alu test

```
# time =                   0, a =00000000000000000000000000000010, b=00000000000000000000000000000110, sel=001, out=00000000000000000000000000001000, zero=0
# time =                  20, a =01111111111111111111111111111111, b=01111111111111111111111111111111, sel=001, out=11111111111111111111111111111110, zero=0
# time =                  40, a =00000000000000000000000000001111, b=00000000000000000000000000000011, sel=010, out=00000000000000000000000000001100, zero=0
# time =                  60, a =00000000000000000000000000000000, b=01111111111111111111111111111111, sel=010, out=10000000000000000000000000000001, zero=0
# time =                  80, a =00000000000000000000000010101010, b=00000000000000001111111111111111, sel=000, out=00000000000000000000000010101010, zero=0
# time =                 100, a =00000000000000000000000010101010, b=00000000000000001111111111111111, sel=101, out=00000000000000001111111111111111, zero=0
# time =                 120, a =00000000000000000000000010101010, b=00000000000000001111111111111111, sel=011, out=00000000000000001111111110101010, zero=0
# time =                 140, a =00000000000000000000000000000010, b=00000000000000000000000000000010, sel=010, out=00000000000000000000000000000000, zero=1
# time =                 160, a =00000000000000000000000010101010, b=00000000000000001111111111111111, sel=100, out=11111111111111110000000000000000, zero=0
# time =                 180, a =00000000000000000000000000001001, b=00000000000000000000000000001000, sel=110, out=00000000000000000000000000000000, zero=1
```

0	0	0	AND
0	0	1	ADD
0	1	0	SUB
0	1	1	XOR
1	0	0	NOR
1	0	1	OR
0	0	1	ADD
0	0	0	AND
1	0	1	OR
1	0	0	NOR
0	1	0	SUB
1	1	0	SLT

control unit test

```
# time= 0, op=0000, regdst=1, alusrc=0, memtoreg=0, regwrite=1, memread=0, memwrite=0, branch=0, branchn=0, aluop=000
# time=20, op=0001, regdst=0, alusrc=1, memtoreg=0, regwrite=1, memread=0, memwrite=0, branch=0, branchn=0, aluop=001
# time=40, op=0010, regdst=0, alusrc=1, memtoreg=0, regwrite=1, memread=0, memwrite=0, branch=0, branchn=0, aluop=010
# time=60, op=0011, regdst=0, alusrc=1, memtoreg=0, regwrite=1, memread=0, memwrite=0, branch=0, branchn=0, aluop=011
# time=80, op=0100, regdst=0, alusrc=1, memtoreg=0, regwrite=1, memread=0, memwrite=0, branch=0, branchn=0, aluop=100
# time=100, op=0101, regdst=0, alusrc=0, memtoreg=0, regwrite=0, memread=0, memwrite=0, branch=1, branchn=0, aluop=101
# time=120, op=0110, regdst=0, alusrc=0, memtoreg=0, regwrite=0, memread=0, memwrite=0, branch=0, branchn=1, aluop=101
# time=140, op=0111, regdst=0, alusrc=1, memtoreg=0, regwrite=1, memread=0, memwrite=0, branch=0, branchn=0, aluop=110
# time=160, op=1000, regdst=0, alusrc=1, memtoreg=1, regwrite=1, memread=1, memwrite=0, branch=0, branchn=0, aluop=001
# time=180, op=1001, regdst=0, alusrc=1, memtoreg=0, regwrite=0, memread=0, memwrite=1, branch=0, branchn=0, aluop=001
```

Instr	RegDst	RegWrite	ALUSrc	ALUOp2	ALUOp1	ALUOp0	MemWrite	MemRead	MemToReg	BranchNot	Branch
Rtype	1	1	0	0	0	0	0	0	0	0	0
ADDI	0	1	1	0	0	1	0	0	0	0	0
ANDI	0	1	1	0	1	0	0	0	0	0	0
ORI	0	1	1	0	1	1	0	0	0	0	0
NORI	0	1	1	1	0	0	0	0	0	0	0
BEQ	x	0	0	1	0	1	0	0	0	0	1
BNE	x	0	0	1	0	1	0	0	0	1	0
SLTI	x	1	1	1	1	0	0	0	0	0	0
LW	0	1	1	0	0	1	0	1	1	0	0
SW	x	0	1	0	0	1	1	0	0	0	0

Instr	Opcode
AND	0000
ADD	0000
SUB	0000
XOR	0000
NOR	0000
OR	0000
ADDI	0001
ANDI	0010
ORI	0011
NORI	0100
BEQ	0101
BNE	0110
SLTI	0111
LW	1000
SW	1001

