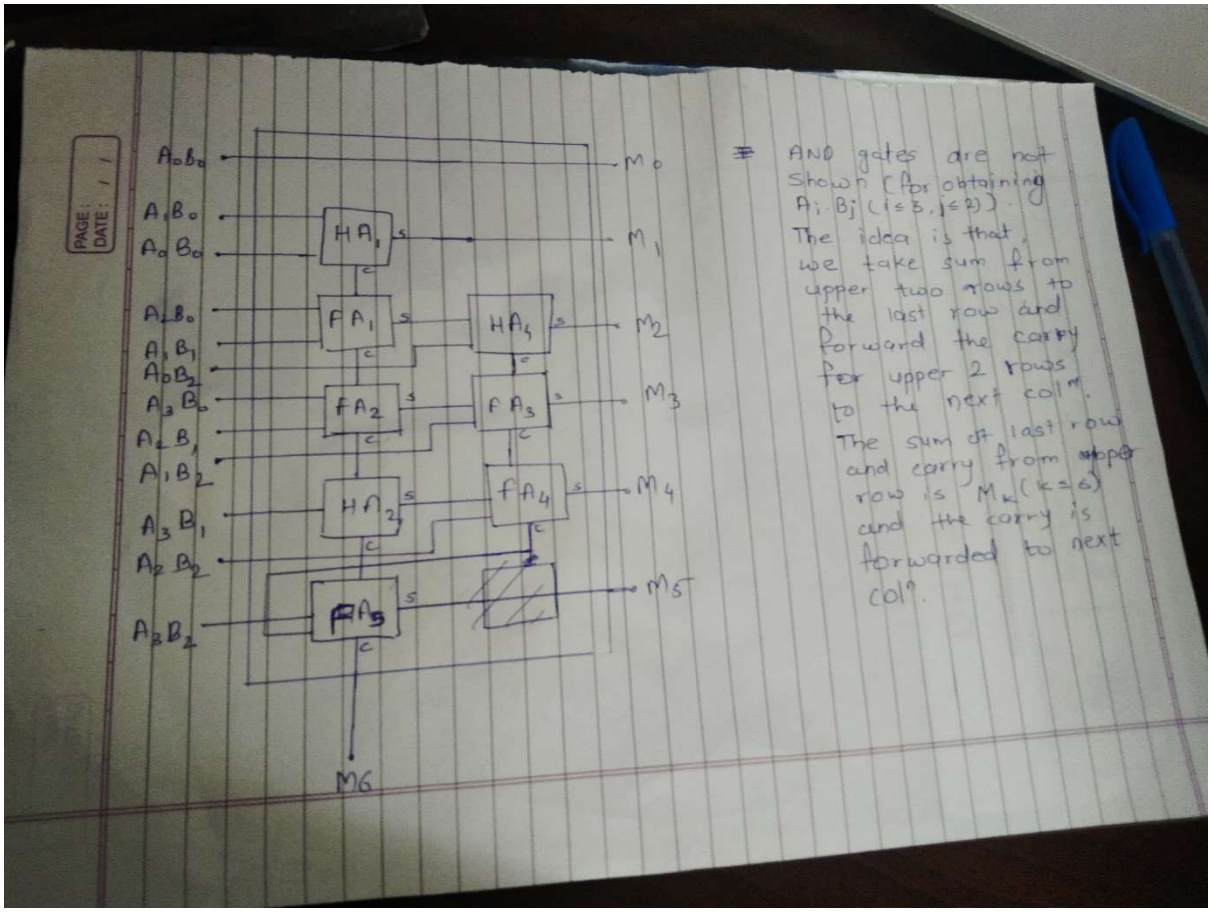
EXP_03 : 4 bit by 3 bit MULTIPLIER

In this lab, we designed a 4bit by 3bit Multiplier , which multiplies a 4 bit binary no. by a 3 bit binary no. .

Design includes 12 AND gates(for product terms) , 4 Half Adders and 5 Full Adders .



