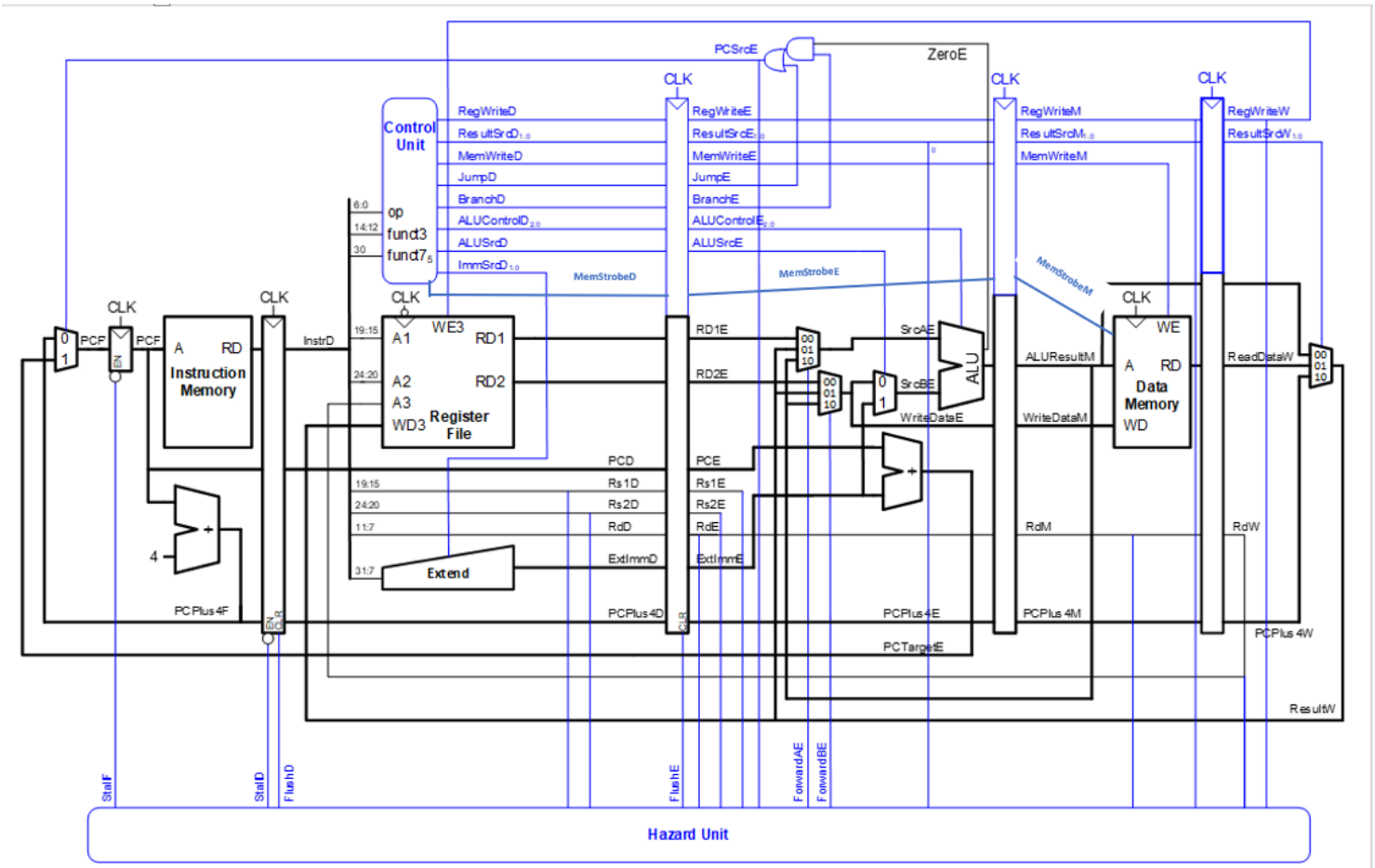
Pipelined RV32I Processor

With Hazard Unit

M.B: Kareem Atef

Full Datapath



Note: I added MemStrobe Signal to handle (load/store) byte, half word and word.

