

The results for the implementation including call and return for the following code

```
//.pos 0
// irmovq stack %rsp
    instr_mem[0]=8'h30; //3 0
    instr_mem[1]=8'hF4; //F rB=0
    instr_mem[2]=8'h00;
    instr_mem[3]=8'h00;
    instr_mem[4]=8'h00;
    instr_mem[5]=8'h00;
    instr_mem[6]=8'h00;
    instr_mem[7]=8'h00;
    instr_mem[8]=8'h03;
    instr_mem[9]=8'hff; //V=1023

// call main
    instr_mem[10]=8'h80; //8 0
    instr_mem[11]=8'h00;
    instr_mem[12]=8'h00;
    instr_mem[13]=8'h00;
    instr_mem[14]=8'h00;
    instr_mem[15]=8'h00;
    instr_mem[16]=8'h00;
    instr_mem[17]=8'h00;
    instr_mem[18]=8'h14;

// halt
    instr_mem[19]=8'h00; //0 0


//main:

// irmovq $0x10 %rdx
    instr_mem[20]=8'h30; //3 0
    instr_mem[21]=8'hF2; //F rB=2
    instr_mem[22]=8'h00;
    instr_mem[23]=8'h00;
    instr_mem[24]=8'h00;
    instr_mem[25]=8'h00;
    instr_mem[26]=8'h00;
    instr_mem[27]=8'h00;
    instr_mem[28]=8'h00;
    instr_mem[29]=8'h10; //V=16
// irmovq $0xc %rbx
    instr_mem[30]=8'h30; //3 0
    instr_mem[31]=8'hF3; //F rB=3
    instr_mem[32]=8'h00;
    instr_mem[33]=8'h00;
    instr_mem[34]=8'h00;
    instr_mem[35]=8'h00;
    instr_mem[36]=8'h00;
    instr_mem[37]=8'h00;
    instr_mem[38]=8'h00;
    instr_mem[39]=8'h0c; //V=12
// call gcd
    instr_mem[40]=8'h80; // 8 0
    instr_mem[41]=8'h00;
    instr_mem[42]=8'h00;
    instr_mem[43]=8'h00;
    instr_mem[44]=8'h00;
    instr_mem[45]=8'h00;
    instr_mem[46]=8'h00;
    instr_mem[47]=8'h00;
    instr_mem[48]=8'h32;
// ret
    instr_mem[49]=8'h90; // 9 0

// gcd(%rdx,%rbx)
// gcd:
//irmovq $0x0, %rax
instr_mem[50]=8'b00110000; //3 0
instr_mem[51]=8'b00000000; //F rB=0
instr_mem[52]=8'b00000000;
instr_mem[53]=8'b00000000;
instr_mem[54]=8'b00000000;
instr_mem[55]=8'b00000000;
instr_mem[56]=8'b00000000;
instr_mem[57]=8'b00000000;
instr_mem[58]=8'b00000000;
instr_mem[59]=8'b00000000; //V=0
//jmp check
instr_mem[60]=8'b01110000; //7 fn
instr_mem[61]=8'b00000000; //Dest
instr_mem[62]=8'b00000000; //Dest
instr_mem[63]=8'b00000000; //Dest
instr_mem[64]=8'b00000000; //Dest
instr_mem[65]=8'b00000000; //Dest
instr_mem[66]=8'b00000000; //Dest
instr_mem[67]=8'b00000000; //Dest
instr_mem[68]=8'h45; //Dest=69

// check:
// addq %rax, %rbx
instr_mem[69]=8'b01100000; //5 fn
instr_mem[70]=8'b00000011; //rA=0 rB=3
// je rbxres
instr_mem[71]=8'b01110011; //7 fn=3
instr_mem[72]=8'b00000000; //Dest
instr_mem[73]=8'b00000000; //Dest
instr_mem[74]=8'b00000000; //Dest
instr_mem[75]=8'b00000000; //Dest
instr_mem[76]=8'b00000000; //Dest
instr_mem[77]=8'b00000000; //Dest
instr_mem[78]=8'b00000000; //Dest
instr_mem[79]=8'h98; //Dest=152
// addq %rax, %rdx
```

```
instr_mem[80]=8'b01100000; //5 fn
instr_mem[81]=8'b00000010; //rA=0 rB=2
// je rdxres
instr_mem[82]=8'b01110011; //7 fn=3
instr_mem[83]=8'b00000000; //Dest
instr_mem[84]=8'b00000000; //Dest
instr_mem[85]=8'b00000000; //Dest
