CSE 331/503

Computer Organization

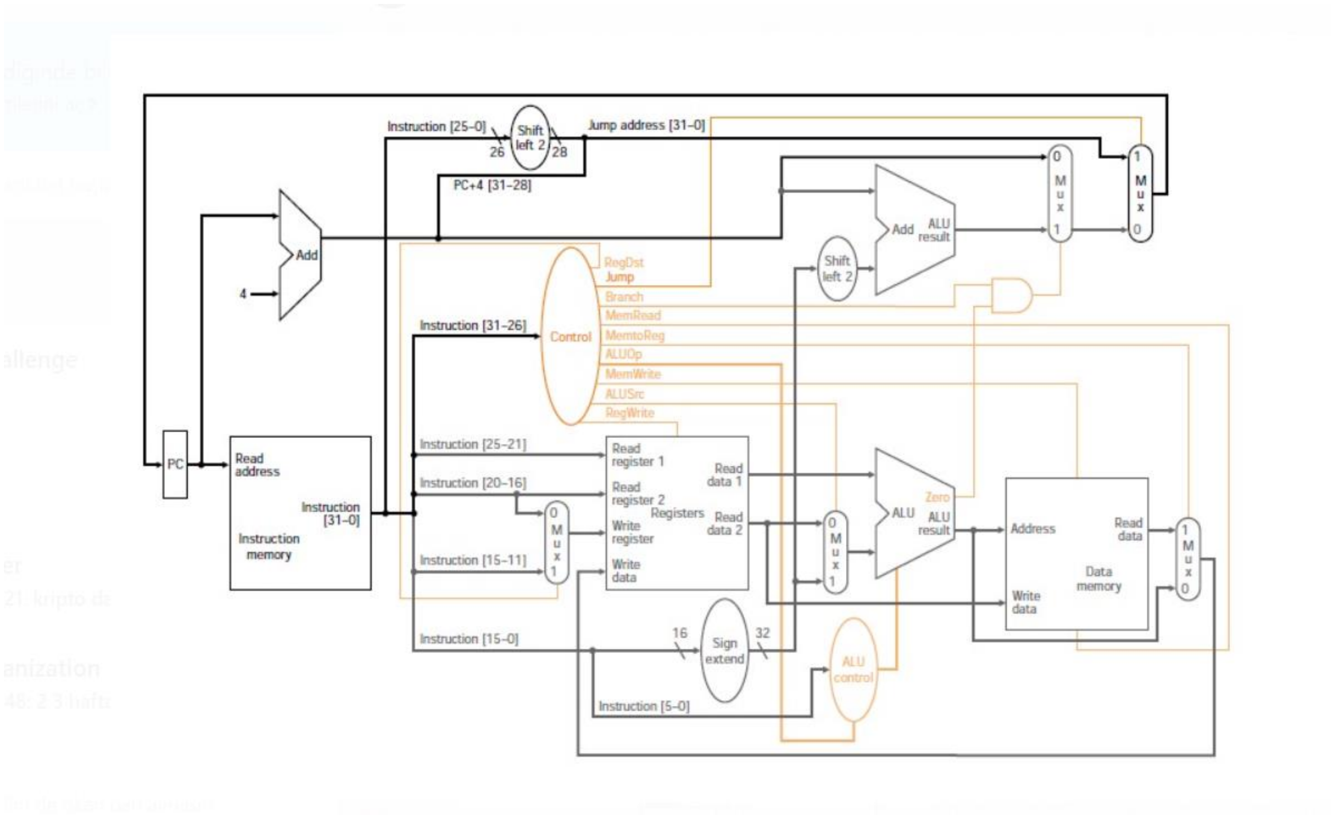
Final Project – MiniMIPS Design

Okan Torun

1801042662

NOTE : register_block,data_memory ve instruction_memory dosyalarındaki dosya uzantılarını kişisel bilgisayarıma göre ayarladım,o uzantıları koymadan çalışmıyordu.