Supported Instruction set

- My Design supports 27 different instructions here are the following supported instructions formats:

imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW

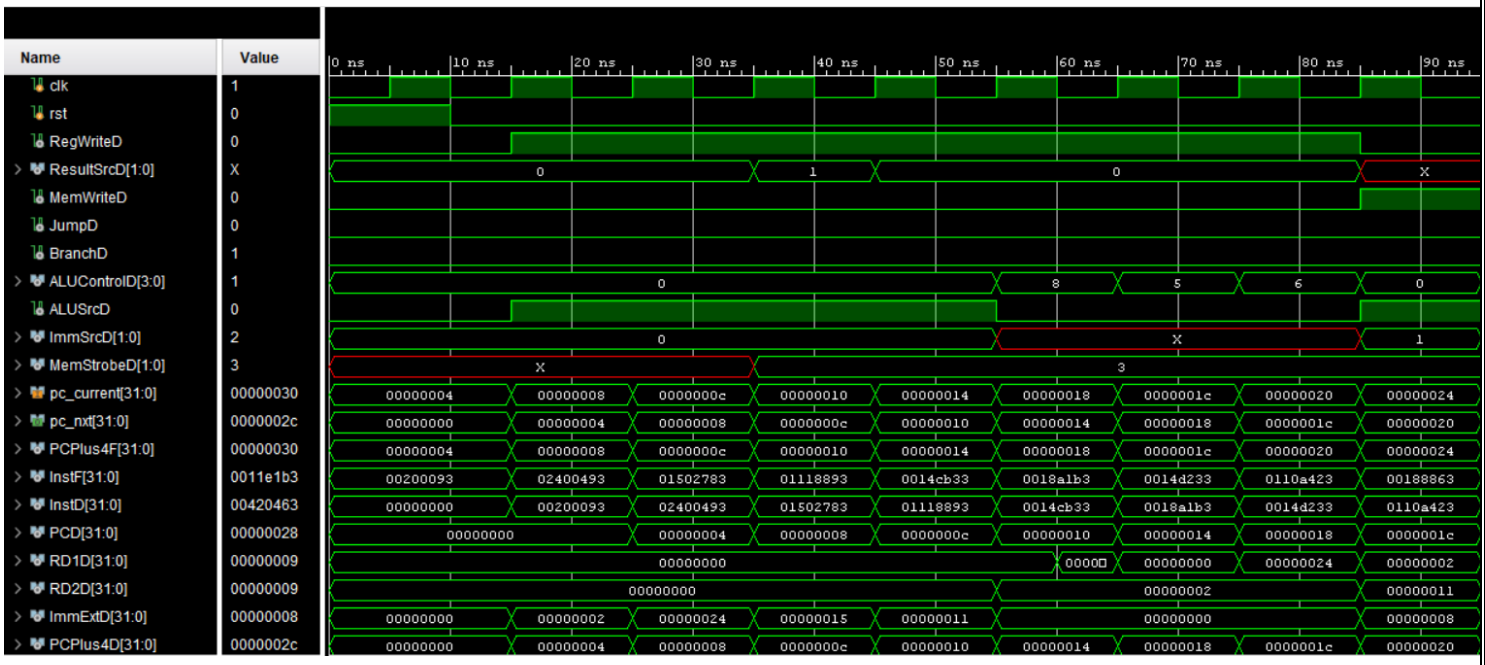
Program test for the design

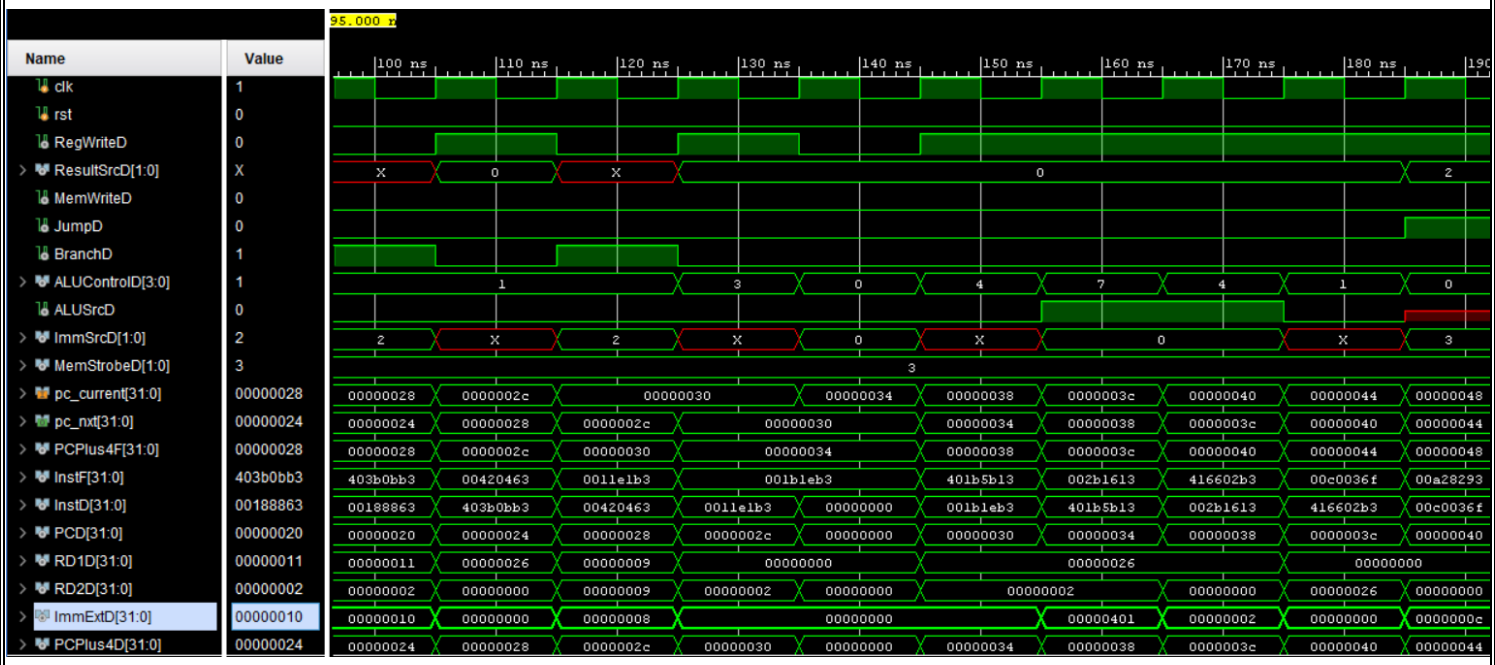
	Address	Assembly	Comment	Machine Code	
	0x0	addi x1,x0,2	x1=2	00200093	
	0x4	addi x9,x0,36	x9=36	02400493	
	0x8	lw x15,21(x0)	x15=DM[21]=124	01502783	
	0xC	addi x17,x3,17	x17=17	01118893	
	0x10	xor x22,x9,x1	x22=38	0014CB33	
	0x14	slt x3,x17,x1	x3=0	0018A1B3	
	0x18	sra x4,x9,x1	x4=9	0014D233	
	0x1C	sw x17,8(x1)	DM[10]=17	0110A423	
	0x20	beq x17,x1,branch	not executed	00188863	
	0x24	sub x23,x22,x3	x22=38	403B0BB3	
	0x28	beq,x4,x4,branch	executed	00420463	
	0x2C	or x3,x3,x1	not executed	0011E1B3	