instr_mem[86]=8'b00000000; //Dest
instr_mem[87]=8'b00000000; //Dest
instr_mem[88]=8'b00000000; //Dest
instr_mem[89]=8'b00000000; //Dest
instr_mem[90]=8'h9B; //Dest=155
// jmp loop2
instr_mem[91]=8'b01110000; //7 fn=0
instr_mem[92]=8'b00000000; //Dest
instr_mem[93]=8'b00000000; //Dest
instr_mem[94]=8'b00000000; //Dest
instr_mem[95]=8'b00000000; //Dest
instr_mem[96]=8'b00000000; //Dest
instr_mem[97]=8'b00000000; //Dest
instr_mem[98]=8'b00000000; //Dest
instr_mem[99]=8'h64; //Dest=100

// loop2:
// rrmovq %rdx, %rsi
instr_mem[100]=8'b00100000; //2 fn=0
instr_mem[101]=8'b00100110; //rA=2 rB=6
// rrmovq %rbx, %rdi
instr_mem[102]=8'b00100000; //2 fn=0
instr_mem[103]=8'b00110111; //rA=3 rB=7
// subq %rbx, %rsi
instr_mem[104]=8'b01100001; //5 fn=1
instr_mem[105]=8'b00110110; //rA=3 rB=6
// jge ab1
instr_mem[106]=8'b01110101; //7 fn=5
instr_mem[107]=8'b00000000; //Dest
instr_mem[108]=8'b00000000; //Dest
instr_mem[109]=8'b00000000; //Dest
instr_mem[110]=8'b00000000; //Dest
instr_mem[111]=8'b00000000; //Dest
instr_mem[112]=8'b00000000; //Dest
instr_mem[113]=8'b00000000; //Dest
instr_mem[114]=8'h7E; //Dest=126
// subq %rdx, %rdi
instr_mem[115]=8'b01100001; //5 fn
instr_mem[116]=8'b00100111; //rA=2 rB=7
// jge ab2
instr_mem[117]=8'b01110101; //7 fn=5
instr_mem[118]=8'b00000000; //Dest
instr_mem[119]=8'b00000000; //Dest
instr_mem[120]=8'b00000000; //Dest
instr_mem[121]=8'b00000000; //Dest
instr_mem[122]=8'b00000000; //Dest
instr_mem[123]=8'b00000000; //Dest
instr_mem[124]=8'b00000000; //Dest
instr_mem[125]=8'h8B; //Dest=139
```

```
// ab1:
// rrmovq %rbx, %rdx
instr_mem[126]=8'b00100000; //2 fn=0
instr_mem[127]=8'b00110010; //rA=3 rB=2
// rrmovq %rsi, %rbx
instr_mem[128]=8'b00100000; //2 fn=0
instr_mem[129]=8'b01100011; //rA=6 rB=3
// jmp check
instr_mem[130]=8'b01110000; //7 fn=0
instr_mem[131]=8'b00000000; //Dest
instr_mem[132]=8'b00000000; //Dest
instr_mem[133]=8'b00000000; //Dest
instr_mem[134]=8'b00000000; //Dest
instr_mem[135]=8'b00000000; //Dest
instr_mem[136]=8'b00000000; //Dest
instr_mem[137]=8'b00000000; //Dest
instr_mem[138]=8'h45; //Dest=69

// ab2:
// rrmovq %rbx, %rdx
instr_mem[139]=8'b00100000; //2 fn=0
instr_mem[140]=8'b00110010; //rA=3 rB=2
// rrmovq %rdi, %rbx
instr_mem[141]=8'b00100000; //2 fn=0
instr_mem[142]=8'b01110011; //rA=7 rB=3
// jmp check
instr_mem[143]=8'b01110000; //7 fn=0
instr_mem[144]=8'b00000000; //Dest
instr_mem[145]=8'b00000000; //Dest
instr_mem[146]=8'b00000000; //Dest
instr_mem[147]=8'b00000000; //Dest
instr_mem[148]=8'b00000000; //Dest
instr_mem[149]=8'b00000000; //Dest
instr_mem[150]=8'b00000000; //Dest
instr_mem[151]=8'h45; //Dest=39

// rbxres:
// rrmovq %rdx, %rcx
instr_mem[152]=8'b00100000; //2 fn=0
instr_mem[153]=8'b00100001; //rA=2 rB=1
// ret
instr_mem[154]=8'h90; // 9 0

// rdxres:
// rrmovq %rbx, %rcx
instr_mem[155]=8'b00100000; //2 fn=0
instr_mem[156]=8'b00110001; //rA=3 rB=1
// ret
instr_mem[157]=8'h90; // 9 0
```

