HW4 REPORT

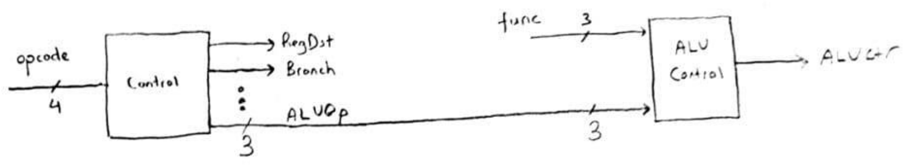
Ömer Faruk Bitikçioğlu - 161044010

I have already implemented ALU in the previous homework. In this homework I implemented the other components. Main control, ALU control and sign extend works correctly. But I am not sure about the register and data parts. When I connect all the components it didn't read instructions and compute anything. Unfortunately, I didn't have time to fix this issue.

I add the designs and calculations below. Also, you can check all my Verilog codes by using the .qar file.

Main Control

Main Control



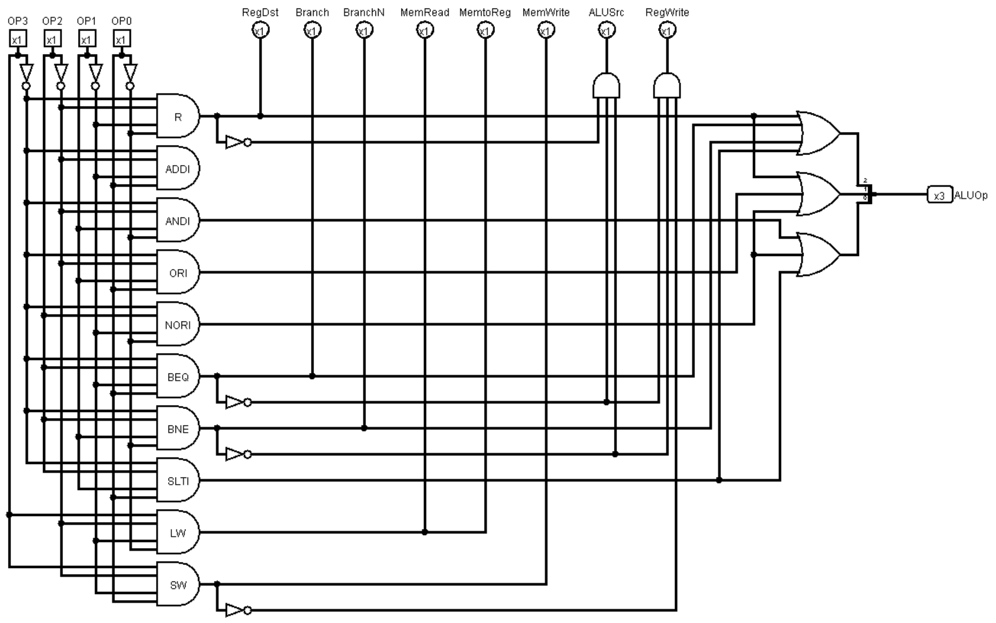
opcode	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
	R-type	addi	andi	ori	nori	beq	bne	slti	lw	sw
RegDst	1	0	0	0	0	X	X	0	0	X
Branch/N	0	0	0	0	0	1/0	0/1	0	0	0
MemRead	0	0	0	0	0	0	0	0	1	0
MemtoReg	0	0	0	0	0	X	X	0	1	X
MemWrite	0	0	0	0	0	0	0	0	0	1
ALUSrc	0	1	1	1	1	0	0	1	1	1
RegWrite	1	1	1	1	1	0	0	1	1	0
ALUOp[2]	1	0	0	0	0	1	1	1	0	0
ALUOp[1]	1	0	0	1	1	0	0	0	0	0
ALUOp[0]	0	0	1	0	1	0	0	1	0	0

RegDst = R
Branch = BEQ
BranchN = BNE

MemRead = LW
MemtoReg = LW
MemWrite = SW

ALUSrc = R' BEQ' BNE'
RegWrite = BEQ' BNE' SW'