branch:	0x30	sll x29,x22,x1	x29=152	001B1EB3	// The following 3 instruction to test data forwarding
	0x34	srai x22,x22,1	x22=19	401B5B13	
	0x38	slli x12,x22,2	x12=76	002B1613	
	0x3C	sub x5,x12,x22	x5=57	416602B3	
	0x40	jal x6,test1	x6=pc+4=44	00C0036F	
	0x44	addi x5,x5,10	not executed	00A28293	
	0x48	sub x5,x5,x22	not executed	016282B3	
test1:	0x4C	sltiu x25,x4,-6	x25=0	FFA23C93	
	0x50	beq x25,x0,test2	executed	000C8463	
	0x54	xor x17,x17,x9	not executed	0098C8B3	// The remaining instructions to test lb,lh,lw,sb,sh,sw
test2:	0x58	addi x16,x0,4076	x16=4076	FEC00813	
	0x5C	slli,x19,x16,5	x19=(1FD80)hex	00581993	
	0x60	sb x19,38(x0)	DM[38]=(80)hex	03300323	
	0x64	sh x19,39(x0)	DM[39]=(FD80)hex	033013A3	
	0x68	sw x19,40(x0)	DM[40]=(fffffd80)	03302423	
	0x6C	lb x24,40(x0)	x24=DM[40]=(80)hex	02800C03	
	0x70	lh x25,40(x0)	x25=(FD80)hex	02801C83	
	0x74	lw x26,40(x0)	x26(fffffd80)	02802D03	

