Gebze Technical University
Computer Engineering

CSE 331
HOMEWORK 04

Derviş Ali DUMAN

1801042626

8 Register 32 bit

Instructions are 16 bit → R type and I type

R → Op (4 bit) Rs(3) Rt(3) Rd(3) Func(3)

I → Op (4) Rs(3) Rt(3) Imm(6)

Immediate extend to 32 bit (16→32)

Zero register always zero

	and	000
	add	001
	sub	010
	xor	011
	nor	100
	or	101
	slt	110

	INST	Opcode	Func	ALU Action	ALUCTR	ALU OP	
	AND	0000	000	and	000	000	000
	ADD	0000	001	add	001	000	001
	SUB	0000	010	sub	010	000	010
	XOR	0000	011	xor	011	000	011
	NOR	0000	100	nor	100	000	100
	OR	0000	101	or	101	000	101
I	ADDI	0001	xxx	add	001	001	X
I	ANDI	0010	xxx	and	000	010	
I	ORI	0011	xxx	or	101	101	X
I	NORI	0100	xxx	nor	100	100	
I	BEQ	0101	xxx	X sub	010	110	X
I	BNE	0110	xxx	X sub	010	110	
I	SLTI	0111	xxx	sub - slt	110	111	X
I	LW	1000	xxx	add	001	001	X
I	SW	1001	xxx	add	001	001	