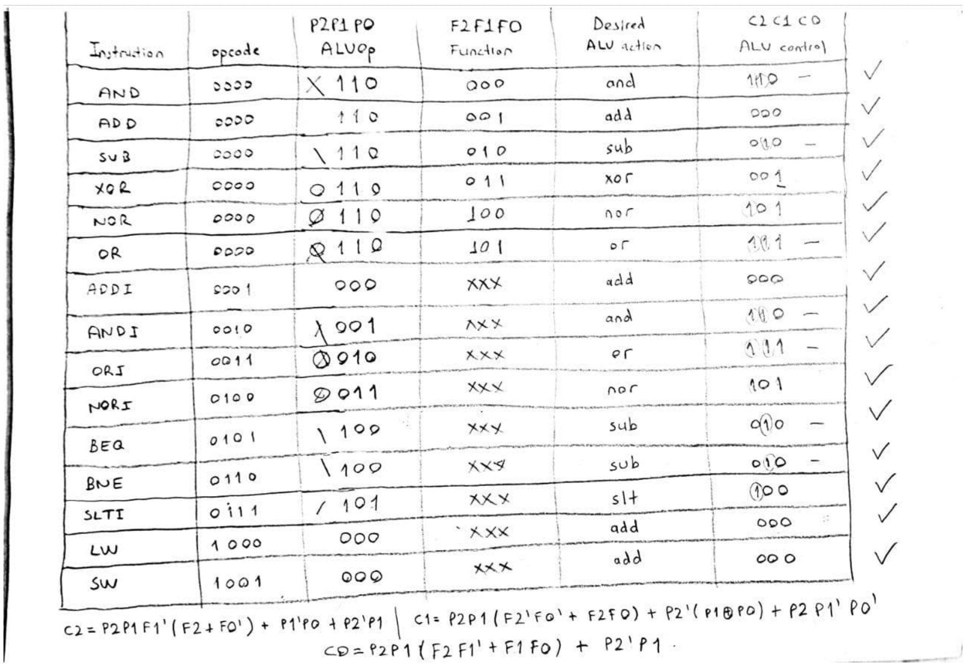
ALUOp(2) = R+BEQ+BNE+SLTI
ALUOp(1) = R+ORI+NORI
ALUOp(0) = ANDI+NORI+SLTI



Control Unit Testbench

```
# time =   0, opcode= 0000, RegDst=1, Branch=0, BranchN=0, MemRead=0, MemtoReg=0, MemWrite=0, ALUSrc=0, RegWrite=1, ALUOp= 110
# time = 20, opcode= 0001, RegDst=0, Branch=0, BranchN=0, MemRead=0, MemtoReg=0, MemWrite=0, ALUSrc=1, RegWrite=1, ALUOp= 000
# time = 40, opcode= 0010, RegDst=0, Branch=0, BranchN=0, MemRead=0, MemtoReg=0, MemWrite=0, ALUSrc=1, RegWrite=1, ALUOp= 001
# time = 60, opcode= 0011, RegDst=0, Branch=0, BranchN=0, MemRead=0, MemtoReg=0, MemWrite=0, ALUSrc=1, RegWrite=1, ALUOp= 010
# time = 80, opcode= 0100, RegDst=0, Branch=0, BranchN=0, MemRead=0, MemtoReg=0, MemWrite=0, ALUSrc=1, RegWrite=1, ALUOp= 011
# time = 100, opcode= 0101, RegDst=0, Branch=1, BranchN=0, MemRead=0, MemtoReg=0, MemWrite=0, ALUSrc=0, RegWrite=0, ALUOp= 100
# time = 120, opcode= 0110, RegDst=0, Branch=0, BranchN=1, MemRead=0, MemtoReg=0, MemWrite=0, ALUSrc=0, RegWrite=0, ALUOp= 100
# time = 140, opcode= 0111, RegDst=0, Branch=0, BranchN=0, MemRead=0, MemtoReg=0, MemWrite=0, ALUSrc=1, RegWrite=1, ALUOp= 101
# time = 160, opcode= 1000, RegDst=0, Branch=0, BranchN=0, MemRead=1, MemtoReg=1, MemWrite=0, ALUSrc=1, RegWrite=1, ALUOp= 000
# time = 180, opcode= 1001, RegDst=0, Branch=0, BranchN=0, MemRead=0, MemtoReg=0, MemWrite=1, ALUSrc=1, RegWrite=0, ALUOp= 000
```