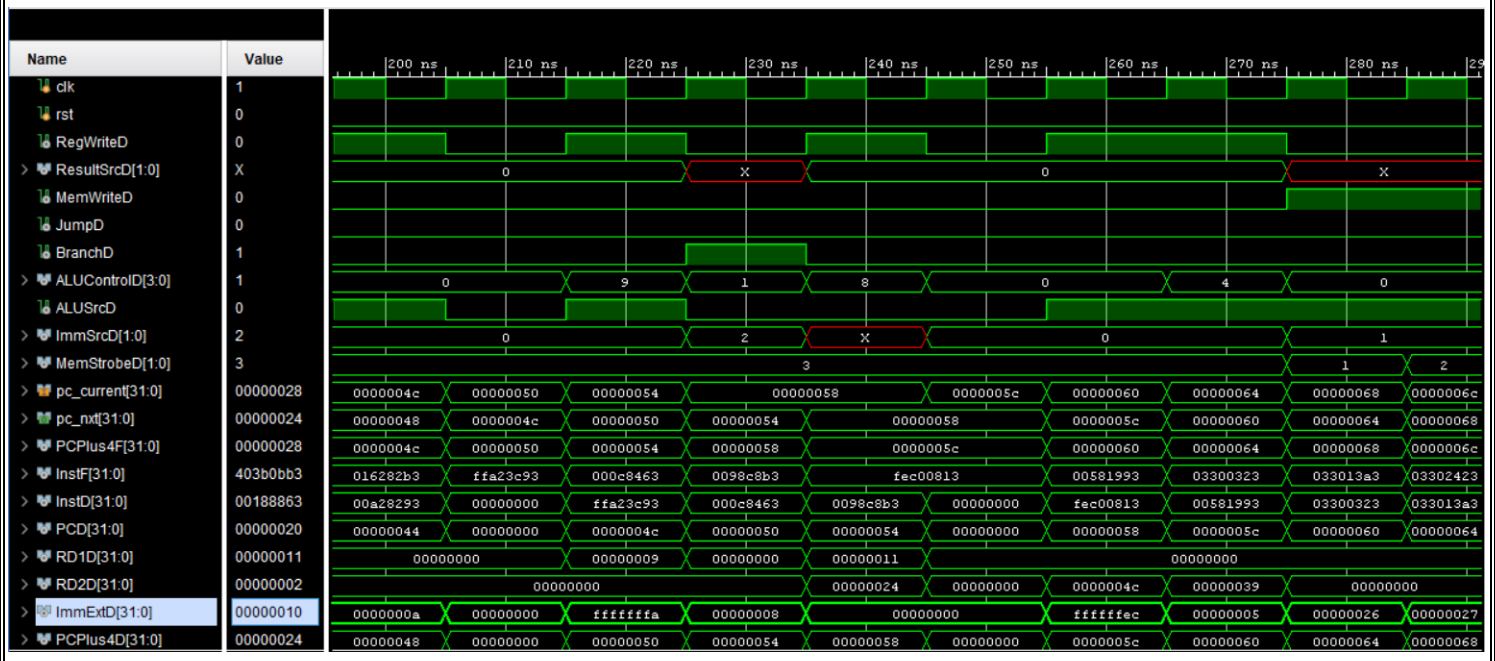
Note: The Machine Codes are stored in this order in the instruction memory.

