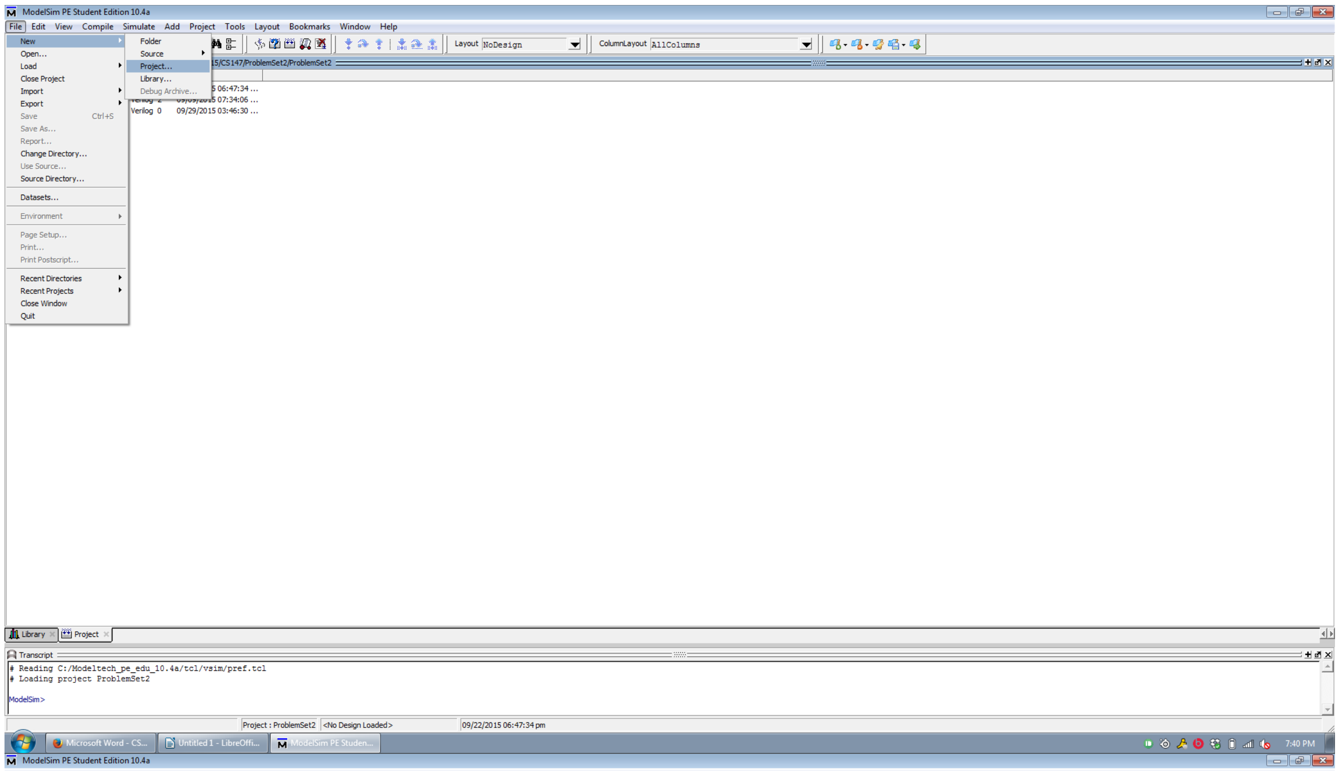
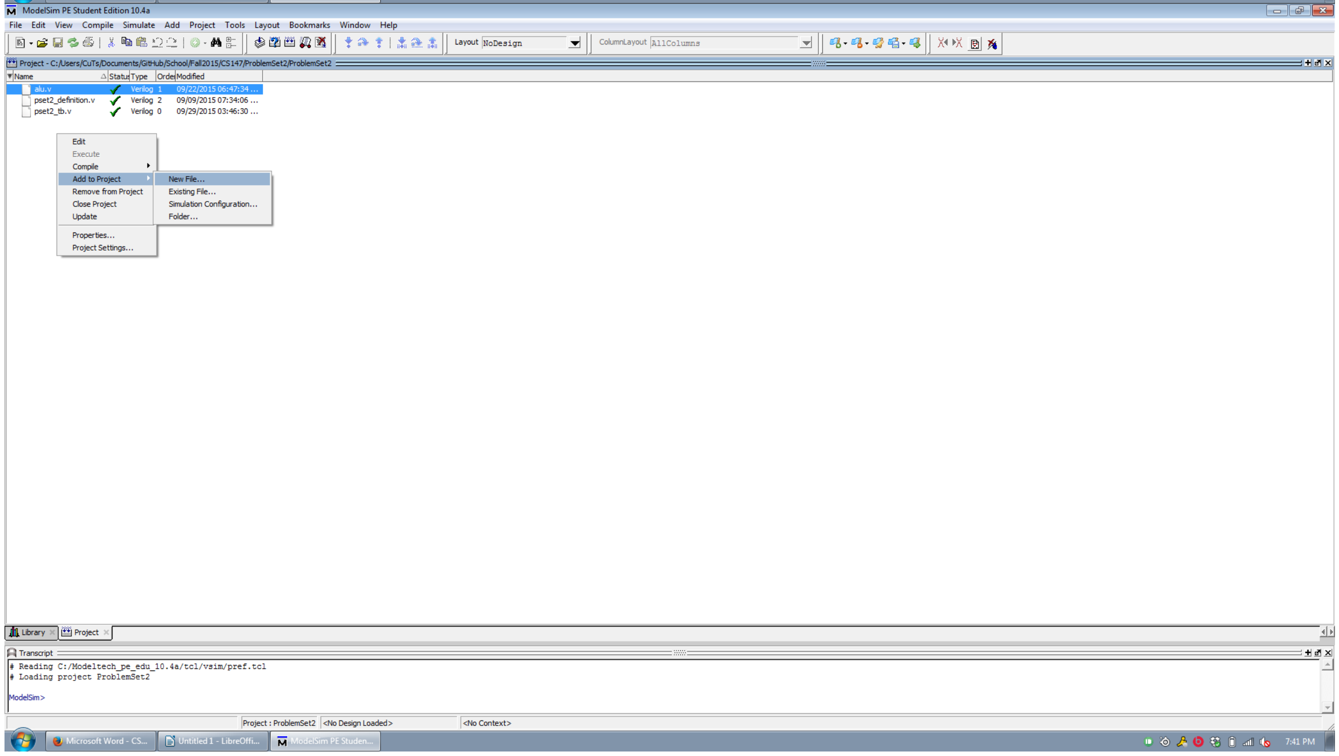


```
# Reading C:/Modeltech_pe_edu_10.4a/tcl/vsim/pref.tcl
# Loading project ProblemSet2

ModelSim>
```

Project : ProblemSet2 <No Design Loaded> 09/22/2015 06:47:34 pm


ModelSim PE Student Edition 10.4a
File Edit View Compile Simulate Add Project Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - C:/Users/CuTs/Documents/GitHub/School/Fall2015/CS147/ProblemSet2/ProblemSet2
Edit
Execute
Compile
Add to Project
Remove from Project
Close Project
Update
Properties...
Project Settings...
New File...
Existing File...
Simulation Configuration...
Folder...
Library
Project
Transcript

Name	Status	Type	Order	Modified
alu.v	✓	Verilog	1	09/22/2015 06:47:34 ...
pset2_definition.v	✓	Verilog	2	09/09/2015 07:34:06 ...
pset2_tb.v	✓	Verilog	0	09/29/2015 03:46:30 ...

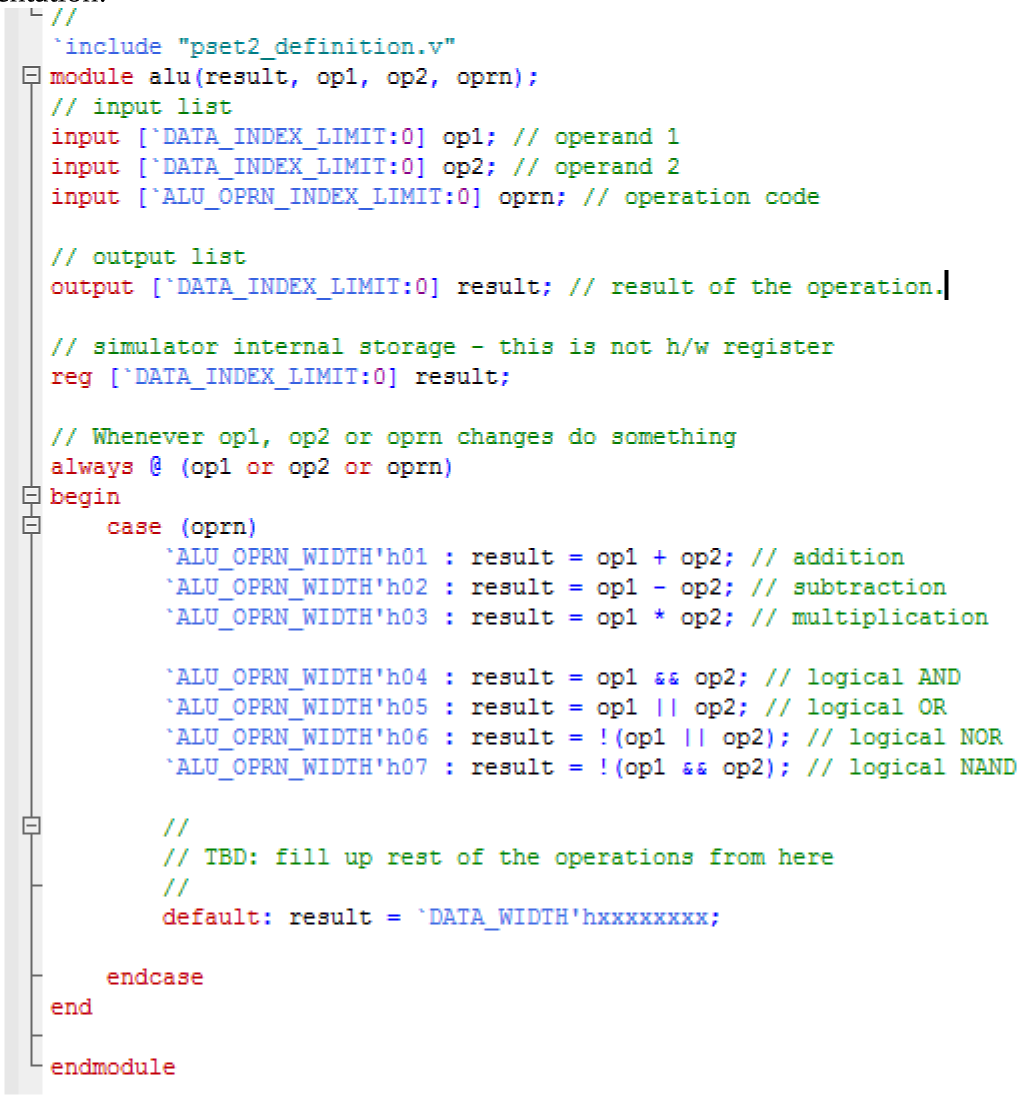
```
# Reading C:/Modeltech_pe_edu_10.4a/tcl/vsim/pref.tcl
# Loading project ProblemSet2

ModelSim>
```