Datapath



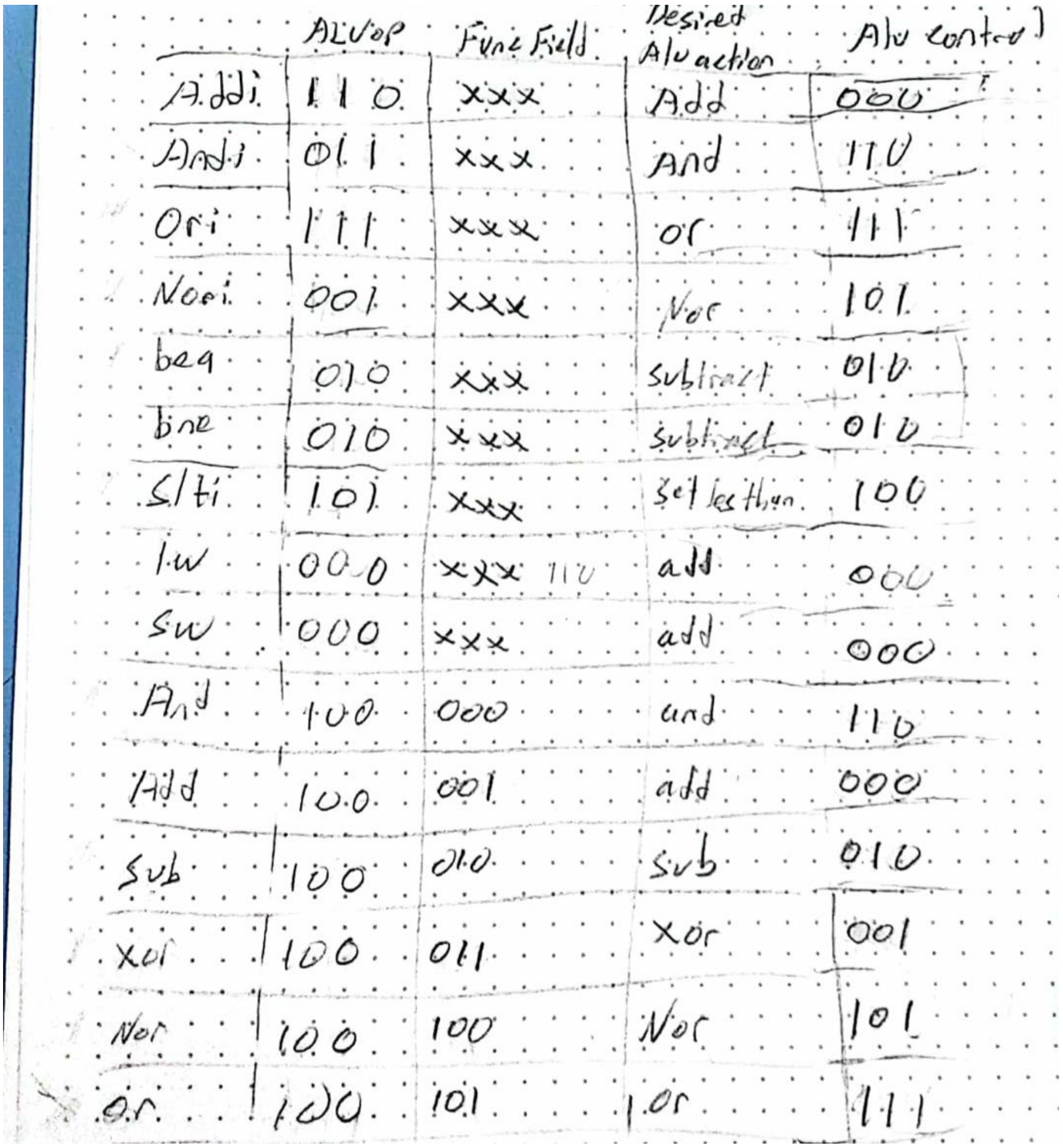
The intructions of MiniMIPS are given in below table:

Instr	Opcode	Func
AND	0000	000
ADD	0000	001
SUB	0000	010
XOR	0000	011
NOR	0000	100
OR	0000	101
ADDI	0001	XXX
ANDI	0010	XXX
ORI	0011	XXX
NORI	0100	XXX
BEQ	0101	XXX
BNE	0110	XXX
SLTI	0111	XXX
LW	1000	XXX
SW	1001	XXX

Instruction Memory

In the instruction memory module, the instruction in the instruction file is retrieved according to the value of the program counter.