is as follows

```
clk=0 f_predPC=                   10 F_predPC=                    0 D_icode= x,E_icode= x,
M_icode= x, ifun= x,rax=                    x,rdx=                    x,rbx=
x,rcx=                   x

clk=1 f_predPC=                   20 F_predPC=                   10 D_icode= 3,E_icode= x,
M_icode= x, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=0 f_predPC=                   20 F_predPC=                   10 D_icode= 3,E_icode= x,
M_icode= x, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=1 f_predPC=                   30 F_predPC=                   20 D_icode= 8,E_icode= 3,
M_icode= x, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=0 f_predPC=                   30 F_predPC=                   20 D_icode= 8,E_icode= 3,
M_icode= x, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=1 f_predPC=                   40 F_predPC=                   30 D_icode= 3,E_icode= 8,
M_icode= 3, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=0 f_predPC=                   40 F_predPC=                   30 D_icode= 3,E_icode= 8,
M_icode= 3, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=1 f_predPC=                   50 F_predPC=                   40 D_icode= 3,E_icode= 3,
M_icode= 8, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=0 f_predPC=                   50 F_predPC=                   40 D_icode= 3,E_icode= 3,
M_icode= 8, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=1 f_predPC=                   60 F_predPC=                   50 D_icode= 8,E_icode= 3,
M_icode= 3, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=0 f_predPC=                   60 F_predPC=                   50 D_icode= 8,E_icode= 3,
M_icode= 3, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=1 f_predPC=                   69 F_predPC=                   60 D_icode= 3,E_icode= 8,
M_icode= 3, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=0 f_predPC=                   69 F_predPC=                   60 D_icode= 3,E_icode= 8,
M_icode= 3, ifun= 0,rax=                    2,rdx=                    2,rbx=
5,rcx=                   1

clk=1 f_predPC=                   71 F_predPC=                   69 D_icode= 7,E_icode= 3,
M_icode= 8, ifun= 0,rax=                    2,rdx=                   16,rbx=
5,rcx=                   1

clk=0 f_predPC=                   71 F_predPC=                   69 D_icode= 7,E_icode= 3,
M_icode= 8, ifun= 0,rax=                    2,rdx=                   16,rbx=
5,rcx=                   1

clk=1 f_predPC=                  152 F_predPC=                   71 D_icode= 6,E_icode= 7,
M_icode= 3, ifun= 0,rax=                    2,rdx=                   16,rbx=
12,rcx=                   1
```

```
clk=0 f_predPC=                 152 F_predPC=                  71 D_icode= 6,E_icode= 7,
M_icode= 3, ifun= 0,rax=                   2,rdx=                  16,rbx=
12,rcx=                   1

clk=1 f_predPC=                 154 F_predPC=                 152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   2,rdx=                  16,rbx=
12,rcx=                   1

clk=0 f_predPC=                 154 F_predPC=                 152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   2,rdx=                  16,rbx=
12,rcx=                   1

clk=1 f_predPC=                 154 F_predPC=                 154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=0 f_predPC=                 154 F_predPC=                 154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=1 f_predPC=                  82 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=0 f_predPC=                  82 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=1 f_predPC=                 155 F_predPC=                  82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=0 f_predPC=                 155 F_predPC=                  82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=1 f_predPC=                 157 F_predPC=                 155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=0 f_predPC=                 157 F_predPC=                 155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=1 f_predPC=                 157 F_predPC=                 157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=0 f_predPC=                 157 F_predPC=                 157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=1 f_predPC=                 100 F_predPC=                 157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=0 f_predPC=                 100 F_predPC=                 157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1

clk=1 f_predPC=                 102 F_predPC=                 100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                  16,rbx=
12,rcx=                   1
```

```
clk=0 f_predPC=                102 F_predPC=                100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=1 f_predPC=                104 F_predPC=                102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=0 f_predPC=                104 F_predPC=                102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=1 f_predPC=                106 F_predPC=                104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=0 f_predPC=                106 F_predPC=                104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=1 f_predPC=                126 F_predPC=                106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=0 f_predPC=                126 F_predPC=                106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=1 f_predPC=                128 F_predPC=                126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=0 f_predPC=                128 F_predPC=                126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=1 f_predPC=                130 F_predPC=                128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=0 f_predPC=                130 F_predPC=                128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=1 f_predPC=                 69 F_predPC=                130 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=0 f_predPC=                 69 F_predPC=                130 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=1 f_predPC=                 71 F_predPC=                 69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=0 f_predPC=                 71 F_predPC=                 69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1

clk=1 f_predPC=                152 F_predPC=                 71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                 0,rdx=                16,rbx=
12,rcx=                    1
```

```
clk=0 f_predPC=                152 F_predPC=                 71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                 0,rdx=                 16,rbx=
12,rcx=                  1

clk=1 f_predPC=                154 F_predPC=                152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                 0,rdx=                 12,rbx=
