RESULTS

The chapter explains the results relating to the design and simulation in Xilinx software environment and functional verification on Modelsim 10.0 software. The results are presented for 2D NoC router; 3D NoC router, 2D and 3D mesh NoC, 2D and 3D Torus NoC, and 2D and 3D ring NoC.

6.1 RTL and Waveform Simulation for 2D NoC Router

The RTL shows the all inputs and outputs of the router. It is the description of all the input and output logic used to design the chip. The RTL view of the developed 2D router chip is shown in fig. 6.1 and its internal schematic is depicted in fig.6.2. Internal schematic presents the internal structure having different logic gates and internal hardware required in the design. The functional verification is carried for 256 bit data transfer and simulation waveforms are shown in fig. 6.3 and fig. 6.4 Table 6.1 is used to discuss the use, size and detail of the pins used.

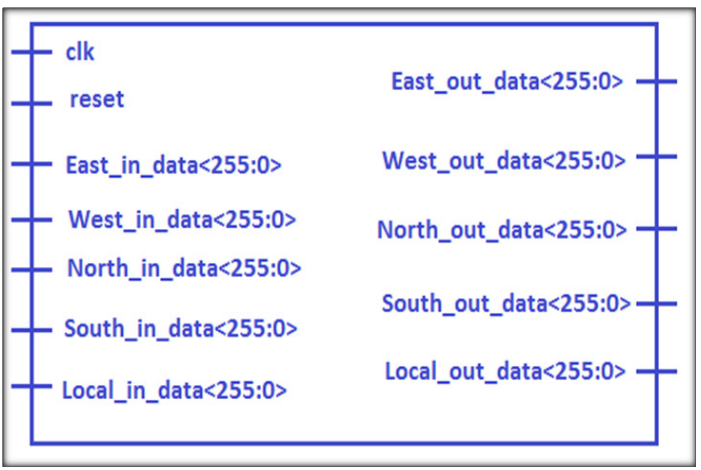


Figure 6.1 RTL of Router in 2D

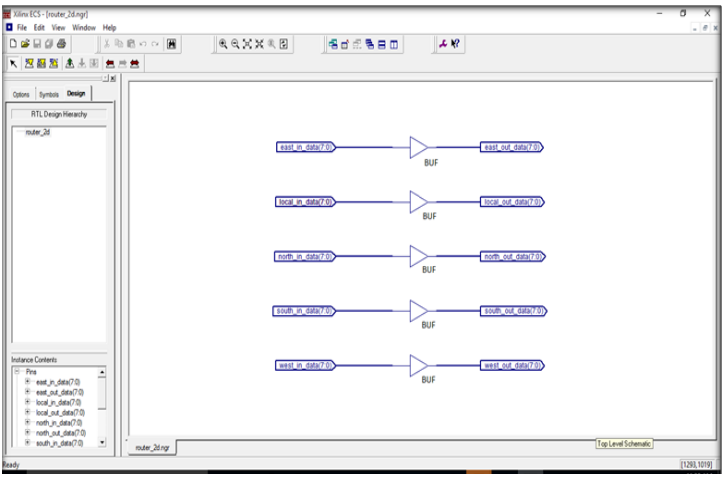


Figure 6.2 Internal schematic of the 2D router

Table 6.1 Pin explanation for the 2D router chip

Pin	**Size**	**Functional Description**
Reset	1 bit	It is the input pin of std_logic, used to reset all the data and contents of all the ports and registers in the design.

Clk	1 bit	It is 1 bit input and default associated with design used to give the rising edge (+ve) edge in the simulation results and works on 50 % duty cycle.
East_in_data<255:0>	256 bit	It is the input data packet coming in the router from east input port (256 bit) and controlled through the control unit associated with network.
West_in_data<255:0>	256 bit	It is the input data packet coming in the router from west input port (256 bit) and controlled through the control unit associated with network.

North_in_data<255:0>	256 bit	It is the input data packet coming in the router from north input port (256 bit) and controlled through the control unit associated with network.

South_in_data<255:0>	256 bit	It is the input data packet coming in the router from south input port (256 bit) and controlled through the control unit associated with network.
Local_in_data<255:0>	256 bit	It is the input data packet coming in the router from local input port (256 bit) and controlled through the control unit associated with network.
East_out_data<255:0>	256 bit	It is the output port having 256 bit data on east out data and stored in the corresponding register.

West_out_data<255:0>	256 bit	It is the output port having 256 bit data on west out data and stored in the corresponding register.
North_out_data<255:0>	256 bit	It is the output port having 256 bit data on north out data and stored in the corresponding register.
South_out_data<255:0>	256 bit	It is the output port having 256 bit data on south out data and stored in the corresponding register.
Local_out_data<255:0>	256 bit	It is the output port having 256 bit data on local out data and stored in the corresponding register.

The simulation is verified in Modelsim 10.0 software. Fig. 6.3 and Fig. 6.4 present Modelsim waveform result for 2D router for 256 bit data transfer in hexadecimal and ASCII. The data is coming from the port east_in_data<255:0> and local_in_ data<255:0> and output is obtained at east_out_data<255:0> and local_out_data<255:0>.

Test-1: Reset = '1' and run the simulation environment. All the ports will get zero value. Reset = '0' and give the rising edge of clk, and East_in_data<255:0> ,Local_in_data<255:0> = "54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72" (heaxdecimal) or "TMU@ TMU@ComputerTMU@TMU@Computer" in ASCII. The

same data is obtained at East_out_data and local_out_data. The data is given as "0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010"

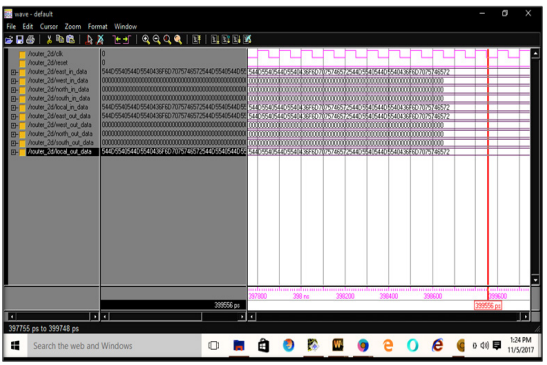


Figure 6.3 Modelsim waveform result for 2D router data transfer, 256 bit (hexadecimal)

6.2 is used to discuss the use, size and

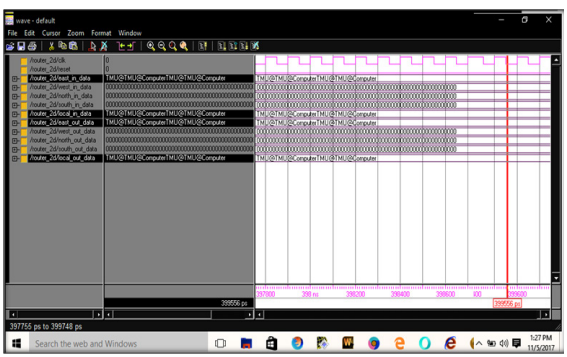


Figure 6.4 Modelsim waveform result for 2D router data transfer, 256 bit (ASCII)

6.2 RTL and Waveform Simulation for 3D NoC Router

The RTL view of the developed 3D router chip is shown in fig. 6.5 and its internal schematic is depicted in fig.6.6. Internal schematic presents the internal structure having different logic gates and internal hardware required in the design. Table detail of the pins used. The functional verification is carried for 256 bit data transfer and simulation waveforms are shown in fig. 6.7 and fig. 6.8. Modelsim waveform in Fig. 6.7 and Fig. 6.8 presents the result for 3D router for 256 bit data transfer in hexadecimal and ASCII. The data is coming from the port local_in_data<255:0>, Up_in_data<255:0> and down_in_ data<255:0> and output is obtained at local_out_data<255:0>, Up_out_data<255:0> and down_out_data<255:0>. The flow chart of the 2D and 3D router is shown in fig. 6.9.

Test-1: Reset = '1' and run the simulation environment. All the ports will get zero value. Reset = '0' and, given+ the rising edge of clk, and East_in_data<255:0>, Local_in_data<255:0> = "54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72" (heaxdecimal) or "TMU@ TMU@ComputerTMU@TMU@Computer" in ASCII. The same data is obtained at East_out_data and local_out_data. The data is given as "0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010"

Table 6.2 Pin explanation for the 3D router chip

Pin	Size	Functional Description
Reset	1 bit	It is the input pin of std_logic, used to reset all the data and contents of all the ports and registers in the design.

Clk	1 bit	It is 1 bit input and default associated with design used to give the rinsing edge (+ve) edge in the simulation results and works on 50 % duty cycle.
East_in_data<255:0>	256 bit	It is the input data packet coming in the router from east input port (256 bit) and controlled through the control unit associated with network.
West_in_data<255:0>	256 bit	It is the input data packet coming in the router from west input port (256 bit) and controlled through the control unit associated with network.
North_in_data<255:0>	256 bit	It is the input data packet coming in the router from north input port (256 bit) and controlled through the control unit associated with network.
South_in_data<255:0>	256 bit	It is the input data packet coming in the router from south input port (256 bit) and controlled through the control unit associated with network.
Local_in_data<255:0>	256 bit	It is the input data packet coming in the router from local input port (256 bit) and controlled through the control unit associated with network.

Up_in_data<255:0>	256 bit	It is the input data packet coming in the router from upward input port (256 bit) and controlled through the control unit associated with network.
Down_in_data<255:0>	256 bit	It is the input data packet coming in the router from downward input port (256 bit) and controlled through the control unit associated with network.
East_out_data<255:0>	256 bit	It is the output port having 256 bit data on east out data and stored in the corresponding register.
West_out_data<255:0>	256 bit	It is the output port having 256 bit data on west out data and stored in the corresponding register.
North_out_data<255:0>	256 bit	It is the output port having 256 bit data on north out data and stored in the corresponding register.
South_out_data<255:0>	256 bit	It is the output port having 256 bit data on south out data and stored in the corresponding register.
Local_out_data<255:0>	256 bit	It is the output port having 256 bit data on local out data and stored in the corresponding register.

Up_out_data<255:0>	256 bit	It is the output port having 256 bit data on upward out data and stored in the corresponding register.
Down_out_data<255:0>	256 bit	It is the output port having 256 bit data on downward out data and stored in the corresponding register.

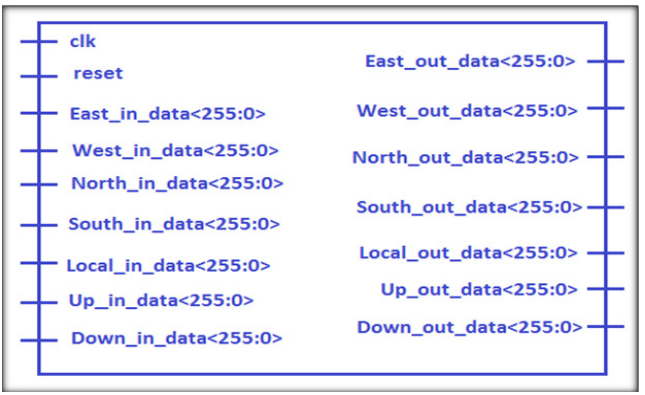


Figure 6.5 RTL of Router in 3D

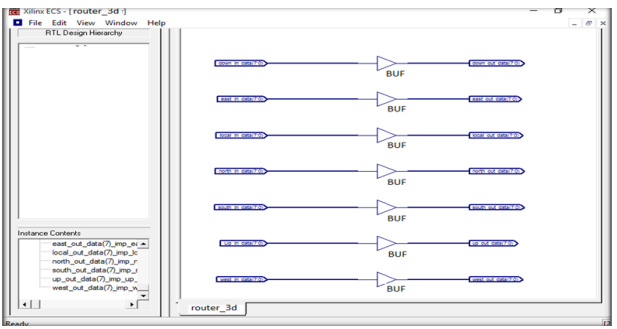


Figure 6.6 Internal schematic of the 3D router

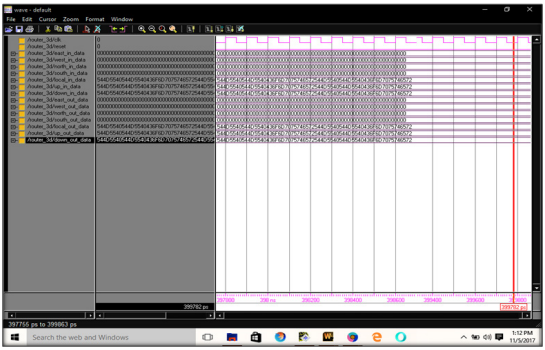


Figure 6.7 Modelsim waveform result for 3D router data transfer, 256 bit (Hexadecimal)

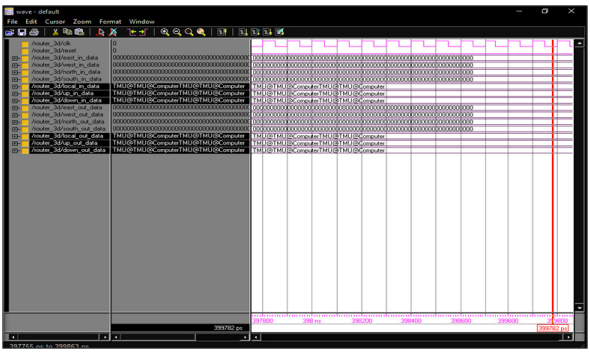


Figure 6.8 Modelsim waveform result for 3D router data transfer, 256 bit (ASCII)