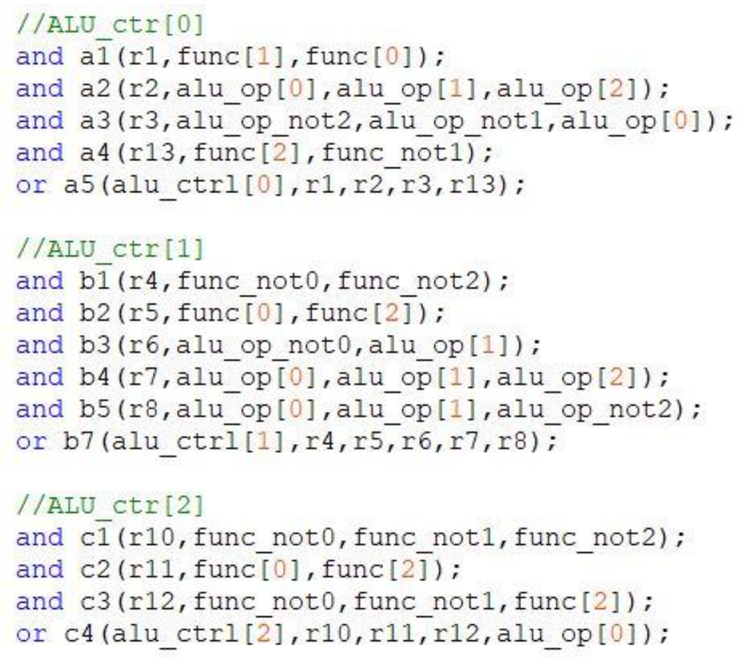
ALU Control Truth Table



	ALUop	Func Field	Desired Alu action	Alu control
Addi	110	xxx	Add	000
Andi	011	xxx	And	110
Ori	111	xxx	or	111
Nori	001	xxx	Nor	101
beq	010	xxx	subtract	010
bne	010	xxx	subtract	010
slti	101	xxx	set less than	100
lw	000	xxx 110	add	000
sw	000	xxx	add	000
And	100	000	and	110
Add	100	001	add	000
Sub	100	010	sub	010
xor	100	011	xor	001
Nor	100	100	Nor	101
or	100	101	or	111

Alu_src values were calculated according to boolean expression values.

Boolean expressions from table

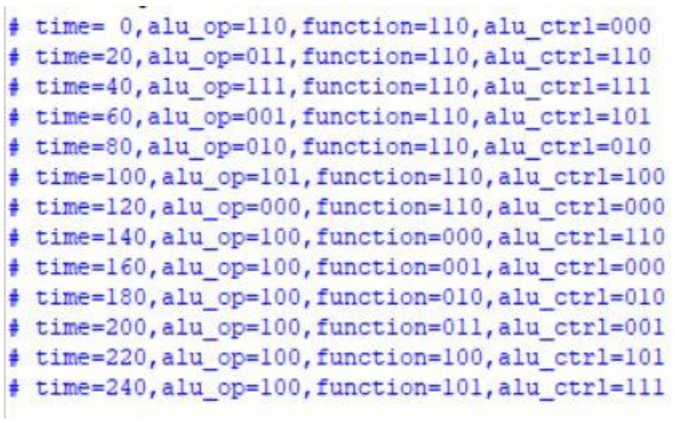


```
//ALU_ctr[0]
and a1(r1,func[1],func[0]);
and a2(r2,alu_op[0],alu_op[1],alu_op[2]);
and a3(r3,alu_op_not2,alu_op_not1,alu_op[0]);
and a4(r13,func[2],func_not1);
or a5(alu_ctrl[0],r1,r2,r3,r13);

//ALU_ctr[1]
and b1(r4,func_not0,func_not2);
and b2(r5,func[0],func[2]);
and b3(r6,alu_op_not0,alu_op[1]);
and b4(r7,alu_op[0],alu_op[1],alu_op[2]);
and b5(r8,alu_op[0],alu_op[1],alu_op_not2);
or b7(alu_ctrl[1],r4,r5,r6,r7,r8);

//ALU_ctr[2]
and c1(r10,func_not0,func_not1,func_not2);
and c2(r11,func[0],func[2]);
and c3(r12,func_not0,func_not1,func[2]);
or c4(alu_ctrl[2],r10,r11,r12,alu_op[0]);
```

