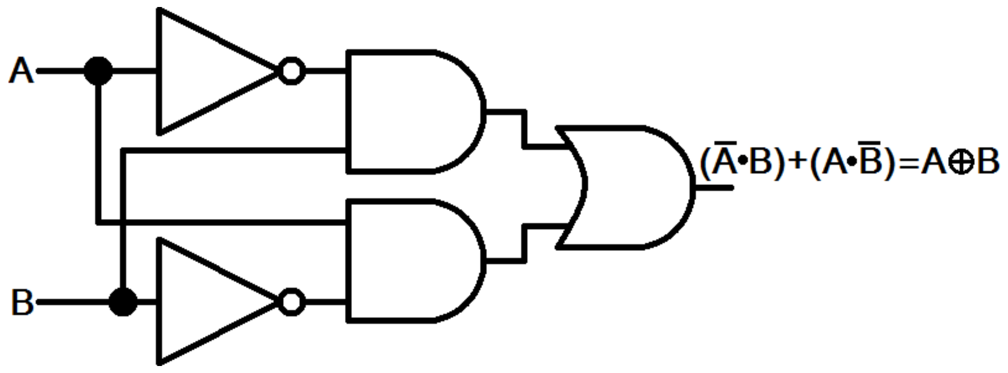
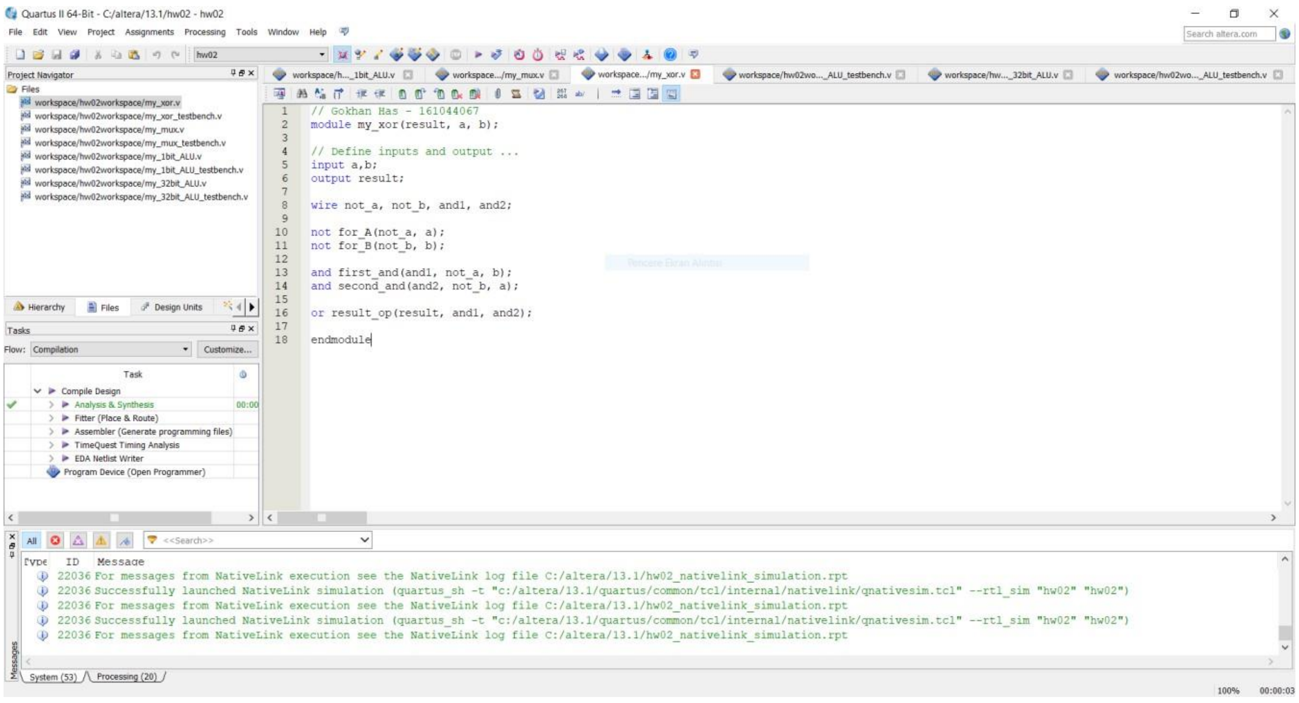
CSE331 COMPUTER ORGANIZATION HOMEWORK 2

Gökhan HAS – 161044067

First, I started to design XOR gate. I used the following circuit for this process. I have used two NOT, two AND and one OR gate.

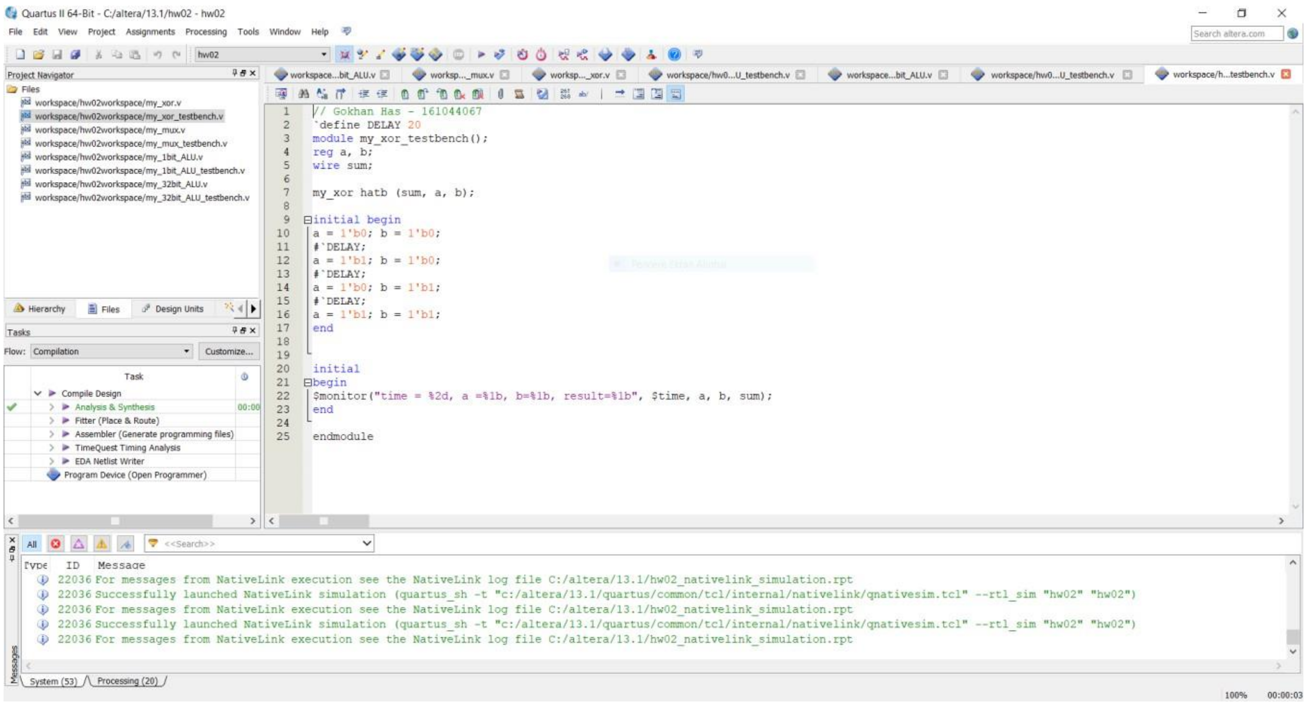




```
// Gokhan Has - 161044067
module my_xor(result, a, b);

// Define inputs and output ...
input a,b;
output result;

wire not_a, not_b, and1, and2;

not for_A(not_a, a);
not for_B(not_b, b);

and first_and(and1, not_a, b);
and second_and(and2, not_b, a);

or result_op(result, and1, and2);

endmodule
```

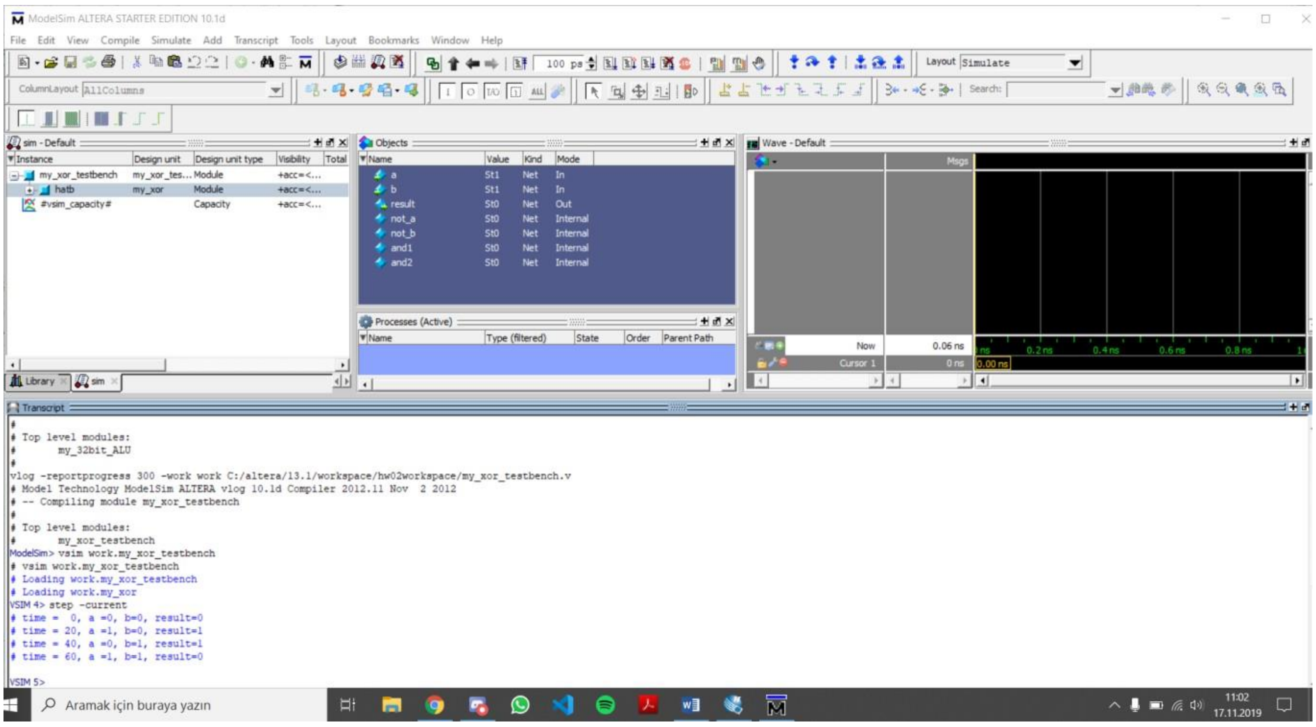
```
Type  ID    Message
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
      22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
      22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
```