Project : ProblemSet2 <No Design Loaded> <No Context>

Requirements:

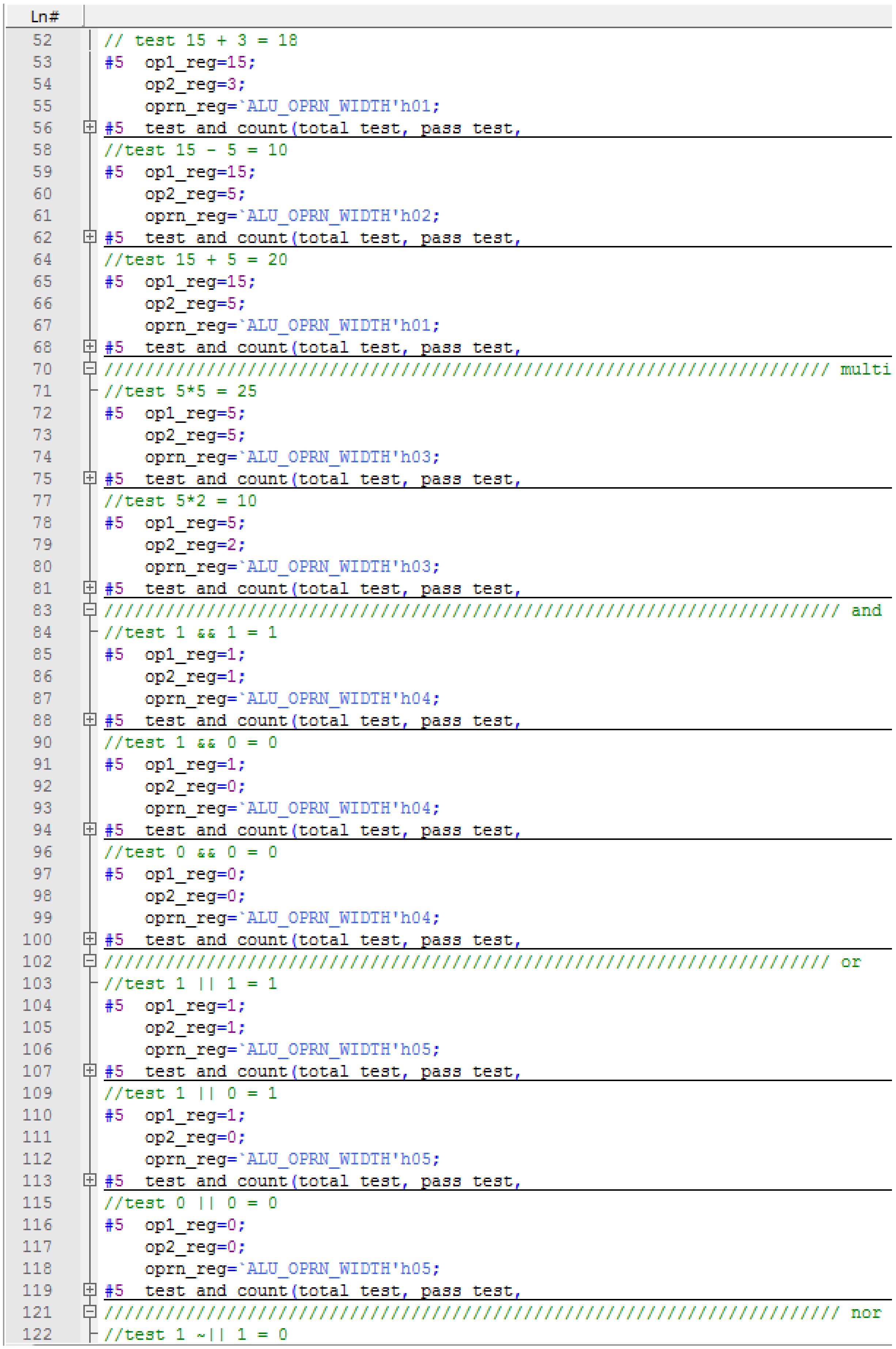
Name	Mnemonic	Format	Operation	OpCode /funct
Addition	add	R	R[rd] = R[rs] + R[rt]	0x00 / 0x20
Subtraction	sub	R	R[rd] = R[rs] - R[rt]	0x00 / 0x22
Multiplication	mul	R	R[rd] = R[rs] * R[rt]	0x00 / 0x2c
Logical AND	and	R	R[rd] = R[rs] & R[rt]	0x00 / 0x24
Logical OR	or	R	R[rd] = R[rs] | R[rt]	0x00 / 0x25
Logical NOR	nor	R	R[rd] = ~(R[rs] | R[rt])	0x00 / 0x27
Logical NAND	Nand	R	R[rd = ~(R[rs] & R[rt])	0x00 / 0x28

Implementation:

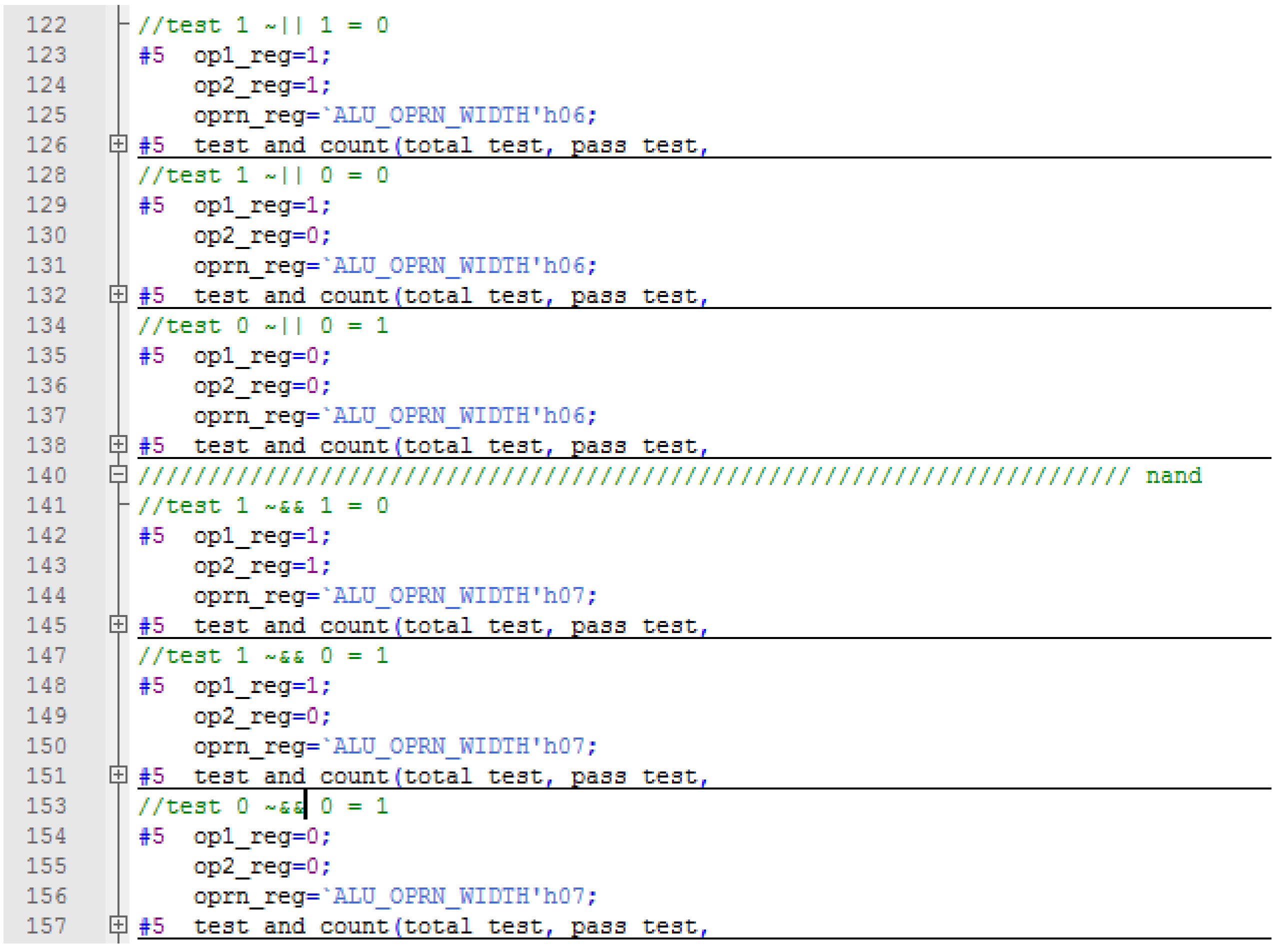


```
//
`include "pset2_definition.v"
module alu(result, op1, op2, oprn);
// input list
input [`DATA_INDEX_LIMIT:0] op1; // operand 1
input [`DATA_INDEX_LIMIT:0] op2; // operand 2
input [`ALU_OPRN_INDEX_LIMIT:0] oprn; // operation code

// output list
output [`DATA_INDEX_LIMIT:0] result; // result of the operation.

// simulator internal storage - this is not h/w register
reg [`DATA_INDEX_LIMIT:0] result;

// Whenever op1, op2 or oprn changes do something