ALU control test result



```
# time= 0,alu_op=110,function=110,alu_ctrl=000
# time=20,alu_op=011,function=110,alu_ctrl=110
# time=40,alu_op=111,function=110,alu_ctrl=111
# time=60,alu_op=001,function=110,alu_ctrl=101
# time=80,alu_op=010,function=110,alu_ctrl=010
# time=100,alu_op=101,function=110,alu_ctrl=100
# time=120,alu_op=000,function=110,alu_ctrl=000
# time=140,alu_op=100,function=000,alu_ctrl=110
# time=160,alu_op=100,function=001,alu_ctrl=000
# time=180,alu_op=100,function=010,alu_ctrl=010
# time=200,alu_op=100,function=011,alu_ctrl=001
# time=220,alu_op=100,function=100,alu_ctrl=101
# time=240,alu_op=100,function=101,alu_ctrl=111
```

1)Addi

2)Andi

3)Ori

4)Nori

5)Beq/Bne

6)Slti

7)lw/sw

8)And

9)Add

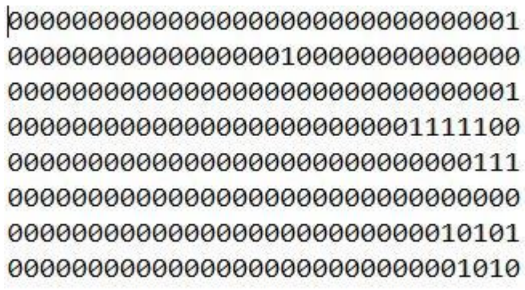
10)Sub

11)Xor

12)Nor

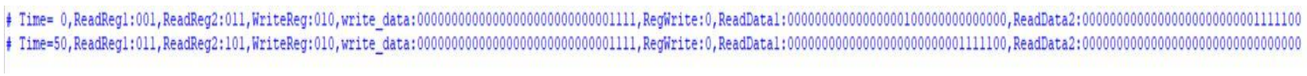
13)Or

Register Data File


```
00000000000000000000000000000001
00000000000000001000000000000000
00000000000000000000000000000001
00000000000000000000000001111100
00000000000000000000000000000111
00000000000000000000000000000000
00000000000000000000000000010101
00000000000000000000000000001010
```

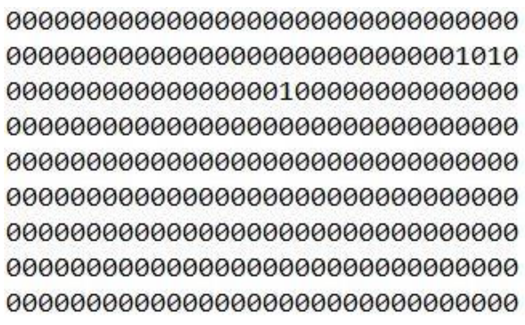
**Eight registers of 32 bits each are used.

Register Block Test Results

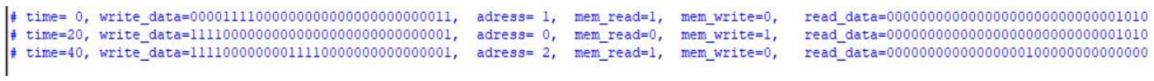

```
# Time= 0,ReadReg1:001,ReadReg2:011,WriteReg:010,write_data:00000000000000000000000000001111,RegWrite:0,ReadData1:00000000000000001000000000000000,ReadData2:00000000000000000000000001111100
# Time=50,ReadReg1:011,ReadReg2:101,WriteReg:010,write_data:00000000000000000000000000001111,RegWrite:0,ReadData1:00000000000000000000000001111100,ReadData2:00000000000000000000000000000000
```