VHDL desc for Multiplier is :

```
 entity Multiplier_4x1  is
  port (B2,B1,B0,A3,A2,A1,A0: in std_logic;
      Y6,Y5,Y4,Y3,Y2,Y1,Y0 : out std_logic);
end entity Multiplier_4x1;
architecture Struct of Multiplier_4x1 is
  signal S11,S10,S9,S8,S7,S6,S5,S4,S3,S2,S1,t10,t9,t8,t7,t6,t5,t4,t3,t2,t1: std_logic;
begin
  -- component instances
  N1 : AND_2 port map(A=>A0,B=>B0,Y=>Y0);
  N2 : AND_2 port map(A=>A1,B=>B0,Y=>S1);
  N3 : AND_2 port map(A=>A2,B=>B0,Y=>S2);
  N4 : AND_2 port map(A=>A3,B=>B0,Y=>S3);
  N5 : AND_2 port map(A=>A0,B=>B1,Y=>S4);
  N6 : AND_2 port map(A=>A1,B=>B1,Y=>S5);
  N7 : AND_2 port map(A=>A2,B=>B1,Y=>S6);
  N8 : AND_2 port map(A=>A3,B=>B1,Y=>S7);
```

```
 N9 : AND_2 port map(A=>A0,B=>B2,Y=>S8);
 N10 :AND_2 port map(A=>A1,B=>B2,Y=>S9);
 N11 :AND_2 port map(A=>A2,B=>B2,Y=>S10);
 N12 :AND_2 port map(A=>A3,B=>B2,Y=>S11);
 HA1 : HALF_ADDER port map(A=>S1,B=>S4,S=>Y1,C=>t1);
 FA1 : FULL_ADDER port map(A=>S2,B=>S5,Cin=>t1,S=>t2,C=>t3);
 HA2 : HALF_ADDER port map(A=>t2,B=>S8,S=>Y2,C=>t4);
 FA2 : FULL_ADDER port map(A=>S3,B=>S6,Cin=>t3,S=>t5,C=>t6);
 FA3 : FULL_ADDER port map(A=>t5,B=>S9,Cin=>t4,S=>Y3,C=>t7);
 HA4 : HALF_ADDER port map(A=>S7,B=>t6,S=>t8,C=>t9);
 FA4 : FULL_ADDER port map(A=>S10,B=>t8,Cin=>t7,S=>Y4,C=>t10);
 FA5 : FULL_ADDER port map(A=>S11,B=>t9,Cin=>t10,S=>Y5,C=>Y6);
end Struct;
```

Input and Output format of DUT:

Input MSB = A3

Input LSB = B0

Output MSB = Y6

Output LSB = Y0

Input-Output format in Tracefile :-

<A3 A2 A1 A0 B2 B1 B0> <Y6 Y5 Y4 Y3 Y2 Y1 Y0>

Testcases from TRACEFILE :-

```
0000000 0000000 1111111
 0000001 0000000 1111111
0000010 0000000 1111111
```