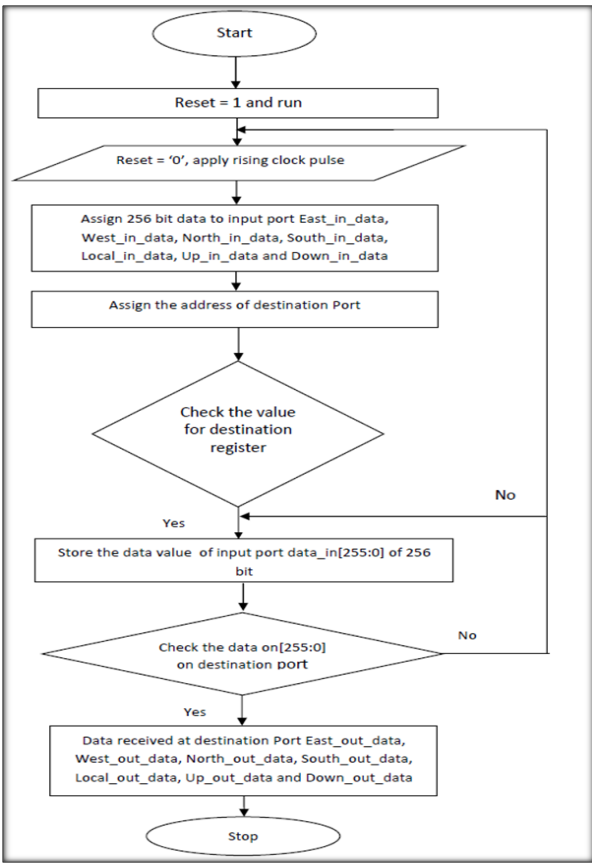


Figure 6.9 Flow Chart for 2D and 3D NoC Router

6.3 RTL and Wave form Simulation for 2D Mesh, Torus, Ring and Fat tree Topology

The RTL view of the developed 2D mesh, torus, ring and fat tree topology is shown in fig. 6.10 and its internal schematic is depicted in fig.6.11. Internal schematic presents the internal structure having different logic gates and internal hardware required in the design. Table 6.3 is used topology. The functional verification is carried for 256 bit data transfer and

simulation waveforms. Fig. 6.12 (a) shows the waveform and simulation off mesh NoC (4 x 4) for 8 bit data communication, Fig. 6.12 (b) 16 bit data communication, Fig. 6.12 (c) 32 bit data communication, Fig. 6.12 (d) 64 bit for 8 bit data communication, Fig. 6.12 (e) 128 bit data communication binary Fig. 6.12 (f) 128 bit data communication in hexadecimal. Fig. 6.13 and Fig. 6.14 present the Modelsim result simulation of 256 bit data in hexadecimal and ASCII for mesh topology. Fig. 6.15 and Fig. 6.16 present the Modelsim result simulation of 256 bit data in hexadecimal and ASCII for torus topology. Fig. 6.17 and Fig. 6.18 present the Modelsim result simulation of 256 bit data in hexadecimal and ASCII for ring topology. Fig. 6.19 and Fig. 6.20 present the Modelsim result simulation of 256 bit data in hexadecimal and ASCII for fat tree topology. The flow chart for the functional behavior is shown in fig. 6.21.to discuss the use, size and detail of the pins used for 2D mesh, torus, ring and fat tree

Table 6.3 Pin explanation for the 2D topology (Mesh, Torus, Ring and Fat tree)

Pin	**Size**	**Functional Description**
Reset	1 bit	It is the input pin of std_logic, used to reset all the data and contents of all the ports and registers in the design.

Clk	1 bit	It is 1 bit input and default associated with design used to give the rinsing edge (+ve) edge in the simulation results and works on 50 % duty cycle.
Data_in<255:0>	256 bit	It is the input data packet (256 bit) by the source router and controlled through the control unit associated with network.
Data_out<255:0>	256 bit	It is the output data packet (256 bit) by the source router and controlled through the control unit associated with network.
In_node_address<3:0>	4 bit	It is the input of source node used to define the input address of the source node need to communicate
Out_node_address<3:0>	4 bit	It is the input for destination node used to define the output address of the target node need to communicate

Row_adddres	2 bit	It is the address of the source and target nodes corresponding to the row for mesh and torus topology

Column_address	2 bit	It is the address of the source and target nodes corresponding to the column for mesh and torus topology
Write_enable	1 bit	It is the input control signal used to write the contents from source node to destination node. If the write_enable = '1' and read_enable = '0', the contents are written from the control unit to register
Read_enable	1 bit	It is the input control signal used to read the contents from source node to destination node. If the read_enable = '1' and write_enable = '0', the contents are written from the control unit to register
M0<255:0>	256 bit	It is the input/output node carrying 256 bit data
M1<255:0>	256 bit	It is the input/output node carrying 256 bit data
M2<255:0>	256 bit	It is the input/output node carrying 256 bit data

M3<255:0>	256 bit	It is the input/output node carrying 256 bit data
M4<255:0>	256 bit	It is the input/output node carrying 256 bit data
M5<255:0>	256 bit	It is the input/output node carrying 256 bit data
M6<255:0>	256 bit	It is the input/output node carrying 256 bit data
M7<255:0>	256 bit	It is the input/output node carrying 256 bit data
M9<255:0>	256 bit	It is the input/output node carrying 256 bit data
M10<255:0>	256 bit	It is the input/output node carrying 256 bit data
M11<255:0>	256 bit	It is the input/output node carrying 256 bit data
M12<255:0>	256 bit	It is the input/output node carrying 256 bit data
M13<255:0>	256 bit	It is the input/output node carrying 256 bit data
M14<255:0>	256 bit	It is the input/output node carrying 256 bit data
M15<255:0>	256 bit	It is the input/output node carrying 256 bit data

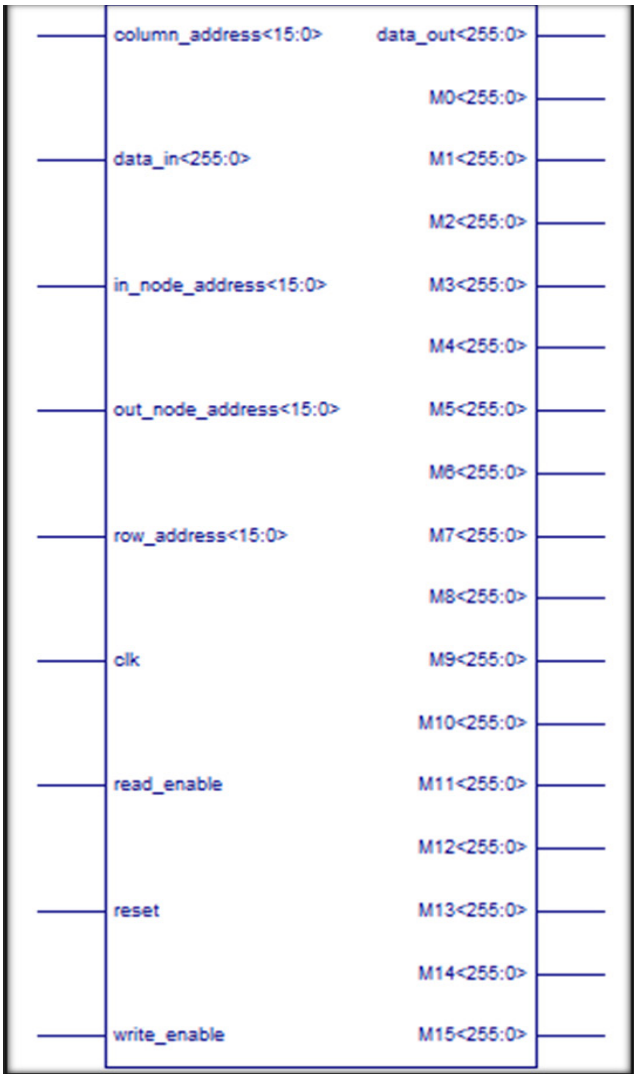


Fig. 6.10 RTL of 2D Mesh, Torus, Ring and Fat tree Topology

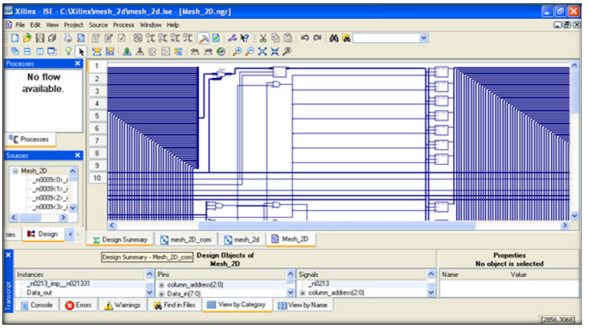
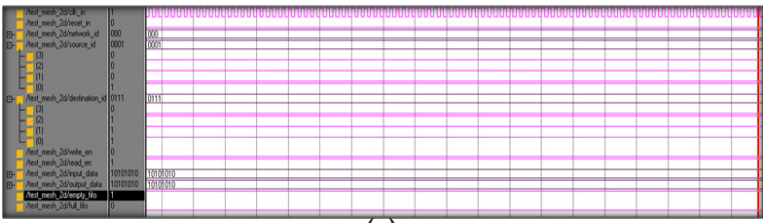
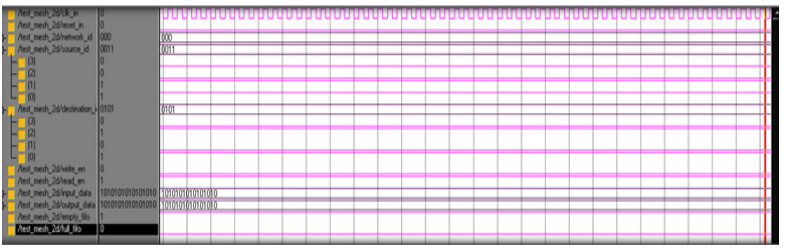


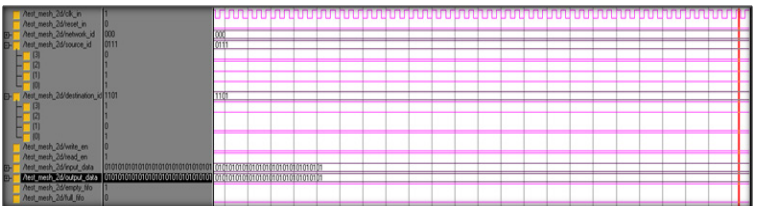
Figure 6.11 RTL of 2D Mesh, Torus, Ring and Fat tree Topology



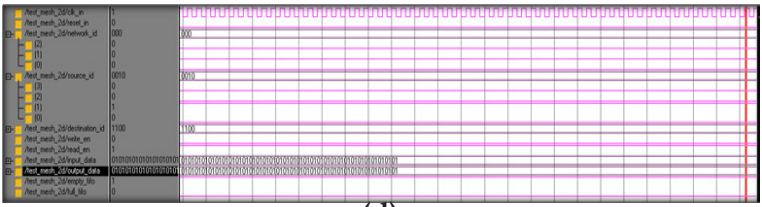
(a)



(b)

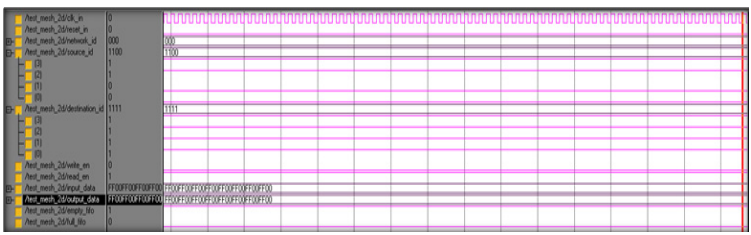


(c)



(d)

(e)



(f)

Figure 6.12 (a) Waveform and simulation of mesh NoC (4 x 4) for 8 bit data communication (b) 16 bit data communication c) 32 bit data communication (d) 64 bit data communication (e) 128 bit data communication (f) 128 bit data communication in hexadecimal.

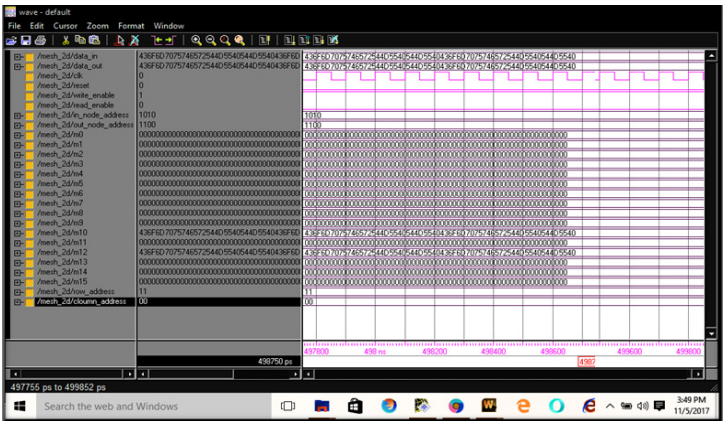


Figure6.13 Modelsim result simulation of 256 bit data in hexadecimal for 2D mesh topology

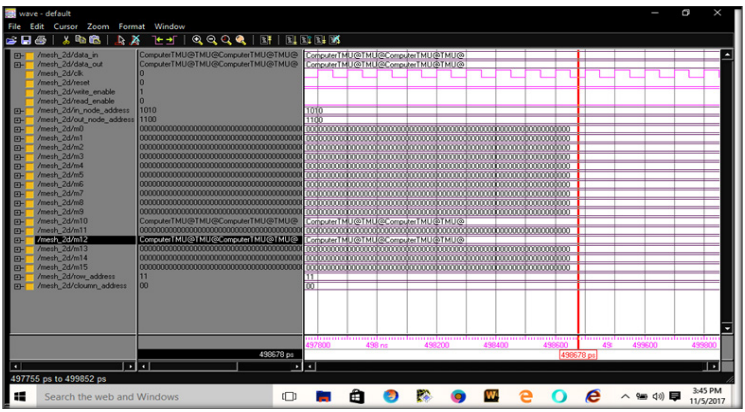


Figure 6.14 Modelsim result simulation of 256 bit data in ASCII for 2D mesh topology

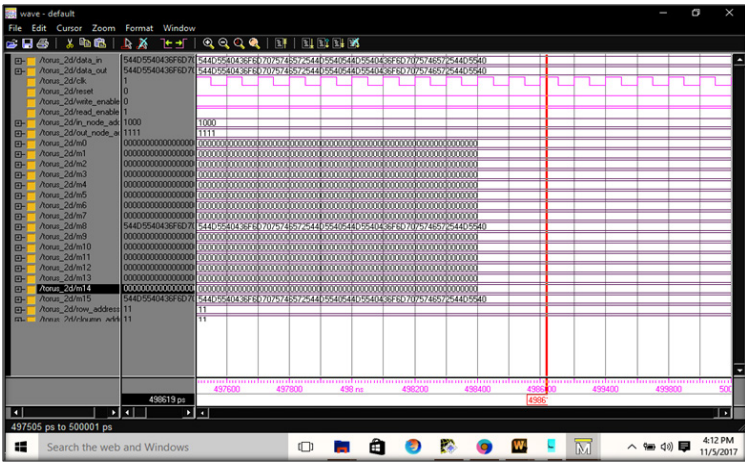


Figure 6.15 Modelsim result simulation of 256 bit data in hexadecimal for 2D torus topology

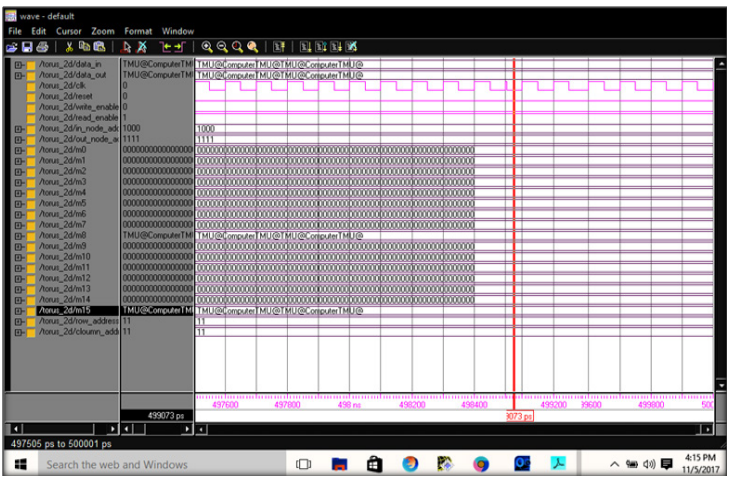


Figure 6.16 Modelsim result simulation of 256 bit data in ASCII for 2D torus topology

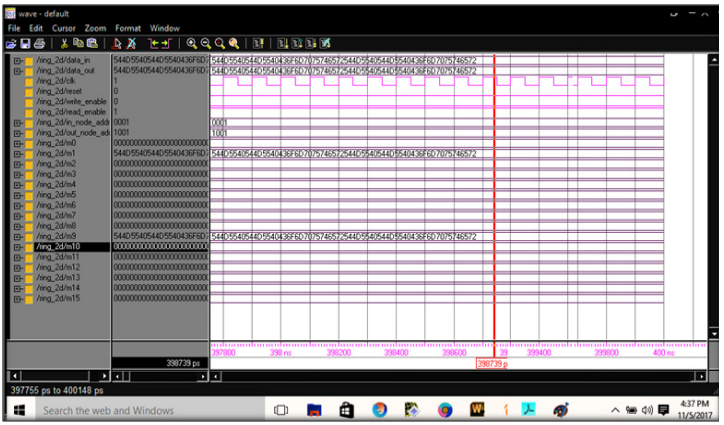


Figure 6.17 Modelsim result simulation of 256 bit data in hexadecimal for 2D ring topology

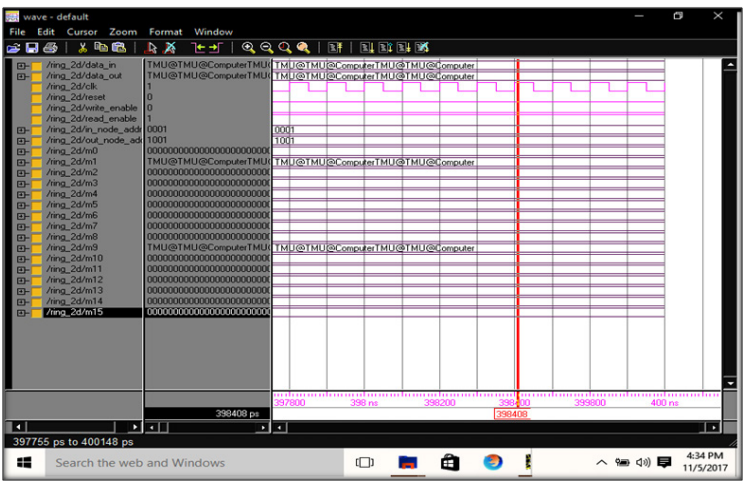


Figure 6.18 Modelsim result simulation of 256 bit data in ASCII for 2D ring topology

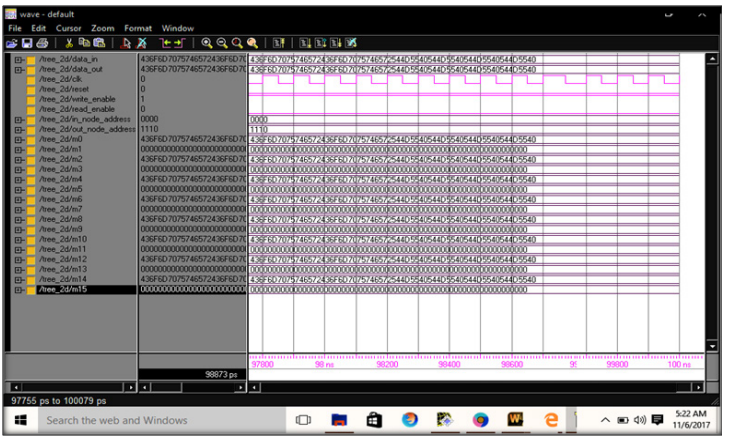


Figure 6.19 Modelsim result simulation of 256 bit data in hexadecimal for 2D fat tree topology

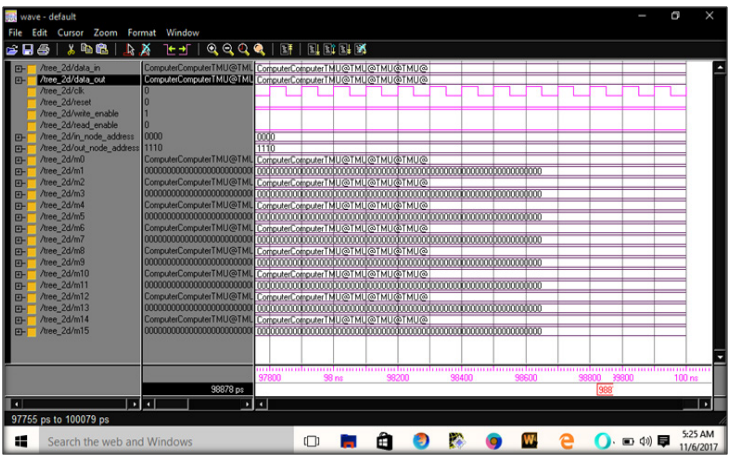


Figure 6.20 Modelsim result simulation of 256 bit data in ASCII for 2D fat free topology

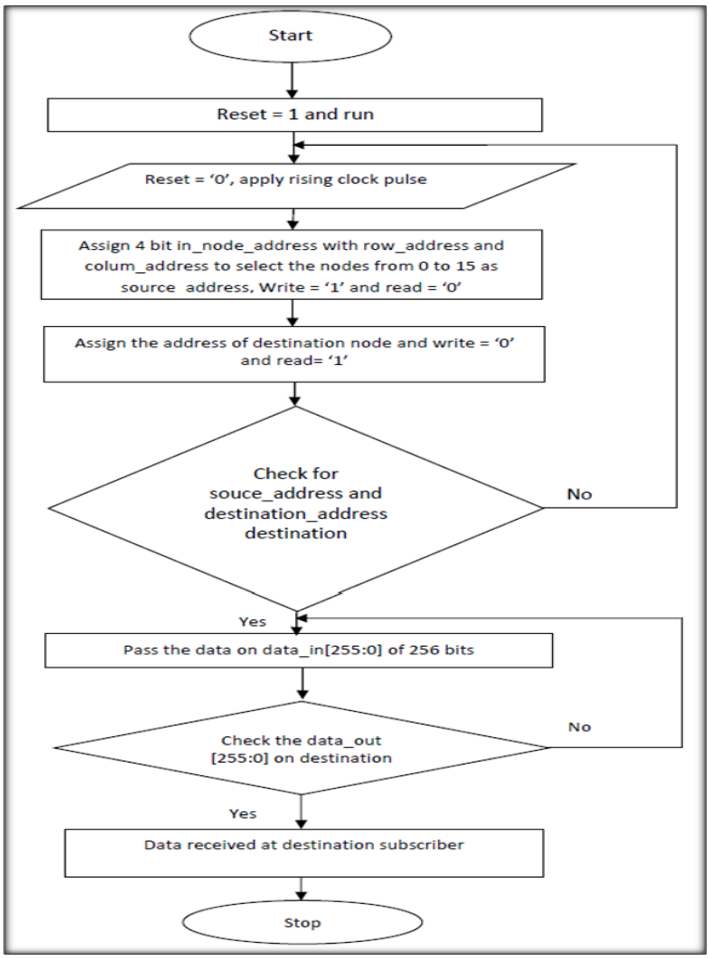


Figure 6.21 Flow chart for 2D Topology (Mesh, Torus, Ring and Fat Tree)

The Simulation of the 2D NoC is completed for the following test cases:

Test -1 (Mesh): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', Source_id = "0001", Destination_id = "0111", Read_in = '0',

input_data = "10101010", in hexadecimal (AA). The data is written in destination_id. Now, Write_en = '0', Read_in = '1', data is transferred to destination node.

Test -2 (Mesh): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', in_ node_address = "1010" out_node_address = "1100", row_ address = "11" and column_addess = "00" based on output node, data_in = "43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40 54 4D 55 40" in hexadecimal or **ComputerTMU@TMU@ComputerTMU@ TMU@ in ASCII.** The same data is from source node M10 <255:0>. When Write_en = '0', Read_in = '1', the destination node M10 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

Test -3 (Torus): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', in_ node_address = "1000" out_node_address = "1111", row_ address = "11" and column_addess = "11" based on output node, data_in = "54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40" in hexadecimal or **TMU@ComputerTMU@TMUComputerTMU@ in ASCII.** The same data is from source node M8 <255:0>. When Write_ en = '0', Read_in = '1', the destination node M15 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

Test -4 (Ring): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', in_ node_address = "0001" out_node_address = "1001", based on output node, data_in = "54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72" in hexadecimal or **TMU@TMU@ComputerTMU@TMU@ Computer in ASCII.** The same data is from source node M1 <255:0>. When Write_en = '0', Read_in = '1', the destination node M9 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010" in binary.

Test -5 (Fat Tree): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', in_ node_address = "0000" out_node_address = "0000", "0010", "0100", "0110", "1000", "1010", "1100" and "1110", based on output node, data_in = "43 6F 6D 70 75 74 65 72 43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 54 4D 55 40 54 4D 55 40" in hexadecimal or **ComputerComputer TMU@TMU@TMU@ TMU@ in ASCII.** The same data is from source node M0 <255:0>. When Write_en = '0', Read_in = '1', the destination node M0 <255:0>, M2 <255:0>, M4 <255:0> , M6 <255:0> , M8 <255:0> , M10 <255:0>, M12 <255:0> and M14 <255:0> . Data_ out< 255:0> are getting the same data. Data_out = "0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

6.4 RTL and Wave form Simulation for 3D Mesh, Torus,

Ring and Fat tree Topology

The RTL view of the developed 3D mesh, torus, ring and fat tree topology is shown in fig. 6.22 and its internal schematic is depicted in fig.6.23. Internal schematic presents the internal structure having different logic gates and internal hardware required in the design. Table 6.4 is used to discuss the use, size and detail of the pins used for 3D mesh, torus, ring and fat tree topology. The functional verification is carried for 256 bit data transfer and simulation waveforms

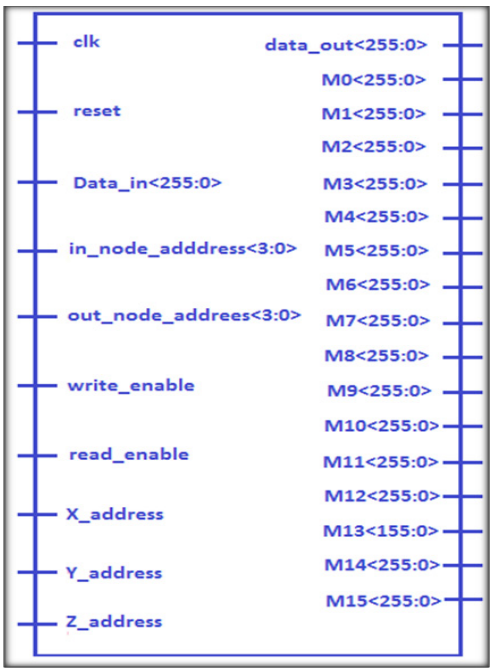


Figure 6.22 RTL view of 3D Topology

Fig. 6.24 and Fig. 6.25 present the Modelsim result simulation of 256 bit data in hexadecimal and ASCII for 3D mesh topology. Fig. 6.26 and Fig. 6.27 and present the Modelsim result simulation of 256 bit data in hexadecimal and ASCII for 3D torus topology. Fig. 6.28 and Fig. 6.29 present the Modelsim result simulation of 256 bit data in hexadecimal

and ASCII for 3D ring topology. Fig. 6.30 and Fig. 6.31 present the Modelsim result simulation of 256 bit data in hexadecimal and ASCII for 3D fat tree topology. The flow chart for the functional behavior is shown in fig. 6.32.

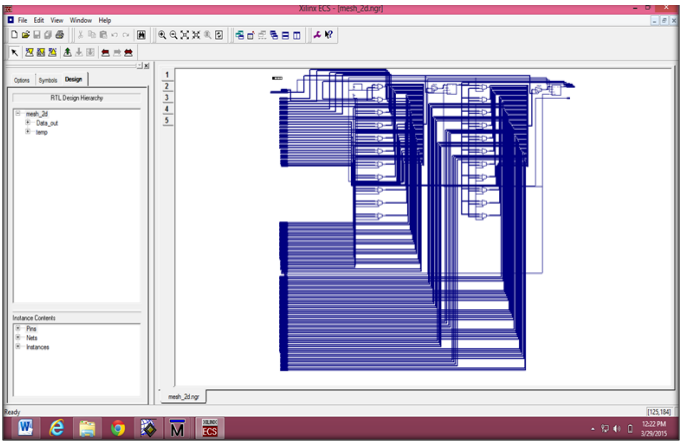


Figure 6.23 Internal schematic of 3D Topology

Table 6.4 Pin explanation for the 3D topology (Mesh, Torus, Ring and Fat tree)

Pin	Size	Functional Description
Reset	1 bit	It is the input pin of std_logic, used to reset all the data and contents of all the ports and registers in the design.
Clk	1 bit	It is 1 bit input and default associated with design used to give the rinsing edge (+ve) edge in the simulation results and works on 50 % duty cycle.

Data_in<255:0>	256 bit	It is the input data packet (256 bit) by the source router and controlled by the control unit of the network.
Data_out<255:0>	256 bit	It is the output data packet (256 bit) by the source router and controlled by the control unit of the network.
In_node_address<8:0>	9 bit	It is the input of source node used to define the input address of the source node need to communicate
Out_node_address<8:0>	9 bit	It is the input for destination node used to define the output address of the target node need to communicate
X_adddres	3 bit	It is the address of the source and target nodes corresponding to X axis
Y_address	3 bit	It is the address of the source and target nodes corresponding to Y axis
Z_address	3 bit	It is the address of the source and target nodes corresponding to Z axis

Write_enable	1 bit	It is the input control signal used to write the contents from source node to destination node. If the write_enable = '1' and read_enable = '0', the contents are written from the control unit to register
Read_enable	1 bit	It is the input control signal used to read the contents from source node to destination node. If the read_enable = '1' and write_enable = '0', the contents are written from the control unit to register
R0<255:0>	256 bit	It is the input/output node carrying 256 bit data
R1<255:0>	256 bit	It is the input/output node carrying 256 bit data
R2<255:0>	256 bit	It is the input/output node carrying 256 bit data
R3<255:0>	256 bit	It is the input/output node carrying 256 bit data
R4<255:0>	256 bit	It is the input/output node carrying 256 bit data

R5<255:0>	256 bit	It is the input/output node carrying 256 bit data
R6<255:0>	256 bit	It is the input/output node carrying 256 bit data
R7<255:0>	256 bit	It is the input/output node carrying 256 bit data
R9<255:0>	256 bit	It is the input/output node carrying 256 bit data
R10<255:0>	256 bit	It is the input/output node carrying 256 bit data
R11<255:0>	256 bit	It is the input/output node carrying 256 bit data
R12<255:0>	256 bit	It is the input/output node carrying 256 bit data
M13<255:0>	256 bit	It is the input/output node carrying 256 bit data
R14<255:0>	256 bit	It is the input/output node carrying 256 bit data
R15<255:0>	256 bit	It is the input/output node carrying 256 bit data
⋮	⋮	⋮
R26<255:0>	256 bit	It is the input/output node carrying 256 bit data

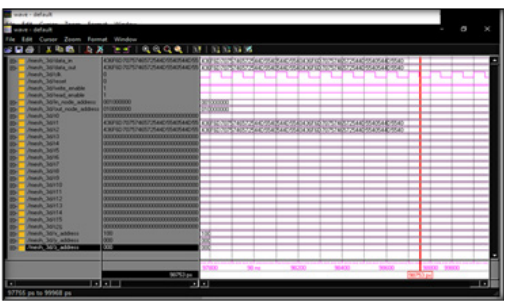


Figure 6.24 Modelsim result simulation of 256 bit data in hexadecimal for 3D mesh topology

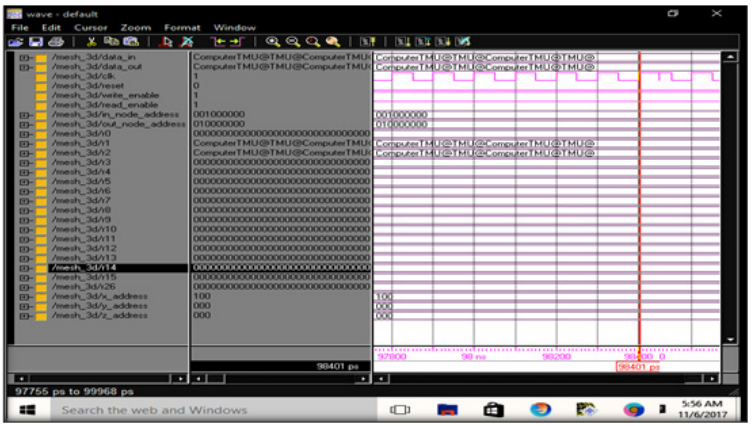


Figure 6.25 Modelsim result simulation of 256 bit data in ASCII for 3D mesh topology

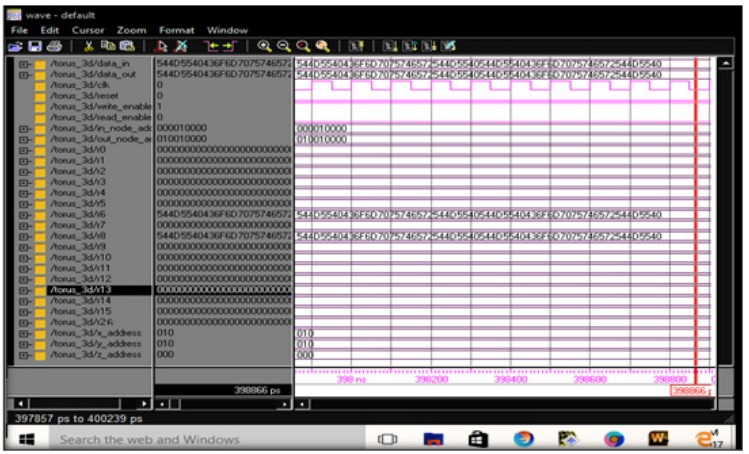


Figure 6.26 Modelsim result simulation of 256 bit data in hexadecimal for 3D Torus topology

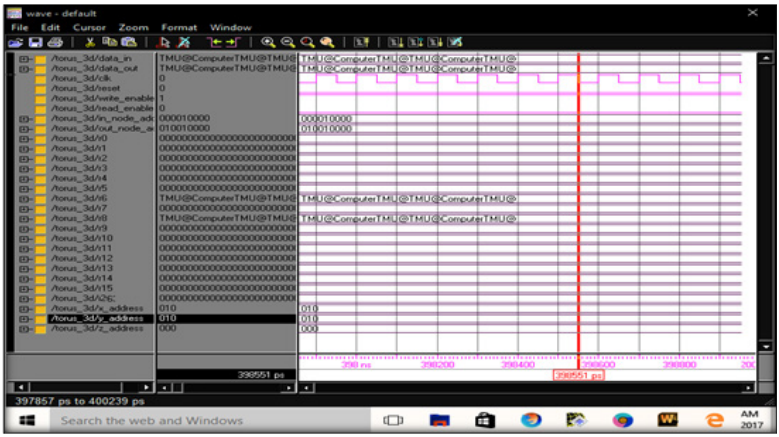


Figure 6.27 Modelsim result simulation of 256 bit data in ASCII for 3D torus topology

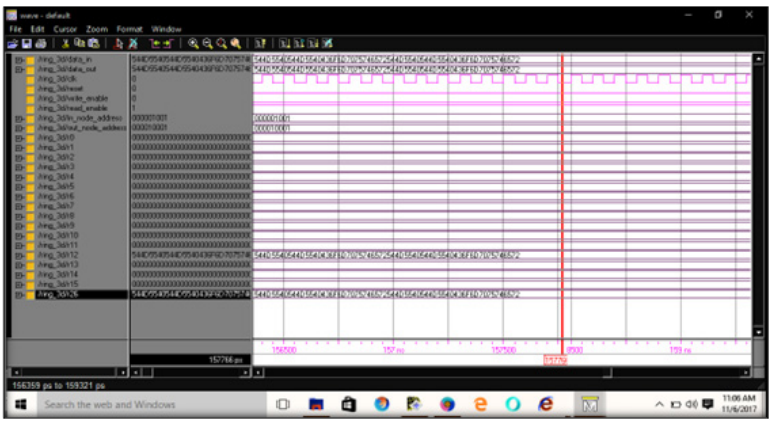


Figure 6.28 Modelsim result simulation of 256 bit data in hexadecimal for 3D ring topology

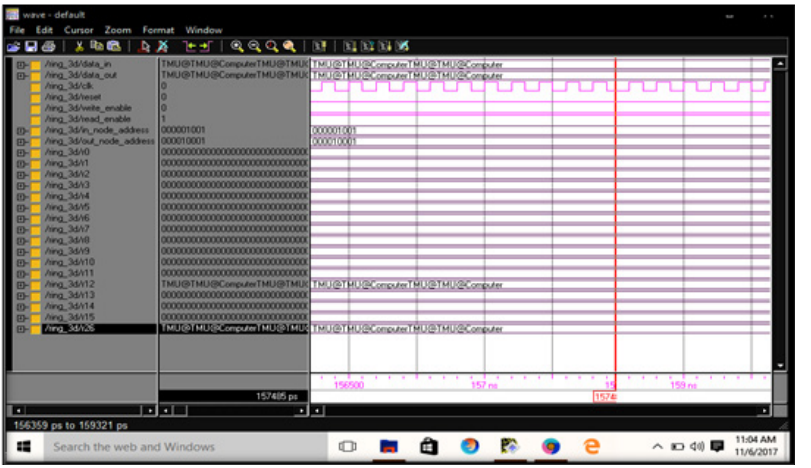


Figure 6.29 Modelsim result simulation of 256 bit data in ASCII for 3D ring topology

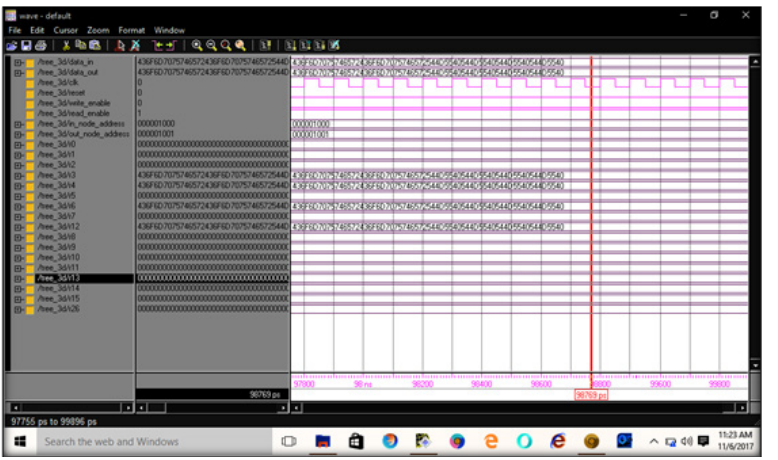


Figure 6.30 Modelsim result simulation of 256 bit data in hexadecimal for 3D tree topology

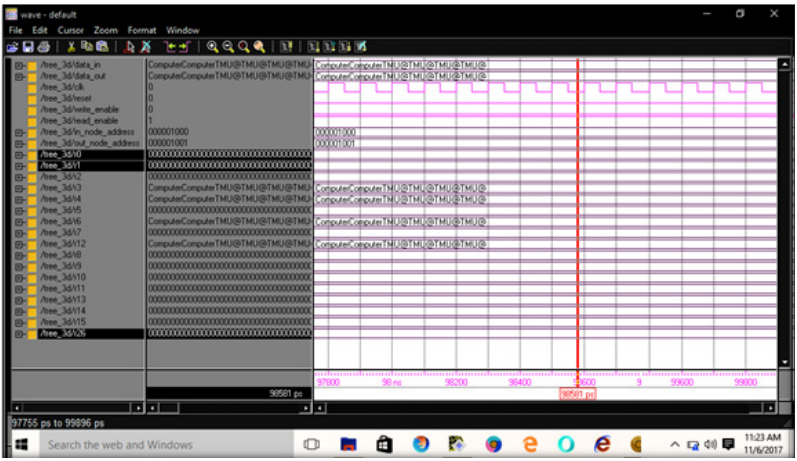


Figure 6.31 Modelsim result simulation of 256 bit data in ASCII for 3D tree topology

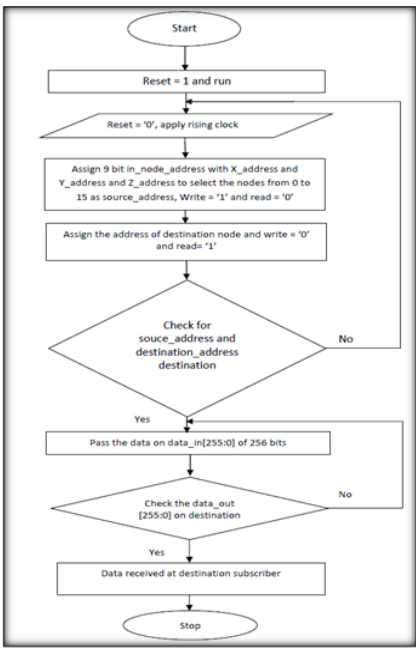


Figure 6.32 Flowchart of 3D NoC topology function

Test -1 (Mesh): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', in_node_address = "001000000" out_node_address = "010000000", X_address = "010" Y_addess = "000" and Z_ address = "000" based on output node, data_in = "43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40 54 4D 55 40" in hexadecimal or **ComputerTMU@ TMU@ComputerTMU@TMU@ in ASCII.** The same data is from source router R1 <255:0>. When Write_en = '0', Read_in = '1', the destination node R2 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

Test -2 (Torus): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', in_node_address = "010000000" out_node_address = "010010000", X_address = "010" Y_addess = "010" and Z_ address = "000" based on output node, data_in = "54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40" in hexadecimal or **TMU@ComputerTMU@ TMUComputerTMU@ in ASCII.** The same data is from source router R6 <255:0>. When Write_en = '0', Read_in = '1', the destination node R8 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

Test -3 (Ring): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', in_node_ address = "000001001" out_node_address = "000010001", based on output node, data_in = "54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72" in hexadecimal or **TMU@TMU@ComputerTMU@ TMU@Computer @ in ASCII.** The same data is from source router R12 <255:0>. When Write_en = '0', Read_in = '1', the destination node R15 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010" in binary.

Test -4 (Fat Tree): First of all reset = '1' and run. It will make all the router or nodes data to zero. Then reset = '0' and give direct positive clock signal. Assign the Write_en = '1', in_node_address = "000001000" out_node_address = "001001000", "000010000", "000001001 based on output node, data_in = "43 6F 6D 70 75 74 65 72 43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 54 4D 55 40 54 4D 55 40" in hexadecimal or **ComputerComputer TMU@TMU@TMU@ TMU@ in ASCII.** The same data is from source node M3 <255:0>. When Write_en = '0', Read_in = '1', the destination node M4 <255:0>, M6 <255:0> , M12 <255:0> getting the data. Data_out< 255:0> are getting the same data. Data_out = "0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

6.5 Hardware Detail on FPGA Synthesis

All the chips are synthesized on Virtex 5 FPGA. The hardware utilization depends on the most of the parameters. The parameters relating to the hardware chip are Number of Slice Flip-flops, Number of Slices,Number of GCLK, Number of 4 input LUTs and Number of Slice Flip-flops. The hardware parameter summary for 2D and 3D NoC router is listed in table 6.5.

Table 6.5 Hardware parameters summary for 2D and 3d NoC Router

Hardware Parameter	2D Router	3D Router
No. of Slices	130 /12480	131/12480
No. of Slice Flipflops	145/12480	148/12480
No. of 4 input LUTs	8/512	9/512
No. of Bounded IOBs	130/172	131/172
No. of GCLK	1	1

The description of the required values of FPGA hardware for 2D mesh, 2D torus, 2D, ring and 2D fat tree topology for cluster size (N= 2, 4, 8, 16, 32, 64, 128, 256) is listed in table 6.6, table 6.7, table 6.8, and table 6.9 respectively. The corresponding graphs for Hardware utilization with cluster size in 2D mesh NoC, 2D torus NoC, 2D ring NoC and 2D fat tree NoC are listed in Fig. 6.33, Fig. 6.34, Fig. 6.35, and Fig. 6.36 respectively.

Table 6.6 Hardware parameters summary for 2D mesh NoC

Network Size	Slices Number	Slice Flip-flops number	Four input LUTs number	Bounded IOBs numbers	Number of GCLK
N =2	12	34	6	8	1
N=4	20	39	10	16	1
N=8	32	52	24	30	1
N=16	40	60	30	56	1
N=32	46	72	42	84	1
N=64	50	78	52	112	1
N=128	61	86	64	136	1
N=256	82	104	80	148	1

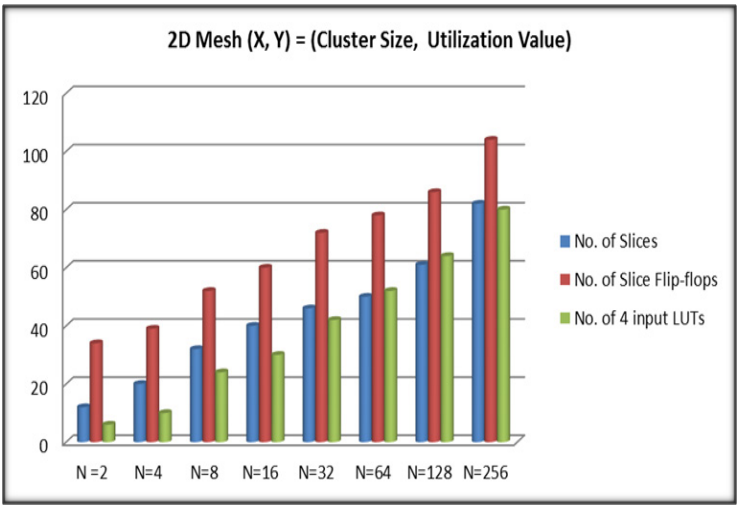


Figure 6.33 Hardware utilization with cluster size in 2D mesh NoC

Table 6.7 Hardware parameters summary for 2D Torus NoC

Network Size	Slices Number	Slice Flip-flops number	Four input LUTs number	Bounded IOBs numbers	Number of GCLK
N =2	16	38	8	8	1
N=4	22	44	16	16	1
N=8	35	58	26	30	1
N=16	42	65	35	56	1
N=32	49	78	48	84	1
N=64	56	82	55	112	1
N=128	66	90	68	136	1
N=256	**86**	110	92	148	1

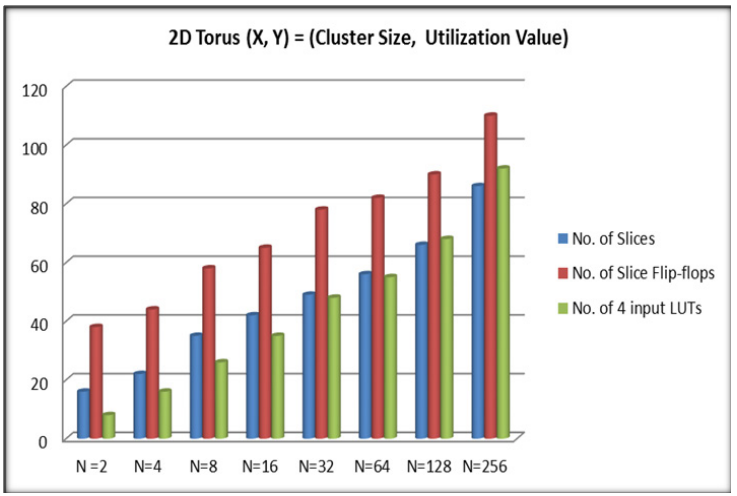


Figure 6.34 Hardware utilization with cluster size in 2D torus NoC

Table 6.8 Hardware parameters summary for 2D ring NoC

Network Size	Slices Number	Slice Flip-flops number	Four input LUTs number	Bounded IOBs numbers	Number of GCLK
N =2	19	40	9	8	1
N=4	24	46	18	16	1
N=8	36	60	28	30	1
N=16	48	68	36	56	1
N=32	51	84	50	84	1
N=64	58	92	58	112	1
N=128	70	104	72	136	1
N=256	92	110	98	148	1

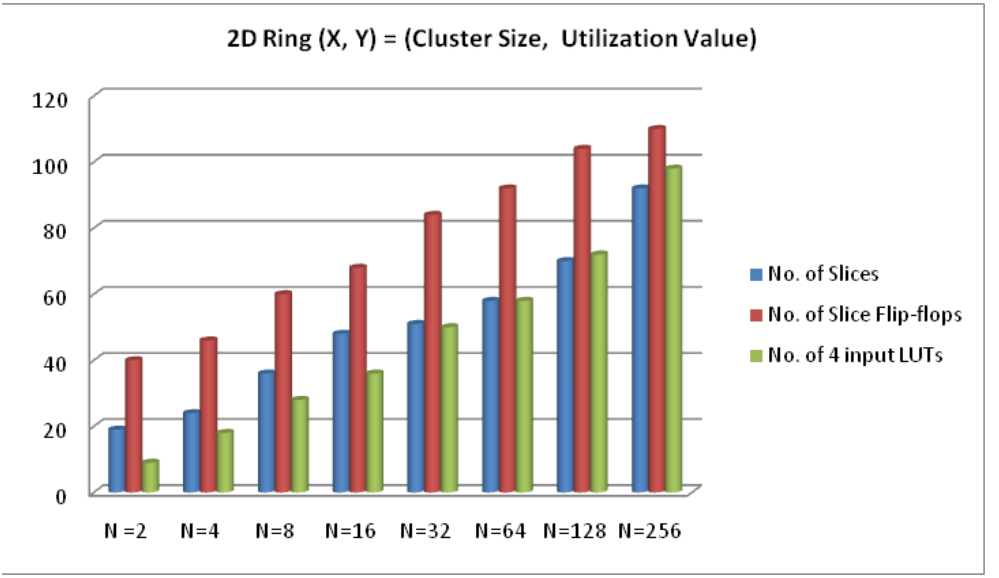


Figure 6.35 Hardware utilization with cluster size in 2D Ring NoC

Table 6.9 Hardware parameters summary for 2D Tree NoC

Network Size	Slices Number	Slice Flip-flops number	Four input LUTs number	Bounded IOBs numbers	Number of GCLK
N =2	24	42	10	8	1
N=4	28	48	18	16	1
N=8	40	61	30	30	1
N=16	52	69	36	56	1
N=32	55	84	52	84	1
N=64	60	96	58	112	1
N=128	74	110	76	136	1
N=256	96	112	104	148	1

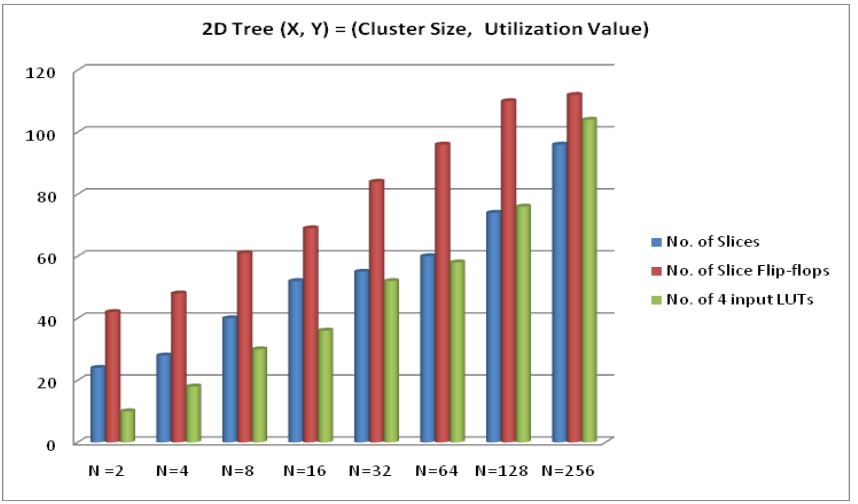


Figure 6.36 Hardware utilization with cluster size in 2D tree NoC

The detailed FPGA hardware for 3D mesh, 3D torus, 3D ring and 3D fat tree topology for cluster size (N= 2, 4, 8, 16, 32, 64, 128, 256) is listed in table 6.10, table 6.11, table 6.12, and table 6.13 respectively. The corresponding graphs for Hardware utilization with cluster size in 3D mesh NoC, 3D torus NoC, 3D ring NoC and 3D fat tree NoC are listed in Fig. 6.37, Fig. 6.38, Fig. 6.39, and Fig. 6.40 respectively.

Table 6.10 Hardware parameters summary for 3D mesh NoC

Network Size	Slices Number	Slice Flip-flops number	Four input LUTs number	Bounded IOBs numbers	Number of GCLK
N =2	10	33	6	9	1
N=4	18	36	8	18	1
N=8	28	50	23	32	1

N=16	38	57	28	58	1
N=32	45	69	40	86	1
N=64	48	75	51	116	1
N=128	60	83	62	146	1
N=256	80	102	74	160	1

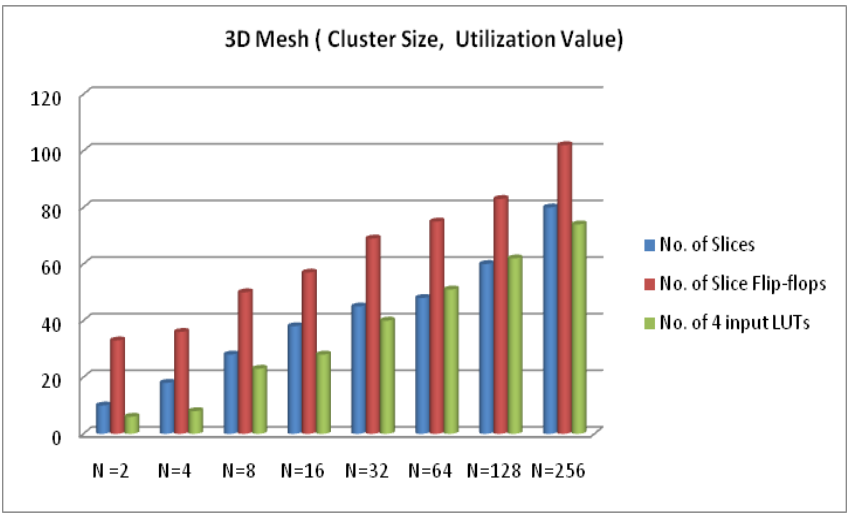


Figure 6.37 Hardware utilization with cluster size in 3D mesh NoC

Table 6.11 Hardware parameters summary for 3D Torus NoC

Network Size	Slices Number	Slice Flip-flops number	Four input LUTs number	Bounded IOBs numbers	Number of GCLK
N =2	15	36	7	9	1
N=4	20	43	15	18	1
N=8	34	56	25	32	1
N=16	40	64	32	58	1

N=32	46	76	45	86	1
N=64	54	80	52	116	1
N=128	65	88	64	146	1
N=256	82	104	90	160	1

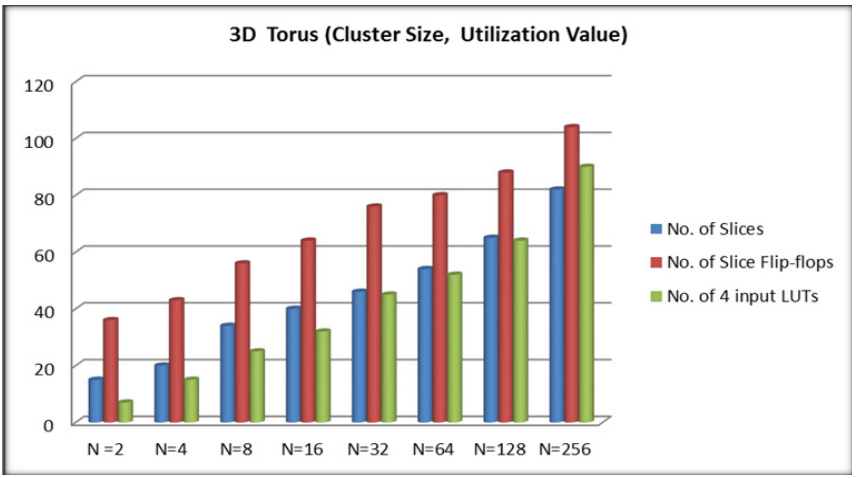


Figure 6.38 Hardware utilization with cluster size in 3D torus NoC

Table 6.12 Hardware parameters summary for 3D ring NoC

Network Size	Slices Number	Slice Flip-flops number	Four input LUTs number	Bounded IOBs numbers	Number of GCLK
N =2	18	38	8	9	1
N=4	22	45	18	18	1
N=8	34	58	26	32	1
N=16	45	66	35	58	1
N=32	50	80	45	86	1
N=64	56	90	55	116	1
N=128	68	102	70	146	1

N=256	90	106	96	160	1

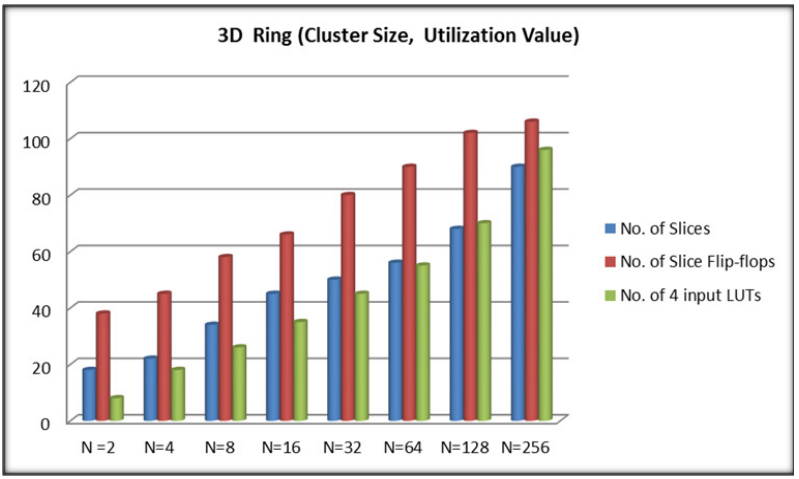


Figure 6.39 Hardware utilization with cluster size in 3D torus NoC

Table 6.13 Hardware parameters summary for 3D Tree NoC

Network Size	Slices Number	Slice Flip-flops number	Four input LUTs number	Bounded IOBs numbers	Number of GCLK
N =2	22	40	10	9	1
N=4	25	46	18	18	1
N=8	39	60	28	32	1
N=16	50	65	36	58	1
N=32	52	82	46	86	1
N=64	58	92	56	116	1
N=128	72	105	74	146	1
N=256	95	110	98	160	1

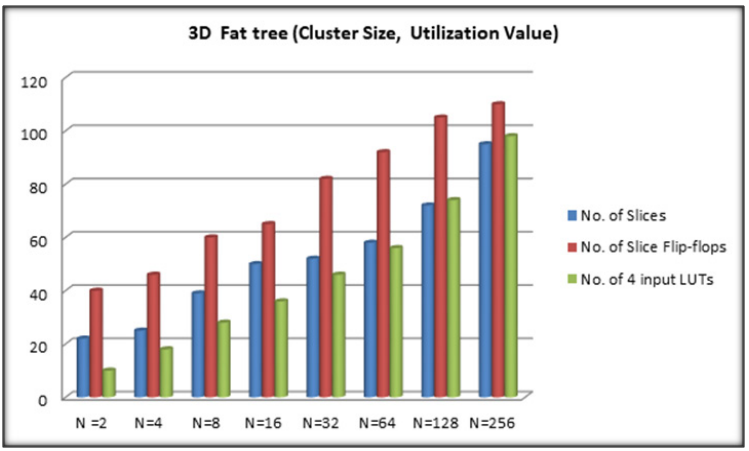


Figure 6.40 Hardware utilization with cluster size in 3D fat tree NoC

It is clear from the figures that the hardware related parameters values is increasing with cluster size of the topology network

6.6 Timing Summary

Timing parameters are very important to understand the network behaviour and performance. The timing parameters are Max Frequency, Time before clk (minimum) (ns), Time after clock (maximum) (ns) and Combinational Path delay (ns). The other parameters are memory usage depends on hardware resources.

Table 6.14 Timing results for 2D and 3D NoC Router

Timing Parameter	2D Router	3D Router
Max Frequency	180.00 MHz	215 MHz
Min Period (ns)	1.496 ns	1.210 ns
Time before clk (minimum) (ns)	4.003 ns	4.870 ns
Time after clock (maximum) (ns)	5.916 ns	5.624 ns
Combinational Path delay (ns)	8.197 ns	9.810 ns
Memory Usage	135411 kB	154923 kB

Table 6.14 Timing results for 2D and 3D NoC Router. The table 6.15, table 6.16, table 6.17, table 6.18 are presenting the results corresponding to 2D mesh, 2D torus, 2D ring and 2D fat tree topology. The timing analysis graph for all topology is presented in Fig. 6.41, Fig. 6.42, Fig. 6.43 and Fig. 6.44 with respect to cluster size (N =2, 4, 8, 16, 32, 64, 128 and 256). From the graph figure it is clear that the timing values such as time before clk (minimum) (ns), time after clock (maximum) (ns) and min period (ns) are increasing with the cluster size of the network. In the same way, table 6.19, Table 6.20, Table 6.21, Table 6.22 are presenting the results corresponding to 3D mesh, 3D torus, 3D ring and 3D fat tree topology. The timing analysis graph for all topology is presented in Fig. 6.45, Fig. 6.46, Fig. 6.47 and Fig. 6.48 with respect to cluster size (N =2, 4, 8, 16, 32, 64, 128 and 256). From these graphs also, it is clear that the timing values such as time before clk (minimum) (ns), time after clock (maximum) (ns) and min period (ns) are increasing with the cluster size of the network.

Table 6.15 Timing detailed parameters for 2D mesh NoC

Network Size	Max Frequency	Min period(ns)	Time (ns) before clk (minimum)	Time (ns) after clock (maximum)	Memory Usage (kB)
N =2	135	0.906	2.045	3.011	9123
N=4	145	0.912	2.056	3.089	101450
N=8	170	0.923	2.145	3.110	124781
N=16	195	0.982	2.187	3.188	135190
N=32	215	1.004	2.210	3.214	201012
N=64	235	1.101	2.406	3.248	201130
N=128	400	1.211	2.451	3.349	210412
N=256	925	1.320	2.504	3.451	221005

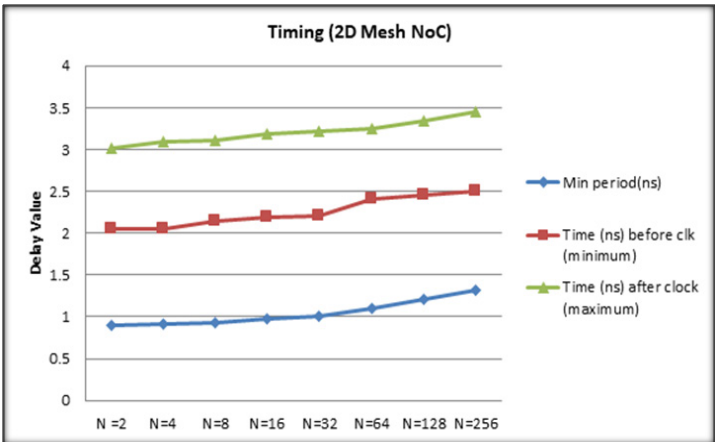


Figure 6.41 Timing values with cluster size in 2D mesh NoC

Table 6.16 Timing detailed parameters for 2D torus NoC

Network Size	Max Frequency	Min period(ns)	Time (ns) before clk (minimum)	Time (ns) after clock (maximum)	Memory Usage (kB)
N =2	130	0.880	2.045	3.011	9420
N=4	140	0.902	2.056	3.089	114510
N=8	170	0.911	2.145	3.110	135121
N=16	190	0.971	2.187	3.188	136120
N=32	211	1.000	2.210	3.214	210201
N=64	234	1.009	2.417	3.248	218019
N=128	400	1.194	2.420	3.349	215129
N=256	910	1.318	2.504	3.451	228107

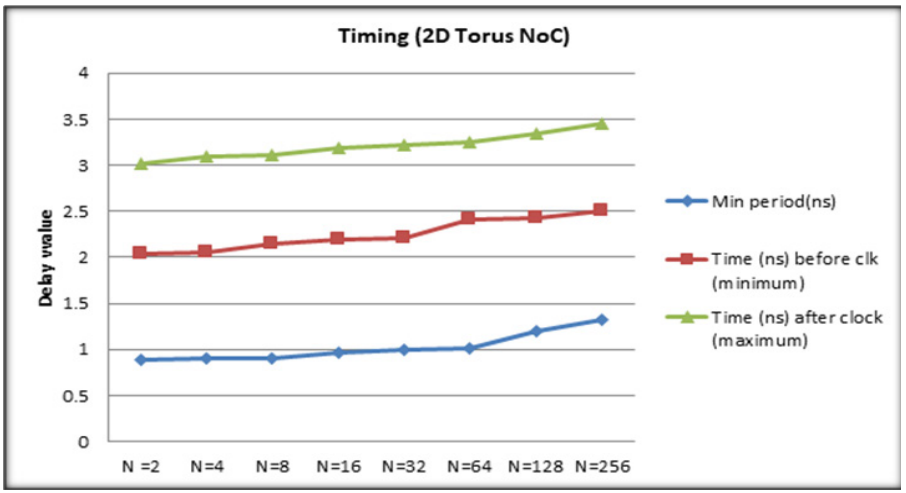


Figure 6.42 Timing values with cluster size in 2D torus NoC

Table 6.17 Timing detailed parameters for 2D ring NoC

Network Size	Max Frequency	Min period(ns)	Time (ns) before clk (minimum)	Time (ns) after clock (maximum)	Memory Usage (kB)
N =2	130	0.781	2.012	3.000	9587
N=4	138	0.819	2.051	3.080	122529
N=8	170	0.891	2.089	3.101	138121
N=16	190	0.969	2.178	3.108	136127
N=32	210	0.998	2.200	3.210	214510
N=64	230	1.004	2.416	3.218	218240
N=128	400	1.094	2.417	3.320	223412
N=256	900	1.214	2.459	3.381	251318

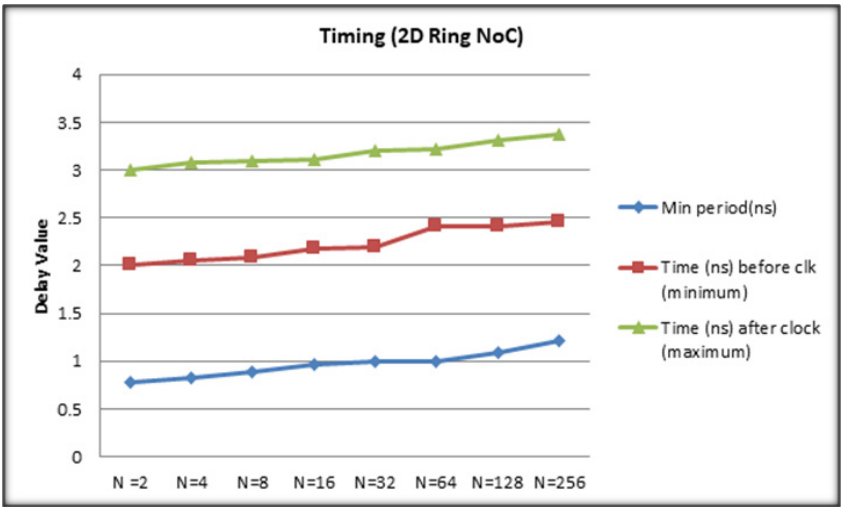


Figure 6.43 Timing values with cluster size in 2D ring NoC

Table 6.18 Timing detailed parameters for 2D tree NoC

Network Size	Max Frequency	Min period(ns)	Time (ns) before clk (minimum)	Time (ns) after clock (maximum)	Memory Usage (kB)	Speed Grade
N =2	130	0.779	2.008	2.981	9540	-5
N=4	135	0.790	2.001	3.001	122295	-5
N=8	175	0.882	2.010	3.019	137256	-5
N=16	185	0.911	2.150	3.100	135210	-5
N=32	210	0.921	2.195	3.182	213219	-5
N=64	225	0.917	2.391	3.205	219615	-5
N=128	390	1.002	2.398	3.209	231297	-5
N=256	875	1.179	2.410	3.211	251200	-5

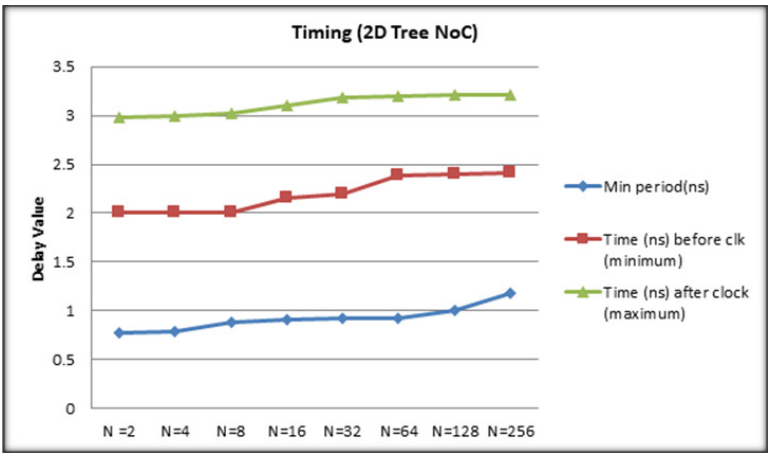


Figure 6.44 Timing values with cluster size in 2D tree NoC

Table 6.19 Timing detailed parameters for 3D mesh NoC

Network Size	Max Frequency	Min period(ns)	Time (ns) before clk (minimum)	Time (ns) after clock (maximum)	Memory Usage (kB)	Speed Grade
N =2	235	0.912	2.251	4.239	125681	-5
N=4	320	1.023	2.348	4.312	231467	-5
N=8	365	1.187	2.451	4.417	281369	-5
N=16	400	1.267	2.488	4.512	312034	-5
N=32	425	1.371	2.512	4.621	383412	-5
N=64	675	1.429	2.598	4.901	421628	-5
N=128	825	1.467	2.613	5.014	456721	-5
N=256	987	1.560	2.718	5.128	491234	-5

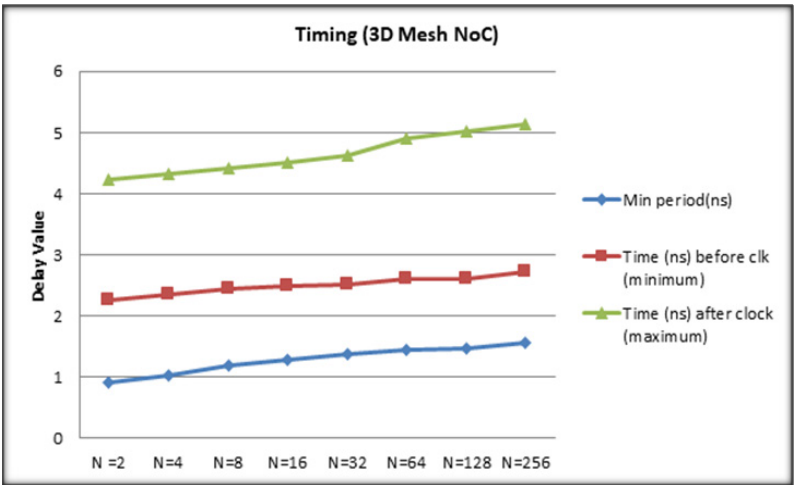


Figure 6.45 Timing values with cluster size in 3D mesh NoC

Table 6.20 Timing detailed parameters for 3D torus NoC

Network Size	Max Frequency	Min period(ns)	Time (ns) before clk (minimum)	Time (ns) after clock (maximum)	Memory Usage (kB)	Speed Grade
N =2	232	0.907	2.119	4.230	135967	-5
N=4	315	1.008	2.241	4.300	233456	-5
N=8	360	1.117	2.317	4.399	291249	-5
N=16	395	1.216	2.417	4.401	320120	-5
N=32	418	1.370	2.448	4.599	394312	-5
N=64	667	1.410	2.490	4.812	437827	-5
N=128	821	1.450	2.531	5.001	467612	-5
N=256	950	1.500	2.614	5.119	501234	-5

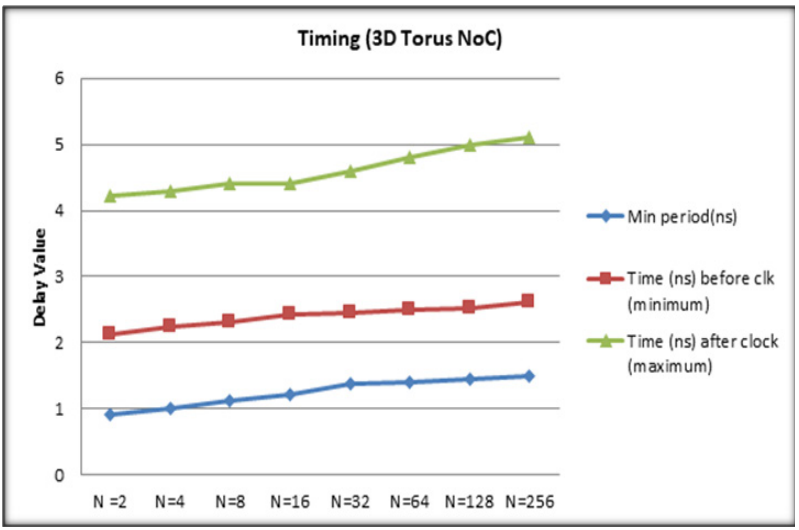


Figure 6.46 Timing values with cluster size in 3D torus NoC

Table 6.21 Timing detailed parameters for 3D ring NoC

Network Size	Max Frequency	Min period(ns)	Time (ns) before clk (minimum)	Time (ns) after clock (maximum)	Memory Usage (kB)	Speed Grade
N =2	230	0.900	2.110	4.190	144176	-5
N=4	314	0.991	2.214	4.212	240065	-5
N=8	356	1.017	2.210	4.291	300347	-5
N=16	387	1.196	2.317	4.301	330001	-5
N=32	412	1.370	2.390	4.412	403419	-5
N=64	650	1.391	2.399	4.617	448772	-5
N=128	810	1.450	2.500	4.990	486721	-5
N=256	925	1.500	2.599	5.091	521009	-5

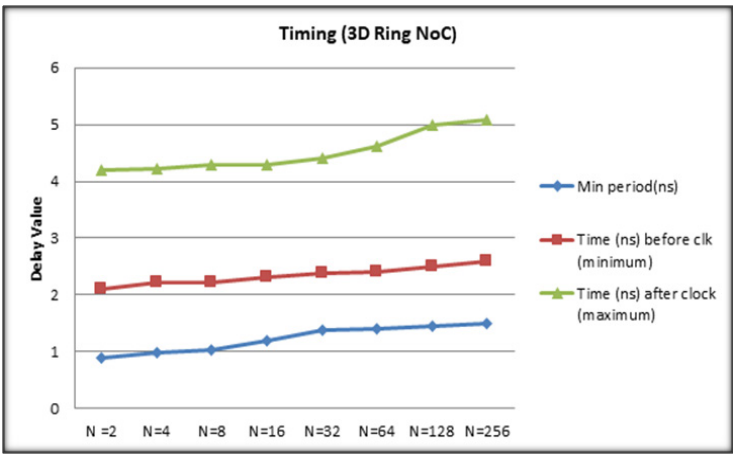


Figure 6.47 Timing values with cluster size in 3D ring NoC

Table 6.22 Timing detailed parameters for 3D fat tree NoC

Network Size	Max Frequency	Min period(ns)	Time (ns) before clk (minimum)	Time (ns) after clock (maximum)	Memory Usage (kB)	Speed Grade
N =2	214	0.812	1.959	3.790	159176	-5
N=4	310	0.928	2.140	3.794	252129	-5
N=8	350	1.009	2.199	4.091	324560	-5
N=16	378	1.161	2.371	4.191	354519	-5
N=32	400	1.271	2.310	4.239	444523	-5
N=64	610	1.311	2.319	4.587	483219	-5
N=128	799	1.405	2.419	4.782	524217	-5
N=256	900	1.500	2.581	4.961	551756	-5

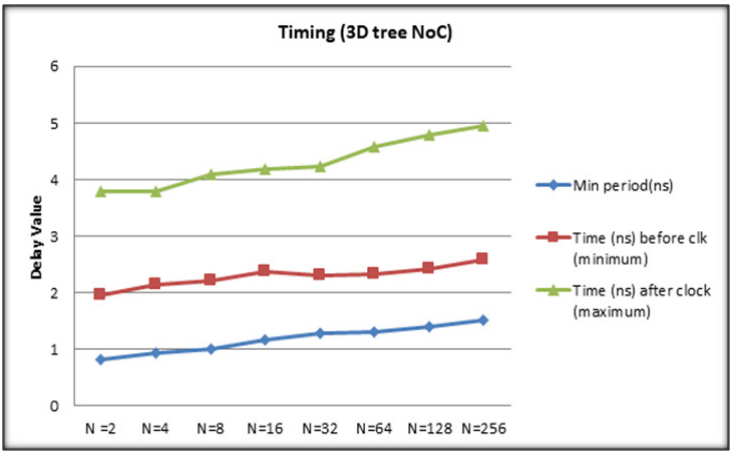


Figure 6.48 Timing values with cluster size in 3D fat tree NoC

6.7 Comparative Analysis

The comparative analysis of the 2D and 3D mesh, torus, ring and fat tree is done with respect to hardware usage and timing values obtained from the simulation on FPGA synthesis results. The comparison among the topology is done based on Frequency support, memory usage, No. of slices and flip-flops required. Fig. 6.49 and Fig. 6.50 presented the frequency (MHz) analysis for 2D (mesh_torus_ring_fatree) and 3D (mesh_torus_ring_fatree) respectively. From the graph, it is clear that mesh has higher frequency support with cluster size. The frequency of 2D and 3D mesh, torus, ring and fat tree is increasing with cluster size. It signifies that the system is becoming faster with cluster size. It is also important to note that the maximum frequency support is in mesh in compassion to torus, ring and fat tree. Fig. 6.51 and fig. 6.52 are presenting the memory usage for 2D (mesh_torus_ring_ fatree) and 3D (mesh_torus_ring_fatree).It is clear that mesh topology is having less memory in comparison to torus, ring and fat tree. It is also noticed that the memory is increasing with cluster size increment. It is obvious that it will increase

because memory directly related to slices and flip-flops. Fig. 6.53 and fig. 6.54 are presenting the slices utilization for 2D (mesh_torus_ring_fatree)and 3D (mesh_torus_ring_fatree). Fig. 6.55 and fig. 6.56 are presenting the flip flops usage for 2D (mesh_torus_ring_fatree)and 3D (mesh_torus_ring_fatree). The slices and flip flops are increasing with cluster size but found of less utilization in mesh topology in comparison to torus, ring and fat tree topology.

When the designer is designing the chip then the slices and flip-flops values can be optimized based on the hardware optimization techniques. The details of all the parameters are extracted directly from the chip design software Xilinx 14.2. The other sub parameters are number of gates utilization, number of multiplexers, number of DSS an element by those are not so important in comparison to memory usage, slices utilization and flip flips. All the simulation results concluded that the 3D mesh topology is the optimal solution in terms of performance, hardware and memory in comparison to the 3D torus, ring and fat tree topology. Another important fact it that the nodes can communicate with equal distance and throughput in the topology is more.

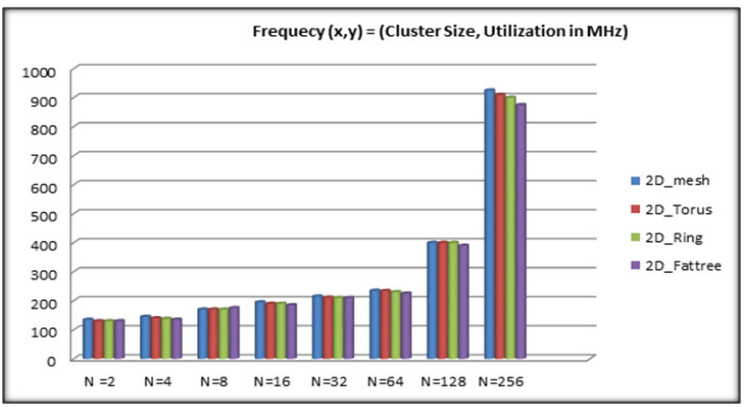


Figure 6.49 Frequency analysis for 2D (mesh_torus_ring_ fatree)

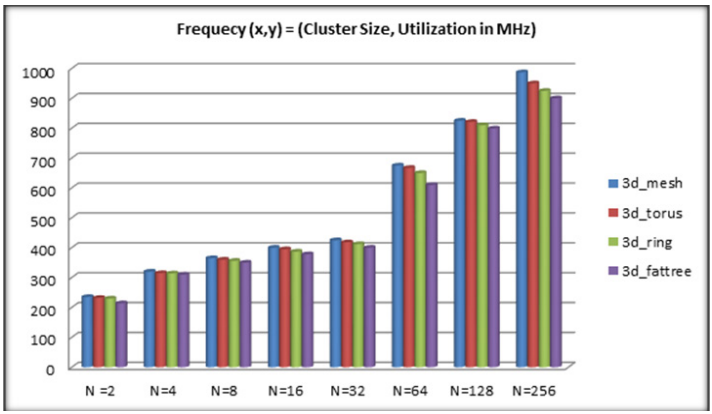


Figure 6.50 Frequency analysis for 3D (mesh_torus_ring_ fatree)

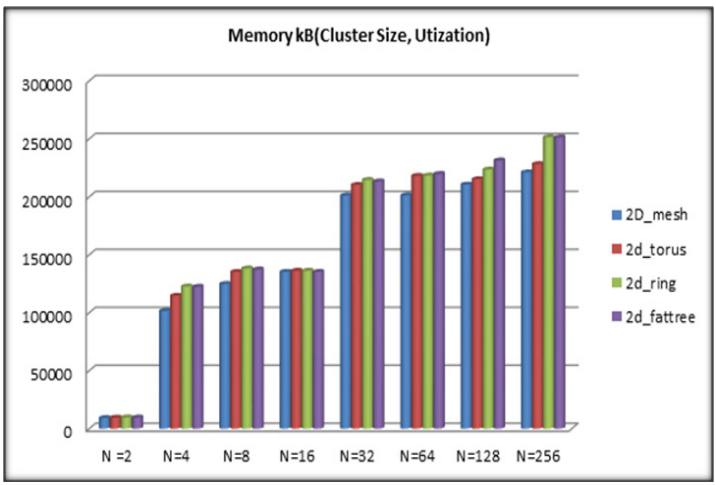


Figure 6.51 Memory usage for 2D (mesh_torus_ring_fatree)

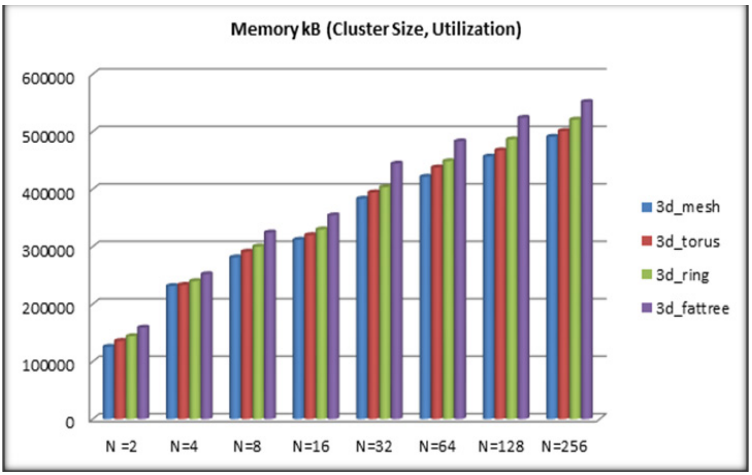


Figure 6.52 Memory usage for 3D (mesh_torus_ring_fatree)

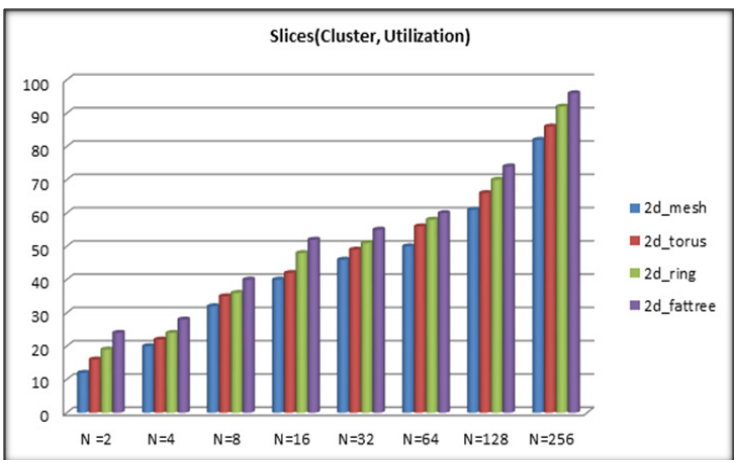


Figure 6.53 Slices utilization for 2D (mesh_torus_ring_ fatree)

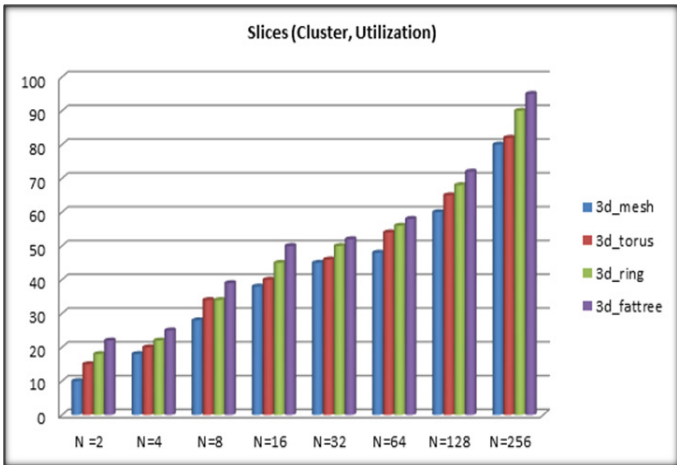


Figure 6.54 Slices utilization for 3D (mesh_torus_ring_ fatree)

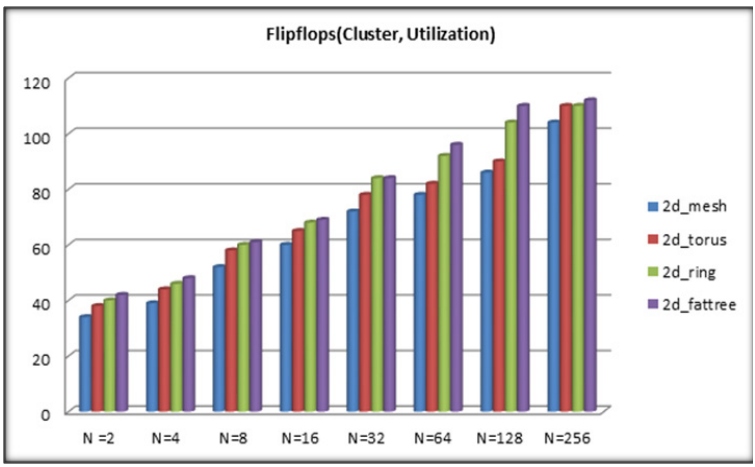


Figure 6.55 Flipflop utilization for 2D (mesh_torus_ring_ fatree)

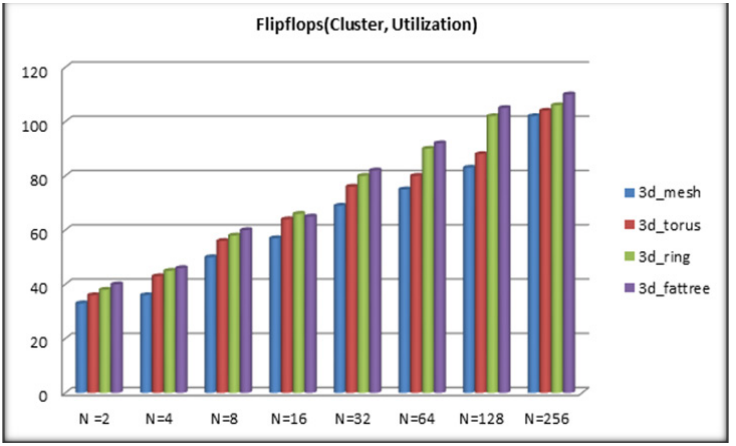


Figure 6.56 Flipflop for 3D (mesh_torus_ring_fatree)

6.8 Verification and Validation

The design of all the developed chip is verified using some test input and validate in the FPGA hardware shown in fig. 6.57 and Fig. 6.58. The control input is given using the VHDL code is burned in Virtex 5 FPGA and the data of the destination node is shown output screen of another PC. It is Virtex 5 FPGA having XC5VLX110T device, manufactured by Digilent Company. Test input are also given using input switches and verified on LEDs as output for small data. The data is also shown on the PC using VGA. Fig.6.59 shows the successful hardware burning of the VHDL program. Fig. 6.60 shows the experimental validation of the receiving the data **TMU@TMU@ComputerTMU@TMU@**Computeron Computer Screen using VGA cable. The verification is done for the following test samples.

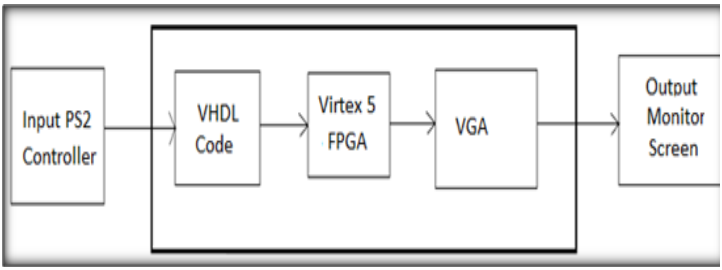


Figure 6.57 FPGA verification process

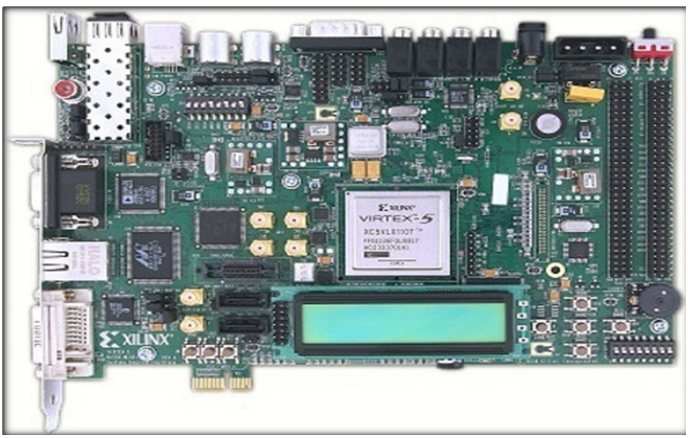


Figure 6.58 Virtex 5 FPGA in pictorial View

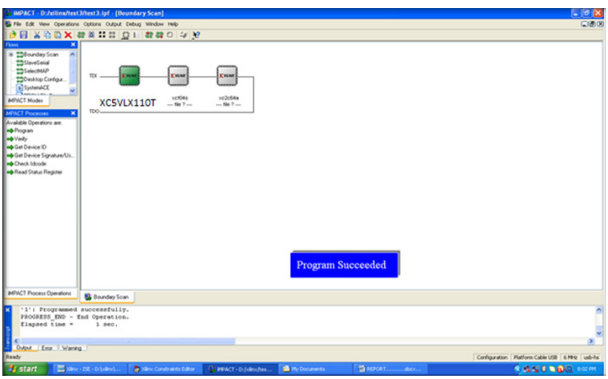


Figure 6.59 FPGA with successful program synthesis

Figure 6.60 Experimental validation and verification

Test -1 (Mesh): in_node_address = "001000000" out_node_address = "010000000", X_address = "010" Y_addess = "000" and Z_address = "000" based on output node, data_in = "43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40 54 4D 55 40" in hexadecimal or **ComputerTMU@TMU@ComputerTMU@TMU@ in ASCII.** Source router R1 <255:0>. The destination node R2 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

Test -2 (Torus): in_node_address = "010000000" out_node_address = "010010000", X_address = "010" Y_addess = "010" and Z_address = "000" based on output node, data_in = "54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 7254 4D 55 40" in hexadecimal or TMU@ ComputerTMU@TMUComputerTMU@ in ASCII. The same data is from source router R6 <255:0>. The destination node

R8 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

Test -3 (Ring): in_node_address = "000001001" out_node_ address = "000010001", based on output node, data_in = "54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 43 6F 6D 70 75 74 65 72" in hexadecimal or TMU@TMU@ComputerTMU@TMU@Computer @ in ASCII. The destination node R15 <255:0> and data_out< 255:0> are getting the same data. Data_out = "0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0100 0110 0101 0111 0010" in binary

Test -4 (Fat Tree): in_node_address = "000001000" out_node_ address = "001001000", "000010000", "000001001 based on output node, data_in = "43 6F 6D 70 75 74 65 72 43 6F 6D 70 75 74 65 72 54 4D 55 40 54 4D 55 40 54 4D 55 40 54 4D 55 40" in hexadecimal or ComputerComputer TMU@TMU@TMU@ TMU@ in ASCII. The same data is from source node M3 <255:0>, the destination node M4 <255:0>, M6 <255:0> , M12 <255:0> getting the data. Data_out< 255:0> are getting the same data. Data_out = "0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0100 0011 0110 1111 0110 1101 0111 0000 0111 0101 0111 0101 0110 0101 0111 0010 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000 0101 0100 0100 1101 0101 0101 0100 0000" in binary.

The data of all the test input is verified on FPGA connected on monitor screen