**There are 256 memory blocks, each with a 32-bit address value in mem_memory file.

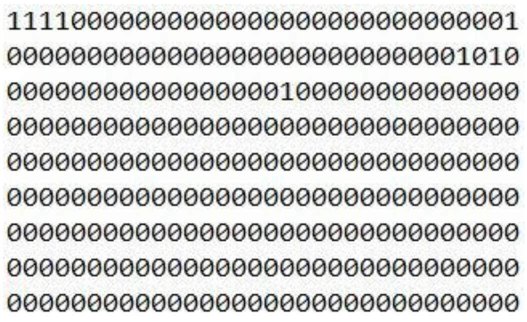
Data Memory File Before


```
00000000000000000000000000000000
00000000000000000000000000001010
00000000000000001000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

Data Memory Test Results


```
# time= 0, write_data=00001111000000000000000000000011,  adress= 1,  mem_read=1,  mem_write=0,   read_data=00000000000000000000000000001010
# time=20, write_data=11110000000000000000000000000001,  adress= 0,  mem_read=0,  mem_write=1,   read_data=00000000000000000000000000001010
# time=40, write_data=11110000000111100000000000000001,  adress= 2,  mem_read=1,  mem_write=0,   read_data=00000000000000001000000000000000
```

Data Memory File After

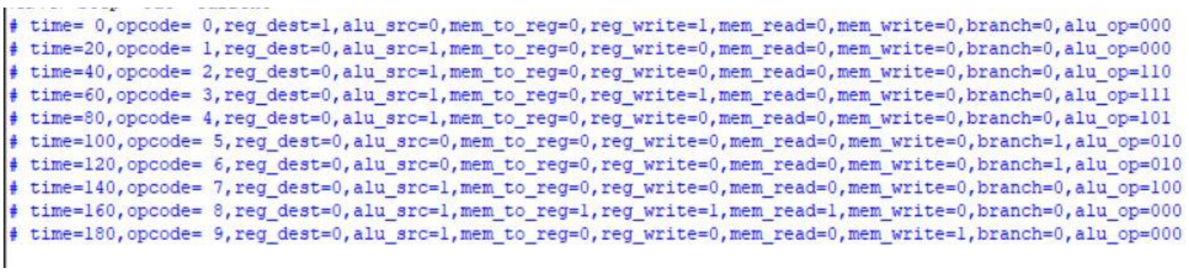


```
11110000000000000000000000000001
00000000000000000000000000001010
00000000000000000100000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

If the MemToReg signal is not active, the value from the ALU passes through the mux and is written to the register.

Control Unit Test Results

All signals are obtained with the op_code entering the control unit and the results from it**.**



```
# time= 0,opcode= 0,reg_dest=1,alu_src=0,mem_to_reg=0,reg_write=1,mem_read=0,mem_write=0,branch=0,alu_op=000
# time=20,opcode= 1,reg_dest=0,alu_src=1,mem_to_reg=0,reg_write=0,mem_read=0,mem_write=0,branch=0,alu_op=000
# time=40,opcode= 2,reg_dest=0,alu_src=1,mem_to_reg=0,reg_write=0,mem_read=0,mem_write=0,branch=0,alu_op=110
# time=60,opcode= 3,reg_dest=0,alu_src=1,mem_to_reg=0,reg_write=1,mem_read=0,mem_write=0,branch=0,alu_op=111
# time=80,opcode= 4,reg_dest=0,alu_src=1,mem_to_reg=0,reg_write=0,mem_read=0,mem_write=0,branch=0,alu_op=101
# time=100,opcode= 5,reg_dest=0,alu_src=0,mem_to_reg=0,reg_write=0,mem_read=0,mem_write=0,branch=1,alu_op=010
# time=120,opcode= 6,reg_dest=0,alu_src=0,mem_to_reg=0,reg_write=0,mem_read=0,mem_write=0,branch=1,alu_op=010
# time=140,opcode= 7,reg_dest=0,alu_src=1,mem_to_reg=0,reg_write=0,mem_read=0,mem_write=0,branch=0,alu_op=100
# time=160,opcode= 8,reg_dest=0,alu_src=1,mem_to_reg=1,reg_write=1,mem_read=1,mem_write=0,branch=0,alu_op=000
# time=180,opcode= 9,reg_dest=0,alu_src=1,mem_to_reg=0,reg_write=0,mem_read=0,mem_write=1,branch=0,alu_op=000
```