0000011 0000000 1111111

0000100 0000000 1111111

0000101 0000000 1111111

0000110 0000000 1111111

0000111 0000000 1111111

0001000 0000000 1111111

0001001 0000001 1111111

0001010 0000010 1111111

0001011 0000011 1111111

0001100 0000100 1111111

0001101 0000101 1111111

0001110 0000110 1111111

0001111 0000111 1111111

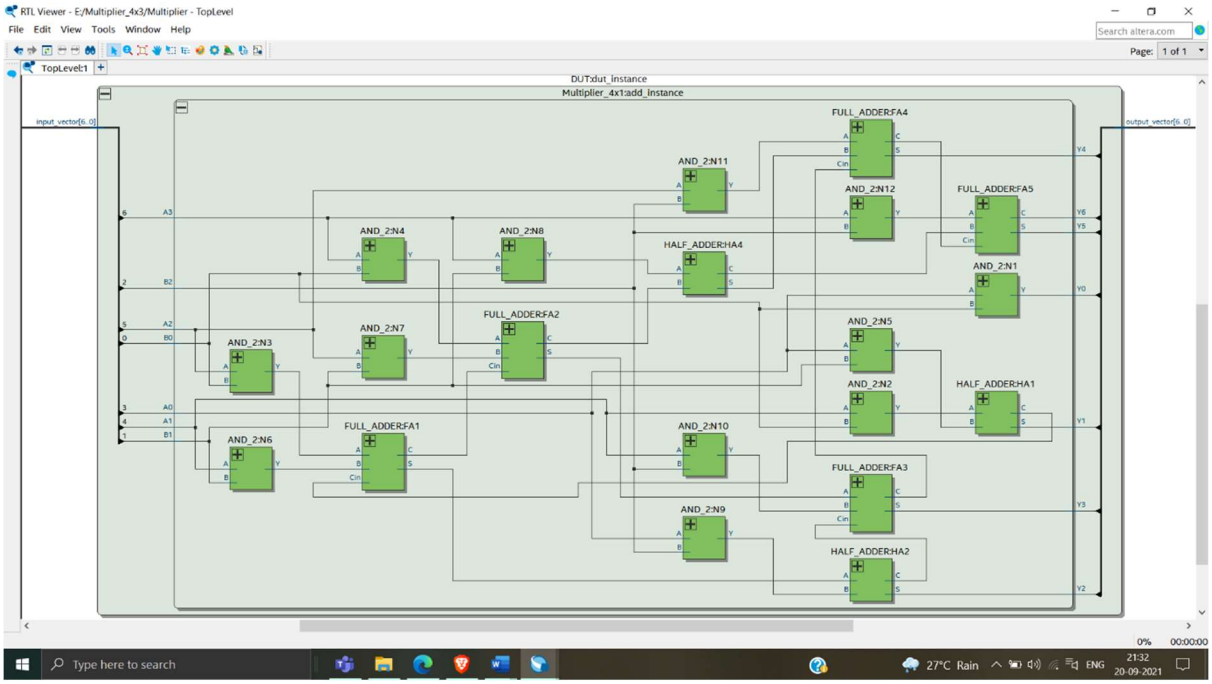
0010000 0000000 1111111

0010001 0000010 1111111

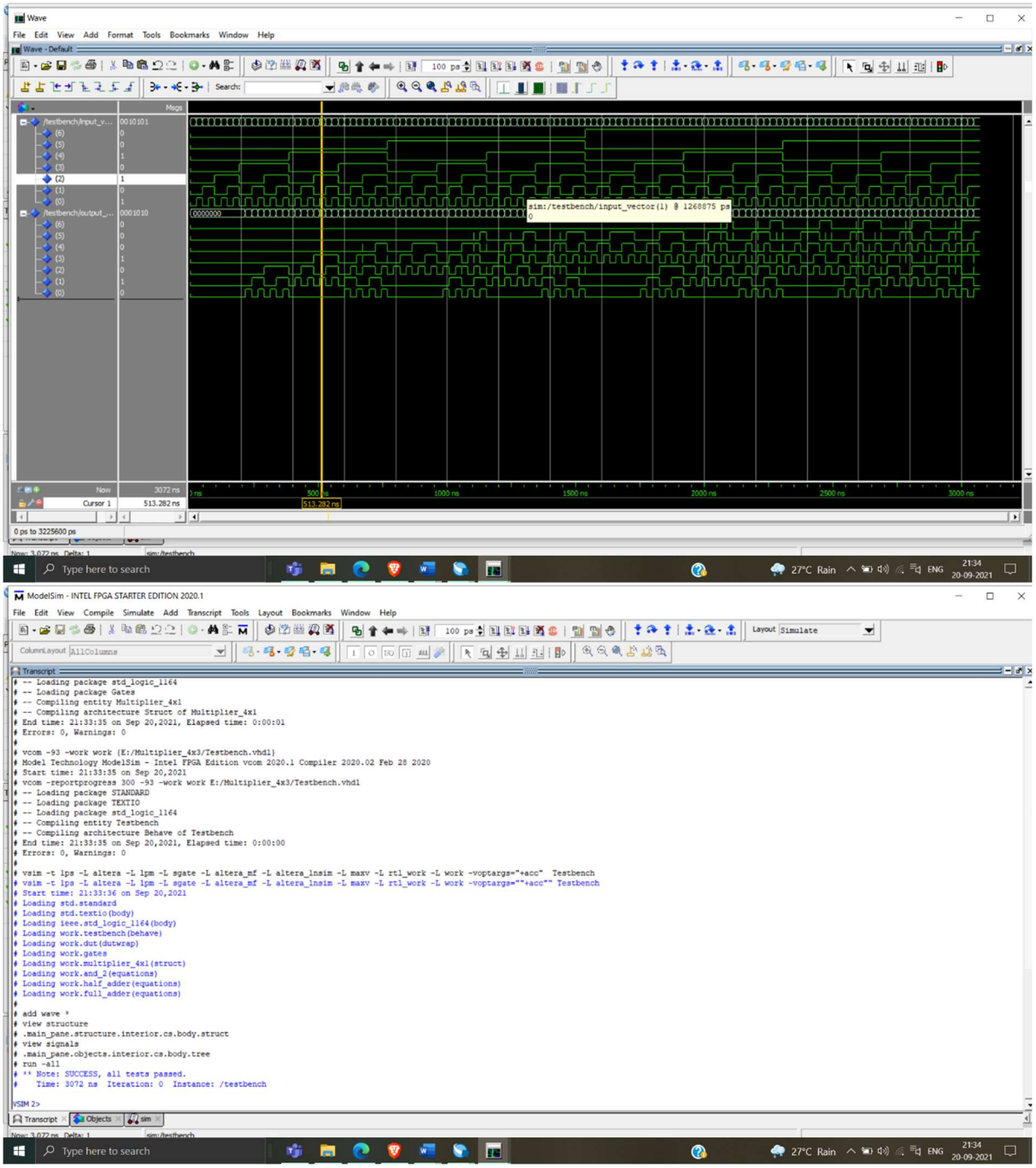
0010010 0000100 1111111

0010011 0000110 1111111

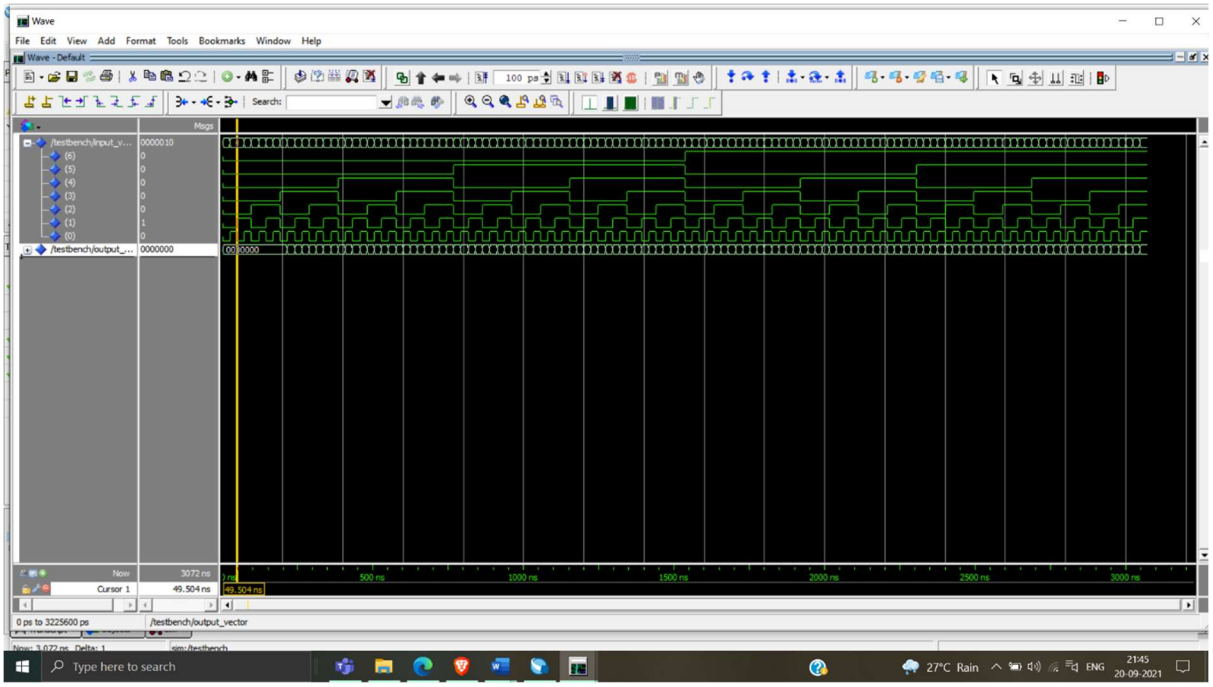
RTL View of Multiplier :

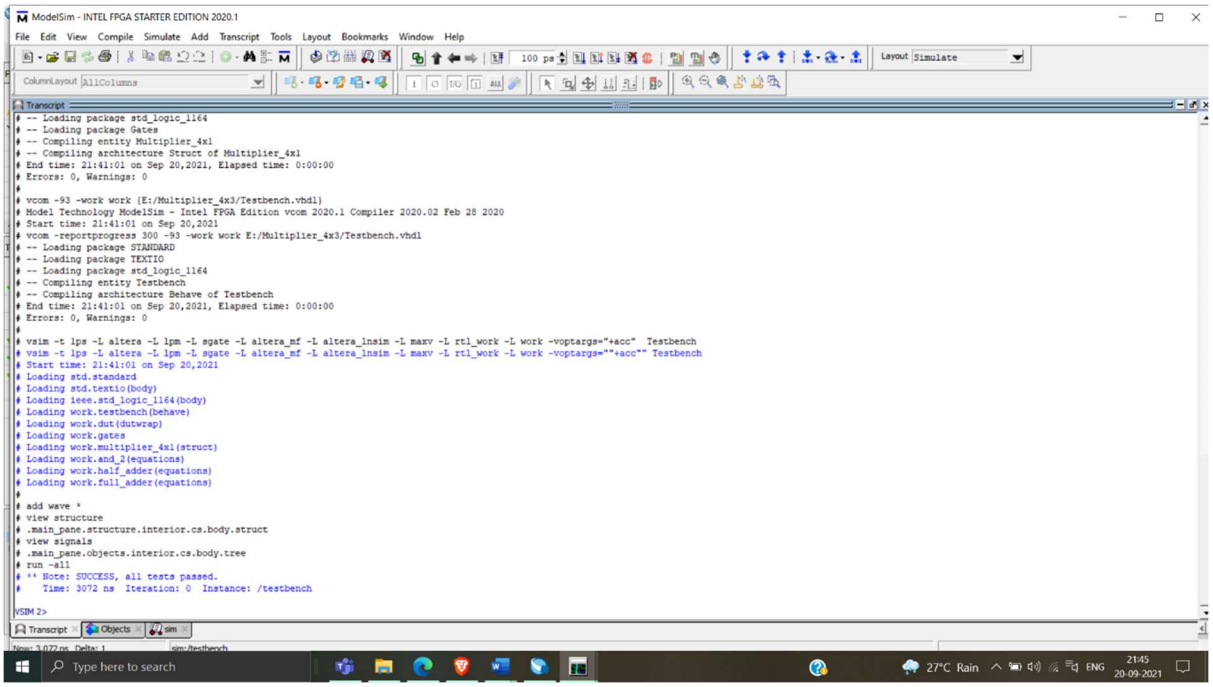


RTL Simulation :

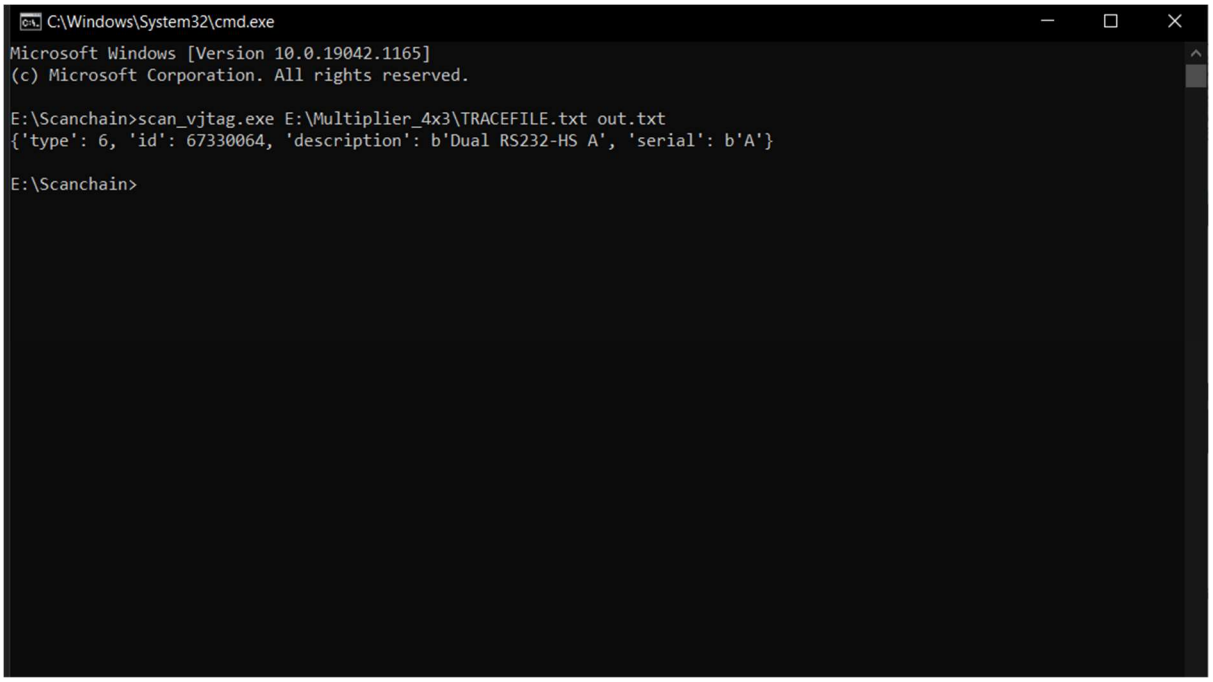


GATE level Simulation:


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
/testbench/input_v...
/testbench/output_...
Now
Cursor 1
0 ps to 3225600 ps
/testbench/output_vector
Type here to search
27°C Rain
ENG
21:45
20-09-2021


ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
Transcript
-- Loading package std_logic_1164
-- Loading package Gates
-- Compiling entity Multiplier_4x1
-- Compiling architecture Struct of Multiplier_4x1
End time: 21:41:01 on Sep 20,2021, Elapsed time: 0:00:00
Errors: 0, Warnings: 0
#
vcom -93 -work work {E:/Multiplier_4x3/Testbench.vhd1}
Model Technology ModelSim - Intel FPGA Edition vcom 2020.1 Compiler 2020.02 Feb 28 2020
Start time: 21:41:01 on Sep 20,2021
vcom -reportprogress 300 -93 -work work E:/Multiplier_4x3/Testbench.vhd1
-- Loading package STANDARD
-- Loading package TEXTIO
-- Loading package std_logic_1164
-- Compiling entity Testbench
-- Compiling architecture Behave of Testbench
End time: 21:41:01 on Sep 20,2021, Elapsed time: 0:00:00
Errors: 0, Warnings: 0
#
vsim -t 1ps -L altera -L lpm -L sgate -L altera_mf -L altera_lnsim -L maxv -L rtl_work -L work -voptargs="+acc" Testbench
vsim -t 1ps -L altera -L lpm -L sgate -L altera_mf -L altera_lnsim -L maxv -L rtl_work -L work -voptargs=""+acc"" Testbench
Start time: 21:41:01 on Sep 20,2021
Loading std.standard
Loading std.textio(body)
Loading ieee.std_logic_1164(body)
Loading work.testbench(behave)
Loading work.dut(dutwrap)
Loading work.gates
Loading work.multiplier_4x1(struct)
Loading work.and_2(equations)
Loading work.half_adder(equations)
Loading work.full_adder(equations)
#
add wave *
view structure
.main_pane.structure.interior.cs.body.struct
view signals
.main_pane.objects.interior.cs.body.tree
run -all
** Note: SUCCESS, all tests passed.
Time: 3072 ns Iteration: 0 Instance: /testbench
VSIM 2>
Transcript
Objects
sim
Type here to search
27°C Rain
ENG
21:45
20-09-2021

Testing on Scanchain:



out.txt file generated after Scanchain testing:-

0000000 0000000 Success

0000001 0000000 Success

0000010 0000000 Success

0000011 0000000 Success

0000100 0000000 Success

0000101 0000000 Success

0000110 0000000 Success

0000111 0000000 Success

0001000 0000000 Success

0001001 0000001 Success

0001010 0000010 Success

0001011 0000011 Success

0001100 0000100 Success

0001101 0000101 Success

0001110 0000110 Success

0001111 0000111 Success

0010000 0000000 Success

0010001 0000010 Success

0010010 0000100 Success

0010011 0000110 Success

0010100 0001000 Success

0010101 0001010 Success

0010110 0001100 Success

0010111 0001110 Success

0011000 0000000 Success

0011001 0000011 Success

0011010 0000110 Success

0011011 0001001 Success

0011100 0001100 Success

0011101 0001111 Success

0011110 0010010 Success

0011111 0010101 Success

0100000 0000000 Success

0100001 0000100 Success

0100010 0001000 Success

0100011 0001100 Success

0100100 0010000 Success

0100101 0010100 Success

0100110 0011000 Success

0100111 0011100 Success

0101000 0000000 Success

0101001 0000101 Success

0101010 0001010 Success

0101011 0001111 Success

0101100 0010100 Success

0101101 0011001 Success

0101110 0011110 Success

0101111 0100011 Success

0110000 0000000 Success

0110001 0000110 Success

0110010 0001100 Success

0110011 0010010 Success

0110100 0011000 Success

0110101 0011110 Success

0110110 0100100 Success

0110111 0101010 Success

0111000 0000000 Success

0111001 0000111 Success

0111010 0001110 Success

0111011 0010101 Success

0111100 0011100 Success

0111101 0100011 Success

0111110 0101010 Success

0111111 0110001 Success

1000000 0000000 Success

1000001 0001000 Success

1000010 0010000 Success

1000011 0011000 Success

1000100 0100000 Success

1000101 0101000 Success

1000110 0110000 Success

1000111 0111000 Success

1001000 0000000 Success

1001001 0001001 Success

1001010 0010010 Success

1001011 0011011 Success

1001100 0100100 Success

1001101 0101101 Success

1001110 0110110 Success

1001111 0111111 Success

1010000 0000000 Success

1010001 0001010 Success

1010010 0010100 Success

1010011 0011110 Success

1010100 0101000 Success

1010101 0110010 Success

1010110 0111100 Success

1010111 1000110 Success

1011000 0000000 Success

1011001 0001011 Success

1011010 0010110 Success

1011011 0100001 Success

1011100 0101100 Success

1011101 0110111 Success

1011110 1000010 Success

1011111 1001101 Success

1100000 0000000 Success

1100001 0001100 Success

1100010 0011000 Success

1100011 0100100 Success

1100100 0110000 Success

1100101 0111100 Success

1100110 1001000 Success

1100111 1010100 Success

1101000 0000000 Success

1101001 0001101 Success

1101010 0011010 Success

1101011 0100111 Success

1101100 0110100 Success

1101101 1000001 Success

1101110 1001110 Success

1101111 1011011 Success

1110000 0000000 Success

1110001 0001110 Success

1110010 0011100 Success

1110011 0101010 Success

1110100 0111000 Success

1110101 1000110 Success

1110110 1010100 Success

1110111 1100010 Success

1111000 0000000 Success

1111001 0001111 Success

1111010 0011110 Success

1111011 0101101 Success

1111100 0111100 Success

1111101 1001011 Success

1111110 1011010 Success

1111111 1101001 Success