IMPLEMENTATION OF HIGH PERFORMANCE ROUTER USING FSM, SYNCHRONIZER AND FIFO

PROJECT REPORT
MASTER OF TECHNOLOGY IN
VLSI DESIGN AND EMBEDDED SYSTEMS

Submitted by:
Manshi Singh
2K20/VLS/11

Under the supervision of
Mr. LOKESH GAUTAM
Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) New Delhi-110042

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) New Delhi-110042

CANDIDATE'S DECLARATION

I MANSHI SINGH student of M.Tech (VLSI and Embedded Systems), hereby declare that the project Dissertation titled **"IMPLEMENTATION OF HIGH PERFORMANCE ROUTER USING FSM, SYNCHRONIZER AND FIFO"** which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, for open area seminar, is original and not copied from any source without proper citation.

Place: Delhi
Date: 22.05.2022

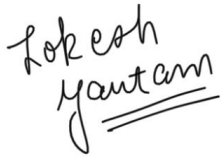
Manshi Singh
(2K20/VLS/11)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) New Delhi-110042

CERTIFICATE

I hereby certify that the Project Report titled **"IMPLEMENTATION OF HIGH PERFORMANCE ROUTER USING FSM, SYNCHRONIZER AND FIFO"** which is submitted by **Manshi Singh (2K20/VLS/11)** of Electronics and Communication Department, Delhi Technological University, Delhi for the Innovative work , is a record of the project work carried out by the students under my supervision.



Place: Delhi
Date: 31.05.2022

Mr. LOKESH GAUTAM
Assistant Professor

ACKNOWLEDGEMENT

A successful project can never be prepared by the efforts of the person to whom the project is assigned, but it also demands the help and guardianship of people who helped in completion of the project. I would like to thank all those people who have helped us in this research and inspired me during my study.

With profound sense of gratitude, I thank Mr. Lokesh Gautam my Research Supervisor, for his encouragement, support, patience and his guidance in this project work. I heartily appreciate the guidance given by him in the project presentation that has improved my presentation skills with his comments and advices. I take immense delight in extending my acknowledgement to my family and friends who have helped me throughout this project work.

Manshi Singh
Roll no: 2K20/VLS/11

ABSTRACT

Routing is defined as procedure of transferring any sort of data in compact form from sender to receiver and this is a step by step process i.e. one after another until target receiver is found. This process requires some sort of data that has the address of the client or destination. This process of routing message from source to client or destination happens at networking layer of OSI model. Hence it follows all the protocols of networking layer. The transmission or transition of data through router occurs in the form of packets, which usually contains header byte, payload and parity (trailer). Header will contain destination or client address and the upcoming size of payload byte. Payloads are the actual message or data which needs to be forwarded to destination network by the router. Trailer byte is for error detection. We use router as a networking device that advances data frames with computer and networks on other side. Router can be linked to more than two lines from various networks (this process is totally opposite of working of network switch that is used to establish a link data line and network). Project that is performed by me is basically based on few topic like working of router, its top model architecture, working of its other parts i.e. FIFO, FSM and Register.

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

A router is a device used in networking, universally dedicated hardware that forward data package from one computer networks to another. This forms overlay between internetworks, as routers are connected in between two or multiple data lines from various networks. Whenever a packet which contains any information (data) is encountered at any network line, router will try to get the address of the its receiver that is stored in that data packet which was encountered. After collecting information this will send the packet to next network according to the its table. Functioning of routers are similar to "traffic flow guide". The process of sending data packet is from sender to receiver that is routers with the help of different network till it reaches to receiver side router. Home-based and minor office routers are most familiar sort of routers which only allows data like email, web page, videos, and IM between a household computers and interent.

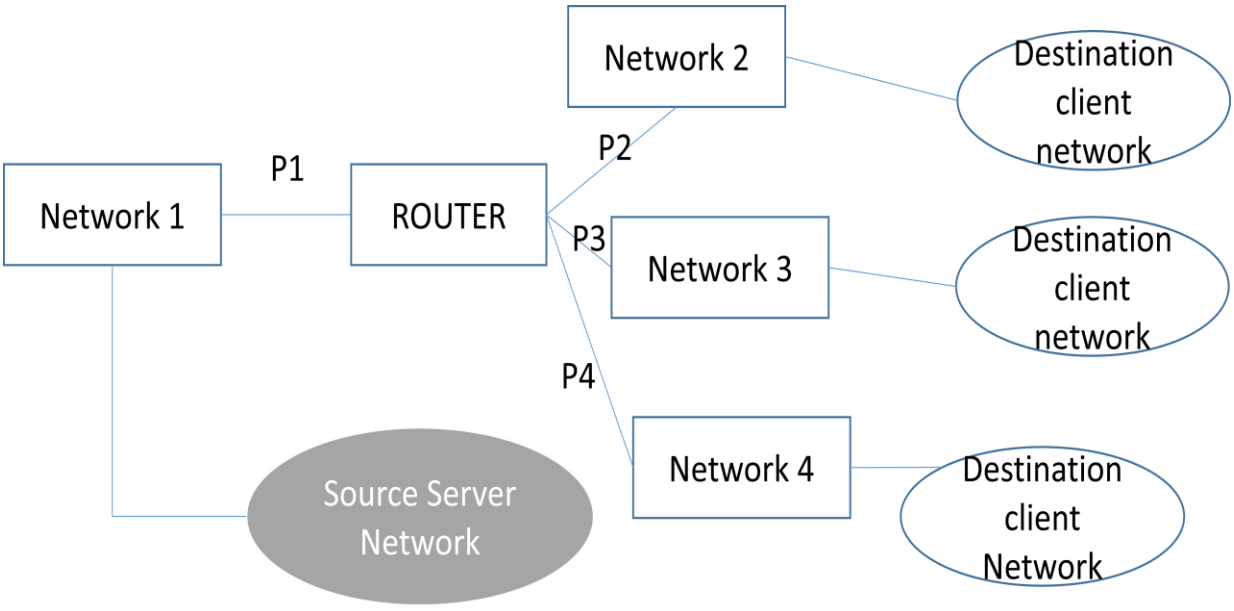


Figure1.1 – Connectivity of Router

1.1 Types of Router:

1) On basis of Configuration or Flexibility:

1.1.1 Non-modular router

Generally, Low-end routers are referred as Non-modular router which have interfaces which cannot be changed, like if we want to add or subtract ports or interface at later stage then we cannot do so. These routers are mainly accommodated in houses or small groups to connect. Non-modular type of router gives SLIP or PPP connection as well as it is compatible with some other protocols and few of them are virtual private network protocols PPTP and IPSec, etc. Technology such as ADSL will shortly improve the whole situation existing for every broadband, and this leads will lead to increment in the access router afford. Due to scenarios like this, Non-modular router will be able establish compatibility between various heterogeneous and high-speed ports in the future, and with the capability to execute more than one protocols on every port.

1.1.2 Modular router

Router which is able to extendable with particular components are described by the term Modular router, we can add router with features such as encryption accelerators, sound processing modules, interfaces, etc. This type of routers basically denotes routers like interface type and some extension which are accommodated as per requirements of user router and configurable at the factory, these types of devices mainly gives the basic routing features, like users can establish a connection to a fixed sort of network to choose the useful module, different connectivity and management capabilities can be given by various modules. Taking an example, majorly modular router provides flexibility to users so that they can choose network interface type, modular routers are able to offer VPN and various functional modules, and few of the this type of routers are eligible to give firewall functions, and much more. Major number of present routers are modular router. These devices basically establishes connection among enterprises, one enterprise and other the Internet, this is inside an Internet Service Provider (ISP). As an example Cisco Juniper T1600 or CRS-1 provides interconnection between ISP, and they are incorporated inside ISP, either this can be accommodated in large scale enterprise network.

Smallest router are basically used to establish connections inside the small and home offices.

2) On the basis of Applications:

1.1.3 Edge Router

Basically this kind i.e. Edge router is incorporated at edges of any Internet Service Provider network, edge routers are majorly organized between protocols like from Border gateway protocol to one other BGP of one other Internet Service Providers or big association.

1.1.4 Subscriber Edge Router

Subscriber Edge is a kind of routers which have its place an edge of a subscribers network. This type of router are constructed to broadcast External BGP to its provider's.

1.1.5 Inter Provider Border Router

It is a kind of router which is incorporated to establish a connection between ISPs from one another. It is working like a BGP dialogue router which can maintains BGP sessions between same type of BGP dialogue routers in other provider's.

1.1.6 Core Router

Core routers is a type of router that is placed in the center or backbone structure of any LAN networks, that means it is not placed at any boundary. In illustrations, core router delivers a step-down backbone, establishing connection between distribution router begins at various office block (LAN), or Large enterprise Location (WAN). These are incorporated in any system when we want optimization for high bandwidth.

1.1.7 Wired and Wireless Routers

The widespread of houses and small scale industries networking is increasing rapidly by using this kind of routers. These kinds of routers has the power so that it can organize routing and

configuration data in particular table used for routing. It can also give the facility of segregating circulation of data package on the basis of their IP addresses. Few of wireless kind of routers syndicates the purpose of router with network switch and an firewall as an single entity.

1.2 Methodology:

Methodology can be classified into 3 main stages:-

- Xilinx ISE (Integrated Synthesis Environment) consists of total three kinds of lineups for Designing, Simulating and Implementing respectively.
- For Design and simulation simple Verilog HDL code file which is of .v type is castoff (also known as Testbench file). For Implementation Constraint File .ucf is used to design. The code for designing of module has to be set as a Top module.
- We can use any kind of modelling out of three i.e. Dataflow, Behavioral and structural modeling. Testbench file deliver different users input comprises clock signal to design which is inevitable to support simulate the design and analyze that designed module is working as per our requirement.

1.3 Motivation:

- Objective of the project is to provide efficient router architecture having three in no. output port selected according to header, which must contains address of the port.
- Old router uses OSI model; which have seven in numbers of layer so for transmitting a data package it takes additional time in compare to TCP model, which have only four in numbers of layer, so time taken for transmitting a package is less, therefore overall speed increases.

Chapter 2

Router Design

2.1 Router: Top Level block

The construction of a top level router is depicted in a diagram. SYNCHRONIZER, FSM, FIFO-0, FIFO-1, FIFO-2, and REGISTER are the components of a Router package. During the planning and implementation of the entire module, we develop each sub-module separately via RTL coding in a Verilog; and at that time we instantiate all submodules from the top using structural modelling and certain advanced Verilog structures.

2.2 Top Block of Router

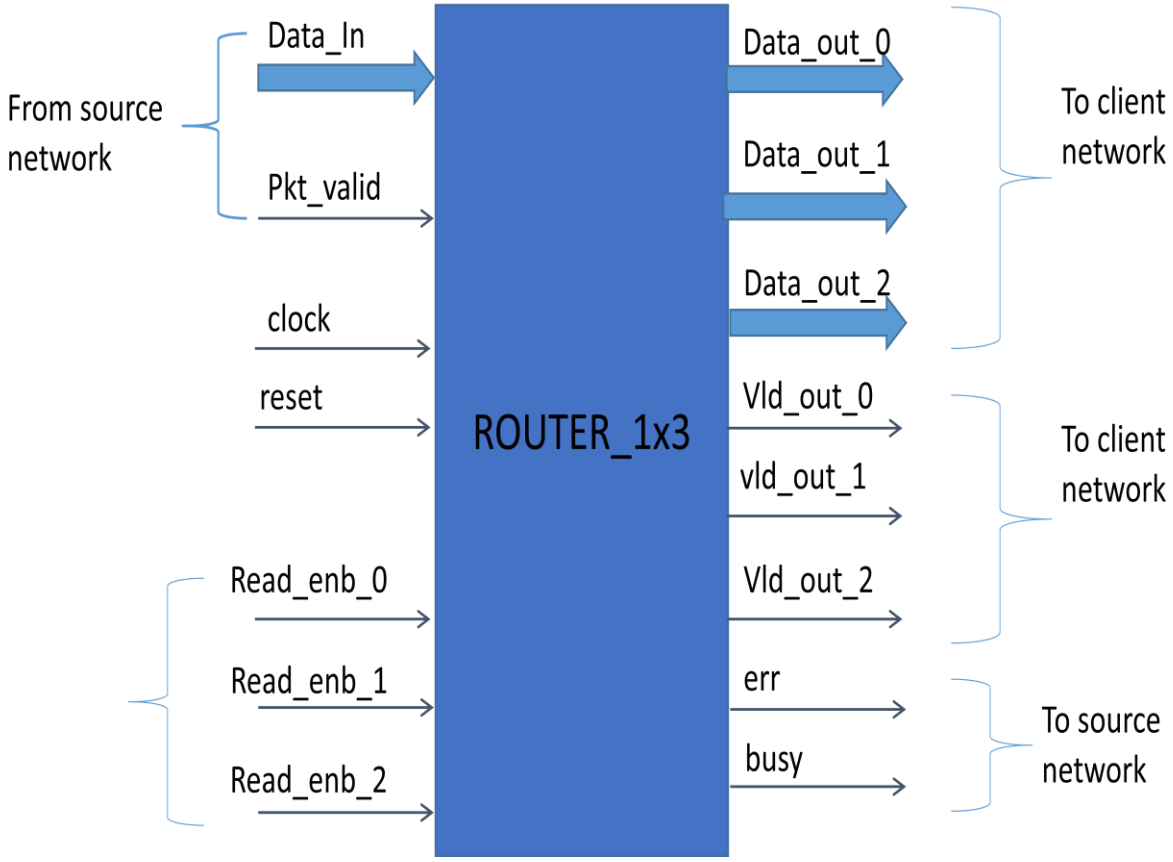


Figure2.1– Top level block

2.3 Top Level Architecture

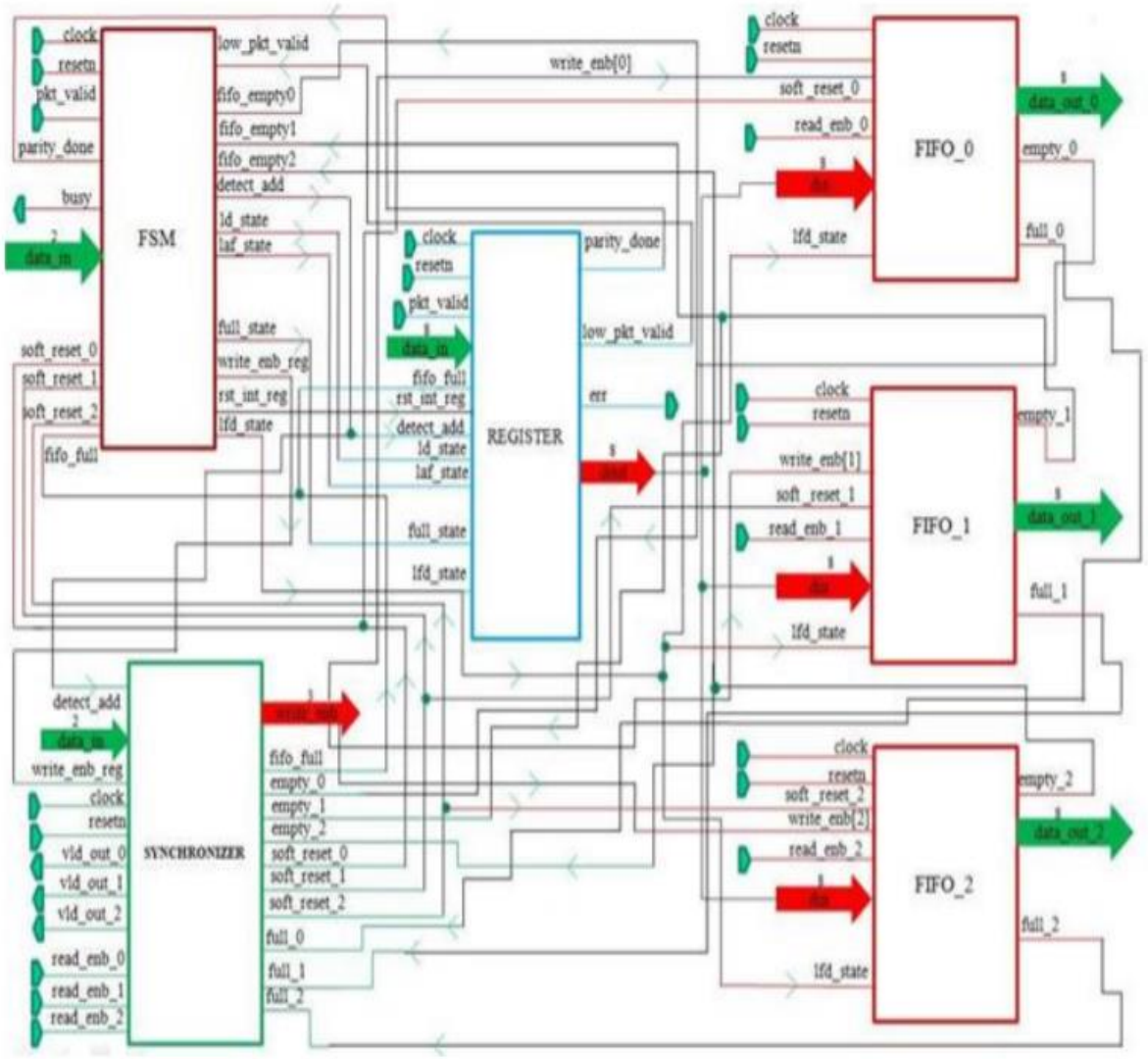


Figure2.2– Top level Architecture of Router

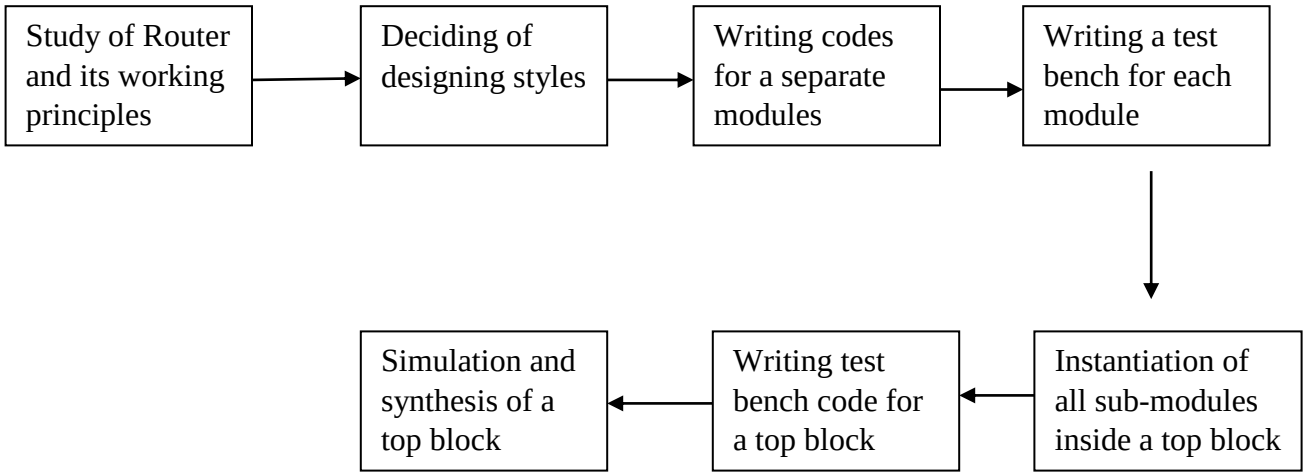
2.4 Progress of Work:

- Studied basic fundamentals of router and it's working principle.
- Studied about OSI Layers (Specially Network layer).
- Installed Xilinx ISE 14.7. Design Suite.

- For simulation and designing a simple .v form of Verilog module file is used(also acknowledged as Test bench file). A .ucf file is used for designing implementation constraints. The top module is designated as design file.

- The concept can be implemented using behavioral, dataflow, and structural modelling techniques. The Test bench file allows many different users to submit data, including a clock to the design file, which is required for simulation and testing of the design.

2.5 Work Plan:



2.6 Router Features:

- **Packet Routing:-** The packets are from obtained an input ports and is routed towards an output port dependent on an receivers networks address.

- **Parity Checking:-** An error-recognition approach that verifies the truthfulness of digital data which has been sent between the server and the client. This is a technique that guarantees that data sent by the server network is correctly received and without corruption by a client network.

- **Reset**:- The router's condition is reset by a low active synchronous input, the routers FIFOs are rendered vacant, and a valid signal goes low; indicative of none of the valid packets are recognised on output data bus.

- **Sending Packet**:- Router input protocol.

- **Reading Packet:-** Router output protocol.

2.7 Packet Format:

The packets contains 3 parts-

- Header
- Payload
- Parity

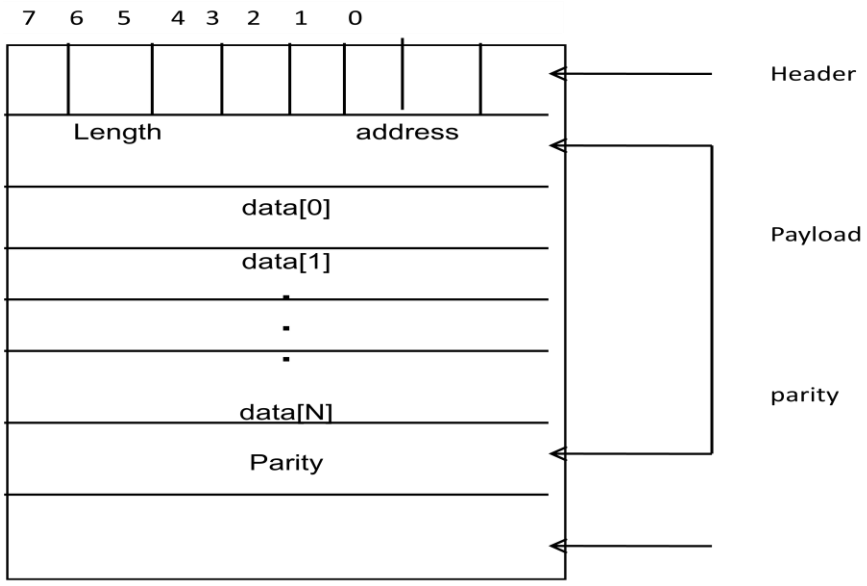


Figure1.3 – Packet Format

Each payload length and breadth of 8 bits can be expanded from 1 to 63 bytes.

Header: Package headers have 2 dimensions:- Destination Address and length.

- **Destination Address (DA)**: A package's destination address is 2-bit long. Based on the package's destination address, the router directs the package to its appropriate port. A 2-bit port address is assigned to each output port. If a package's destination address matches the address of a port, the router will send the package to the appropriate output port. '3' is not a valid address.

- **Length**: Six bits make up the data length. The number of data bytes is specified. A packet's data size cannot be less than 1 byte or greater than 63 bytes.
- If the length is equal to one, the data length is one byte.
- If the length is equal to two, the data is two bytes long.
- If the length equals three, the data length is three bytes.
- If the length equals 63, that signifies the data is 63*8 bits long..
- **Payload:** The payload contains data statistics. Bytes are used to represent the data.
- **Parity**: This parameter verifies the package's security. For header and payload bytes, parity is determined bitwise.

2.8 Router :- Input Protocol

The following are some of the features of DUT input protocols:

- **Testbench Notes:** Except for the low reset, all inputs of the module are in sync with the negative edge of systems clock. This is owing to the DUT routers sensitivity to the clock's turning point i.e. when the clock is changing from 0 to 1. As a result, inputs of the module which are driving at the clocks edge when the clock is going from 1 to 0 in the test bench ensures setup and holds time. Signals which are obtained at the clock's another edge i.e. changing from 0 to 1, can be obtained using the clocking block itself in a Verilog/UVM system-based test-bench, avoiding meta-stability.
- At the clock's turning edge when header byte is being driven into the input bus, package valid signal is asserted.
- In the meantime, the header byte provides address of receiver, followed by the router's address to which package is being directed (data_out 0; data _out 1; data_out 2).
- For every new falling edge of the clock, each following byte of payload shall be fed onto the data bus which is being used as input.
- After driving the preceding payload 8 bits to upcoming negative turning point of clock, the package-signal which is valid needs to be de-asserted, parity of the package should needs to be driven. This is being sensed as package is finished.

- When a busy signal is detected, the test-bench should not drive any of the bytes; as an alternative, it needs to have the values which are driven recently.
- If any of the incoming byte (8 bits) of information is dropped, then 'busy' signal is being asserted in the cycle.
- An "err" signal is inserted in the cycle if any of the packet's parity bit is not as required. This is done at the time when it is recognized that parity bits is mismatched.

2.9 Router :- Output Protocol

Below mentioned are the features of any output protocol:

- Test-bench note: - All the signals which are being used as output in the router module are actively high, they must be synchronized with the clock's rising edge i.e. when clock is getting from 0 to 1.
- A FIFO with a 16x9 capacity buffers each data out_x (data out_0, data out_1, and data out_2) output port.
- If any data which is genuine reaches at the vld out_x (data out_0, data out_1, and data out_2) output line, the router asserts vld out_x (vld out_0, vld out_1, and vld out_2) signal. It is in one of that signal which is used to direct to the receiver's client indicating that data is existing at specific output data line.
- If receiver's package want to grasp the package's bits before inserting read enb_x (read enb_0, read enb_1, and read enb_2) then it should be in waiting state till is has enough memory space to store data.
- When clock negative turning point i.e. transition from 1 to 0 comes, the read enb_x (read enb_0, read enb_1, or read enb_2) input signals could be inserted, allowing data to be read from data out_x (data out_0, data out_1, or data out_2) line.
- When read enb_x (read enb_0, read out_1, or read out_2) is inserted in the range of 30 clock cycles of vld out_x (vld out_0, vld out_1, or vld out_2), the FIFO will be reset.
- When a package header byte becomes corrupted due to a time-out event, then data out_x (data out_0, data out_1 or data out_2) line is in Z state i.e. high impedance state throughout the process.

Chapter 3

Sub Module Functionality Description

3.1 Router: FIFO

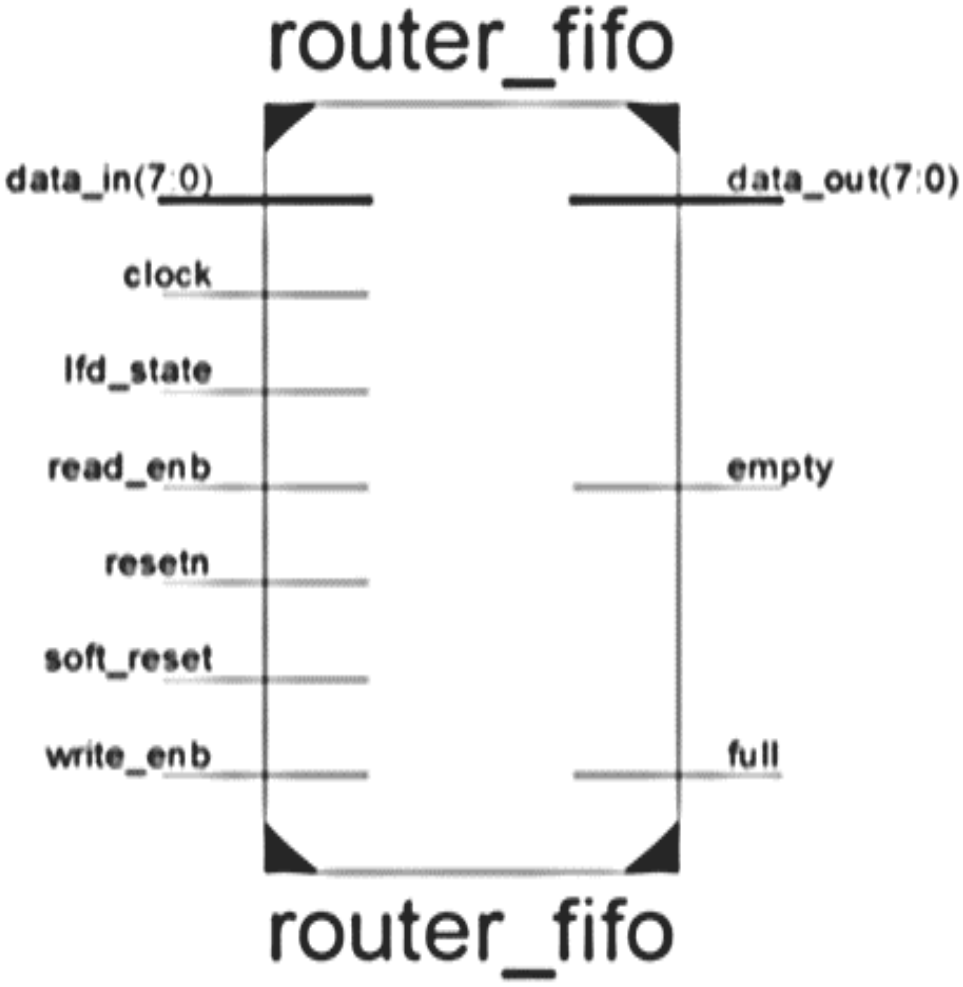


Figure3.1– RTL Schematic of Router FIFO

3.1.1 Functionality

In the router design there are 3 FIFOs used. Each FIFO size is of 16 bits or 2 bytes depth and 9 bits wide. FIFO can be reset using a synchronizer which is actively low reset and is powered by the clock which is used in the system. Internally, a soft reset signal also resets the FIFO. The SYNCHRONIZER block generates a **soft_reset** active high signal during the ROUTER's time

out state. When the value of **resetn** port's value is zero, empty=1, full is equal to 0 and data out is equal to zero 0 are the values.

Dimensions of FIFO m/m is 16*9 bits. Detecting the header byte requires the following step: extra bit in a data width is added. The header byte i.e. 8 bits of a package is sensed by **Lfd_state**. For remaining bytes, the bit which is at ninth position is 0 and for the header byte, it's value is equal to 1.

Write Operation :-

- **Data_in** signal's samples are being taken at positive transition of clock i.e. from 0 to 1 change, but only if the value **write_enb** is equal to logic 1.
- To avoid an over run condition, write operation will be performed only if FIFO is vacant i.e. it does not has any data.

3.1.2 Read operation :-

- D**ata_out** signal is being recited at positive transition of clock i.e. from 0 to 1 change, but only but only if the value **read_enb** is equal to logic 1.
- To avoid an under run state, read operation will be performed only if FIFO is not in the vacant state.
- When the header byte is read, the internal counter will have a value 1 more than the payload length (parity byte), and this value will be decreased by one after each clock until 0 is observed. Until the counter is reloaded with a fresh packet payload length, it will remain at 0.
- All through time-out condition, empty=1 and value of full is equal to 0.
- **Data_out** is being driven into a Z state i.e. high-impedance state beneath two conditions:

1. When **read_enb** reads the fifo m/m entirely (payload + header + parity).
2. When Router is under time-out conditions.

- Status Full-FIFO can be sensed as that every place within FIFO has been occupied or written .
- All locations inside FIFO are being analyzed and made empty is indicated by **Empty-FIFO** status.

3.2 Router: Synchronizer

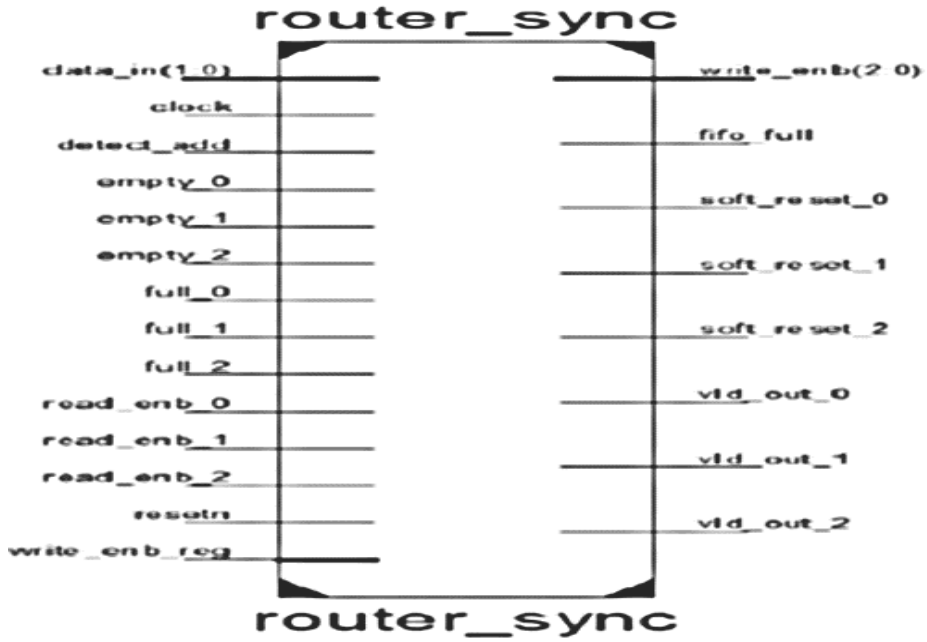


Figure3.2 – RTL Schematic of Router Synchronizer

3.2.1 Functionality:

This sort of module is used to establish synchronization among two modules of router i.e. FSM and FIFO. It's useful for establishing secure communication between a single input and three output ports.

- The **data_in** and **detect_add** signals are being utilized to pick FIFO while waiting for the package routing for that FIFO is complete.
- F**ull_status** of **FIFO_0, FIFO_1** or **FIFO_2** can be used to determine the signal **fifo _full** value.
- When **data_in** has the value equals to 2‟b00 then **full_0** value will be assigned to **fifo_full.**
- When **data_in** has the value equals to 2‟b01 then **full_1** value will be assigned to **fifo_full.**
- When **data_in** has the value equals to 2‟b02 then **full_2** value will be assigned to **fifo_full.**

- Signal **vld_out_x** is calculated which is being created on the FIFO's vacant condition as written:
- Value of **empty_0** inverted and assigned to **vld_out_0**.
- Value of **empty_1** inverted and assigned to **vld_out_1**.
- Value of **empty_2** inverted and assigned to **vld_out_2**.
- In process of writing operation in selected FIFO; for producing the **write_enb** signal, a **write_enb** register indication is also used.
- **soft_reset_0**, **soft_reset_1, soft_reset_2** are the 3 internal reset signal for each of the FIFO respectively. It becomes equal to logic 1 if **read_enb_x (read_enb_0**,**read_out_1**, **read_out_2**) is not inserted inside 30 cycle of clock of **vld_out_x (vld out 0,vld out 1 or vld out 2).**
- Simultaneously we can read and write.

3.3 Router: FSM

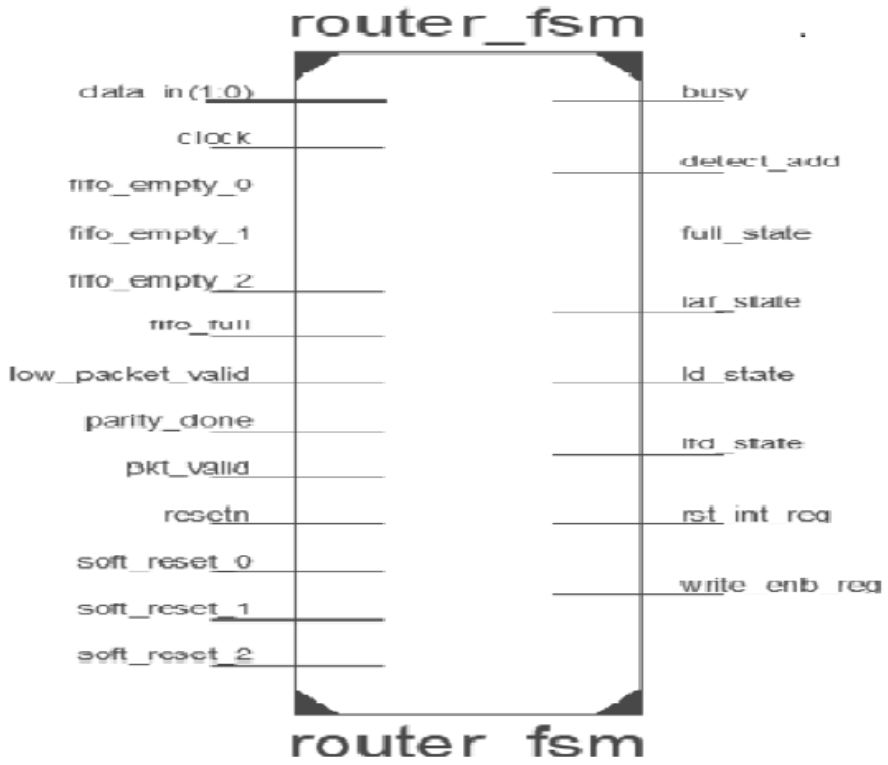


Figure3.3 – RTL Schematic of Router FSM

3.3.1 STATE -DECODE_ADDRESS

• It is being used as the initial state.

• The **detect_add** is added to state and utilized to sense any arriving packets. This is useful in latching first byte which is treated as header byte.

3.3.2 STATE-LOAD_FIRST_DATA

- The **lfd_state** signal is injected into above mentioned block and is being utilized in accommodating first 8 bits data to FIFO.
- Signal **busy** can be used in above condition to appropriately latch the header byte and prevent it from being modified with a new value for the current packet.
- In the upcoming clock, this state will be renamed LOAD DATA without any condition checking.

3.3.3 STATE-LOAD_DATA

- The **lfd_state** signal is injected into above mentioned block and is being utilized in loading payload data into FIFO.
- In above condition, busy state is de inserted, making the ROUTER ready to receive new data from the source.
- In above mentioned state, the signal **write_enb_reg** is being utilized in writing packet information i.e. summation of header, payload and parity to appropriate FIFO.
- When **pkt_valid** is low, it switches to LAOD PARITY, and when the FIFO is full, it switches to IFO_FULL_STATE. Signal busy is de inserted in above state to make ROUTER ready to accept upcoming data from sender.

3.3.4 STATE-LOAD_PARITY

- Above mentioned state is utilized to latch the last byte i.e. parity 8 bits.
- This goes without any condition to CHECK_PARITY_ERROR state.
- Signal **busy** is inserted to stop the ROUTER from receiving any upcoming data.

- **write_enb_reg** is set as logic 1 to latch the parity 8 bits to FIFO.

3.3.5 STATE-FIFO_FULL_STATE

- Signal **busy** alloted as logic 1 but the signal **write_enb_reg** is set as logic 0.
- Signal **full_state** is inserted to check for the state of FIFO whether it is full or not.

3.3.6 STATE-LOAD_AFTER_FULL

- After FIFO_FULL_STATE, the **laf_state** signal is introduced, and it is utilized for latching out the information.
- **Write_enb_reg** is accommodated despite the signal **busy**.
- It looks for the **parity_done** signal, and if it is logic 1, it indicates that the LOAD_PARITY state has encountered, then proceeds towards DECODE_ADDRESS.
- When **low_packet_valid** is sensed as logic 1, the state will be LOAD PARITY; otherwise, it will be LOAD DATA state.

3.3.7 STATE-WAIT_TILL_EMPTY

- Signal **busy** is set at logic 1 and signal **write_enb_reg** is set as logic 0.

3.3.8 STATE-CHECK_PARITY_ERROR

- At this point, the rst_int_reg signal is generated, which is then utilized to reset **low_packet_valid** signal.
- When the FIFO is not full, this stage switches to DECODE ADDRESS, and when the FIFO is full, it switches to FIFO FULL STATE.
- In this stage, the **busy** is inserted.

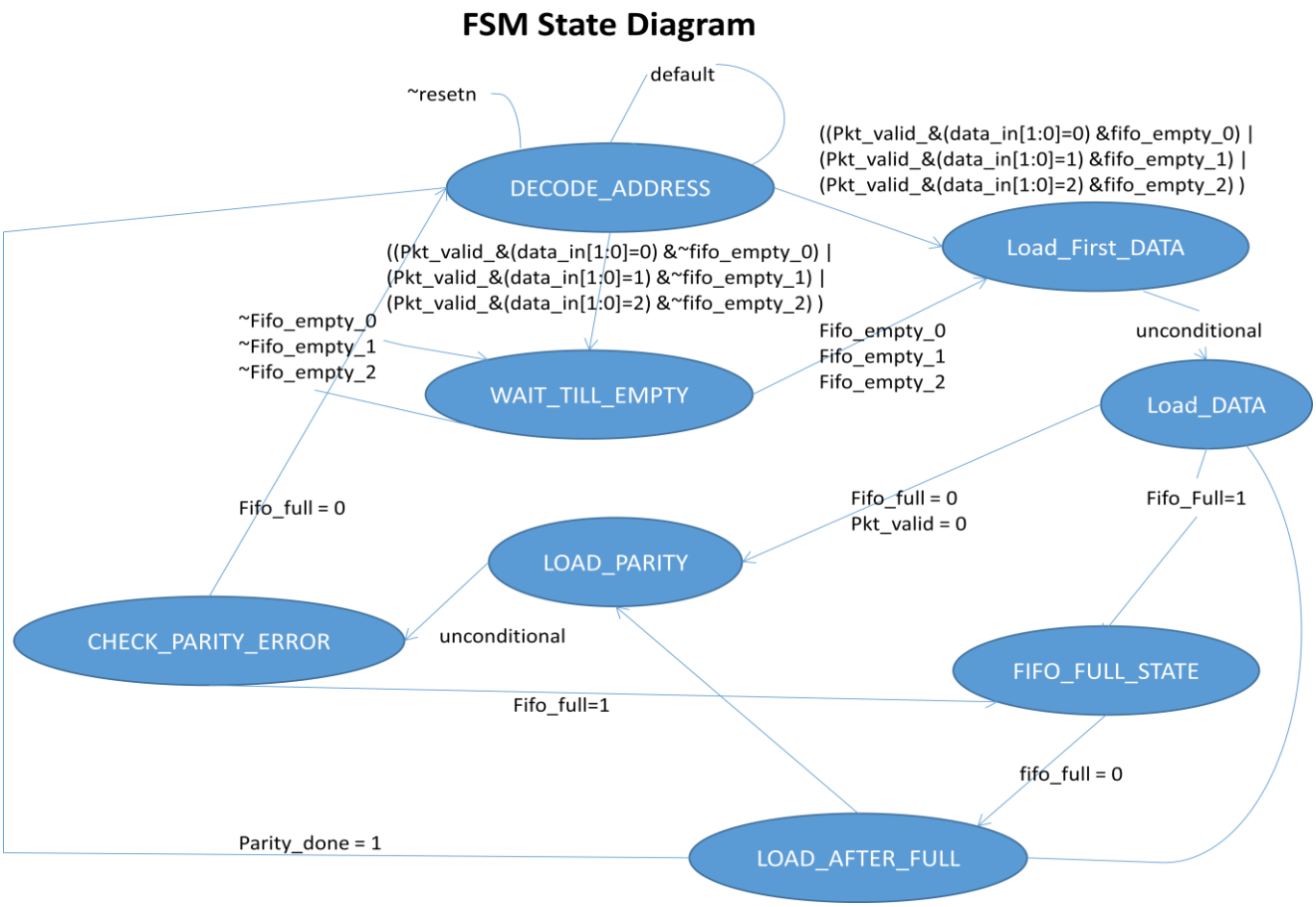


Figure3.4 – Diagram of FSM

3.4 Router: Register

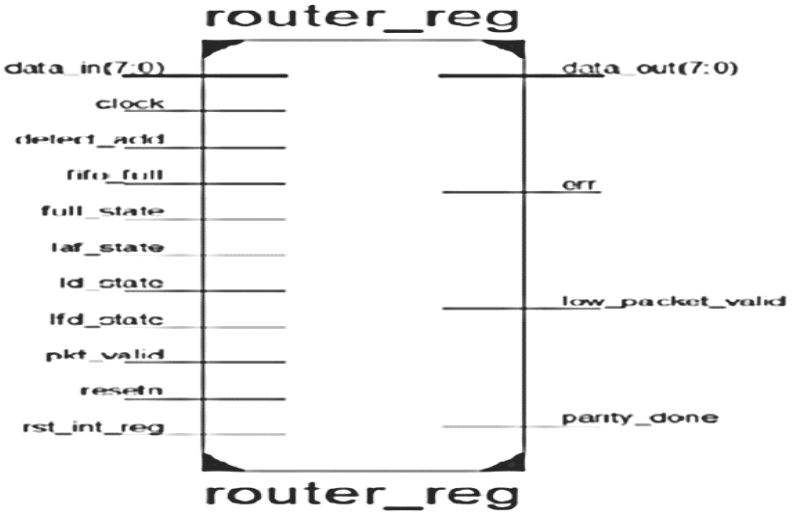


Figure3.5 – RTL Schematic of Router Register Block

3.4.1 Functionality:

This module implements 4 internal registers in order to hold a header byte, FIFO full state byte, internal parity and packet parity byte. All the registers in this module are latched on the rising edge of the **clock**.

- If **resetn** is low then the signals (**dout,err,parity_done and low_pkt_valid**) are low.
- The signal **parity_done** is high under the following conditions:
- When signal **ld_state** is high and signals **(fifo_full and pkt_valid)** are low.
- When signals **laf_state** and **low_pkt_valid** both are high and the previous value of **parity_done** is low.
- **rst_int_reg** signal is used to reset **low_pkt_valid** signal.
- **detect_add** signal is used to reset **parity_done** signal.
- **Signal low_pkt_valid** is high when **ld_state** is high and **pkt_valid** is low. **Low_packet_valid** shows that pkt_valid for current state has been deasserted.
- First data byte i.e., header is latched inside an internal register when **detect_add** and **pkt_valid** signals are high. This data is latched to the output **dout** when **lfd_state** goes high.
- Then signal **data_in** i.e. Payload is latched to **dout** if **ld_state** signal is high and **fifo_full** is low.
- Signal data_in is latched to an internal register when **ld_state** and **fifo_full** are high. This data is latched to output **dout** when laf_state goes high.
- **Full_state** is used to calculate internal parity.
- Another internal register is used to store internal parity for parity matching. Internal parity is calculated using the bit-wise xor operation between header byte, payload byte and previous parity values as shown below:

parity_reg=parity_reg_previous^header_byte ---- t1 clock cycle
parity_reg=parity_reg_previous^header_byte ---- t1 clock cycle
parity_reg=parity_reg_previous^header_byte ---- t1 clock cycle
parity_reg=parity_reg_previous^header_byte ---- t1 clock cycle **Last payload byte**

 The **err** is calculated only after packet parity is loaded and goes high if the packet parity doesn‟t match with the internal parity.

Chapter 4

Simulation Results

4.1 Router: Top Level

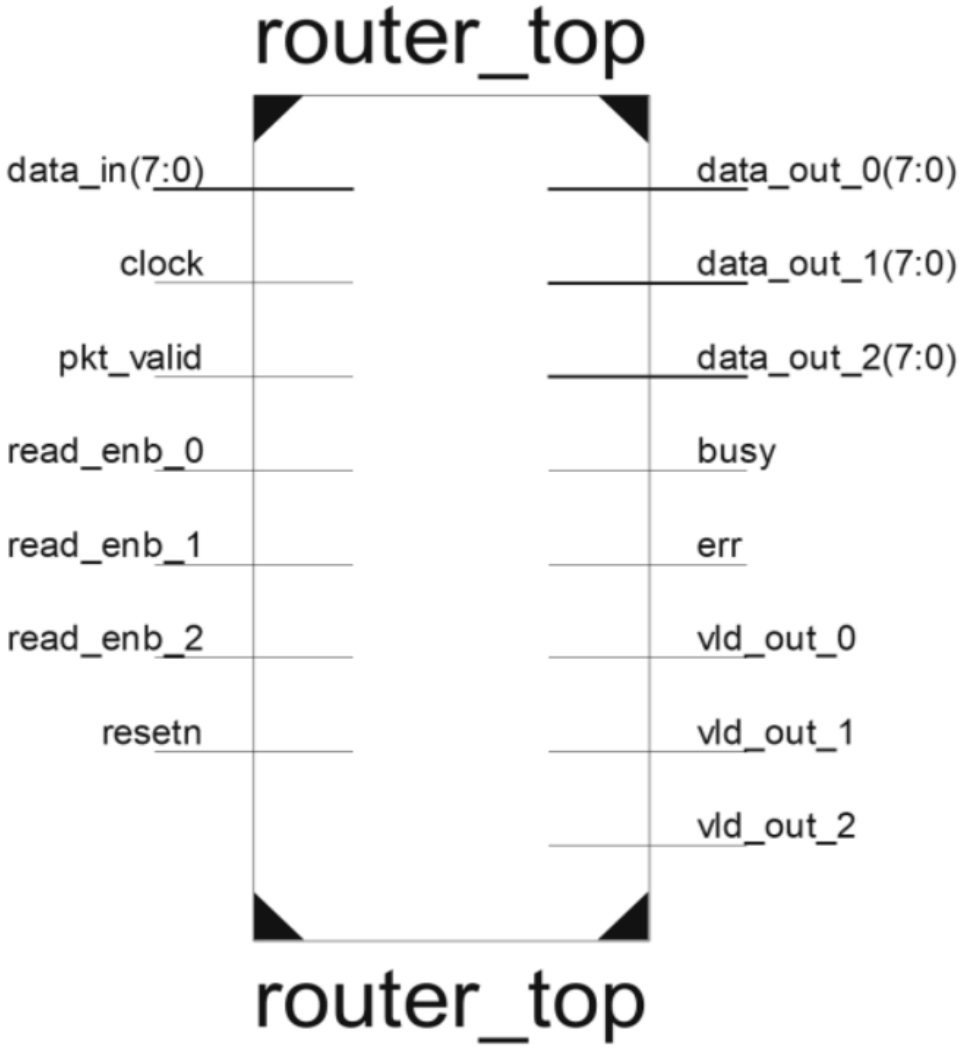


Figure4.1 –Block Diagram of Router

4.2 Overall Schematic

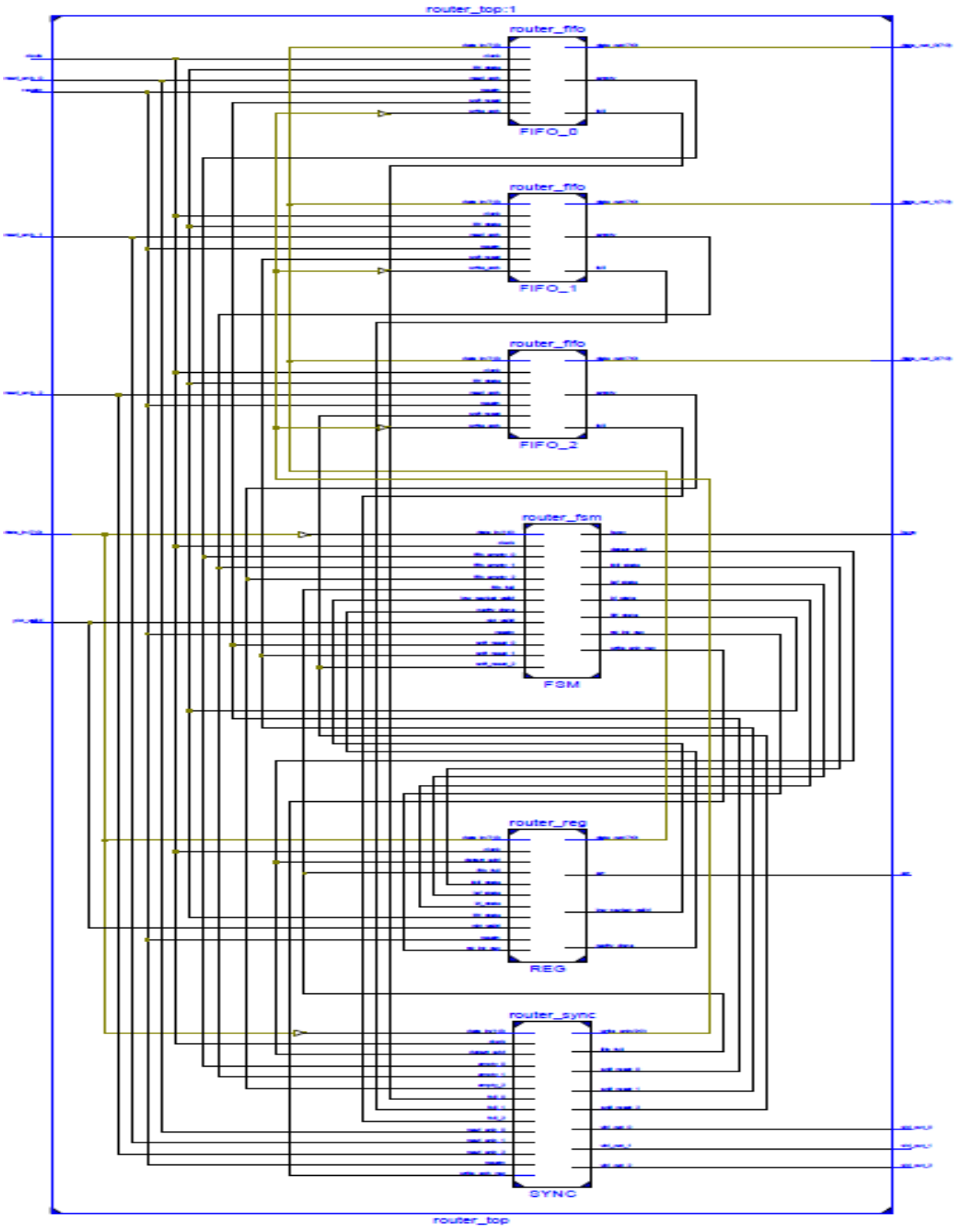


Figure4.2–Overall RTL Schematic of Router

4.3 Simulation Results

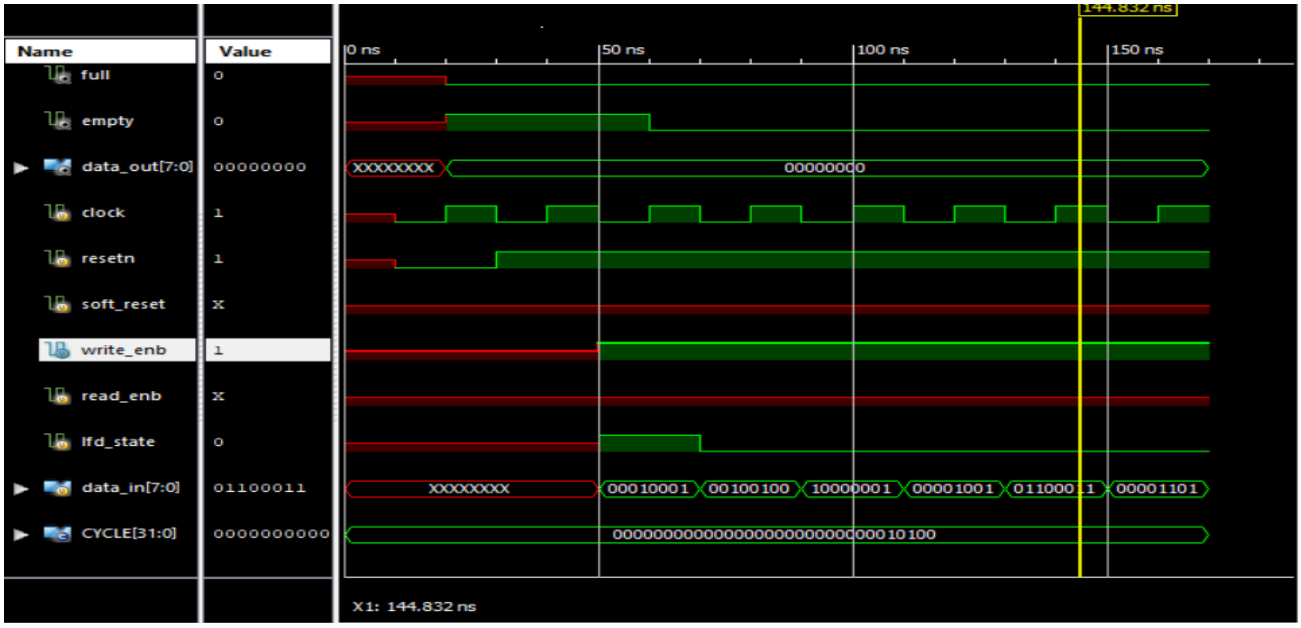


Figure4.3 – Output of FIFO

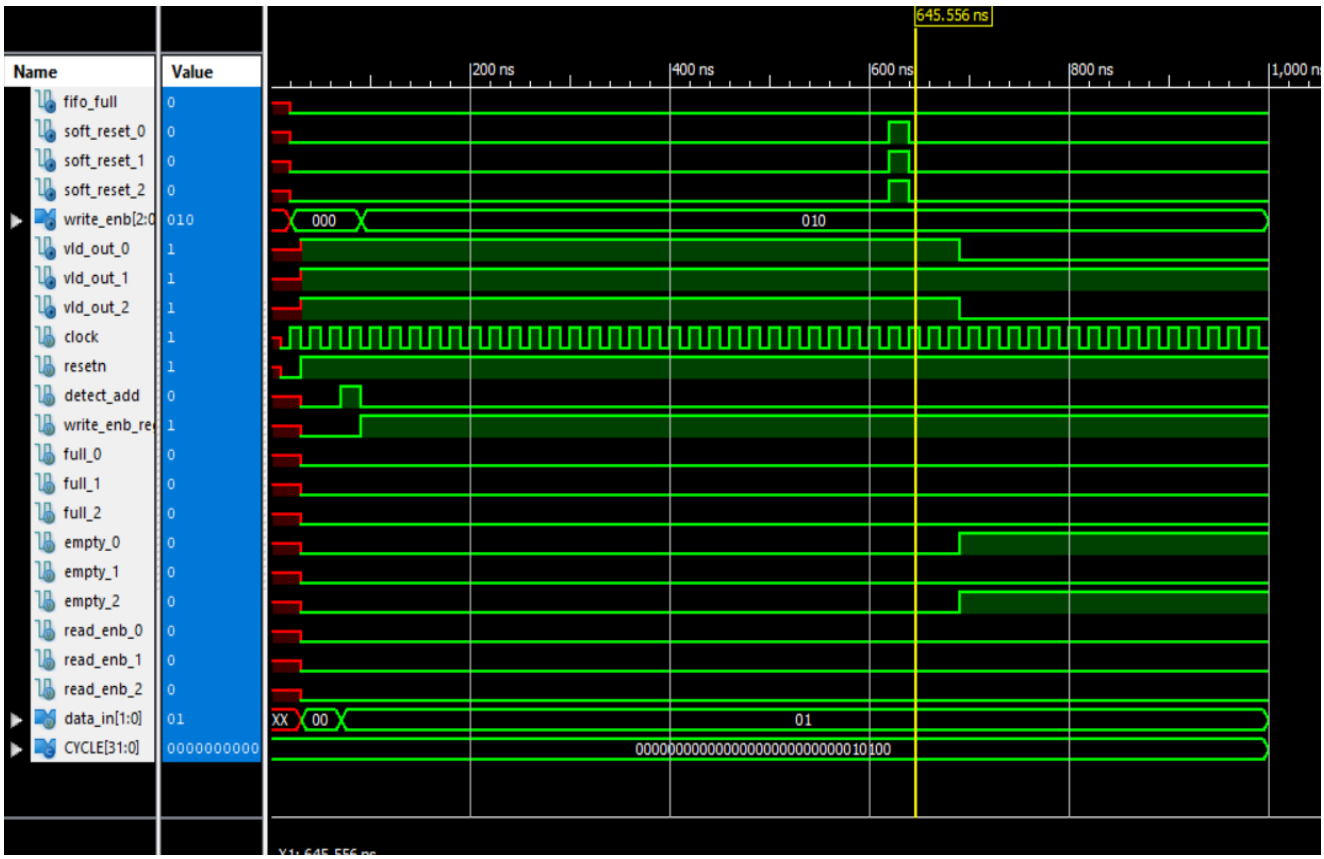


Figure4.4 – Output of Synchronizer

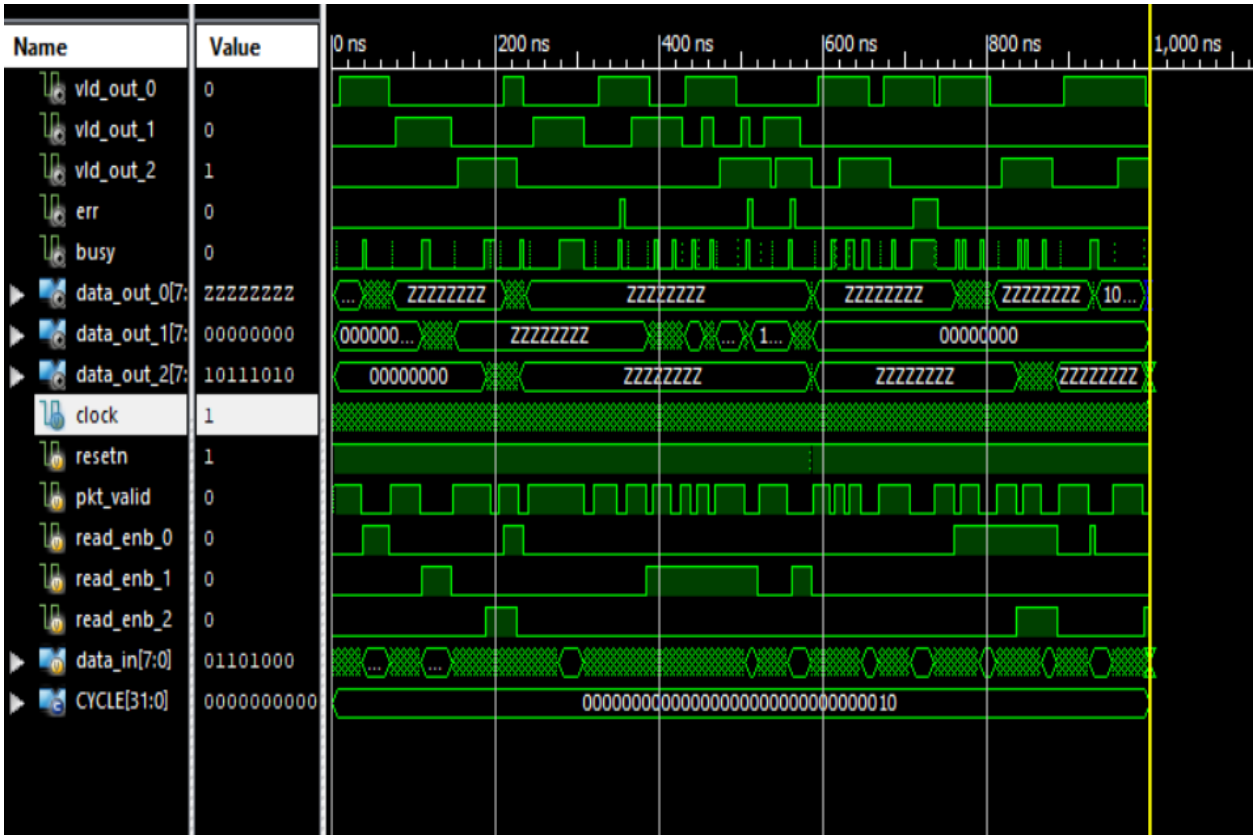


Figure4.5 – Output of Synchronizer

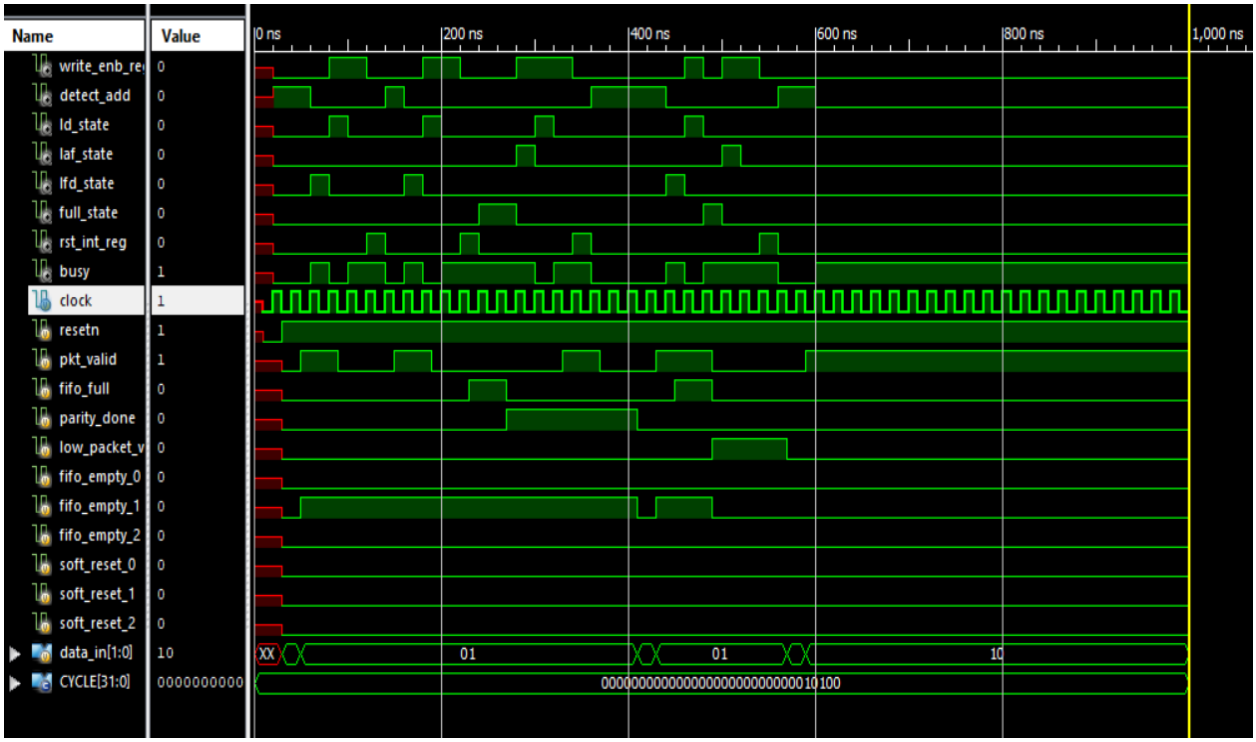


Figure4.6 – Output of FSM

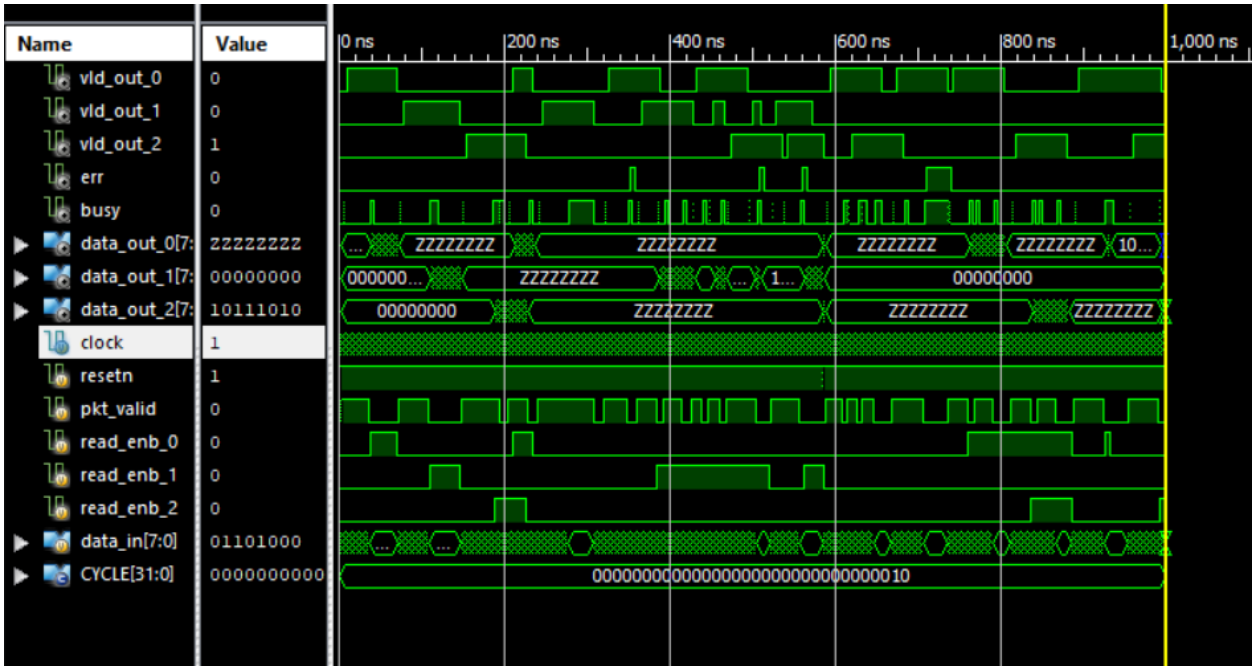


Figure4.7 – Output of FSM

4.4 Synthesis Reports:

```
=========================================================================