always @ (op1 or op2 or oprn)
begin
    case (oprn)
        `ALU_OPRN_WIDTH'h01 : result = op1 + op2; // addition
        `ALU_OPRN_WIDTH'h02 : result = op1 - op2; // subtraction
        `ALU_OPRN_WIDTH'h03 : result = op1 * op2; // multiplication

        `ALU_OPRN_WIDTH'h04 : result = op1 && op2; // logical AND
        `ALU_OPRN_WIDTH'h05 : result = op1 || op2; // logical OR
        `ALU_OPRN_WIDTH'h06 : result = !(op1 || op2); // logical NOR
        `ALU_OPRN_WIDTH'h07 : result = !(op1 && op2); // logical NAND

        //
        // TBD: fill up rest of the operations from here
        //
        default: result = `DATA_WIDTH'hxxxxxxxx;

    endcase
end

endmodule
```

Testing Strategy:

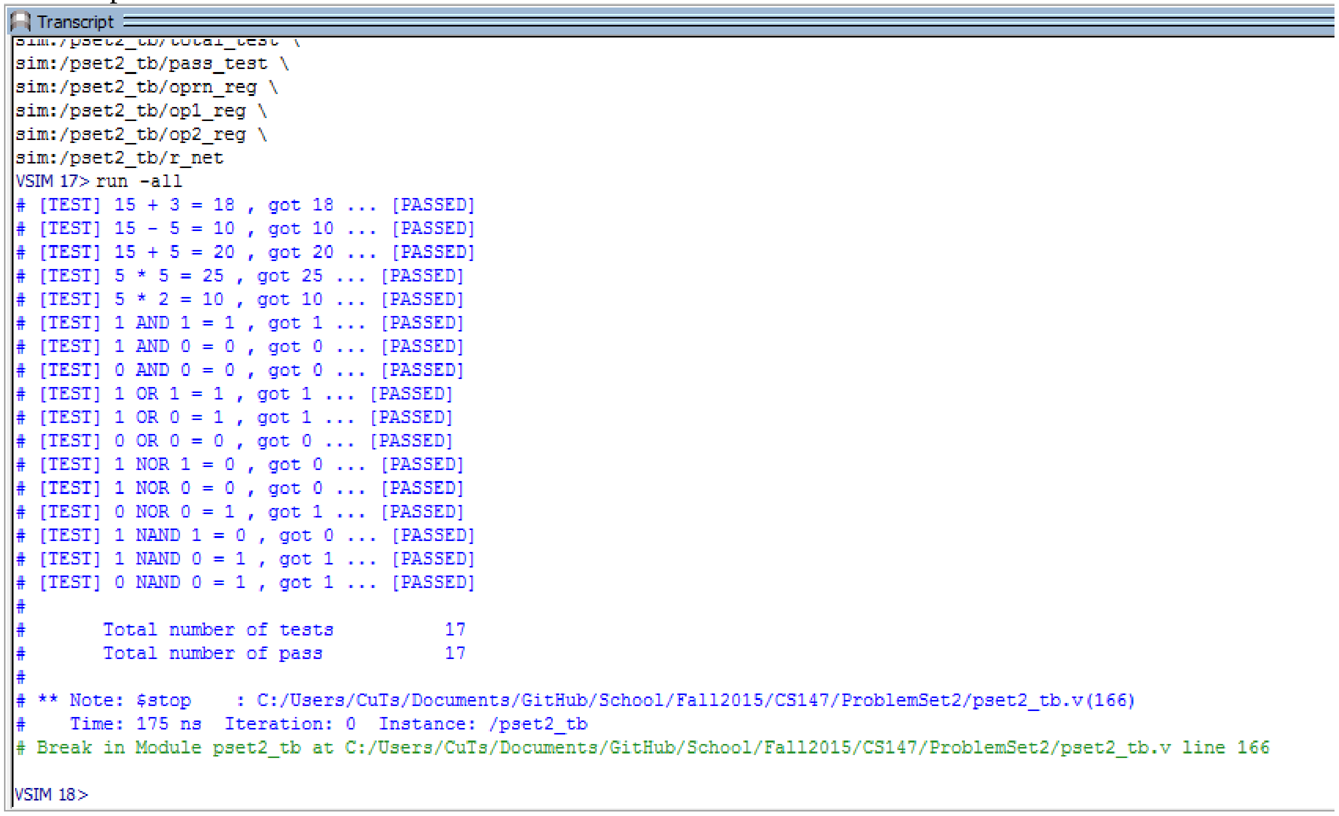


```
// test 15 + 3 = 18
#5   op1_reg=15;
     op2_reg=3;
     oprn_reg=`ALU_OPRN_WIDTH'h01;
#5   test_and_count(total_test, pass_test,
//test 15 - 5 = 10
#5   op1_reg=15;
     op2_reg=5;
     oprn_reg=`ALU_OPRN_WIDTH'h02;