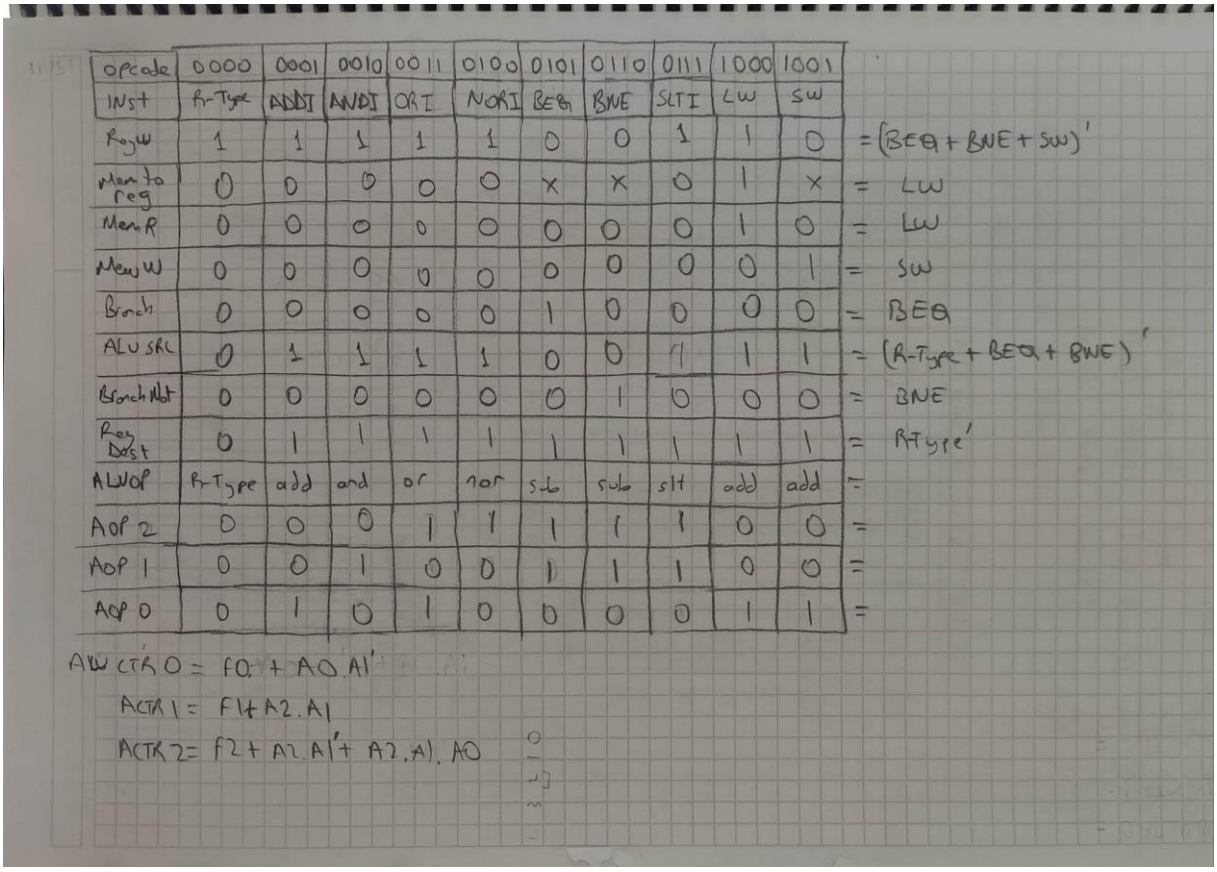
P3 P2 P1 P0 F2 F1 F0 — A2 A1 A0

ALUOp0 = P3 + P3'.P2'.P1'.P0 + .P1.P0

ALUOp1 = P2.P1 + P2.P1'.P0 + P2'.P1.P0'

ALUP02 = P3'.P2 + P2'.P1.P0

	Reg write	Mem to Reg	Mem Read	Mem write	Branch	ALU SRC	ALUOP	Branch not	Reg Dest
RTypes	1	0	0	0	0	0	000	0	0
ADDI	1	0	0	0	0	1	001	0	
ANDI	1	0	0	0	0	1	010	0	1
ORI	1	0	0	0	0	1	101	0	
NORI	1	0	0	0	0	1	100	0	1
BEQ	0	x	0	0	1	0	110	0	
BNE	0	x	0	0	0	0	110	1	1
SLTI	1	0	0	0	0	1	110	0	
LW	1	1	1	0	0	1	001	0	1
SW	0	x	0	1	0	1	001	0	



Opcode	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	
Inst	R-Type	ADDI	ANDI	ORI	NORI	BEQ	BNE	SLTI	LW	SW	
RegW	1	1	1	1	1	0	0	1	1	0	= (BEQ + BNE + SW)'
Mem to reg	0	0	0	0	0	x	x	0	1	x	= LW
MemR	0	0	0	0	0	0	0	0	1	0	= LW
MemW	0	0	0	0	0	0	0	0	0	1	= SW
Branch	0	0	0	0	0	1	0	0	0	0	= BEQ
ALUSRC	0	1	1	1	1	0	0	1	1	1	= (R-Type + BEQ + BNE)'
BranchNot	0	0	0	0	0	0	1	0	0	0	= BNE
Reg Dst	0	1	1	1	1	1	1	1	1	1	= RType'
ALUop	R-Type	add	and	or	nor	sub	sub	slt	add	add	=
Aop 2	0	0	0	1	1	1	1	1	0	0	=
Aop 1	0	0	1	0	0	1	1	1	0	0	=
Aop 0	0	1	0	1	0	0	0	0	1	1	=

ALU CTR 0 = F0 + A0.A1'

ACTR 1 = F1 + A2.A1

ACTR 2 = F2 + A2.A1' + A2.A1.A0

Extender Test:

```
VSIM 45> step -current
# Time: 0 i:11111, out:00000000000000000000000000011111
# Time:100 i:10111, out:00000000000000000000000000010111
```

Shift Test:

```
# Loading work.shiftleft2_test
# Loading work.shiftleft2
VSIM 41> step -current
# Time: 0 i:00000000000000000000000001111111, out: 00000000000000000000000111111100
```