12,rcx=                  1

clk=0 f_predPC=                154 F_predPC=                152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                 0,rdx=                 12,rbx=
12,rcx=                  1

clk=1 f_predPC=                154 F_predPC=                154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=0 f_predPC=                154 F_predPC=                154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=1 f_predPC=                 82 F_predPC=                154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=0 f_predPC=                 82 F_predPC=                154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=1 f_predPC=                155 F_predPC=                 82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=0 f_predPC=                155 F_predPC=                 82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=1 f_predPC=                157 F_predPC=                155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=0 f_predPC=                157 F_predPC=                155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=1 f_predPC=                157 F_predPC=                157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=0 f_predPC=                157 F_predPC=                157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=1 f_predPC=                100 F_predPC=                157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=0 f_predPC=                100 F_predPC=                157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1

clk=1 f_predPC=                102 F_predPC=                100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                 0,rdx=                 12,rbx=
4,rcx=                  1
```

```
clk=0 f_predPC=                 102 F_predPC=                 100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=1 f_predPC=                 104 F_predPC=                 102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=0 f_predPC=                 104 F_predPC=                 102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=1 f_predPC=                 106 F_predPC=                 104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=0 f_predPC=                 106 F_predPC=                 104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=1 f_predPC=                 126 F_predPC=                 106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=0 f_predPC=                 126 F_predPC=                 106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=1 f_predPC=                 128 F_predPC=                 126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=0 f_predPC=                 128 F_predPC=                 126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=1 f_predPC=                 130 F_predPC=                 128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=0 f_predPC=                 130 F_predPC=                 128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=1 f_predPC=                  69 F_predPC=                 130 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=0 f_predPC=                  69 F_predPC=                 130 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=1 f_predPC=                  71 F_predPC=                  69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=0 f_predPC=                  71 F_predPC=                  69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1

clk=1 f_predPC=                 152 F_predPC=                  71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   12,rbx=
4,rcx=                   1
```

```
clk=0 f_predPC=                  152 F_predPC=                   71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                    0,rdx=                  12,rbx=
4,rcx=                   1

clk=1 f_predPC=                  154 F_predPC=                  152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                    0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                  154 F_predPC=                  152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                    0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                  154 F_predPC=                  154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                  154 F_predPC=                  154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                   82 F_predPC=                  154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                   82 F_predPC=                  154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                  155 F_predPC=                   82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                  155 F_predPC=                   82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                  157 F_predPC=                  155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                  157 F_predPC=                  155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                  157 F_predPC=                  157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                  157 F_predPC=                  157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                  100 F_predPC=                  157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                  100 F_predPC=                  157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                  102 F_predPC=                  100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                    0,rdx=                   4,rbx=
8,rcx=                   1
```

```
clk=0 f_predPC=                 102 F_predPC=                 100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 104 F_predPC=                 102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 104 F_predPC=                 102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 106 F_predPC=                 104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 106 F_predPC=                 104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 126 F_predPC=                 106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 126 F_predPC=                 106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 128 F_predPC=                 126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 128 F_predPC=                 126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 130 F_predPC=                 128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 130 F_predPC=                 128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 117 F_predPC=                 130 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 117 F_predPC=                 130 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 139 F_predPC=                 117 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 1,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 139 F_predPC=                 117 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 1,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 141 F_predPC=                 139 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 5,rax=                   0,rdx=                   4,rbx=