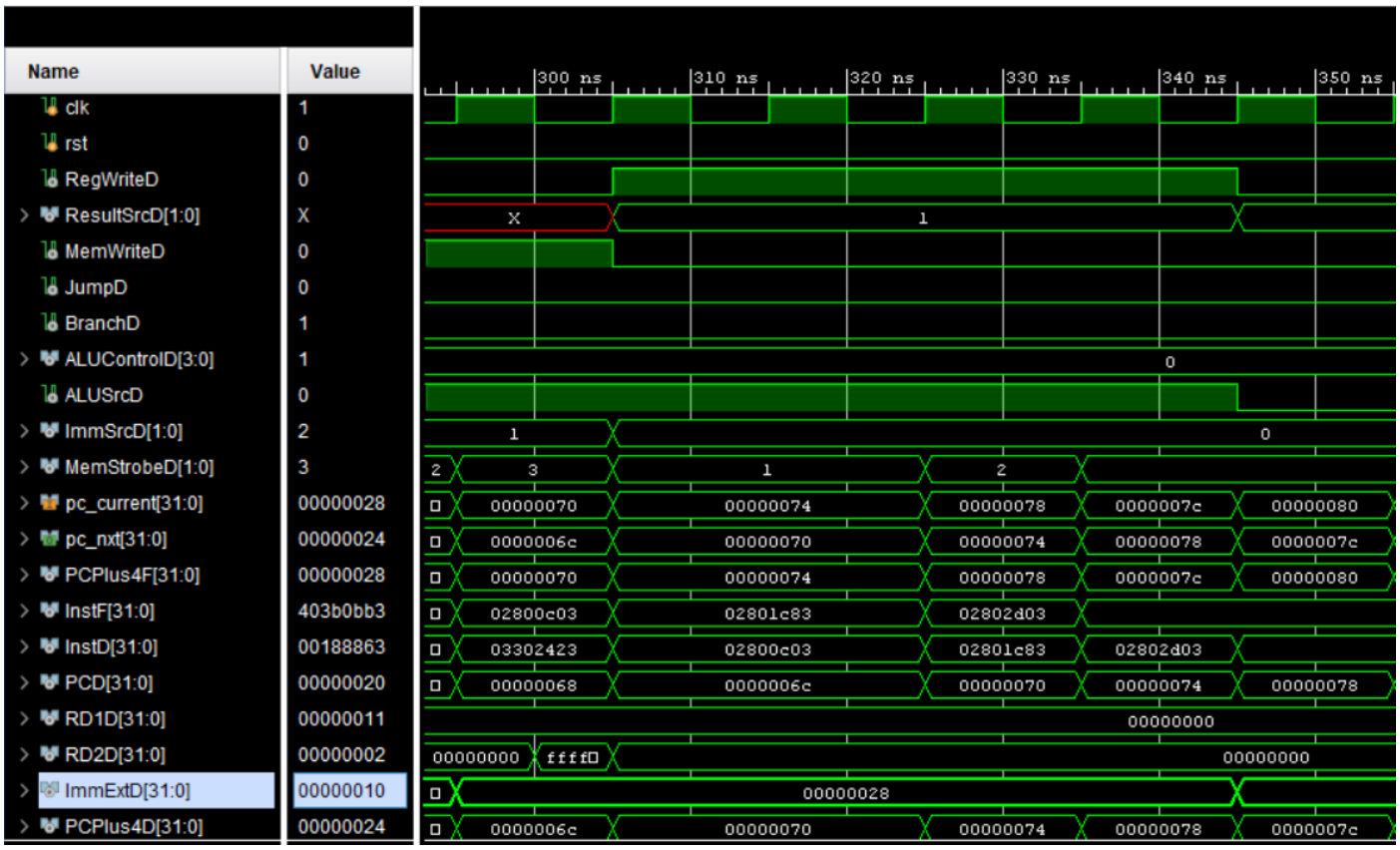
Simulation snippets

A) D-Stage:

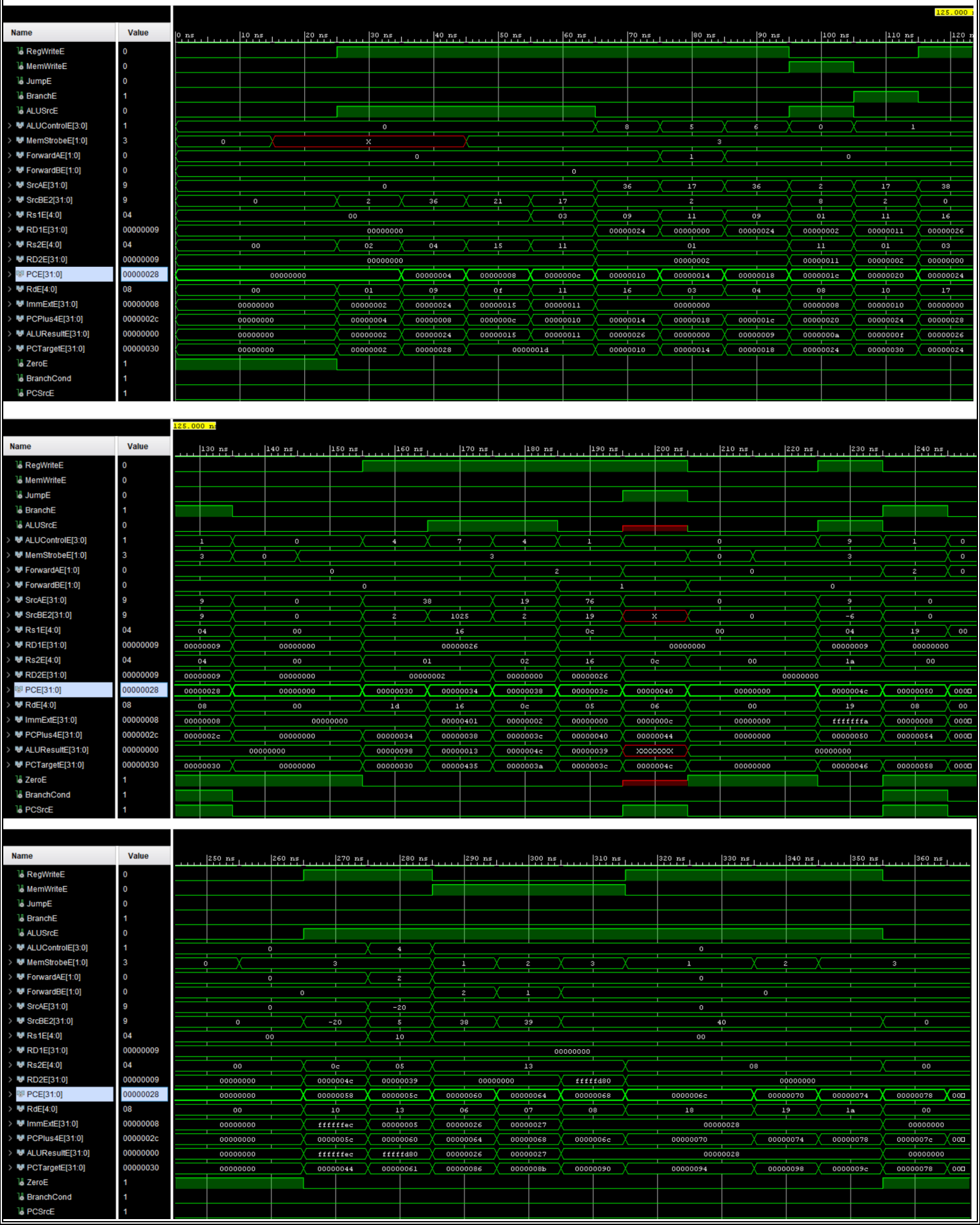




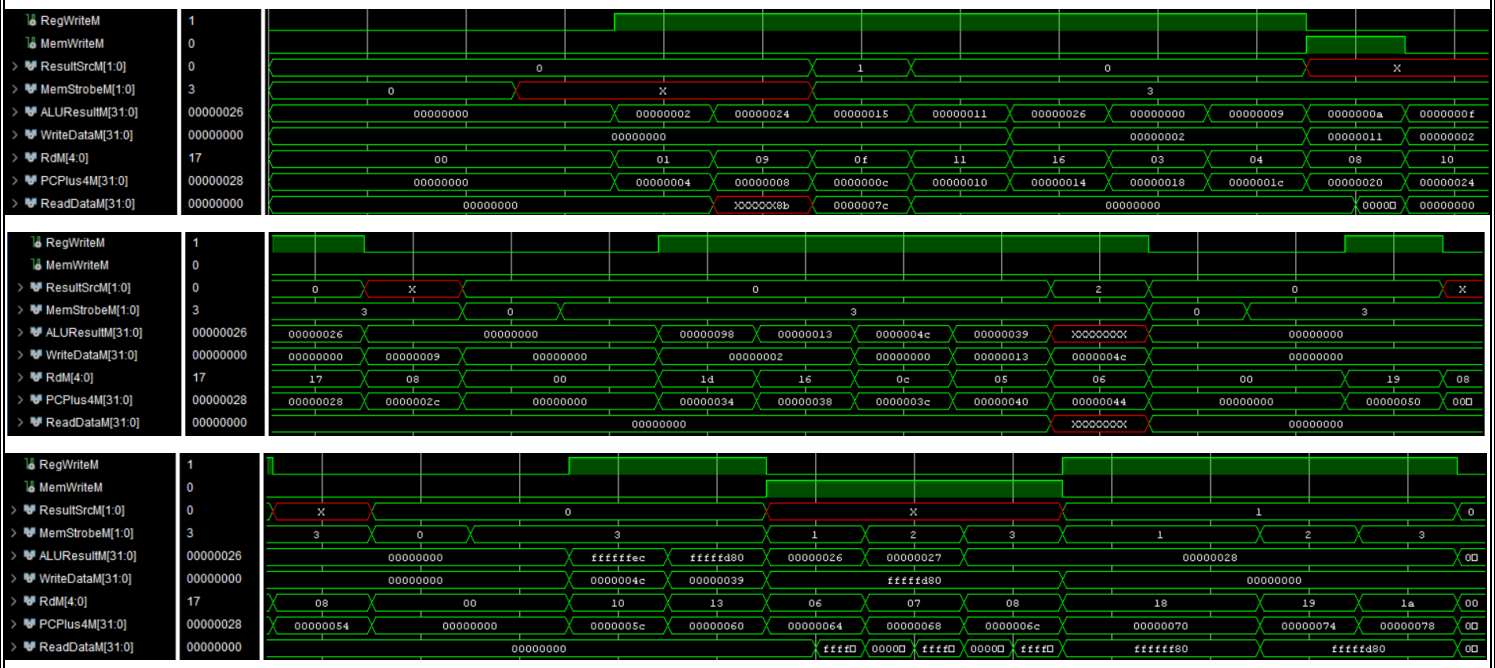

Name
Value
clk 1
rst 0
RegWriteD 0
ResultSrcD[1:0] X
MemWriteD 0
JumpD 0
BranchD 1
ALUControlD[3:0] 1
ALUSrcD 0
ImmSrcD[1:0] 2
MemStrobeD[1:0] 3
pc_current[31:0] 00000028
pc_nxt[31:0] 00000024
PCPlus4F[31:0] 00000028
InstF[31:0] 403b0bb3
InstD[31:0] 00188863
PCD[31:0] 00000020
RD1D[31:0] 00000011
RD2D[31:0] 00000002
ImmExtD[31:0] 00000010
PCPlus4D[31:0] 00000024
200 ns
210 ns
220 ns
230 ns
240 ns
250 ns
260 ns
270 ns
280 ns


Name
Value
clk 1
rst 0
RegWriteD 0
ResultSrcD[1:0] X
MemWriteD 0
JumpD 0
BranchD 1
ALUControlD[3:0] 1
ALUSrcD 0
ImmSrcD[1:0] 2
MemStrobeD[1:0] 3
pc_current[31:0] 00000028
pc_nxt[31:0] 00000024
PCPlus4F[31:0] 00000028
InstF[31:0] 403b0bb3
InstD[31:0] 00188863
PCD[31:0] 00000020
RD1D[31:0] 00000011
RD2D[31:0] 00000002
ImmExtD[31:0] 00000010
PCPlus4D[31:0] 00000024
300 ns
310 ns
320 ns
330 ns
340 ns
350 ns