1)r-types

2)addi

3)andi

4)ori

5)nori

6)beq

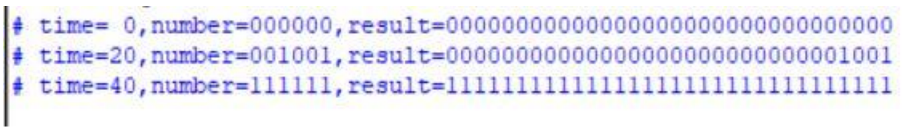
7)bne

8)slti

9)lw

10)sw

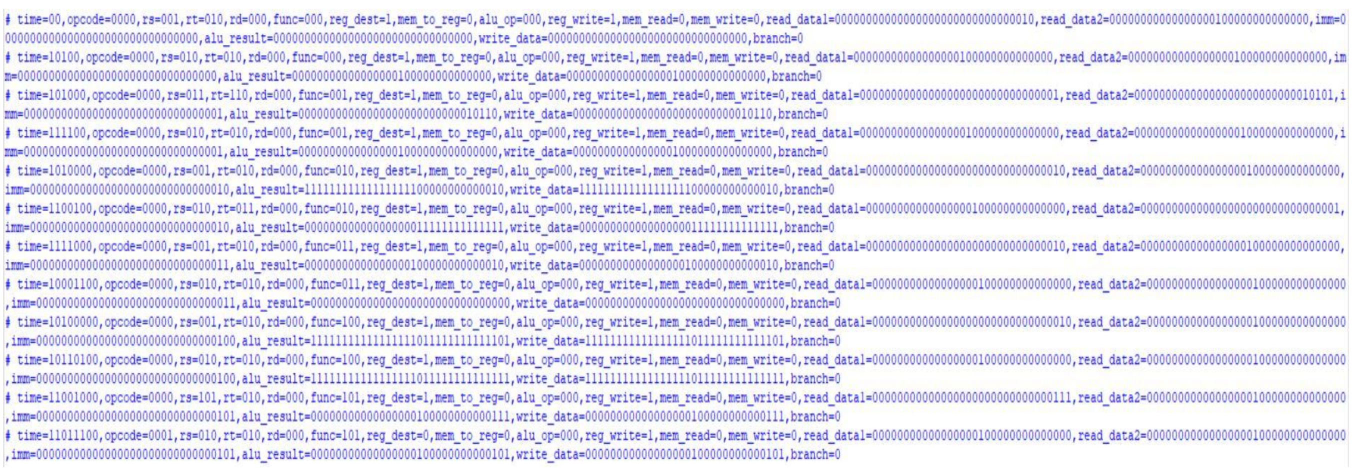
Sign Extender Test Results



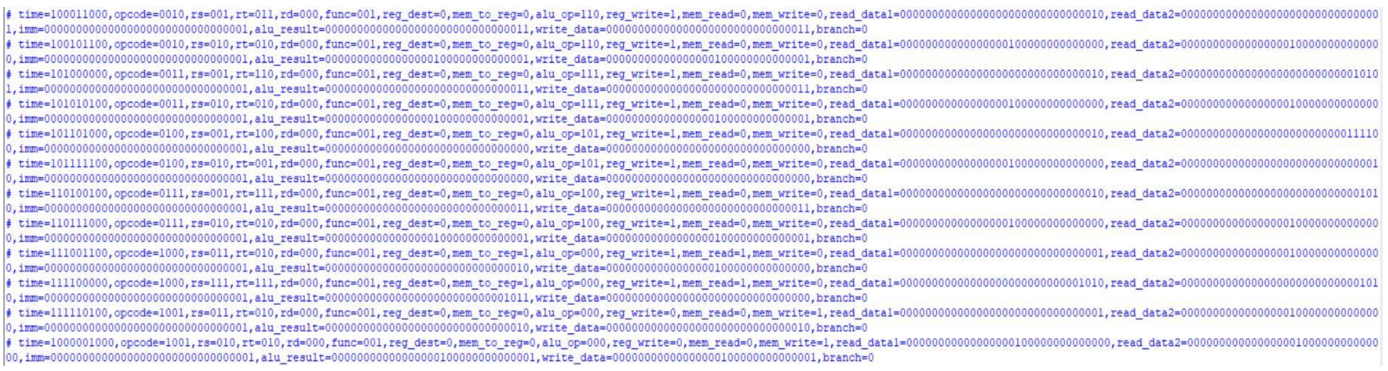
```
# time= 0,number=000000,result=00000000000000000000000000000000
# time=20,number=001001,result=00000000000000000000000000001001
# time=40,number=111111,result=11111111111111111111111111111111
```