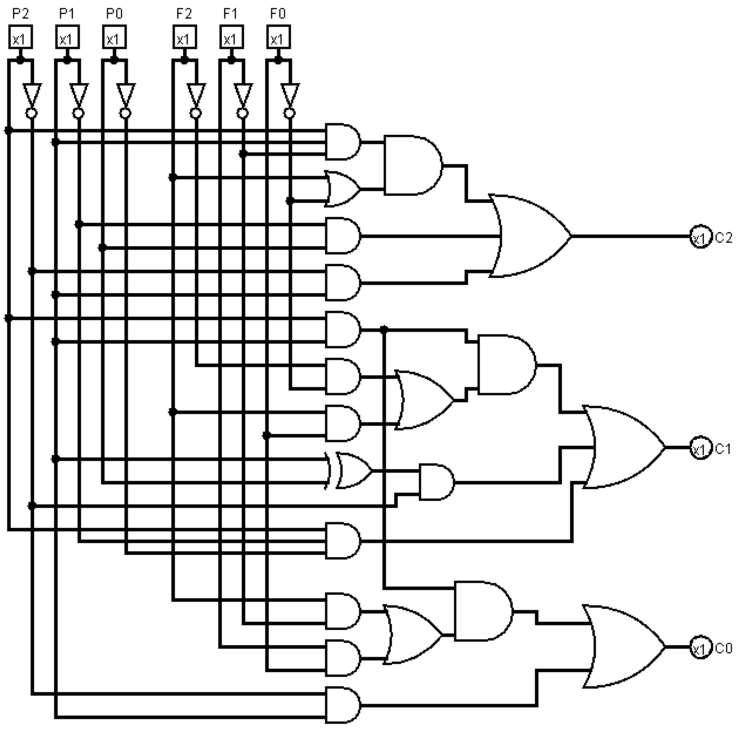
ALU Control



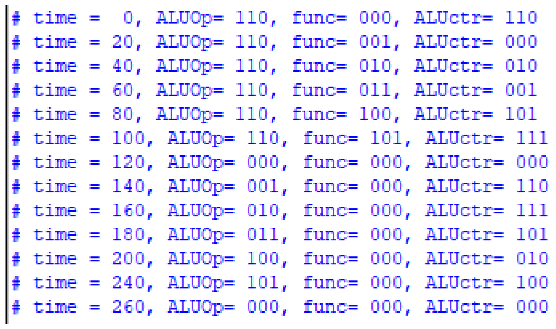
Instruction	opcode	P2P1P0 ALUOp	F2F1F0 Function	Desired ALU action	C2 C1 C0 ALU control	
AND	0000	110	000	and	110 –	✓
ADD	0000	110	001	add	000	✓
SUB	0000	110	010	sub	010 –	✓
XOR	0000	110	011	xor	001	✓
NOR	0000	110	100	nor	101	✓
OR	0000	110	101	or	111 –	✓
ADDI	0001	000	XXX	add	000	✓
ANDI	0010	001	XXX	and	110 –	✓
ORI	0011	010	XXX	or	111 –	✓
NORI	0100	011	XXX	nor	101	✓
BEQ	0101	100	XXX	sub	010 –	✓
BNE	0110	100	XXX	sub	010 –	✓
SLTI	0111	101	XXX	slt	100	✓
LW	1000	000	XXX	add	000	✓
SW	1001	000	XXX	add	000	✓

$C2 = P2P1F1'(F2+F0') + P1'P0 + P2'P1$ | $C1 = P2P1(F2'F0' + F2F0) + P2'(P1 \oplus P0) + P2P1'P0'$

$C0 = P2P1(F2F1' + F1F0) + P2'P1$.

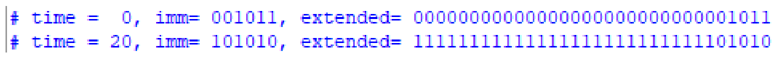


ALU control testbench



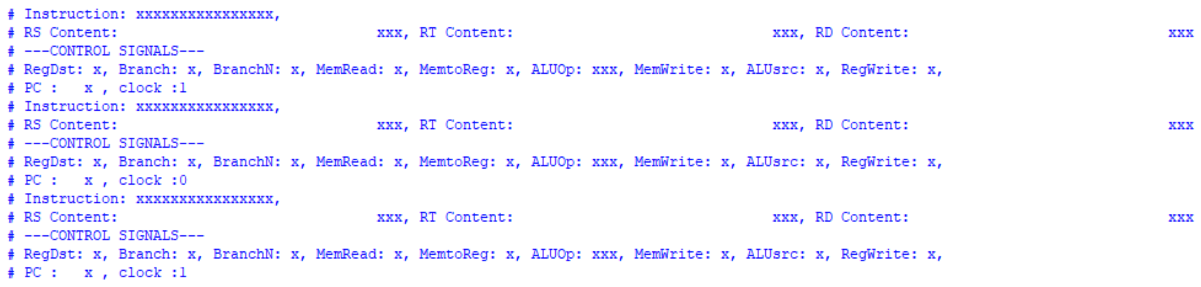
```
# time =  0, ALUOp= 110, func= 000, ALUctr= 110
# time = 20, ALUOp= 110, func= 001, ALUctr= 000
# time = 40, ALUOp= 110, func= 010, ALUctr= 010
# time = 60, ALUOp= 110, func= 011, ALUctr= 001
# time = 80, ALUOp= 110, func= 100, ALUctr= 101
# time = 100, ALUOp= 110, func= 101, ALUctr= 111
# time = 120, ALUOp= 000, func= 000, ALUctr= 000
# time = 140, ALUOp= 001, func= 000, ALUctr= 110
# time = 160, ALUOp= 010, func= 000, ALUctr= 111
# time = 180, ALUOp= 011, func= 000, ALUctr= 101
# time = 200, ALUOp= 100, func= 000, ALUctr= 010
# time = 240, ALUOp= 101, func= 000, ALUctr= 100
# time = 260, ALUOp= 000, func= 000, ALUctr= 000
```

Sign Extend Testbench



```
# time =  0, imm= 001011, extended= 00000000000000000000000000001011
# time = 20, imm= 101010, extended= 11111111111111111111111111101010
```

Testbench of the broken processor:



```
# Instruction: xxxxxxxxxxxxxxxx,
# RS Content:                              xxx, RT Content:                              xxx, RD Content:                              xxx
# ---CONTROL SIGNALS---
# RegDst: x, Branch: x, BranchN: x, MemRead: x, MemtoReg: x, ALUOp: xxx, MemWrite: x, ALUsrc: x, RegWrite: x,
# PC :   x , clock :1
# Instruction: xxxxxxxxxxxxxxxx,
# RS Content:                              xxx, RT Content:                              xxx, RD Content:                              xxx
# ---CONTROL SIGNALS---
# RegDst: x, Branch: x, BranchN: x, MemRead: x, MemtoReg: x, ALUOp: xxx, MemWrite: x, ALUsrc: x, RegWrite: x,
# PC :   x , clock :0
# Instruction: xxxxxxxxxxxxxxxx,
# RS Content:                              xxx, RT Content:                              xxx, RD Content:                              xxx
# ---CONTROL SIGNALS---
# RegDst: x, Branch: x, BranchN: x, MemRead: x, MemtoReg: x, ALUOp: xxx, MemWrite: x, ALUsrc: x, RegWrite: x,
# PC :   x , clock :1
```