Here is a testbench code for xor;



```
// Gokhan Has - 161044067
`define DELAY 20
module my_xor_testbench();
reg a, b;
wire sum;

my_xor hatb (sum, a, b);

initial begin
a = 1'b0; b = 1'b0;
#`DELAY;
a = 1'b1; b = 1'b0;
#`DELAY;
a = 1'b0; b = 1'b1;
#`DELAY;
a = 1'b1; b = 1'b1;
end


initial
begin
$monitor("time = %2d, a =%1b, b=%1b, result=%1b", $time, a, b, sum);
end

endmodule
```

```
Type   ID    Message
  22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
  22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
  22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
  22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
  22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
```

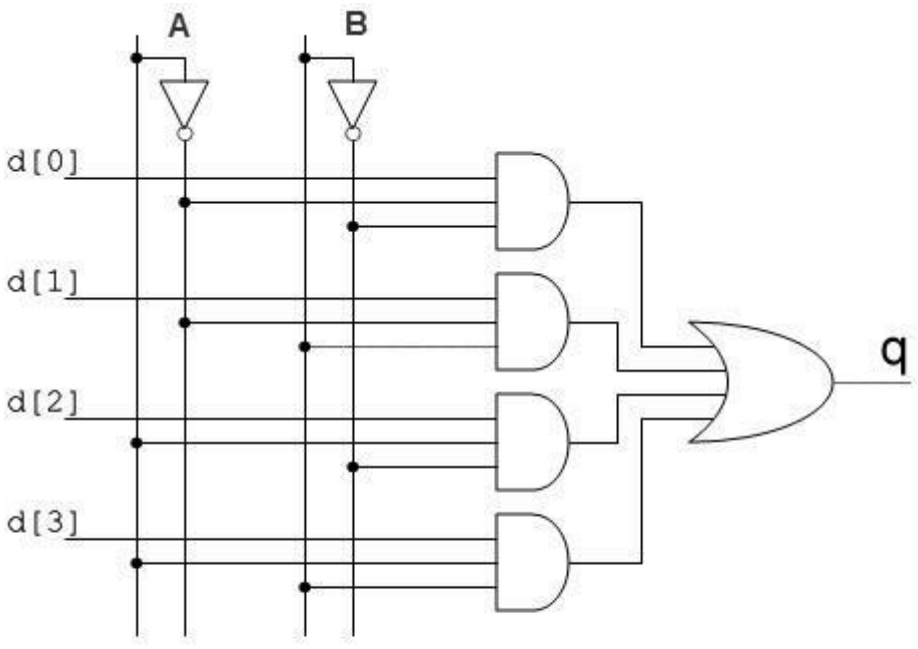


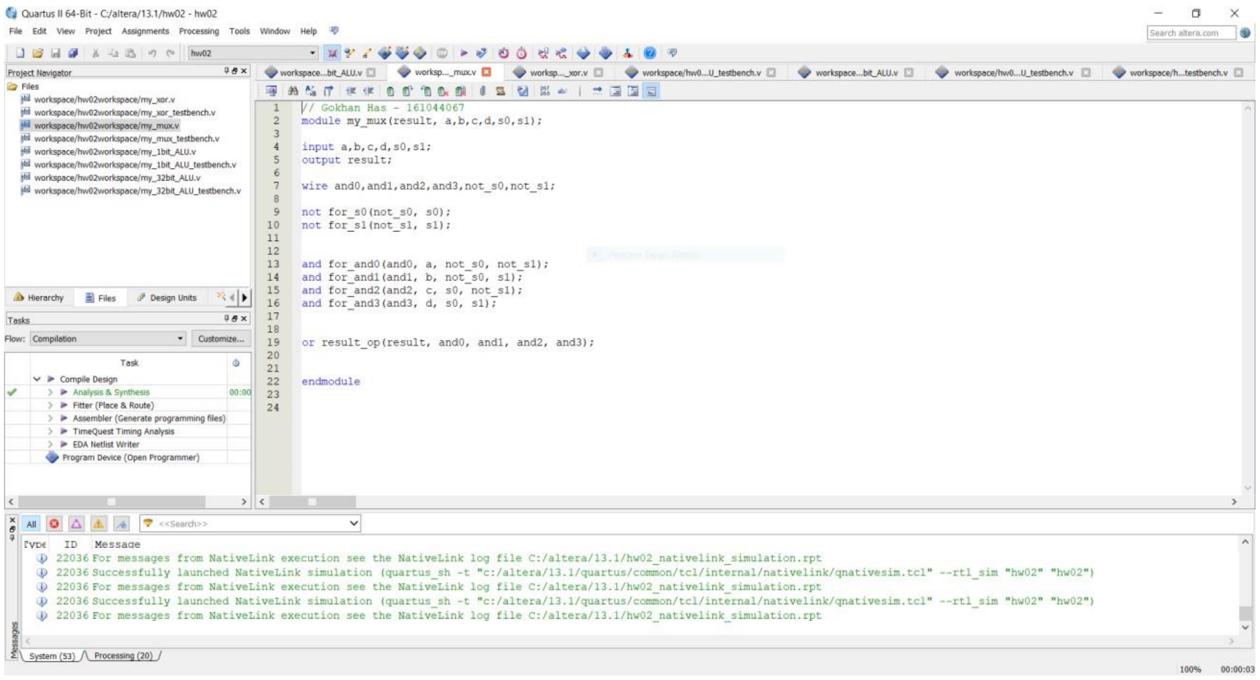
Instance	Design unit	Design unit type	Visibility
my_xor_testbench	my_xor_tes...	Module	+acc=<...
hatb	my_xor	Module	+acc=<...
#vsim_capacity#		Capacity	+acc=<...

Name	Value	Kind	Mode
a	St1	Net	In
b	St1	Net	In
result	St0	Net	Out
not_a	St0	Net	Internal
not_b	St0	Net	Internal
and1	St0	Net	Internal
and2	St0	Net	Internal

```
#
# Top level modules:
#       my_32bit_ALU
#
vlog -reportprogress 300 -work work C:/altera/13.1/workspace/hw02workspace/my_xor_testbench.v
# Model Technology ModelSim ALTERA vlog 10.1d Compiler 2012.11 Nov  2 2012
# -- Compiling module my_xor_testbench
#
# Top level modules:
#       my_xor_testbench
ModelSim> vsim work.my_xor_testbench
# vsim work.my_xor_testbench
# Loading work.my_xor_testbench
# Loading work.my_xor
VSIM 4> step -current
# time =  0, a =0, b=0, result=0
# time = 20, a =1, b=0, result=1
# time = 40, a =0, b=1, result=1
# time = 60, a =1, b=1, result=0

VSIM 5>
```

My_Mux 4x1 :