MiniMIPS Test Results

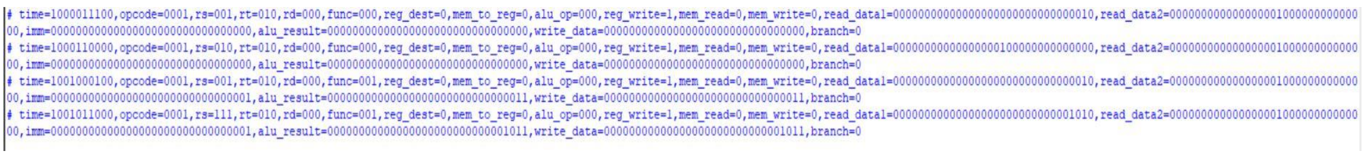
I processed all the instructions by increasing the program counter with 30 instructions in the instruction file.



```
# time=00,opcode=0000,rs=001,rt=010,rd=000,func=000,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=00000000000000001000000000000000,imm=0
0000000000000000000000000000000,alu_result=00000000000000000000000000000000,write_data=00000000000000000000000000000000,branch=0
# time=10100,opcode=0000,rs=010,rt=010,rd=000,func=000,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=00000000000000001000000000000000,im
m=00000000000000000000000000000000,alu_result=00000000000000001000000000000000,write_data=00000000000000001000000000000000,branch=0
# time=101000,opcode=0000,rs=011,rt=110,rd=000,func=001,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000001,read_data2=00000000000000000000000000010101,i
mm=00000000000000000000000000000001,alu_result=00000000000000000000000000010110,write_data=00000000000000000000000000010110,branch=0
# time=111100,opcode=0000,rs=010,rt=010,rd=000,func=001,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=00000000000000001000000000000000,i
mm=00000000000000000000000000000001,alu_result=00000000000000001000000000000000,write_data=00000000000000001000000000000000,branch=0
# time=1010000,opcode=0000,rs=001,rt=010,rd=000,func=010,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=00000000000000001000000000000000,
imm=00000000000000000000000000000010,alu_result=11111111111111111000000000000010,write_data=11111111111111111000000000000010,branch=0
# time=1100100,opcode=0000,rs=010,rt=011,rd=000,func=010,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=00000000000000000000000000000001,
imm=00000000000000000000000000000010,alu_result=00000000000000000111111111111111,write_data=00000000000000000111111111111111,branch=0
# time=1111000,opcode=0000,rs=001,rt=010,rd=000,func=011,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=00000000000000001000000000000000,
imm=00000000000000000000000000000011,alu_result=00000000000000001000000000000010,write_data=00000000000000001000000000000010,branch=0
# time=10001100,opcode=0000,rs=010,rt=010,rd=000,func=011,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=00000000000000001000000000000000
,imm=00000000000000000000000000000011,alu_result=00000000000000000000000000000000,write_data=00000000000000000000000000000000,branch=0
# time=10100000,opcode=0000,rs=001,rt=010,rd=000,func=100,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=00000000000000001000000000000000
,imm=00000000000000000000000000000100,alu_result=11111111111111110111111111111101,write_data=11111111111111110111111111111101,branch=0
# time=10110100,opcode=0000,rs=010,rt=010,rd=000,func=100,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=00000000000000001000000000000000
,imm=00000000000000000000000000000100,alu_result=11111111111111110111111111111111,write_data=11111111111111110111111111111111,branch=0
# time=11001000,opcode=0000,rs=101,rt=010,rd=000,func=101,reg_dest=1,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000111,read_data2=00000000000000001000000000000000
,imm=00000000000000000000000000000101,alu_result=00000000000000001000000000000111,write_data=00000000000000001000000000000111,branch=0
# time=11011100,opcode=0001,rs=010,rt=010,rd=000,func=101,reg_dest=0,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=00000000000000001000000000000000
,imm=00000000000000000000000000000101,alu_result=00000000000000001000000000000101,write_data=00000000000000001000000000000101,branch=0
```



```
# time=100011000,opcode=0010,rs=001,rt=011,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=110,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=0000000000000000000000000000000
1,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000000011,write_data=00000000000000000000000000000011,branch=0
# time=100101100,opcode=0010,rs=010,rt=010,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=110,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=0000000000000000100000000000000
0,imm=00000000000000000000000000000001,alu_result=00000000000000001000000000000001,write_data=00000000000000001000000000000001,branch=0