#5   test_and_count(total_test, pass_test,
//test 15 + 5 = 20
#5   op1_reg=15;
     op2_reg=5;
     oprn_reg=`ALU_OPRN_WIDTH'h01;
#5   test_and_count(total_test, pass_test,
/////////////////////////////////////////////////////////////////////// multi
//test 5*5 = 25
#5   op1_reg=5;
     op2_reg=5;
     oprn_reg=`ALU_OPRN_WIDTH'h03;
#5   test_and_count(total_test, pass_test,
//test 5*2 = 10
#5   op1_reg=5;
     op2_reg=2;
     oprn_reg=`ALU_OPRN_WIDTH'h03;
#5   test_and_count(total_test, pass_test,
//////////////////////////////////////////////////////////////////////// and
//test 1 && 1 = 1
#5   op1_reg=1;
     op2_reg=1;
     oprn_reg=`ALU_OPRN_WIDTH'h04;
#5   test_and_count(total_test, pass_test,
//test 1 && 0 = 0
#5   op1_reg=1;
     op2_reg=0;
     oprn_reg=`ALU_OPRN_WIDTH'h04;
#5   test_and_count(total_test, pass_test,
//test 0 && 0 = 0
#5   op1_reg=0;
     op2_reg=0;
     oprn_reg=`ALU_OPRN_WIDTH'h04;
#5   test_and_count(total_test, pass_test,
/////////////////////////////////////////////////////////////////////// or
//test 1 || 1 = 1
#5   op1_reg=1;
     op2_reg=1;
     oprn_reg=`ALU_OPRN_WIDTH'h05;
#5   test_and_count(total_test, pass_test,
//test 1 || 0 = 1
#5   op1_reg=1;
     op2_reg=0;
     oprn_reg=`ALU_OPRN_WIDTH'h05;
#5   test_and_count(total_test, pass_test,
//test 0 || 0 = 0
#5   op1_reg=0;
     op2_reg=0;
     oprn_reg=`ALU_OPRN_WIDTH'h05;
#5   test_and_count(total_test, pass_test,
//////////////////////////////////////////////////////////////////////// nor
//test 1 ~|| 1 = 0
```



```
//test 1 ~|| 1 = 0
#5  op1_reg=1;
    op2_reg=1;
    oprn_reg=`ALU_OPRN_WIDTH'h06;
#5  test_and_count(total_test, pass_test,
//test 1 ~|| 0 = 0
#5  op1_reg=1;
    op2_reg=0;
    oprn_reg=`ALU_OPRN_WIDTH'h06;
#5  test_and_count(total_test, pass_test,
//test 0 ~|| 0 = 1
#5  op1_reg=0;
    op2_reg=0;
    oprn_reg=`ALU_OPRN_WIDTH'h06;
#5  test_and_count(total_test, pass_test,
////////////////////////////////////////////////////////////////////////// nand
//test 1 ~&& 1 = 0
#5  op1_reg=1;
    op2_reg=1;
    oprn_reg=`ALU_OPRN_WIDTH'h07;
#5  test_and_count(total_test, pass_test,
//test 1 ~&& 0 = 1
#5  op1_reg=1;
    op2_reg=0;
    oprn_reg=`ALU_OPRN_WIDTH'h07;
#5  test_and_count(total_test, pass_test,
//test 0 ~&& 0 = 1
#5  op1_reg=0;
    op2_reg=0;
    oprn_reg=`ALU_OPRN_WIDTH'h07;
#5  test_and_count(total_test, pass_test,
```

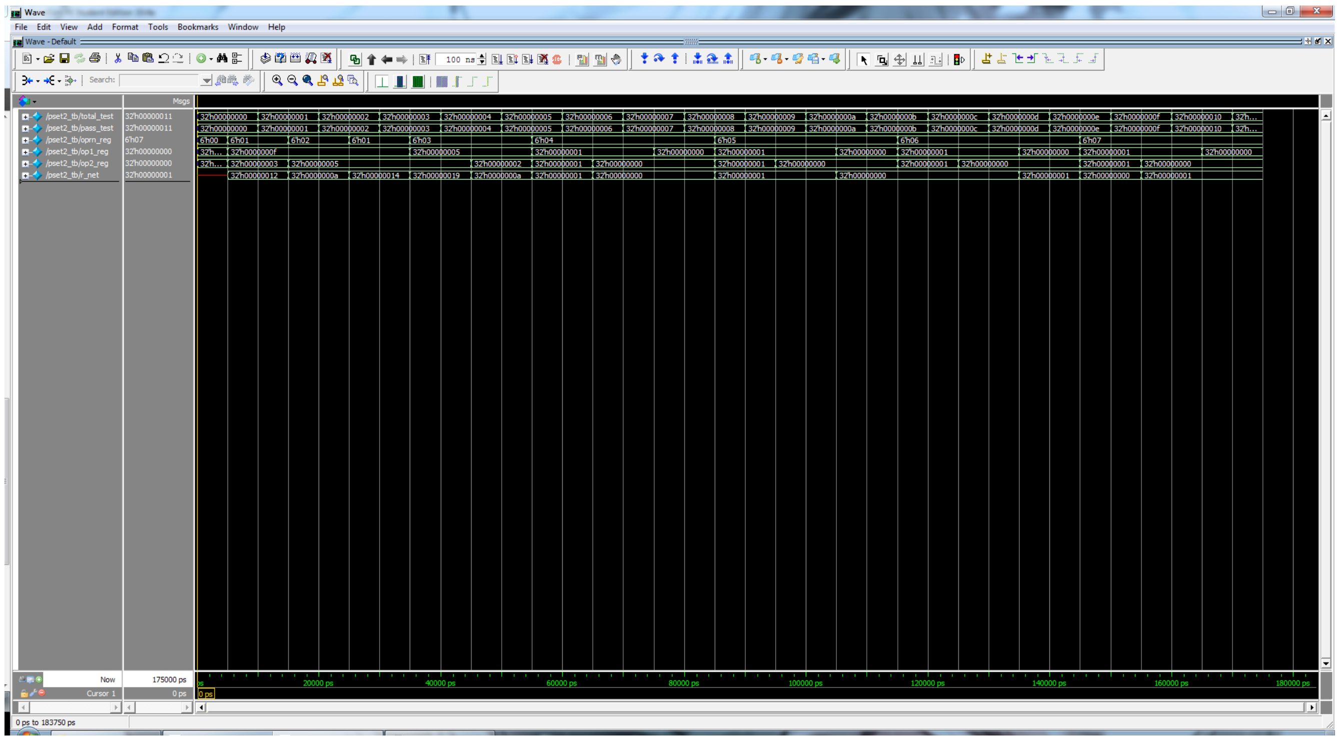
Text output:



Transcript

```
sim:/pset2_tb/total_test \
sim:/pset2_tb/pass_test \
sim:/pset2_tb/oprn_reg \
sim:/pset2_tb/op1_reg \
sim:/pset2_tb/op2_reg \
sim:/pset2_tb/r_net
VSIM 17> run -all
# [TEST] 15 + 3 = 18 , got 18 ... [PASSED]
# [TEST] 15 - 5 = 10 , got 10 ... [PASSED]
# [TEST] 15 + 5 = 20 , got 20 ... [PASSED]
# [TEST] 5 * 5 = 25 , got 25 ... [PASSED]
# [TEST] 5 * 2 = 10 , got 10 ... [PASSED]
# [TEST] 1 AND 1 = 1 , got 1 ... [PASSED]
# [TEST] 1 AND 0 = 0 , got 0 ... [PASSED]
# [TEST] 0 AND 0 = 0 , got 0 ... [PASSED]
# [TEST] 1 OR 1 = 1 , got 1 ... [PASSED]
# [TEST] 1 OR 0 = 1 , got 1 ... [PASSED]
# [TEST] 0 OR 0 = 0 , got 0 ... [PASSED]
# [TEST] 1 NOR 1 = 0 , got 0 ... [PASSED]
# [TEST] 1 NOR 0 = 0 , got 0 ... [PASSED]
# [TEST] 0 NOR 0 = 1 , got 1 ... [PASSED]
# [TEST] 1 NAND 1 = 0 , got 0 ... [PASSED]
# [TEST] 1 NAND 0 = 1 , got 1 ... [PASSED]
# [TEST] 0 NAND 0 = 1 , got 1 ... [PASSED]
#
#       Total number of tests          17
#       Total number of pass           17
#
# ** Note: $stop    : C:/Users/CuTs/Documents/GitHub/School/Fall2015/CS147/ProblemSet2/pset2_tb.v(166)
#    Time: 175 ns  Iteration: 0  Instance: /pset2_tb
# Break in Module pset2_tb at C:/Users/CuTs/Documents/GitHub/School/Fall2015/CS147/ProblemSet2/pset2_tb.v line 166

VSIM 18>
```

Waveform:



```
/pset2_tb/total_test   32'h00000011   32'h00000000 32'h00000001 32'h00000002 32'h00000003 32'h00000004 32'h00000005 32'h00000006 32'h00000007 32'h00000008 32'h00000009 32'h0000000a 32'h0000000b 32'h0000000c 32'h0000000d 32'h0000000e 32'h0000000f 32'h00000010 32'h...