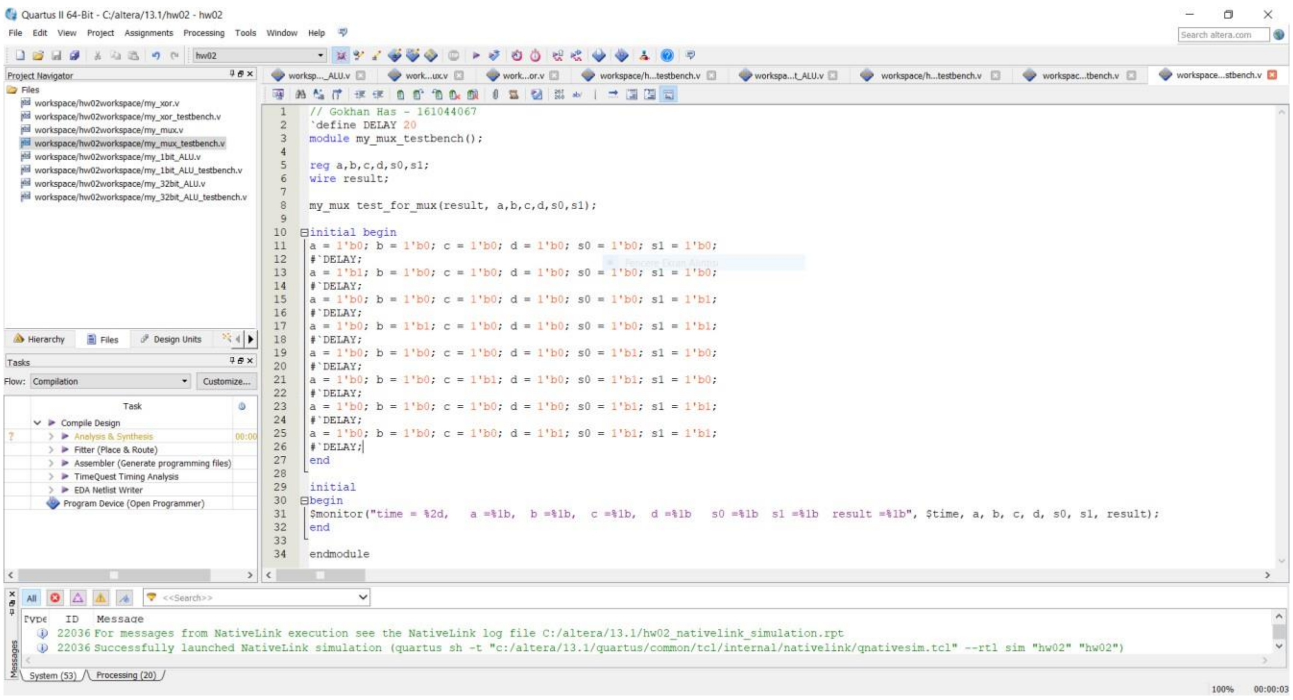




```
// Gokhan Has - 161044067
module my_mux(result, a,b,c,d,s0,s1);

input a,b,c,d,s0,s1;
output result;

wire and0,and1,and2,and3,not_s0,not_s1;

not for_s0(not_s0, s0);
not for_s1(not_s1, s1);


and for_and0(and0, a, not_s0, not_s1);
and for_and1(and1, b, not_s0, s1);
and for_and2(and2, c, s0, not_s1);
and for_and3(and3, d, s0, s1);


or result_op(result, and0, and1, and2, and3);


endmodule
```

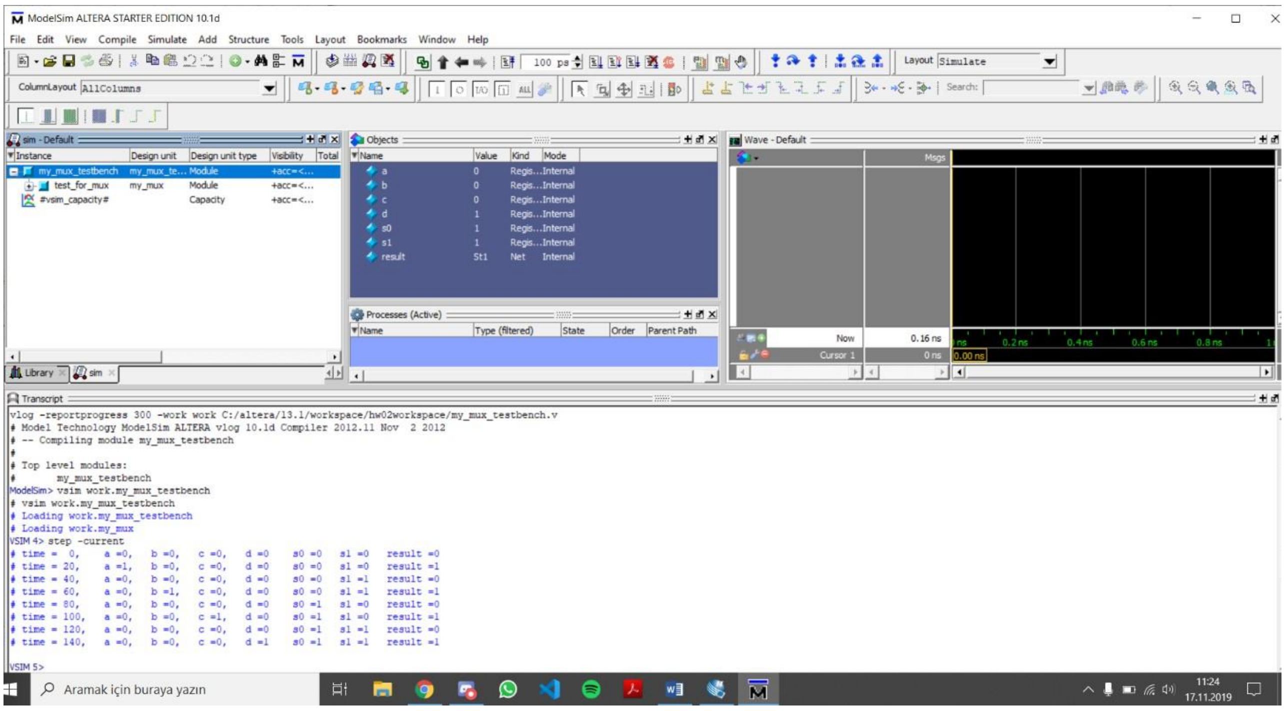
```
Type  ID    Message
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
      22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
      22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
```

MUX Testbench Code:

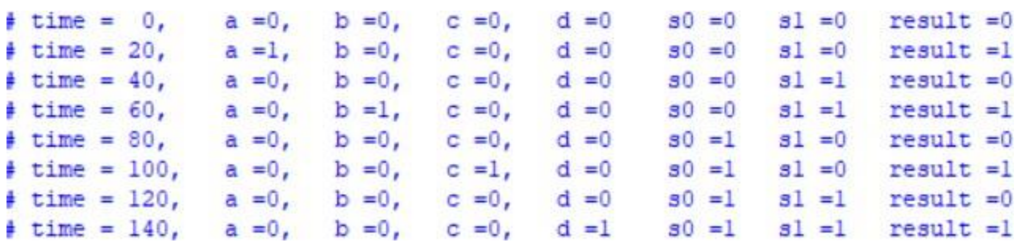


```
// Gokhan Has - 161044067
`define DELAY 20
module my_mux_testbench();

reg a,b,c,d,s0,s1;
wire result;

my_mux test_for_mux(result, a,b,c,d,s0,s1);

initial begin
a = 1'b0; b = 1'b0; c = 1'b0; d = 1'b0; s0 = 1'b0; s1 = 1'b0;
#`DELAY;
a = 1'b1; b = 1'b0; c = 1'b0; d = 1'b0; s0 = 1'b0; s1 = 1'b0;
#`DELAY;
a = 1'b0; b = 1'b0; c = 1'b0; d = 1'b0; s0 = 1'b0; s1 = 1'b1;
#`DELAY;
a = 1'b0; b = 1'b1; c = 1'b0; d = 1'b0; s0 = 1'b0; s1 = 1'b1;
#`DELAY;
a = 1'b0; b = 1'b0; c = 1'b0; d = 1'b0; s0 = 1'b1; s1 = 1'b0;
#`DELAY;
a = 1'b0; b = 1'b0; c = 1'b1; d = 1'b0; s0 = 1'b1; s1 = 1'b0;
#`DELAY;
a = 1'b0; b = 1'b0; c = 1'b0; d = 1'b0; s0 = 1'b1; s1 = 1'b1;
#`DELAY;
a = 1'b0; b = 1'b0; c = 1'b0; d = 1'b1; s0 = 1'b1; s1 = 1'b1;
#`DELAY;
end

initial
begin
$monitor("time = %2d,   a =%1b,  b =%1b,  c =%1b,  d =%1b   s0 =%1b  s1 =%1b  result =%1b", $time, a, b, c, d, s0, s1, result);
end

endmodule
```

```
Type  ID    Message
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
      22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
```