Other tests made in ALU Test:

```
VSIM 52> step -current
# AND :
# Time: 0 il:00000000000000000000000000000101, i2:00000000000000000000000000000101, aluctr: 000, out:00000000000000000000000000000101, zero:1
# ADD :
# Time:100 il:00000000000000000000000000000101, i2:00000000000000000000000000000101, aluctr: 001, out:00000000000000000000000000001010, zero:1
# SUB :
# Time:200 il:00000000000000000000000000000101, i2:00000000000000000000000000000101, aluctr: 010, out:00000000000000000000000000000000, zero:1
# XOR :
# Time:300 il:00000000000000000000000000000101, i2:00000000000000000000000000000101, aluctr: 011, out:00000000000000000000000000000000, zero:1
# NOR :
# Time:400 il:00000000000000000000000000000101, i2:00000000000000000000000000000101, aluctr: 100, out:11111111111111111111111111111010, zero:1
# OR :
# Time:500 il:00000000000000000000000000000101, i2:00000000000000000000000000000101, aluctr: 101, out:00000000000000000000000000000101, zero:1
# SLT :
# Time:600 il:00000000000000000000000000000101, i2:00000000000000000000000000000101, aluctr: 110, out:00000000000000000000000000000000, zero:1
#
# New Set
#
# AND :
# Time:700 il:00000000000000000000000000001010, i2:00000000000000000000000000000101, aluctr: 000, out:00000000000000000000000000000000, zero:0
# ADD :
# Time:800 il:00000000000000000000000000001010, i2:00000000000000000000000000000101, aluctr: 001, out:00000000000000000000000000001111, zero:0
# SUB :
# Time:900 il:00000000000000000000000000001010, i2:00000000000000000000000000000101, aluctr: 010, out:00000000000000000000000000000101, zero:0
# XOR :
# Time:1000 il:00000000000000000000000000001010, i2:00000000000000000000000000000101, aluctr: 011, out:00000000000000000000000000001111, zero:0
# NOR :
# Time:1100 il:00000000000000000000000000001010, i2:00000000000000000000000000000101, aluctr: 100, out:11111111111111111111111111110000, zero:0
# OR :
# Time:1200 il:00000000000000000000000000001010, i2:00000000000000000000000000000101, aluctr: 101, out:00000000000000000000000000001111, zero:0
# SLT :
# Time:1400 il:00000000000000000000000000001010, i2:00000000000000000000000000000101, aluctr: 110, out:00000000000000000000000000000001, zero:0
```

```
# NewSet
#
# AND :
# Time:1500 il:00000000000000000000000000000101, i2:00000000000000000000000000001010, aluctr: 000, out:00000000000000000000000000000000, zero:0
# ADD :
# Time:1600 il:00000000000000000000000000000101, i2:00000000000000000000000000001010, aluctr: 001, out:00000000000000000000000000001111, zero:0
# SUB :
# Time:1700 il:00000000000000000000000000000101, i2:00000000000000000000000000001010, aluctr: 010, out:11111111111111111111111111111011, zero:0
# XOR :
# Time:1800 il:00000000000000000000000000000101, i2:00000000000000000000000000001010, aluctr: 011, out:00000000000000000000000000001111, zero:0
# NOR :
# Time:1900 il:00000000000000000000000000000101, i2:00000000000000000000000000001010, aluctr: 100, out:11111111111111111111111111110000, zero:0
# OR :
# Time:2000 il:00000000000000000000000000000101, i2:00000000000000000000000000001010, aluctr: 101, out:00000000000000000000000000001111, zero:0
# SLT :
# Time:2100 il:00000000000000000000000000000101, i2:00000000000000000000000000001010, aluctr: 110, out:00000000000000000000000000000000, zero:0
```

Alu Control Test:

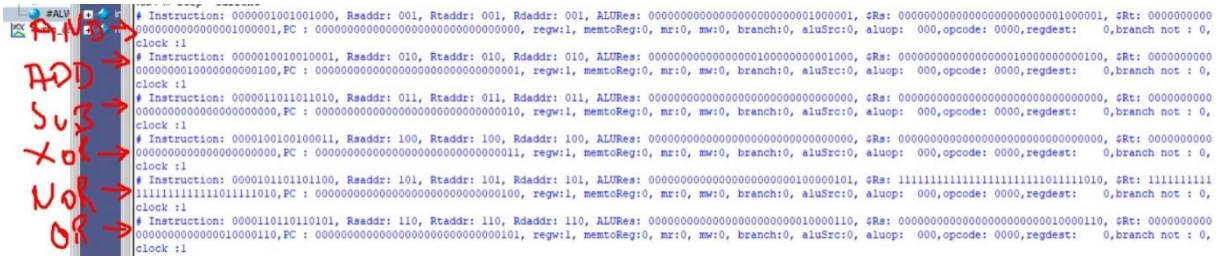
```
VSIM 4> step -current
# Time: 0 _func:000, _aluop:000, _alucontrol:000
# Time:100 _func:001, _aluop:000, _alucontrol:001
# Time:200 _func:010, _aluop:000, _alucontrol:010
# Time:300 _func:011, _aluop:000, _alucontrol:011
# Time:400 _func:100, _aluop:000, _alucontrol:100
# Time:500 _func:101, _aluop:000, _alucontrol:101
# Time:600 _func:110, _aluop:001, _alucontrol:001
# Time:700 _func:110, _aluop:010, _alucontrol:000
# Time:800 _func:110, _aluop:011, _alucontrol:000
# Time:900 _func:110, _aluop:100, _alucontrol:100
# Time:1000 _func:110, _aluop:101, _alucontrol:101
# Time:1100 _func:110, _aluop:110, _alucontrol:010
# Time:1200 _func:110, _aluop:111, _alucontrol:110
```