8,rcx=                   1
```

```
clk=0 f_predPC=                141 F_predPC=               139 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 5,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=1 f_predPC=                143 F_predPC=               141 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=0 f_predPC=                143 F_predPC=               141 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 69 F_predPC=               143 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 69 F_predPC=               143 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=1 f_predPC=                 71 F_predPC=                69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=0 f_predPC=                 71 F_predPC=                69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=1 f_predPC=                152 F_predPC=                71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=0 f_predPC=                152 F_predPC=                71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                      4,rbx=
8,rcx=                   1

clk=1 f_predPC=                154 F_predPC=               152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                      8,rbx=
8,rcx=                   1

clk=0 f_predPC=                154 F_predPC=               152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                      8,rbx=
8,rcx=                   1

clk=1 f_predPC=                154 F_predPC=               154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                      8,rbx=
4,rcx=                   1

clk=0 f_predPC=                154 F_predPC=               154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                      8,rbx=
4,rcx=                   1

clk=1 f_predPC=                 82 F_predPC=               154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                      8,rbx=
4,rcx=                   1

clk=0 f_predPC=                 82 F_predPC=               154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                      8,rbx=
4,rcx=                   1

clk=1 f_predPC=                155 F_predPC=                82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                      8,rbx=
4,rcx=                   1
```

```
clk=0 f_predPC=                155 F_predPC=                82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=1 f_predPC=                157 F_predPC=               155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=0 f_predPC=                157 F_predPC=               155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=1 f_predPC=                157 F_predPC=               157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=0 f_predPC=                157 F_predPC=               157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=1 f_predPC=                100 F_predPC=               157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=0 f_predPC=                100 F_predPC=               157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=1 f_predPC=                102 F_predPC=               100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=0 f_predPC=                102 F_predPC=               100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=1 f_predPC=                104 F_predPC=               102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=0 f_predPC=                104 F_predPC=               102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=1 f_predPC=                106 F_predPC=               104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=0 f_predPC=                106 F_predPC=               104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=1 f_predPC=                126 F_predPC=               106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=0 f_predPC=                126 F_predPC=               106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1

clk=1 f_predPC=                128 F_predPC=               126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                  0,rdx=                    8,rbx=
4,rcx=                   1
```

```
clk=0 f_predPC=                  128 F_predPC=                  126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=1 f_predPC=                  130 F_predPC=                  128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=0 f_predPC=                  130 F_predPC=                  128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=1 f_predPC=                   69 F_predPC=                  130 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=0 f_predPC=                   69 F_predPC=                  130 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=1 f_predPC=                   71 F_predPC=                   69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=0 f_predPC=                   71 F_predPC=                   69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=1 f_predPC=                  152 F_predPC=                   71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=0 f_predPC=                  152 F_predPC=                   71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   8,rbx=
4,rcx=                   1

clk=1 f_predPC=                  154 F_predPC=                  152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                  154 F_predPC=                  152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                  154 F_predPC=                  154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                  154 F_predPC=                  154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                   82 F_predPC=                  154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                   82 F_predPC=                  154 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                  155 F_predPC=                   82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1
```

```
clk=0 f_predPC=                155 F_predPC=                 82 D_icode= 6,E_icode= 1,
M_icode= 1, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=1 f_predPC=                157 F_predPC=                155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=0 f_predPC=                157 F_predPC=                155 D_icode= 7,E_icode= 6,