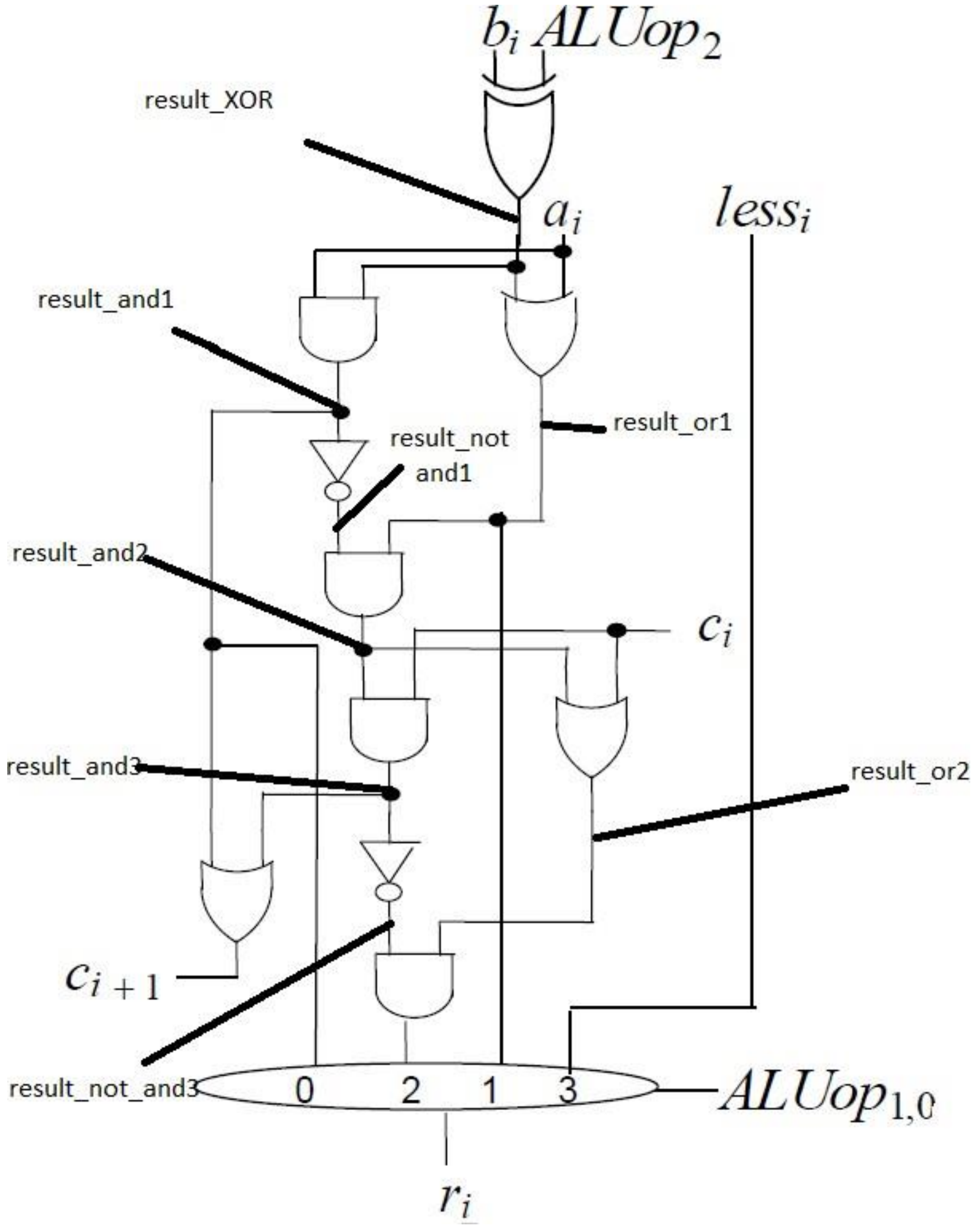


```
vlog -reportprogress 300 -work work C:/altera/13.1/workspace/hw02workspace/my_mux_testbench.v
# Model Technology ModelSim ALTERA vlog 10.1d Compiler 2012.11 Nov  2 2012
# -- Compiling module my_mux_testbench
#
# Top level modules:
#       my_mux_testbench
ModelSim> vsim work.my_mux_testbench
# vsim work.my_mux_testbench
# Loading work.my_mux_testbench
# Loading work.my_mux
VSIM 4> step -current
# time =  0,    a =0,   b =0,   c =0,   d =0    s0 =0   s1 =0   result =0
# time = 20,    a =1,   b =0,   c =0,   d =0    s0 =0   s1 =0   result =1
# time = 40,    a =0,   b =0,   c =0,   d =0    s0 =0   s1 =1   result =0
# time = 60,    a =0,   b =1,   c =0,   d =0    s0 =0   s1 =1   result =1
# time = 80,    a =0,   b =0,   c =0,   d =0    s0 =1   s1 =0   result =0
# time = 100,   a =0,   b =0,   c =1,   d =0    s0 =1   s1 =0   result =1
# time = 120,   a =0,   b =0,   c =0,   d =0    s0 =1   s1 =1   result =0
# time = 140,   a =0,   b =0,   c =0,   d =1    s0 =1   s1 =1   result =1

VSIM 5>
```

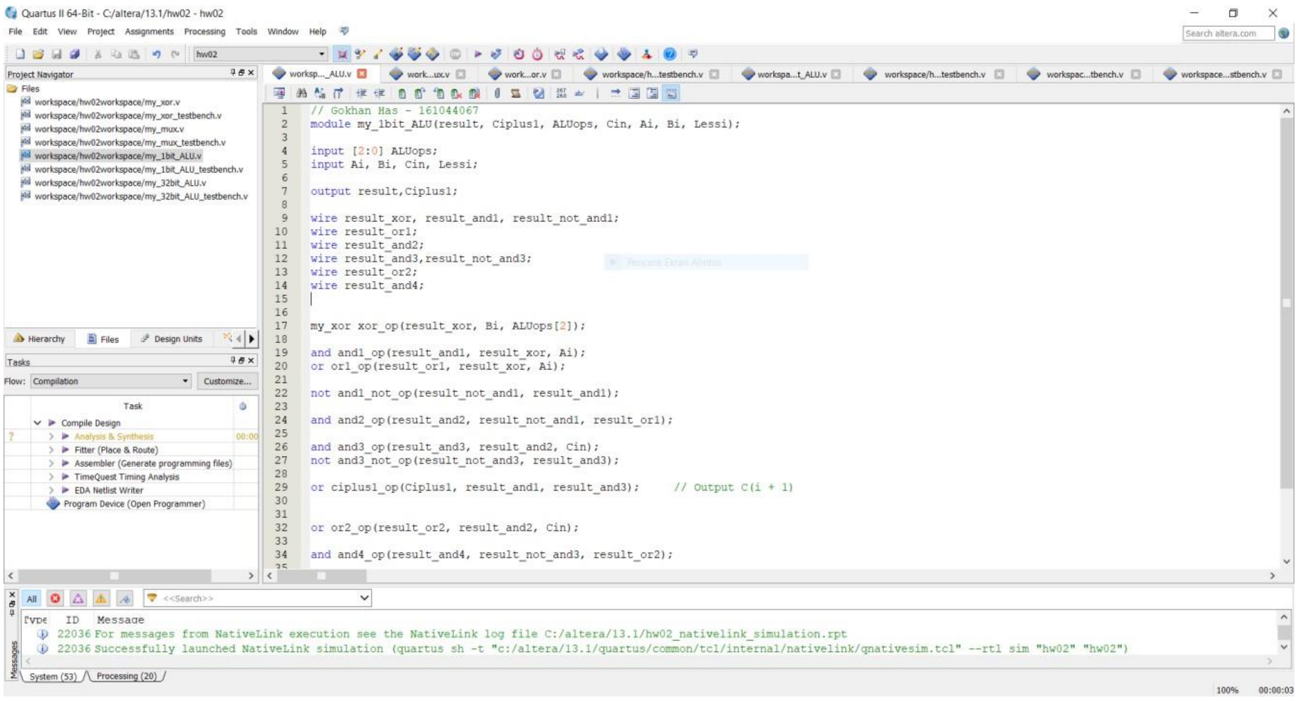


```
# time =  0,    a =0,   b =0,   c =0,   d =0    s0 =0   s1 =0   result =0
# time = 20,    a =1,   b =0,   c =0,   d =0    s0 =0   s1 =0   result =1
# time = 40,    a =0,   b =0,   c =0,   d =0    s0 =0   s1 =1   result =0
# time = 60,    a =0,   b =1,   c =0,   d =0    s0 =0   s1 =1   result =1
# time = 80,    a =0,   b =0,   c =0,   d =0    s0 =1   s1 =0   result =0
# time = 100,   a =0,   b =0,   c =1,   d =0    s0 =1   s1 =0   result =1
# time = 120,   a =0,   b =0,   c =0,   d =0    s0 =1   s1 =1   result =0
# time = 140,   a =0,   b =0,   c =0,   d =1    s0 =1   s1 =1   result =1
```

Now, we can design 1-bit-ALU.



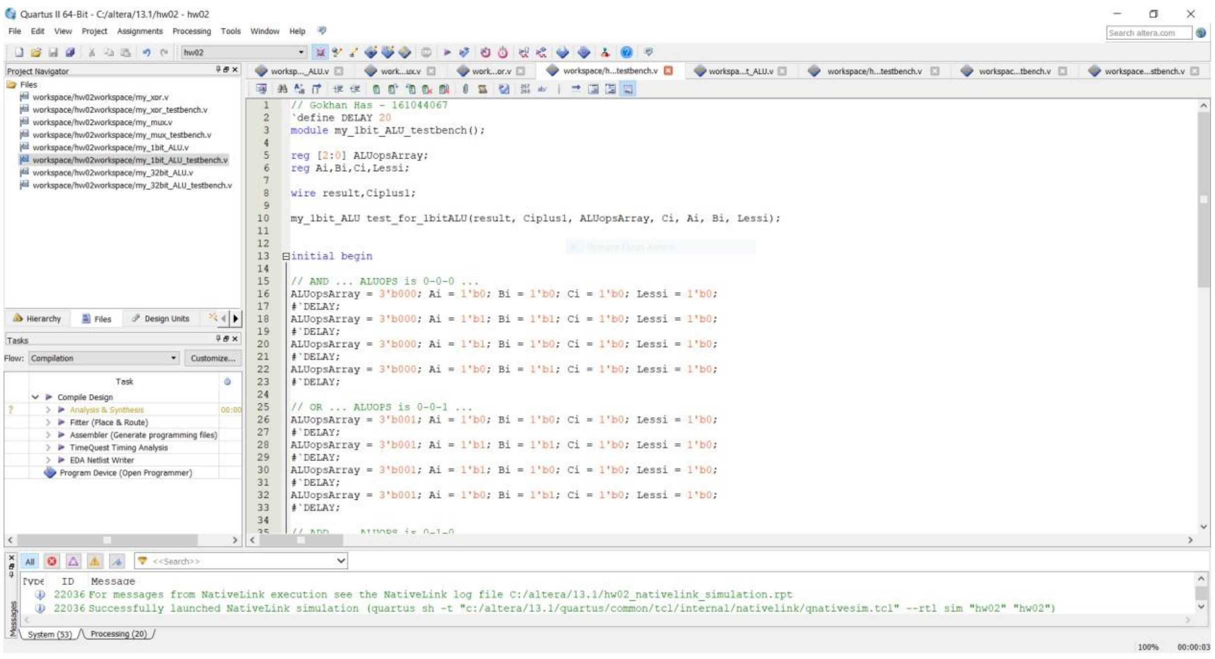
The variable names (wire's name) that I use in Quartus are described above.



```
// Gokhan Has - 161044067
module my_1bit_ALU(result, Ciplus1, ALUops, Cin, Ai, Bi, Lessi);

input [2:0] ALUops;
input Ai, Bi, Cin, Lessi;

output result,Ciplus1;

wire result_xor, result_and1, result_not_and1;
wire result_or1;
wire result_and2;
wire result_and3,result_not_and3;
wire result_or2;
wire result_and4;


my_xor xor_op(result_xor, Bi, ALUops[2]);

and and1_op(result_and1, result_xor, Ai);
or or1_op(result_or1, result_xor, Ai);

not and1_not_op(result_not_and1, result_and1);

and and2_op(result_and2, result_not_and1, result_or1);

and and3_op(result_and3, result_and2, Cin);
not and3_not_op(result_not_and3, result_and3);

or ciplus1_op(Ciplus1, result_and1, result_and3);     // Output C(i + 1)


or or2_op(result_or2, result_and2, Cin);

and and4_op(result_and4, result_not_and3, result_or2);
```

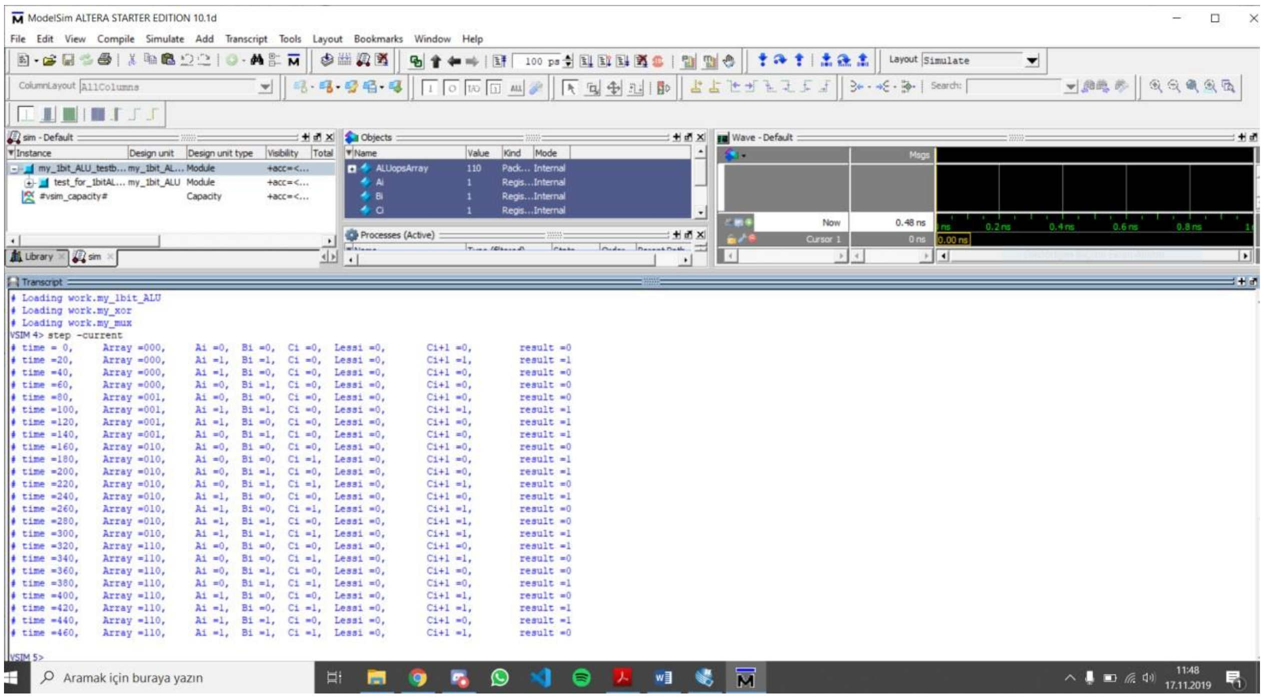
```
Type  ID    Message
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
      22036 Successfully launched NativeLink simulation (quartus sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl sim "hw02" "hw02")
```

1-Bit-ALU TestBench code:



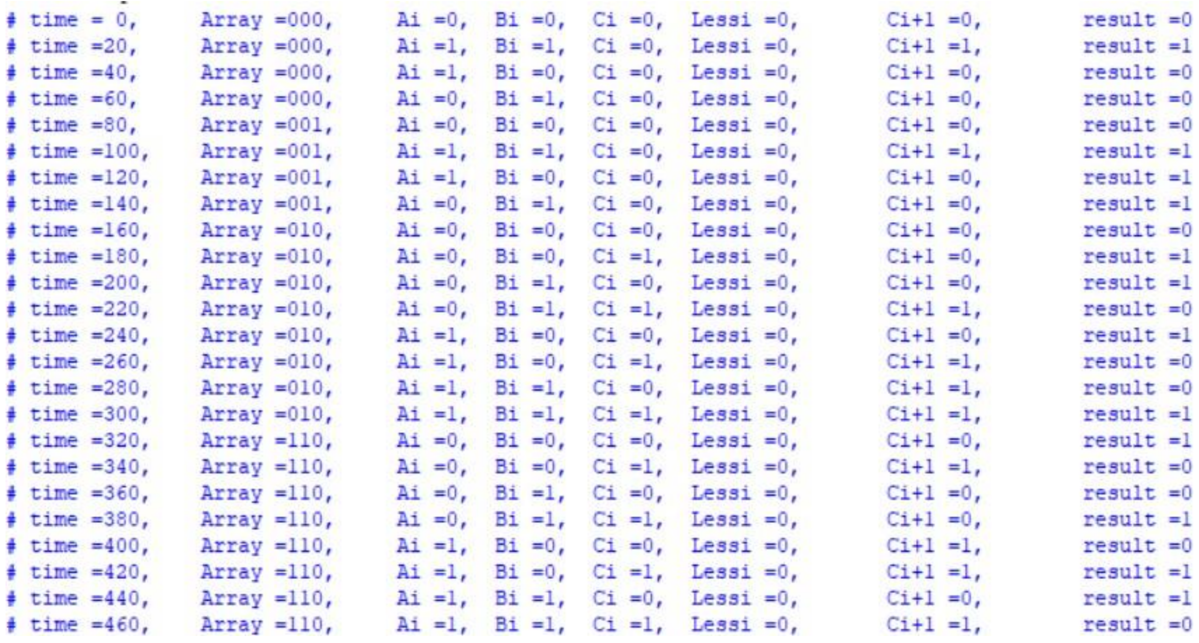
```
// Gokhan Has - 161044067
`define DELAY 20
module my_1bit_ALU_testbench();

reg [2:0] ALUopsArray;
reg Ai,Bi,Ci,Lessi;

wire result,Ciplus1;

my_1bit_ALU test_for_1bitALU(result, Ciplus1, ALUopsArray, Ci, Ai, Bi, Lessi);


initial begin

// AND ... ALUOPS is 0-0-0 ...
ALUopsArray = 3'b000; Ai = 1'b0; Bi = 1'b0; Ci = 1'b0; Lessi = 1'b0;
#`DELAY;
ALUopsArray = 3'b000; Ai = 1'b1; Bi = 1'b1; Ci = 1'b0; Lessi = 1'b0;
#`DELAY;
ALUopsArray = 3'b000; Ai = 1'b1; Bi = 1'b0; Ci = 1'b0; Lessi = 1'b0;
#`DELAY;
ALUopsArray = 3'b000; Ai = 1'b0; Bi = 1'b1; Ci = 1'b0; Lessi = 1'b0;
#`DELAY;

// OR ... ALUOPS is 0-0-1 ...
ALUopsArray = 3'b001; Ai = 1'b0; Bi = 1'b0; Ci = 1'b0; Lessi = 1'b0;
#`DELAY;
ALUopsArray = 3'b001; Ai = 1'b1; Bi = 1'b1; Ci = 1'b0; Lessi = 1'b0;
#`DELAY;
ALUopsArray = 3'b001; Ai = 1'b1; Bi = 1'b0; Ci = 1'b0; Lessi = 1'b0;
#`DELAY;
ALUopsArray = 3'b001; Ai = 1'b0; Bi = 1'b1; Ci = 1'b0; Lessi = 1'b0;
#`DELAY;

// ADD ... ALUOPS is 0-1-0
```

```
Type  ID    Message
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
      22036 Successfully launched NativeLink simulation (quartus sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl sim "hw02" "hw02")
```

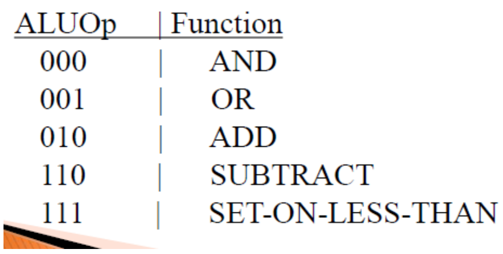

ModelSim ALTERA STARTER EDITION 10.1d
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
sim - Default
Objects
Wave - Default
Transcript
Aramak için buraya yazın
11:48
17.11.2019

Name	Value	Kind	Mode
ALUopsArray	110	Pack...	Internal
Ai	1	Regis...	Internal
Bi	1	Regis...	Internal
Ci	1	Regis...	Internal

```
# Loading work.my_1bit_ALU
# Loading work.my_xor
# Loading work.my_mux
VSIM 4> step -current
# time = 0,    Array =000,   Ai =0,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =20,    Array =000,   Ai =1,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =1,      result =1
# time =40,    Array =000,   Ai =1,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =60,    Array =000,   Ai =0,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =80,    Array =001,   Ai =0,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =100,   Array =001,   Ai =1,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =1,      result =1
# time =120,   Array =001,   Ai =1,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =140,   Array =001,   Ai =0,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =160,   Array =010,   Ai =0,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =180,   Array =010,   Ai =0,  Bi =0,  Ci =1,  Lessi =0,      Ci+1 =0,      result =1
# time =200,   Array =010,   Ai =0,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =220,   Array =010,   Ai =0,  Bi =1,  Ci =1,  Lessi =0,      Ci+1 =1,      result =0
# time =240,   Array =010,   Ai =1,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =260,   Array =010,   Ai =1,  Bi =0,  Ci =1,  Lessi =0,      Ci+1 =1,      result =0
# time =280,   Array =010,   Ai =1,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =1,      result =0
# time =300,   Array =010,   Ai =1,  Bi =1,  Ci =1,  Lessi =0,      Ci+1 =1,      result =1
# time =320,   Array =110,   Ai =0,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =340,   Array =110,   Ai =0,  Bi =0,  Ci =1,  Lessi =0,      Ci+1 =1,      result =0
# time =360,   Array =110,   Ai =0,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =380,   Array =110,   Ai =0,  Bi =1,  Ci =1,  Lessi =0,      Ci+1 =0,      result =1
# time =400,   Array =110,   Ai =1,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =1,      result =0
# time =420,   Array =110,   Ai =1,  Bi =0,  Ci =1,  Lessi =0,      Ci+1 =1,      result =1
# time =440,   Array =110,   Ai =1,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =460,   Array =110,   Ai =1,  Bi =1,  Ci =1,  Lessi =0,      Ci+1 =1,      result =0
VSIM 5>
```



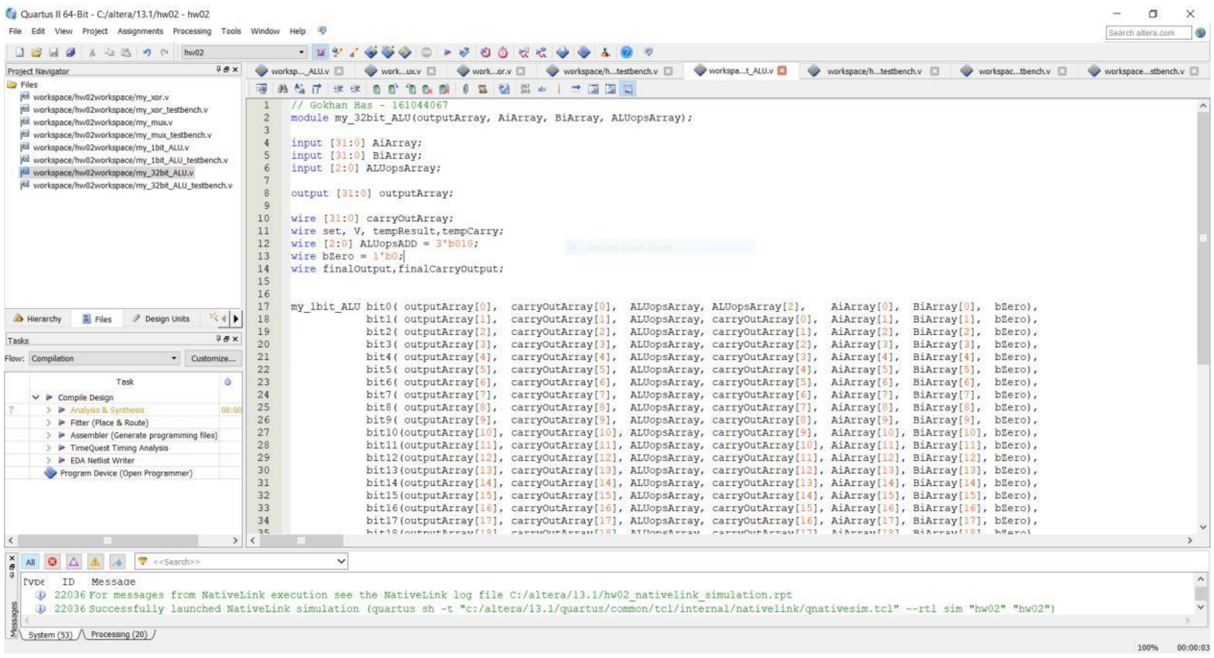
```
# time = 0,    Array =000,   Ai =0,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =20,    Array =000,   Ai =1,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =1,      result =1
# time =40,    Array =000,   Ai =1,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =60,    Array =000,   Ai =0,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =80,    Array =001,   Ai =0,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =100,   Array =001,   Ai =1,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =1,      result =1
# time =120,   Array =001,   Ai =1,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =140,   Array =001,   Ai =0,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =160,   Array =010,   Ai =0,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =180,   Array =010,   Ai =0,  Bi =0,  Ci =1,  Lessi =0,      Ci+1 =0,      result =1
# time =200,   Array =010,   Ai =0,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =220,   Array =010,   Ai =0,  Bi =1,  Ci =1,  Lessi =0,      Ci+1 =1,      result =0
# time =240,   Array =010,   Ai =1,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =260,   Array =010,   Ai =1,  Bi =0,  Ci =1,  Lessi =0,      Ci+1 =1,      result =0
# time =280,   Array =010,   Ai =1,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =1,      result =0
# time =300,   Array =010,   Ai =1,  Bi =1,  Ci =1,  Lessi =0,      Ci+1 =1,      result =1
# time =320,   Array =110,   Ai =0,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =340,   Array =110,   Ai =0,  Bi =0,  Ci =1,  Lessi =0,      Ci+1 =1,      result =0
# time =360,   Array =110,   Ai =0,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =0
# time =380,   Array =110,   Ai =0,  Bi =1,  Ci =1,  Lessi =0,      Ci+1 =0,      result =1
# time =400,   Array =110,   Ai =1,  Bi =0,  Ci =0,  Lessi =0,      Ci+1 =1,      result =0
# time =420,   Array =110,   Ai =1,  Bi =0,  Ci =1,  Lessi =0,      Ci+1 =1,      result =1
# time =440,   Array =110,   Ai =1,  Bi =1,  Ci =0,  Lessi =0,      Ci+1 =0,      result =1
# time =460,   Array =110,   Ai =1,  Bi =1,  Ci =1,  Lessi =0,      Ci+1 =1,      result =0
```



ALUOp	Function
000	AND
001	OR
010	ADD
110	SUBTRACT
111	SET-ON-LESS-THAN

Array is ALUOPScode.

32-bits-ALU :



Quartus II 64-Bit - C:/altera/13.1/hw02 - hw02

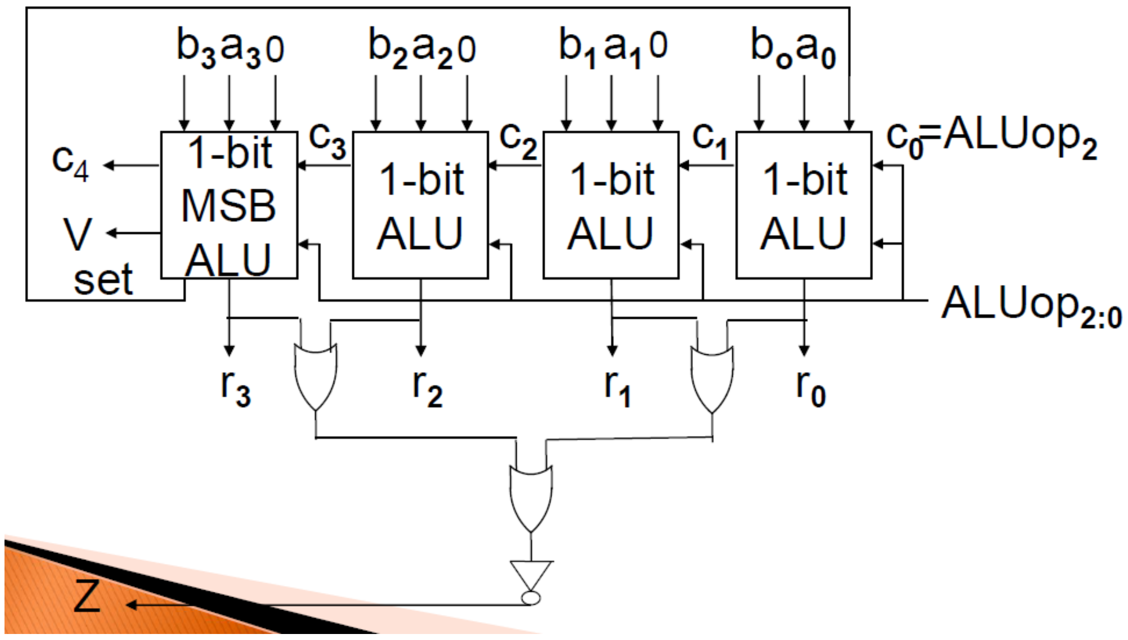
Project Navigator — Files:
- workspace/hw02workspace/my_xor.v
- workspace/hw02workspace/my_xor_testbench.v
- workspace/hw02workspace/my_mux.v
- workspace/hw02workspace/my_mux_testbench.v
- workspace/hw02workspace/my_1bit_ALU.v
- workspace/hw02workspace/my_1bit_ALU_testbench.v
- workspace/hw02workspace/my_32bit_ALU.v
- workspace/hw02workspace/my_32bit_ALU_testbench.v

```
// Gokhan Has - 161044067
module my_32bit_ALU(outputArray, AiArray, BiArray, ALUopsArray);

input [31:0] AiArray;
input [31:0] BiArray;
input [2:0] ALUopsArray;

output [31:0] outputArray;

wire [31:0] carryOutArray;
wire set, V, tempResult,tempCarry;
wire [2:0] ALUopsADD = 3'b010;
wire bZero = 1'b0;
wire finalOutput,finalCarryOutput;


my_1bit_ALU bit0( outputArray[0],  carryOutArray[0],  ALUopsArray, ALUopsArray[2],    AiArray[0],  BiArray[0],  bZero),
            bit1( outputArray[1],  carryOutArray[1],  ALUopsArray, carryOutArray[0],  AiArray[1],  BiArray[1],  bZero),
            bit2( outputArray[2],  carryOutArray[2],  ALUopsArray, carryOutArray[1],  AiArray[2],  BiArray[2],  bZero),
            bit3( outputArray[3],  carryOutArray[3],  ALUopsArray, carryOutArray[2],  AiArray[3],  BiArray[3],  bZero),
            bit4( outputArray[4],  carryOutArray[4],  ALUopsArray, carryOutArray[3],  AiArray[4],  BiArray[4],  bZero),
            bit5( outputArray[5],  carryOutArray[5],  ALUopsArray, carryOutArray[4],  AiArray[5],  BiArray[5],  bZero),
            bit6( outputArray[6],  carryOutArray[6],  ALUopsArray, carryOutArray[5],  AiArray[6],  BiArray[6],  bZero),
            bit7( outputArray[7],  carryOutArray[7],  ALUopsArray, carryOutArray[6],  AiArray[7],  BiArray[7],  bZero),
            bit8( outputArray[8],  carryOutArray[8],  ALUopsArray, carryOutArray[7],  AiArray[8],  BiArray[8],  bZero),
            bit9( outputArray[9],  carryOutArray[9],  ALUopsArray, carryOutArray[8],  AiArray[9],  BiArray[9],  bZero),
            bit10(outputArray[10], carryOutArray[10], ALUopsArray, carryOutArray[9],  AiArray[10], BiArray[10], bZero),
            bit11(outputArray[11], carryOutArray[11], ALUopsArray, carryOutArray[10], AiArray[11], BiArray[11], bZero),
            bit12(outputArray[12], carryOutArray[12], ALUopsArray, carryOutArray[11], AiArray[12], BiArray[12], bZero),
            bit13(outputArray[13], carryOutArray[13], ALUopsArray, carryOutArray[12], AiArray[13], BiArray[13], bZero),
            bit14(outputArray[14], carryOutArray[14], ALUopsArray, carryOutArray[13], AiArray[14], BiArray[14], bZero),
            bit15(outputArray[15], carryOutArray[15], ALUopsArray, carryOutArray[14], AiArray[15], BiArray[15], bZero),
            bit16(outputArray[16], carryOutArray[16], ALUopsArray, carryOutArray[15], AiArray[16], BiArray[16], bZero),
            bit17(outputArray[17], carryOutArray[17], ALUopsArray, carryOutArray[16], AiArray[17], BiArray[17], bZero),
            bit18(outputArray[18], carryOutArray[18], ALUopsArray, carryOutArray[17], AiArray[18], BiArray[18], bZero),
```

```
Type  ID    Message
      22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
      22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
```

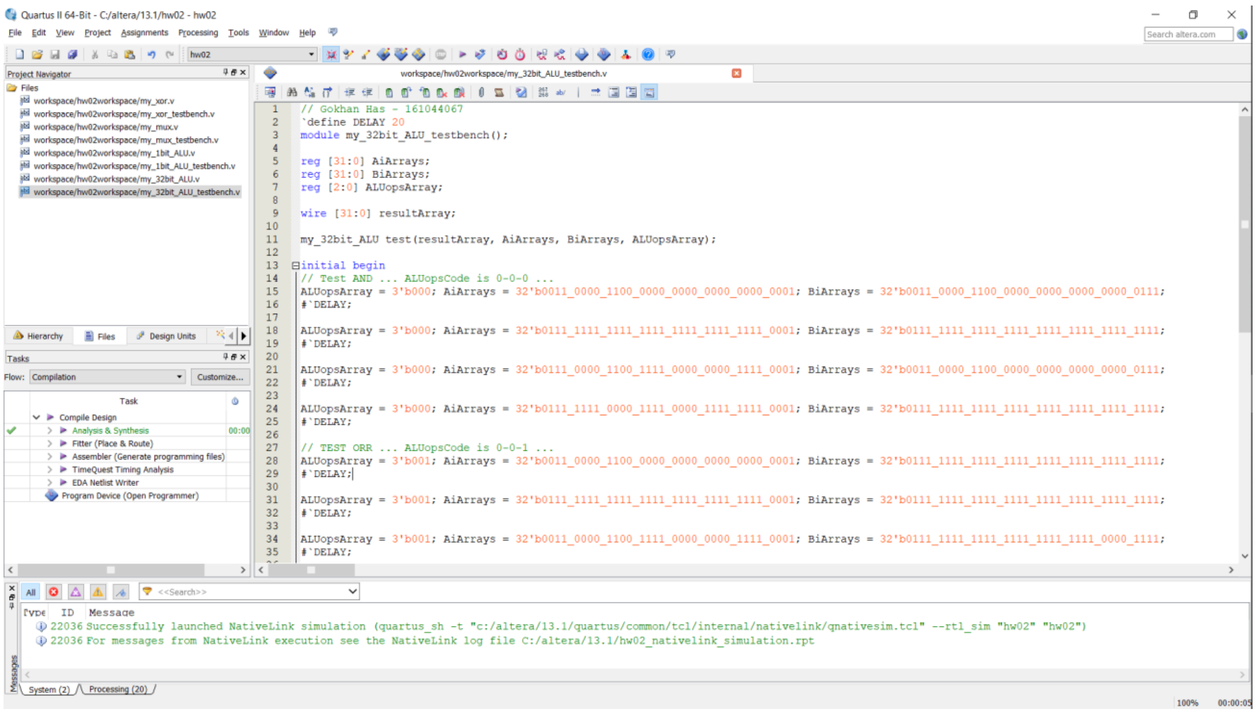
```
my_xor calculate_V(V, carryOutArray[31], carryOutArray[30]);
my_1bit_ALU calculate_ADD(tempResult2, tempCarry, ALUopsSub , carryOutArray[30], AiArray[31], BiArray[31], bZero);
my_xor calculate_Set(set, V, tempResult2);
my_1bit_ALU finalBit(outputArray[0],  finalCarryOutput,  ALUopsArray, ALUopsArray[2],    AiArray[0],  BiArray[0],   set);
```

Here is important. 1 bits from MSB. The ALU for the MSB must also detect overflow and indicate the sign of the result.



Attention the C0 is equal to ALUopArray's third element. (ALUOPS[2])