Control Unit Test:

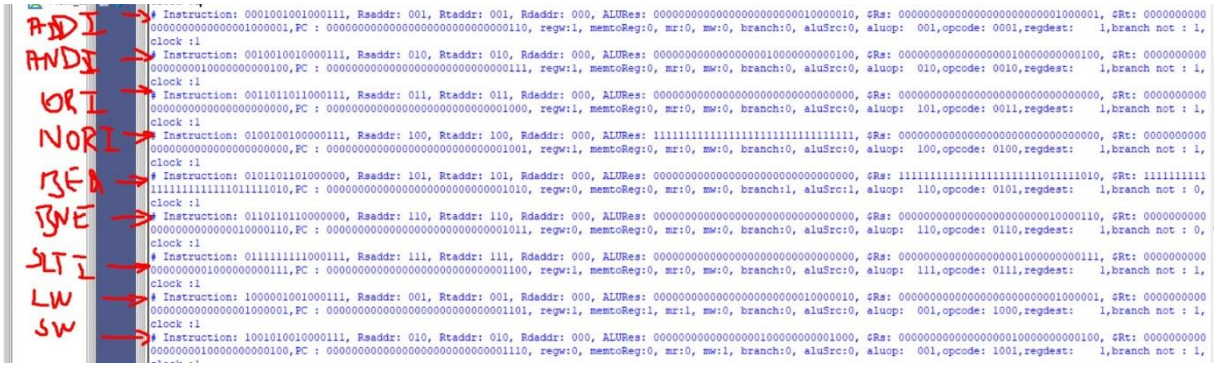
```
VSIM 38> step -current
# Time: 0 Opcode:  0000, RegW:1,  MemtoReg:0, MR:0, MW:0, Branch:0, ALUSrc:0,Branch not:0,regdest:0, ALUOp:000
# Time:100 Opcode:  0001, RegW:1,  MemtoReg:0, MR:0, MW:0, Branch:0, ALUSrc:0,Branch not:1,regdest:1, ALUOp:001
# Time:200 Opcode:  0010, RegW:1,  MemtoReg:0, MR:0, MW:0, Branch:0, ALUSrc:0,Branch not:1,regdest:1, ALUOp:010
# Time:300 Opcode:  0011, RegW:1,  MemtoReg:0, MR:0, MW:0, Branch:0, ALUSrc:0,Branch not:1,regdest:1, ALUOp:101
# Time:400 Opcode:  0100, RegW:1,  MemtoReg:0, MR:0, MW:0, Branch:0, ALUSrc:0,Branch not:1,regdest:1, ALUOp:100
# Time:500 Opcode:  0101, RegW:0,  MemtoReg:0, MR:0, MW:0, Branch:1, ALUSrc:1,Branch not:0,regdest:1, ALUOp:110
# Time:600 Opcode:  0110, RegW:0,  MemtoReg:0, MR:0, MW:0, Branch:0, ALUSrc:0,Branch not:0,regdest:1, ALUOp:110
# Time:700 Opcode:  0111, RegW:1,  MemtoReg:0, MR:0, MW:0, Branch:0, ALUSrc:0,Branch not:1,regdest:1, ALUOp:111
# Time:800 Opcode:  1000, RegW:1,  MemtoReg:1, MR:1, MW:0, Branch:0, ALUSrc:0,Branch not:1,regdest:1, ALUOp:001
# Time:900 Opcode:  1001, RegW:0,  MemtoReg:0, MR:0, MW:1, Branch:0, ALUSrc:0,Branch not:1,regdest:1, ALUOp:001

VSIM 39>
```

MISP TEST: R-types:



MISP TEST: I-types:



Before MISP Test Bench Registers:

After MISP Test Bench Registers: