22bec121

Experiment – 10 Date:- 4/4/2024

Lab Work

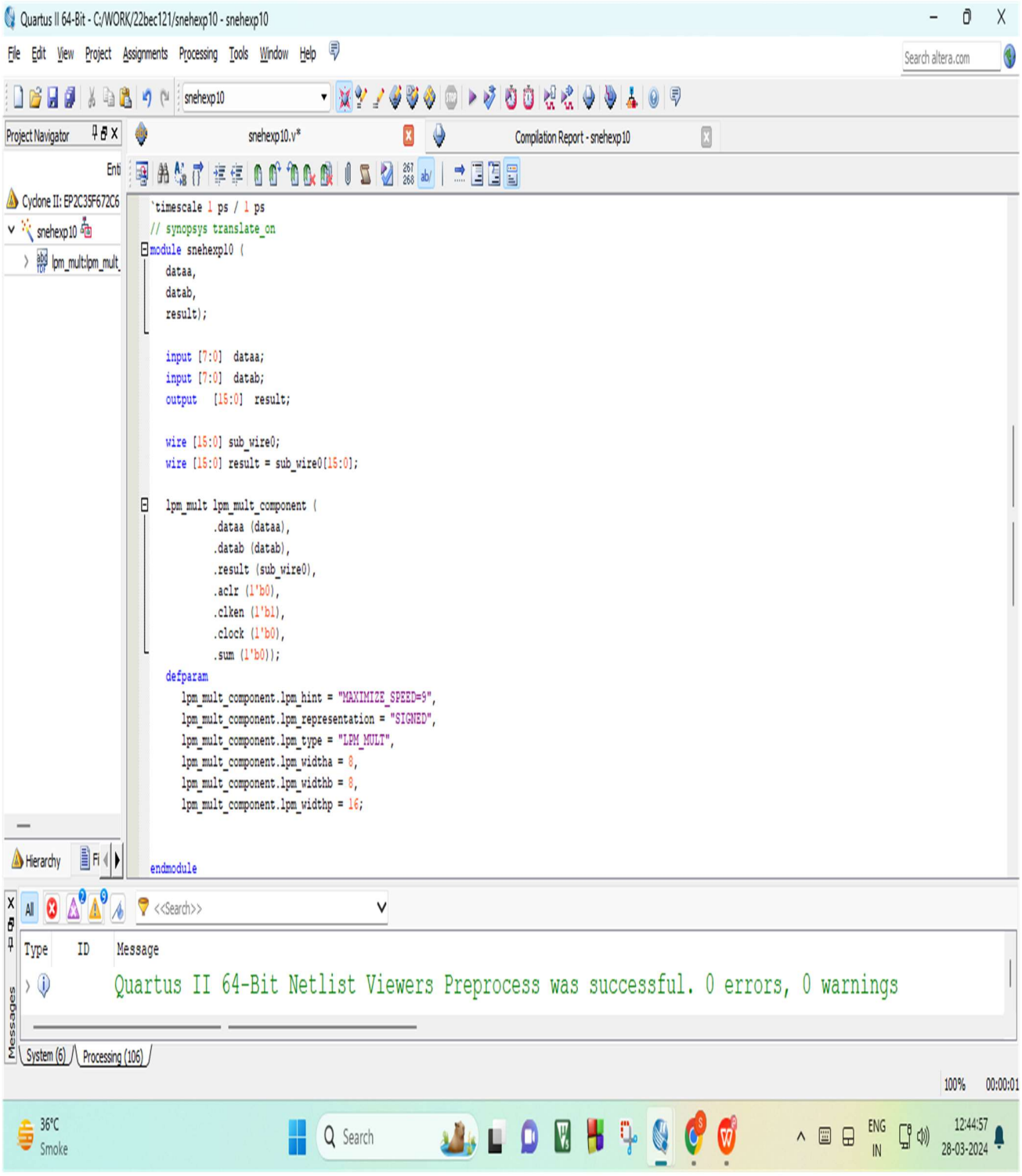
Q1 .Design a 8-Bit Signed Multiplier using IPCores.

Code (IP Core 8-bit Signed Multiplier)

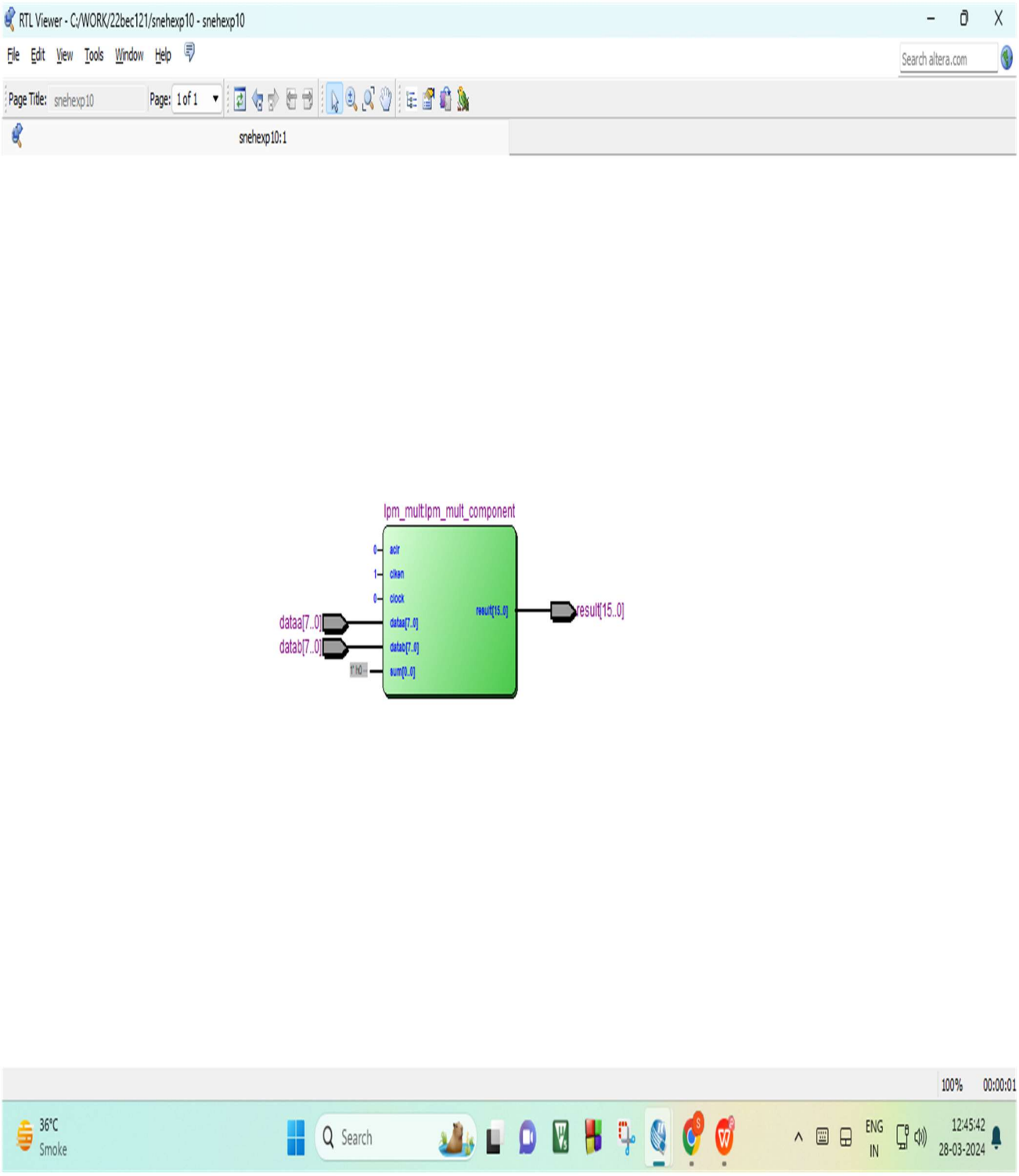
```
//IP Core Code
// megafunction wizard: %LPM_MULT%
// GENERATION: STANDARD
// VERSION: WM1.0
// MODULE: lpm_mult

// ============================================================
// File Name: snehexp10.v
// Megafunction Name(s):
// lpm_mult
//
// Simulation Library Files(s):
// lpm
// ============================================================
// ************************************************************
// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
//
// 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
// ************************************************************


//Copyright (C) 1991-2013 Altera Corporation
//Your use of Altera Corporation's design tools, logic functions
//and other software and tools, and its AMPP partner logic
//functions, and any output files from any of the foregoing
///(including device programming or simulation files), and any
//associated documentation or information are expressly subject
//to the terms and conditions of the Altera Program License
//Subscription Agreement, Altera MegaCore Function License
//Agreement, or other applicable license agreement, including,
//without limitation, that your use is for the sole purpose of
//programming logic devices manufactured by Altera and sold by
//Altera or its authorized distributors.  Please refer to the
```

```
//applicable agreement for further details.


// synopsys translate_off
`timescale 1 ps / 1 ps
// synopsys translate_on
module snehexp10 (
dataa,
datab,
result);

input [7:0]  dataa;
input [7:0]  datab;
output [15:0]  result;

wire [15:0] sub_wire0;
wire [15:0] result = sub_wire0[15:0];

lpm_mult lpm_mult_component (
.dataa (dataa),
.datab (datab),
.result (sub_wire0),
.aclr (1'b0),
.clken (1'b1),
.clock (1'b0),
.sum (1'b0));
defparam
lpm_mult_component.lpm_hint = "MAXIMIZE_SPEED=9",
lpm_mult_component.lpm_representation = "SIGNED",
lpm_mult_component.lpm_type = "LPM_MULT",
lpm_mult_component.lpm_widtha = 8,
lpm_mult_component.lpm_widthb = 8,
lpm_mult_component.lpm_widthp = 16;


endmodule

// ============================================================
// CNX file retrieval info
// ============================================================
// Retrieval info: PRIVATE: AutoSizeResult NUMERIC "1"
// Retrieval info: PRIVATE: B_isConstant NUMERIC "0"
// Retrieval info: PRIVATE: ConstantB NUMERIC "0"
// Retrieval info: PRIVATE: INTENDED_DEVICE_FAMILY STRING "Cyclone II"
// Retrieval info: PRIVATE: LPM_PIPELINE NUMERIC "0"
// Retrieval info: PRIVATE: Latency NUMERIC "0"
// Retrieval info: PRIVATE: SYNTH_WRAPPER_GEN_POSTFIX STRING "1"
```

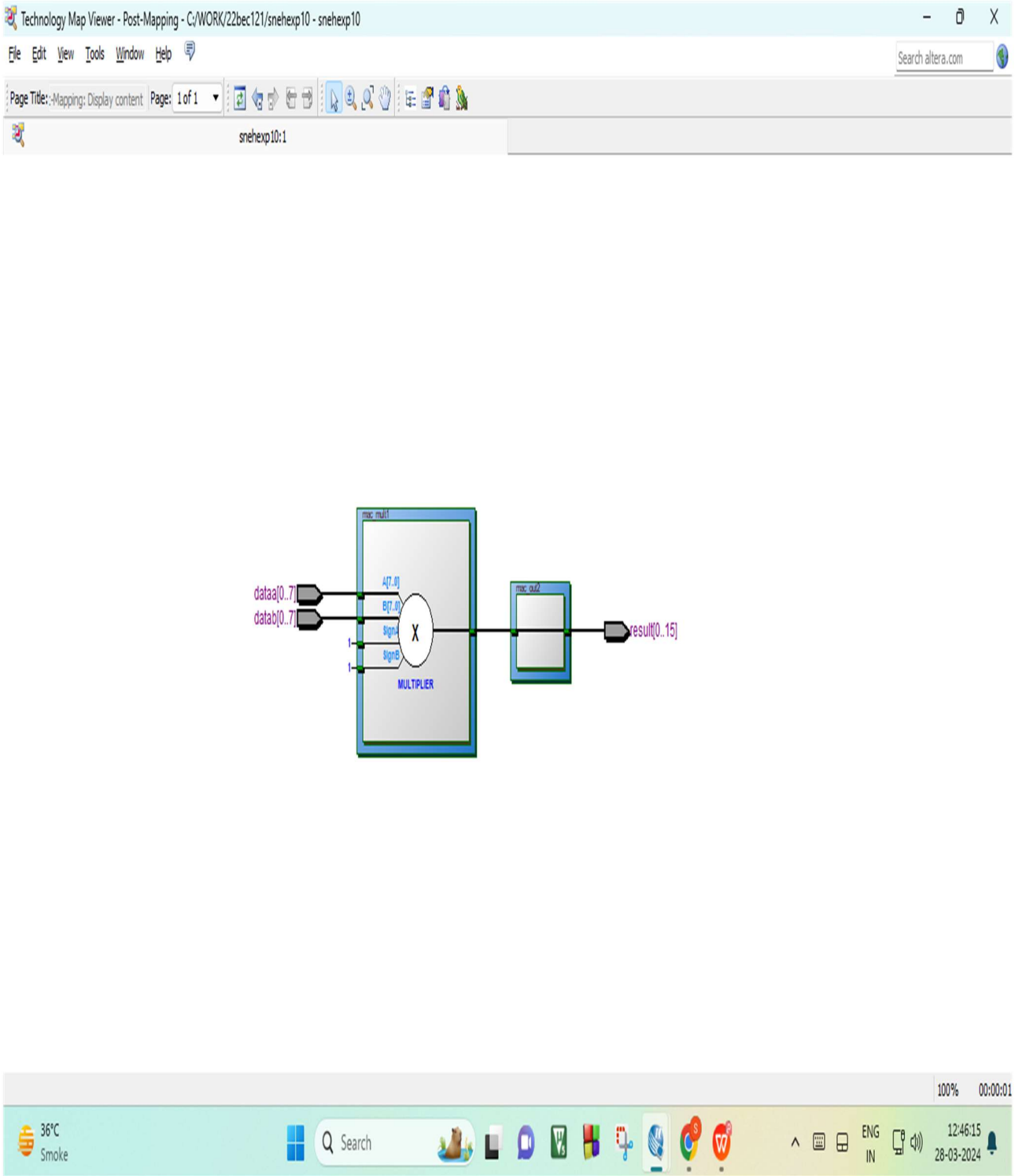
// Retrieval info: PRIVATE: SignedMult NUMERIC "1"
// Retrieval info: PRIVATE: USE_MULT NUMERIC "1"
// Retrieval info: PRIVATE: ValidConstant NUMERIC "0"
// Retrieval info: PRIVATE: WidthA NUMERIC "8"
// Retrieval info: PRIVATE: WidthB NUMERIC "8"
// Retrieval info: PRIVATE: WidthP NUMERIC "16"
// Retrieval info: PRIVATE: aclr NUMERIC "0"
// Retrieval info: PRIVATE: clken NUMERIC "0"
// Retrieval info: PRIVATE: new_diagram STRING "1"
// Retrieval info: PRIVATE: optimize NUMERIC "1"
// Retrieval info: LIBRARY: lpm lpm.lpm_components.all
// Retrieval info: CONSTANT: LPM_HINT STRING "MAXIMIZE_SPEED=9"
// Retrieval info: CONSTANT: LPM_REPRESENTATION STRING "SIGNED"
// Retrieval info: CONSTANT: LPM_TYPE STRING "LPM_MULT"
// Retrieval info: CONSTANT: LPM_WIDTHA NUMERIC "8"
// Retrieval info: CONSTANT: LPM_WIDTHB NUMERIC "8"
// Retrieval info: CONSTANT: LPM_WIDTHP NUMERIC "16"
// Retrieval info: USED_PORT: dataa 0 0 8 0 INPUT NODEFVAL "dataa[7..0]"
// Retrieval info: USED_PORT: datab 0 0 8 0 INPUT NODEFVAL "datab[7..0]"
// Retrieval info: USED_PORT: result 0 0 16 0 OUTPUT NODEFVAL "result[15..0]"
// Retrieval info: CONNECT: @dataa 0 0 8 0 dataa 0 0 8 0
// Retrieval info: CONNECT: @datab 0 0 8 0 datab 0 0 8 0
// Retrieval info: CONNECT: result 0 0 16 0 @result 0 0 16 0
// Retrieval info: GEN_FILE: TYPE_NORMAL snehexp10.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL snehexp10.inc TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL snehexp10.cmp TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL snehexp10.bsf TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL snehexp10_inst.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL snehexp10_bb.v TRUE
// Retrieval info: GEN_FILE: TYPE_NORMAL snehexp10_syn.v TRUE
// Retrieval info: LIB_FILE: lpm


Quartus II 64-Bit - C:/WORK/22bec121/snehexp10 - snehexp10
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
snehexp10
Project Navigator
snehexp10.v*
Compilation Report - snehexp10
Cyclone II: EP2C35F672C6
snehexp10
lpm_mult:lpm_mult_
`timescale 1 ps / 1 ps
// synopsys translate_on
module snehexp10 (
dataa,
datab,
result);
input [7:0] dataa;
input [7:0] datab;
output [15:0] result;
wire [15:0] sub_wire0;
wire [15:0] result = sub_wire0[15:0];
lpm_mult lpm_mult_component (
.dataa (dataa),
.datab (datab),
.result (sub_wire0),
.aclr (1'b0),
.clken (1'b1),
.clock (1'b0),
.sum (1'b0));
defparam
lpm_mult_component.lpm_hint = "MAXIMIZE_SPEED=9",
lpm_mult_component.lpm_representation = "SIGNED",
lpm_mult_component.lpm_type = "LPM_MULT",
lpm_mult_component.lpm_widtha = 8,
lpm_mult_component.lpm_widthb = 8,
lpm_mult_component.lpm_widthp = 16;
endmodule
Hierarchy
<<Search>>
Type ID Message
Quartus II 64-Bit Netlist Viewers Preprocess was successful. 0 errors, 0 warnings
Messages
System (6)
Processing (106)
100% 00:00:01
36°C
Smoke
Search
ENG
IN
12:44:57
28-03-2024

Output 1)Cyclone II (RTL View)



2)Cyclone II (TTL View)



Conclusion :- In this experiment we learnt to use IP Cores made by other users and implement it on the FPGA Kit.

We also learnt how RTL and TTL forms for a 8bit Signed Multiplier . We simulated the IPCore on ModelSim software to view its functioning . We also implemented 8-Bit Signed Multiplier on the FPGA Kit .