# time=101000000,opcode=0011,rs=001,rt=110,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=111,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=0000000000000000000000000001010
1,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000000011,write_data=00000000000000000000000000000011,branch=0
# time=101010100,opcode=0011,rs=010,rt=010,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=111,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=0000000000000000100000000000000
0,imm=00000000000000000000000000000001,alu_result=00000000000000001000000000000001,write_data=00000000000000001000000000000001,branch=0
# time=101101000,opcode=0100,rs=001,rt=100,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=101,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=0000000000000000000000000011110
0,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000000000,write_data=00000000000000000000000000000000,branch=0
# time=101111100,opcode=0100,rs=010,rt=001,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=101,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=0000000000000000000000000000001
0,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000000000,write_data=00000000000000000000000000000000,branch=0
# time=110100100,opcode=0111,rs=001,rt=111,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=100,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=0000000000000000000000000000101
0,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000000011,write_data=00000000000000000000000000000011,branch=0
# time=110111000,opcode=0111,rs=010,rt=010,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=100,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=0000000000000000100000000000000
0,imm=00000000000000000000000000000001,alu_result=00000000000000001000000000000001,write_data=00000000000000001000000000000001,branch=0
# time=111001100,opcode=1000,rs=011,rt=010,rd=000,func=001,reg_dest=0,mem_to_reg=1,alu_op=000,reg_write=1,mem_read=1,mem_write=0,read_data1=00000000000000000000000000000001,read_data2=0000000000000000100000000000000
0,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000000010,write_data=00000000000000001000000000000000,branch=0
# time=111100000,opcode=1000,rs=111,rt=111,rd=000,func=001,reg_dest=0,mem_to_reg=1,alu_op=000,reg_write=1,mem_read=1,mem_write=0,read_data1=00000000000000000000000000001010,read_data2=0000000000000000000000000000101
0,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000001011,write_data=00000000000000000000000000000000,branch=0
# time=111110100,opcode=1001,rs=011,rt=010,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=000,reg_write=0,mem_read=0,mem_write=1,read_data1=00000000000000000000000000000001,read_data2=0000000000000000100000000000000
0,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000000010,write_data=00000000000000000000000000000010,branch=0
# time=1000001000,opcode=1001,rs=010,rt=010,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=000,reg_write=0,mem_read=0,mem_write=1,read_data1=00000000000000001000000000000000,read_data2=000000000000000010000000000000
00,imm=00000000000000000000000000000001,alu_result=00000000000000001000000000000001,write_data=00000000000000001000000000000001,branch=0
```



```
# time=1000011100,opcode=0001,rs=001,rt=010,rd=000,func=000,reg_dest=0,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=000000000000000010000000000000
00,imm=00000000000000000000000000000000,alu_result=00000000000000000000000000000000,write_data=00000000000000000000000000000000,branch=0
# time=1000110000,opcode=0001,rs=010,rt=010,rd=000,func=000,reg_dest=0,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000001000000000000000,read_data2=000000000000000010000000000000
00,imm=00000000000000000000000000000000,alu_result=00000000000000000000000000000000,write_data=00000000000000000000000000000000,branch=0
# time=1001000100,opcode=0001,rs=001,rt=010,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000000010,read_data2=000000000000000010000000000000
00,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000000011,write_data=00000000000000000000000000000011,branch=0
# time=1001011000,opcode=0001,rs=111,rt=010,rd=000,func=001,reg_dest=0,mem_to_reg=0,alu_op=000,reg_write=1,mem_read=0,mem_write=0,read_data1=00000000000000000000000000001010,read_data2=000000000000000010000000000000
00,imm=00000000000000000000000000000001,alu_result=00000000000000000000000000001011,write_data=00000000000000000000000000001011,branch=0
```