alu control test

ins	aluop2	aluop1	aluop0	f2	f1	f0	c2	c1	c0	
rType	0	0	0	0	0	0	0	0	0	AND
rType	0	0	0	0	0	1	0	0	1	ADD
rType	0	0	0	0	1	0	0	1	0	SUB
rType	0	0	0	0	1	1	0	1	1	XOR
rType	0	0	0	1	0	0	1	0	0	NOR
rType	0	0	0	1	0	1	1	0	1	OR
add	0	0	1	x	x	x	0	0	1	ADD
and	0	1	0	x	x	x	0	0	0	AND
or	0	1	1	x	x	x	1	0	1	OR
nor	1	0	0	x	x	x	1	0	0	NOR
sub	1	0	1	x	x	x	0	1	0	SUB
slt	1	1	0	x	x	x	1	1	0	SLT

```
VSIM 27> step -current
# time =                    0, o=000, f=000, out=000,
# time =                   20, o=000, f=001, out=001,
# time =                   40, o=000, f=010, out=010,
# time =                   60, o=000, f=011, out=011,
# time =                   80, o=000, f=100, out=100,
# time =                  100, o=000, f=101, out=101,
# time =                  120, o=001, f=000, out=001,
# time =                  140, o=010, f=000, out=000,
# time =                  160, o=011, f=000, out=101,
# time =                  180, o=100, f=000, out=100,
# time =                  200, o=101, f=000, out=010,
# time =                  220, o=110, f=000, out=110,
```

mips test

```
# ============================
# time =  0, clock=0
# instruction=                 0000001010011001
# memToRegOut=00000000000000000000000000000011
# pc=00000000000000000000000000000000
# pc_plus_1=00000000000000000000000000000001
# mux_branch_out=00000000000000000000000000000001
# alu_control_out=                                001
# sign_extend_out=00000000000000000000000011001
# alu_result=00000000000000000000000000000011
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ============================
# time = 100, clock=1
# instruction=                 0000001010011001
# memToRegOut=00000000000000000000000000000011
# pc=00000000000000000000000000000000
# pc_plus_1=00000000000000000000000000000001
# mux_branch_out=00000000000000000000000000000001
# alu_control_out=                                001
# sign_extend_out=00000000000000000000000011001
# alu_result=00000000000000000000000000000011
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ============================
# time = 100, clock=1
# instruction=                 0000001010100010
# memToRegOut=11111111111111111111111111111111
# pc=00000000000000000000000000000001
# pc_plus_1=00000000000000000000000000000010
# mux_branch_out=00000000000000000000000000000010
# alu_control_out=                                010
# sign_extend_out=11111111111111111111111100010
# alu_result=11111111111111111111111111111111
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ============================
# time = 200, clock=0
# instruction=                 0000001010100010
# memToRegOut=11111111111111111111111111111111
```

```
# pc=00000000000000000000000000000001
# pc_plus_1=00000000000000000000000000000010
# mux_branch_out=00000000000000000000000000000010
# alu_control_out=                                010
# sign_extend_out=11111111111111111111111111100010
# alu_result=11111111111111111111111111111111
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ============================
# time = 300, clock=1
# instruction=                0000001010100010
# memToRegOut=11111111111111111111111111111111
# pc=00000000000000000000000000000001
# pc_plus_1=00000000000000000000000000000010
# mux_branch_out=00000000000000000000000000000010
# alu_control_out=                                010
# sign_extend_out=11111111111111111111111111100010
# alu_result=11111111111111111111111111111111
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ============================
# time = 300, clock=1
# instruction=                0000001010101011
# memToRegOut=00000000000000000000000000000011
# pc=00000000000000000000000000000010
# pc_plus_1=00000000000000000000000000000011
# mux_branch_out=00000000000000000000000000000011
# alu_control_out=                                011
# sign_extend_out=11111111111111111111111111101011
# alu_result=00000000000000000000000000000011
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ============================
# time = 400, clock=0
# instruction=                0000001010101011
# memToRegOut=00000000000000000000000000000011
# pc=00000000000000000000000000000010
# pc_plus_1=00000000000000000000000000000011
# mux_branch_out=00000000000000000000000000000011
# alu_control_out=                                011
# sign_extend_out=11111111111111111111111111101011
# alu_result=00000000000000000000000000000011
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ============================
# time = 500, clock=1
# instruction=                0000001010101011
# memToRegOut=00000000000000000000000000000011
# pc=00000000000000000000000000000010
# pc_plus_1=00000000000000000000000000000011
# mux_branch_out=00000000000000000000000000000011
# alu_control_out=                                011
# sign_extend_out=11111111111111111111111111101011
# alu_result=00000000000000000000000000000011
```

```
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ============================
# time = 500, clock=1
# instruction=                0000001010110100
# memToRegOut=11111111111111111111111111111100
# pc=00000000000000000000000000000011
# pc_plus_1=00000000000000000000000000000100
# mux_branch_out=00000000000000000000000000000100
# alu_control_out=                            100
# sign_extend_out=11111111111111111111111110100
# alu_result=11111111111111111111111111111100
# read_data_1=00000000000000000000000000000001
# read_data_2=00000000000000000000000000000010
# ** Note: $finish    : /home/umit/Documents/quartus/mips32_testbench.v(24)
#    Time: 600 ps  Iteration: 1  Instance: /mips32_testbench
```

instructions

```
0000_001_010_011_001
0000_001_010_100_010
0000_001_010_101_011
0000_001_010_110_100
0000_001_010_111_101
```

registers

```
// memory data file (do not edit the following line - required
// instance=/mips32_testbench/m/register_block/registers
// format=bin addressradix=h dataradix=b version=1.0 wordsperli
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000000010
00000000000000000000000000000011
11111111111111111111111111111111
00000000000000000000000000000011
00000000000000000000000000000000
00000000000000000000000000000000
```