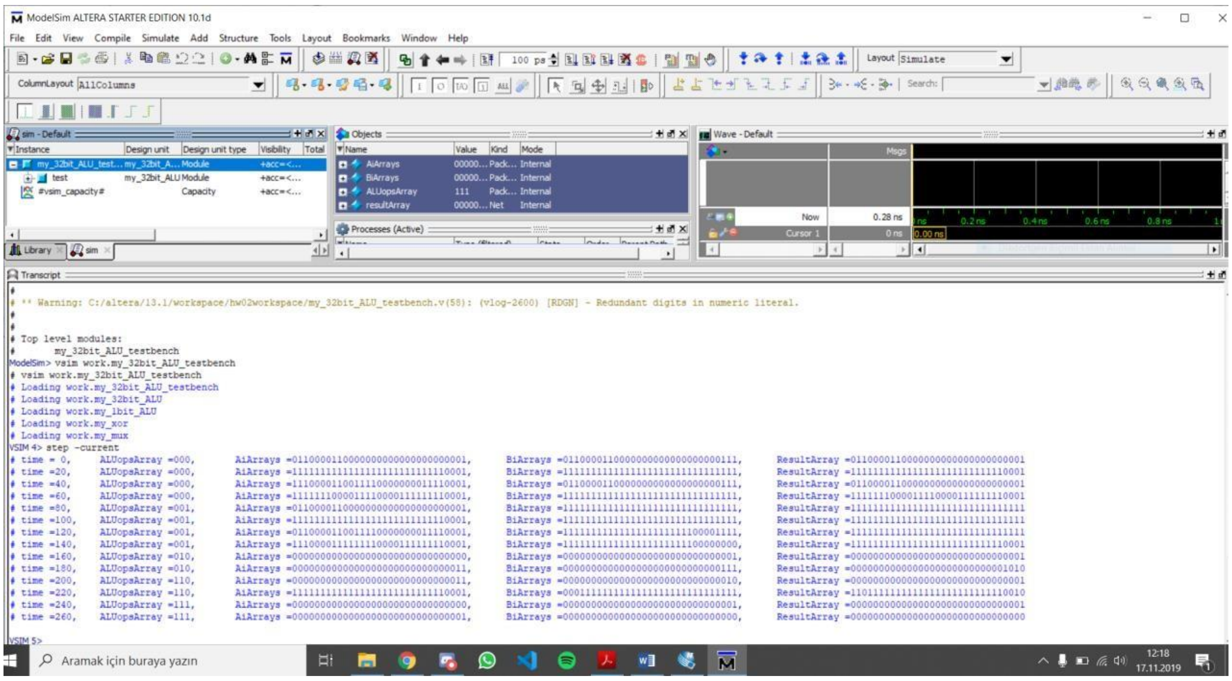
32-bit-ALU- TestBenchCode:



```
// Gokhan Has - 161044067
`define DELAY 20
module my_32bit_ALU_testbench();

reg [31:0] AiArrays;
reg [31:0] BiArrays;
reg [2:0] ALUopsArray;

wire [31:0] resultArray;

my_32bit_ALU test(resultArray, AiArrays, BiArrays, ALUopsArray);

initial begin
// Test AND ... ALUopsCode is 0-0-0 ...
ALUopsArray = 3'b000; AiArrays = 32'b0011_0000_1100_0000_0000_0000_0000_0001; BiArrays = 32'b0011_0000_1100_0000_0000_0000_0000_0111;
#`DELAY;

ALUopsArray = 3'b000; AiArrays = 32'b0111_1111_1111_1111_1111_1111_1111_0001; BiArrays = 32'b0111_1111_1111_1111_1111_1111_1111_1111;
#`DELAY;

ALUopsArray = 3'b000; AiArrays = 32'b0111_0000_1100_1111_0000_0000_1111_0001; BiArrays = 32'b0011_0000_1100_0000_0000_0000_0000_0111;
#`DELAY;

ALUopsArray = 3'b000; AiArrays = 32'b0111_1111_0000_1111_0000_1111_1111_0001; BiArrays = 32'b0111_1111_1111_1111_1111_1111_1111_1111;
#`DELAY;

// TEST ORR ... ALUopsCode is 0-0-1 ...
ALUopsArray = 3'b001; AiArrays = 32'b0011_0000_1100_0000_0000_0000_0000_0001; BiArrays = 32'b0111_1111_1111_1111_1111_1111_1111_1111;
#`DELAY;

ALUopsArray = 3'b001; AiArrays = 32'b0111_1111_1111_1111_1111_1111_1111_0001; BiArrays = 32'b0111_1111_1111_1111_1111_1111_1111_1111;
#`DELAY;

ALUopsArray = 3'b001; AiArrays = 32'b0011_0000_1100_1111_0000_0000_1111_0001; BiArrays = 32'b0111_1111_1111_1111_1111_1111_0000_1111;
#`DELAY;
```

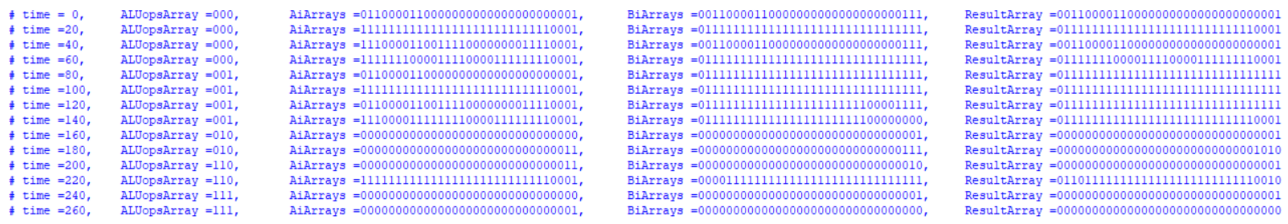
```
Type  ID  Message
22036 Successfully launched NativeLink simulation (quartus_sh -t "c:/altera/13.1/quartus/common/tcl/internal/nativelink/qnativesim.tcl" --rtl_sim "hw02" "hw02")
22036 For messages from NativeLink execution see the NativeLink log file C:/altera/13.1/hw02_nativelink_simulation.rpt
```

You can see which process has been tested by looking at the OP codes. Each process was tested *at least* 2 times.



```
#
# ** Warning: C:/altera/13.1/workspace/hw02workspace/my_32bit_ALU_testbench.v(58): (vlog-2600) [RDGN] - Redundant digits in numeric literal.
#
#
# Top level modules:
#       my_32bit_ALU_testbench
ModelSim> vsim work.my_32bit_ALU_testbench
# vsim work.my_32bit_ALU_testbench
# Loading work.my_32bit_ALU_testbench
# Loading work.my_32bit_ALU
# Loading work.my_1bit_ALU
# Loading work.my_xor
# Loading work.my_mux
VSIM 4> step -current
# time = 0,    ALUopsArray =000,    AiArrays =01100001100000000000000000000001,    BiArrays =01100001100000000000000000000111,    ResultArray =01100001100000000000000000000001
# time =20,    ALUopsArray =000,    AiArrays =11111111111111111111111111110001,    BiArrays =11111111111111111111111111111111,    ResultArray =11111111111111111111111111110001
# time =40,    ALUopsArray =000,    AiArrays =11100001100111100000000011110001,    BiArrays =01100001100000000000000000000111,    ResultArray =01100001100000000000000000000001
# time =60,    ALUopsArray =000,    AiArrays =11111110000111100001111111110001,    BiArrays =11111111111111111111111111111111,    ResultArray =11111110000111100001111111110001
# time =80,    ALUopsArray =001,    AiArrays =01100001100000000000000000000001,    BiArrays =11111111111111111111111111111111,    ResultArray =11111111111111111111111111111111
# time =100,   ALUopsArray =001,    AiArrays =11111111111111111111111111110001,    BiArrays =11111111111111111111111111111111,    ResultArray =11111111111111111111111111111111
# time =120,   ALUopsArray =001,    AiArrays =01100001100111100000000011110001,    BiArrays =11111111111111111111111100001111,    ResultArray =11111111111111111111111111111111
# time =140,   ALUopsArray =001,    AiArrays =11100001111111100001111111110001,    BiArrays =11111111111111111111111100000000,    ResultArray =11111111111111111111111111110001
# time =160,   ALUopsArray =010,    AiArrays =00000000000000000000000000000000,    BiArrays =00000000000000000000000000000001,    ResultArray =00000000000000000000000000000001
# time =180,   ALUopsArray =010,    AiArrays =00000000000000000000000000000011,    BiArrays =00000000000000000000000000000111,    ResultArray =00000000000000000000000000001010
# time =200,   ALUopsArray =110,    AiArrays =00000000000000000000000000000011,    BiArrays =00000000000000000000000000000010,    ResultArray =00000000000000000000000000000001
# time =220,   ALUopsArray =110,    AiArrays =11111111111111111111111111110001,    BiArrays =00011111111111111111111111111111,    ResultArray =11011111111111111111111111110010
# time =240,   ALUopsArray =111,    AiArrays =00000000000000000000000000000000,    BiArrays =00000000000000000000000000000001,    ResultArray =00000000000000000000000000000001
# time =260,   ALUopsArray =111,    AiArrays =00000000000000000000000000000001,    BiArrays =00000000000000000000000000000000,    ResultArray =00000000000000000000000000000000

VSIM 5>
```

32-Bits- Testers …



```
# time = 0,    ALUopsArray =000,     AiArrays =01100001100000000000000000000001,     BiArrays =00110000110000000000000000000111,     ResultArray =00110000110000000000000000000001
# time =20,    ALUopsArray =000,     AiArrays =11111111111111111111111111110001,     BiArrays =01111111111111111111111111111111,     ResultArray =01111111111111111111111111110001
# time =40,    ALUopsArray =000,     AiArrays =11100001100111100000000011110001,     BiArrays =00110000110000000000000000000111,     ResultArray =00110000110000000000000000000001
# time =60,    ALUopsArray =000,     AiArrays =11111110000111100001111111110001,     BiArrays =01111111111111111111111111111111,     ResultArray =01111110000111100001111111110001
# time =80,    ALUopsArray =001,     AiArrays =01100001100000000000000000000001,     BiArrays =01111111111111111111111111111111,     ResultArray =01111111111111111111111111111111
# time =100,   ALUopsArray =001,     AiArrays =11111111111111111111111111110001,     BiArrays =01111111111111111111111111111111,     ResultArray =01111111111111111111111111111111
# time =120,   ALUopsArray =001,     AiArrays =01100001100111100000000011110001,     BiArrays =01111111111111111111111100001111,     ResultArray =01111111111111111111111111111111
# time =140,   ALUopsArray =001,     AiArrays =11100001111111100001111111110001,     BiArrays =01111111111111111111111100000000,     ResultArray =01111111111111111111111111110001
# time =160,   ALUopsArray =010,     AiArrays =00000000000000000000000000000000,     BiArrays =00000000000000000000000000000001,     ResultArray =00000000000000000000000000000001
# time =180,   ALUopsArray =010,     AiArrays =00000000000000000000000000000011,     BiArrays =00000000000000000000000000000111,     ResultArray =00000000000000000000000000001010
# time =200,   ALUopsArray =110,     AiArrays =00000000000000000000000000000011,     BiArrays =00000000000000000000000000000010,     ResultArray =00000000000000000000000000000001
# time =220,   ALUopsArray =110,     AiArrays =11111111111111111111111111110001,     BiArrays =00011111111111111111111111111111,     ResultArray =01101111111111111111111111110010
# time =240,   ALUopsArray =111,     AiArrays =00000000000000000000000000000000,     BiArrays =00000000000000000000000000000001,     ResultArray =00000000000000000000000000000001
# time =260,   ALUopsArray =111,     AiArrays =00000000000000000000000000000001,     BiArrays =00000000000000000000000000000000,     ResultArray =00000000000000000000000000000000
```