M_icode= 1, ifun= 3,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=1 f_predPC=                157 F_predPC=                157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=0 f_predPC=                157 F_predPC=                157 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=1 f_predPC=                100 F_predPC=                157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=0 f_predPC=                100 F_predPC=                157 D_icode= 1,E_icode= 1,
M_icode= 7, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=1 f_predPC=                102 F_predPC=                100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=0 f_predPC=                102 F_predPC=                100 D_icode= 7,E_icode= 1,
M_icode= 1, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=1 f_predPC=                104 F_predPC=                102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=0 f_predPC=                104 F_predPC=                102 D_icode= 2,E_icode= 7,
M_icode= 1, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=1 f_predPC=                106 F_predPC=                104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=0 f_predPC=                106 F_predPC=                104 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=1 f_predPC=                126 F_predPC=                106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=0 f_predPC=                126 F_predPC=                106 D_icode= 6,E_icode= 2,
M_icode= 2, ifun= 1,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1

clk=1 f_predPC=                128 F_predPC=                126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                 0,rdx=                    4,rbx=
4,rcx=                   1
```

```
clk=0 f_predPC=                128 F_predPC=                126 D_icode= 7,E_icode= 6,
M_icode= 2, ifun= 5,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                130 F_predPC=                128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                130 F_predPC=                128 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                 69 F_predPC=                130 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                 69 F_predPC=                130 D_icode= 2,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                 71 F_predPC=                 69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                 71 F_predPC=                 69 D_icode= 7,E_icode= 2,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                152 F_predPC=                 71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                152 F_predPC=                 71 D_icode= 6,E_icode= 7,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                154 F_predPC=                152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=0 f_predPC=                154 F_predPC=                152 D_icode= 7,E_icode= 6,
M_icode= 7, ifun= 3,rax=                   0,rdx=                   4,rbx=
4,rcx=                   1

clk=1 f_predPC=                154 F_predPC=                154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   1

clk=0 f_predPC=                154 F_predPC=                154 D_icode= 2,E_icode= 7,
M_icode= 6, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   1

clk=1 f_predPC=                154 F_predPC=                154 D_icode= 9,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   1

clk=0 f_predPC=                154 F_predPC=                154 D_icode= 9,E_icode= 2,
M_icode= 7, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   1

clk=1 f_predPC=                154 F_predPC=                154 D_icode= 1,E_icode= 9,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   1
```

```
clk=0 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 9,
M_icode= 2, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   1

clk=1 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 9, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   1

clk=0 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 9, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   1

clk=1 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=0 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=1 f_predPC=                 154 F_predPC=                 154 D_icode= 9,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=0 f_predPC=                 154 F_predPC=                 154 D_icode= 9,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=1 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 9,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=0 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 9,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=1 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 9, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=0 f_predPC=                 154 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 9, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=1 f_predPC=                  19 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=0 f_predPC=                  19 F_predPC=                 154 D_icode= 1,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=1 f_predPC=                  19 F_predPC=                  19 D_icode= 0,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=0 f_predPC=                  19 F_predPC=                  19 D_icode= 0,E_icode= 1,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=1 f_predPC=                  19 F_predPC=                  19 D_icode= 0,E_icode= 0,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4
```

```
clk=0 f_predPC=                  19 F_predPC=                  19 D_icode= 0,E_icode= 0,
M_icode= 1, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=1 f_predPC=                  19 F_predPC=                  19 D_icode= 0,E_icode= 0,
M_icode= 0, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

clk=0 f_predPC=                  19 F_predPC=                  19 D_icode= 0,E_icode= 0,
M_icode= 0, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4

Halting
clk=1 f_predPC=                  19 F_predPC=                  19 D_icode= 0,E_icode= 0,
M_icode= 0, ifun= 0,rax=                   0,rdx=                   4,rbx=
0,rcx=                   4
```

This sequence of execution shows that the call and return functionalities are implemented and executed correctly by the pipeline.

Challenges Encountered:

The challenges include the following:

1) Problems with ALU where the same wire was being driven by multiple modules which caused them to go to x state.

2) Keeping a check on clock cycles in which the individual blocks and the registers will get updated.

3) Data forwarding in pipelined implementation dealing with the registers of 0XF.

4) Dealing with the **d_srcA**, **d_srcB**, **d_dstE** and **d_dstM** with the implementation of their logic.

5) Implementing the control logic for data and control hazards.

6) Testing the call and return functions which required the inclusion of a stack in memory.

7) Conversion from Assembly to machine code in instruction memory (can be automated)