/pset2_tb/pass_test    32'h00000011   32'h00000000 32'h00000001 32'h00000002 32'h00000003 32'h00000004 32'h00000005 32'h00000006 32'h00000007 32'h00000008 32'h00000009 32'h0000000a 32'h0000000b 32'h0000000c 32'h0000000d 32'h0000000e 32'h0000000f 32'h00000010 32'h...
/pset2_tb/oprn_reg     6'h07          6'h00 6'h01 6'h02 6'h01 6'h03 6'h04 6'h05 6'h06 6'h07
/pset2_tb/op1_reg      32'h00000000   32'h... 32'h0000000f 32'h00000005 32'h00000001 32'h00000000 32'h00000001 32'h00000000 32'h00000001 32'h00000000 32'h00000001 32'h00000000
/pset2_tb/op2_reg      32'h00000000   32'h... 32'h00000003 32'h00000005 32'h00000002 32'h00000001 32'h00000000 32'h00000001 32'h00000000 32'h00000001 32'h00000000 32'h00000001 32'h00000000
/pset2_tb/r_net        32'h00000001   32'h00000012 32'h0000000a 32'h00000014 32'h00000019 32'h0000000a 32'h00000001 32'h00000000 32'h00000001 32'h00000000 32'h00000001 32'h00000000 32'h00000001

Now       175000 ps
Cursor 1  0 ps

ps  20000 ps  40000 ps  60000 ps  80000 ps  100000 ps  120000 ps  140000 ps  160000 ps  180000 ps
0 ps

0 ps to 183750 ps
```