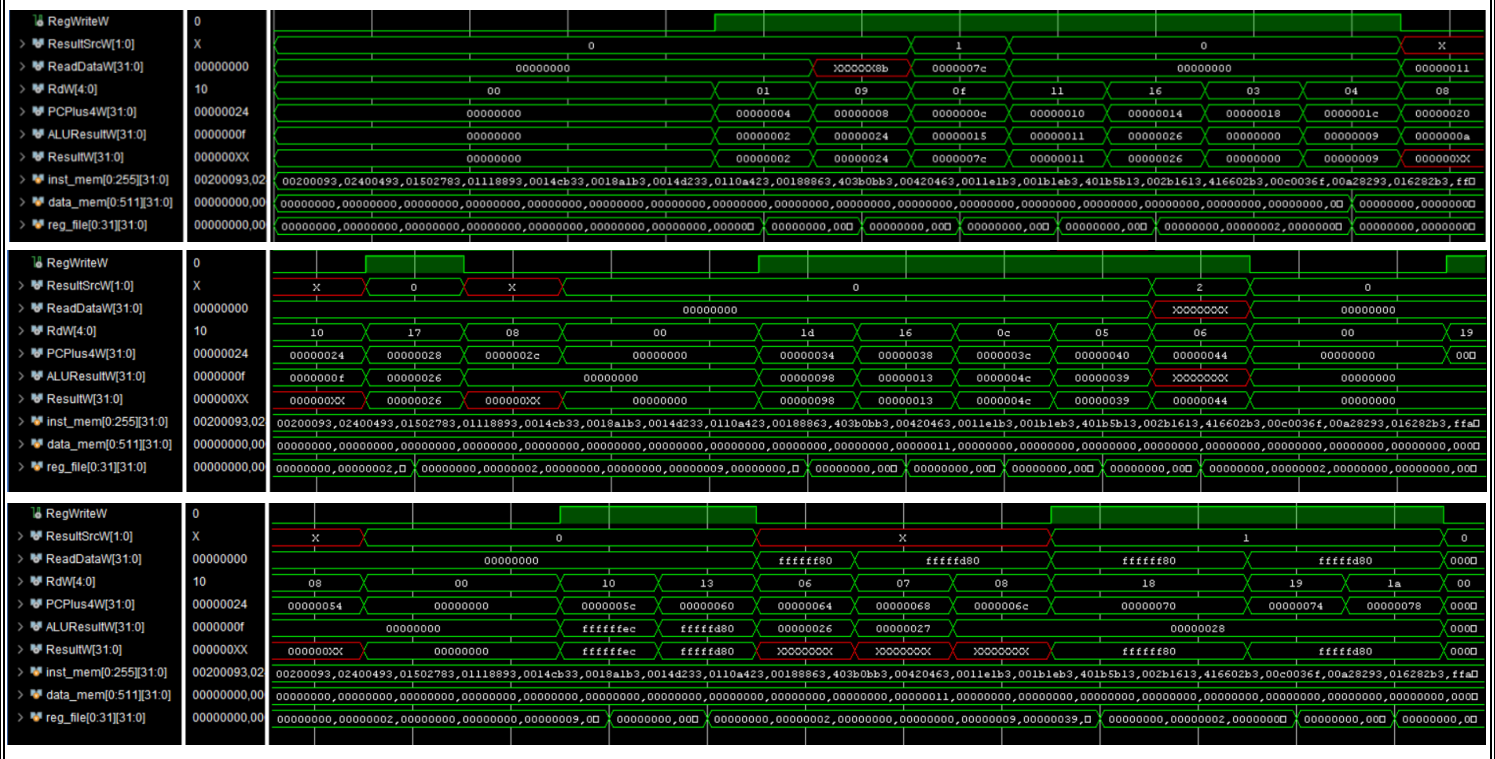
B) E-Stage:


Name
Value
RegWriteE
MemWriteE
JumpE
BranchE
ALUSrcE
ALUControlE[3:0]
MemStrobeE[1:0]
ForwardAE[1:0]
ForwardBE[1:0]
SrcAE[31:0]
SrcBE2[31:0]
Rs1E[4:0]
RD1E[31:0]
Rs2E[4:0]
RD2E[31:0]
PCE[31:0]
RdE[4:0]
ImmExtE[31:0]
PCPlus4E[31:0]
ALUResultE[31:0]
PCTargetE[31:0]
ZeroE
BranchCond
PCSrcE

C) M-Stage:



D) W-Stage:



Note: From The program test section, the column (Comment) contains the final stage of each instruction where can be seen from the ResultW stage and the memories to ease the verification of the correct execution of each instruction.

But the value in the comment column is in decimal while in ResultW is hexa.

Also the final schematic is attached as PDF file with the other files that was provided.