*                      Synthesis Options Summary                        *
=========================================================================
---- Source Parameters
Input File Name                    : "router_top.prj"
Ignore Synthesis Constraint File   : NO

---- Target Parameters
Output File Name                   : "router_top"
Output Format                      : NGC
Target Device                      : xa7a100t-2I-csg324

---- Source Options
Top Module Name                    : router_top
Automatic FSM Extraction           : YES
FSM Encoding Algorithm             : Auto
Safe Implementation                : No
FSM Style                          : LUT
RAM Extraction                     : Yes
RAM Style                          : Auto
ROM Extraction                     : Yes
Shift Register Extraction          : YES
ROM Style                          : Auto
Resource Sharing                   : YES
Asynchronous To Synchronous        : NO
Shift Register Minimum Size        : 2
Use DSP Block                      : Auto
Automatic Register Balancing       : No

---- Target Options
LUT Combining                      : Auto
Reduce Control Sets                : Auto
Add IO Buffers                     : YES
Global Maximum Fanout              : 100000
```

ry (Implemented) | router_sync.v | router_reg.v | router_fifo.v | router_top.v

Figure4.8 – Synthesis options summary

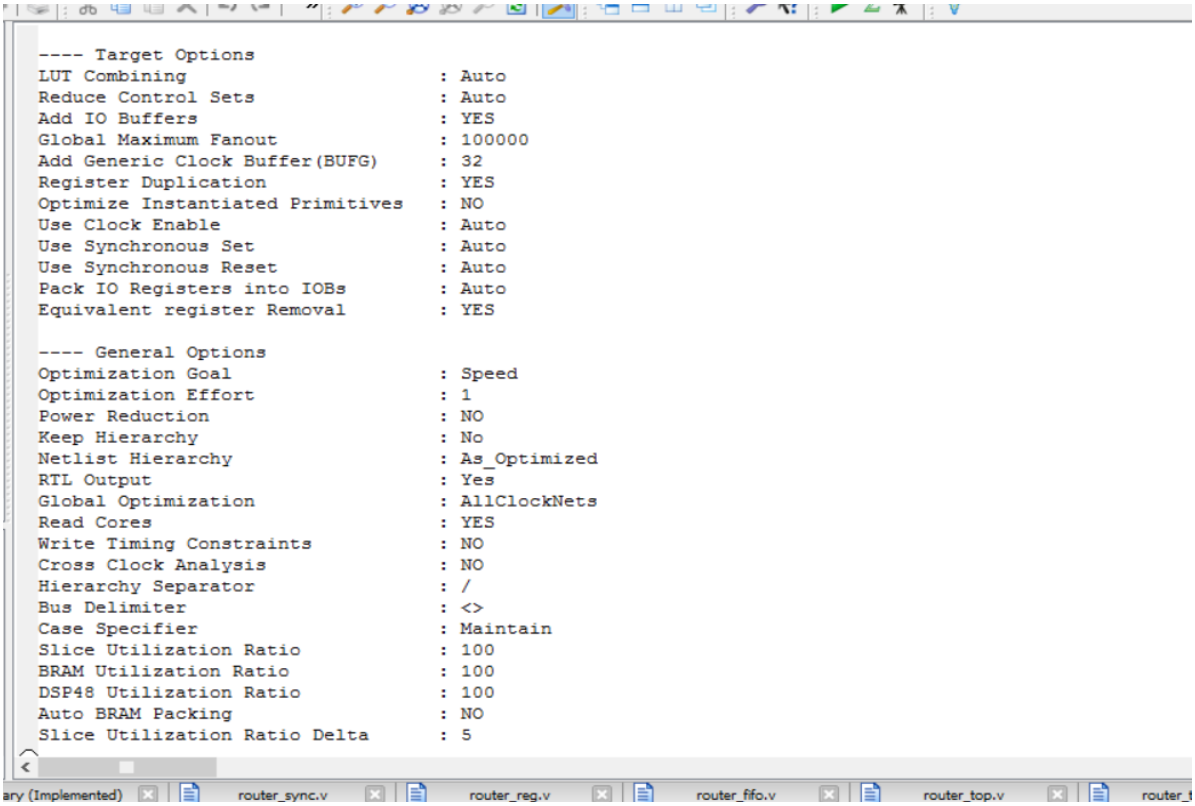


Figure 4.9 – Synthesis options summary

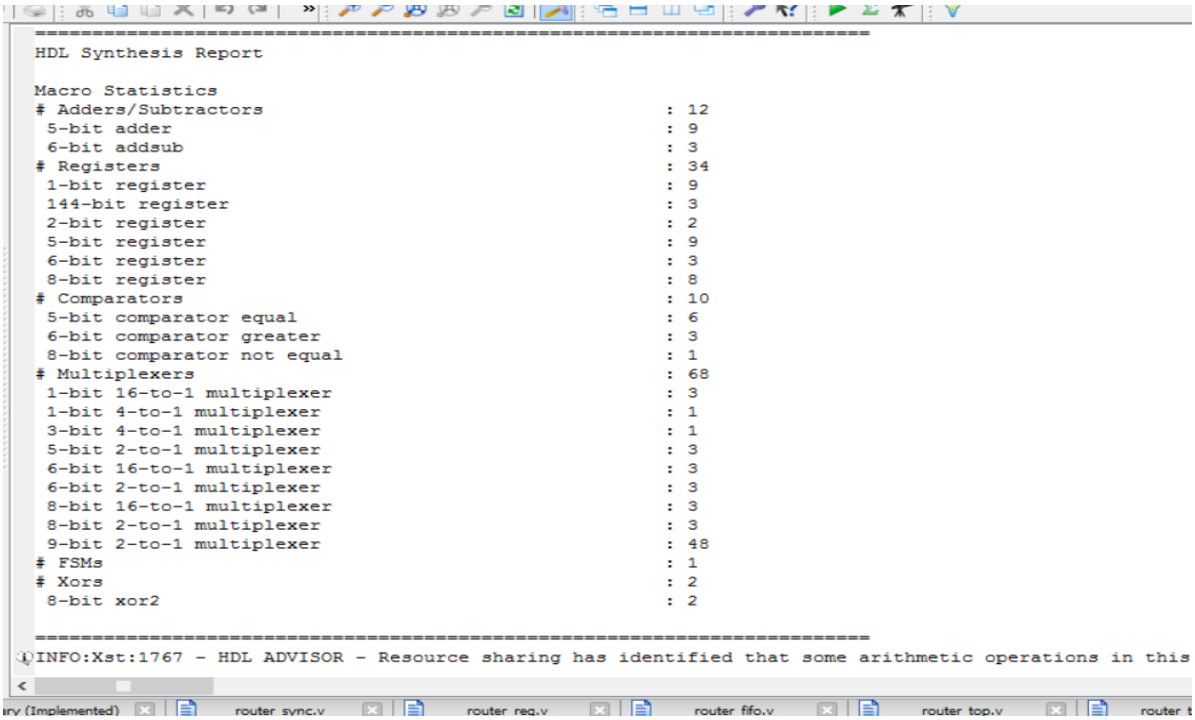
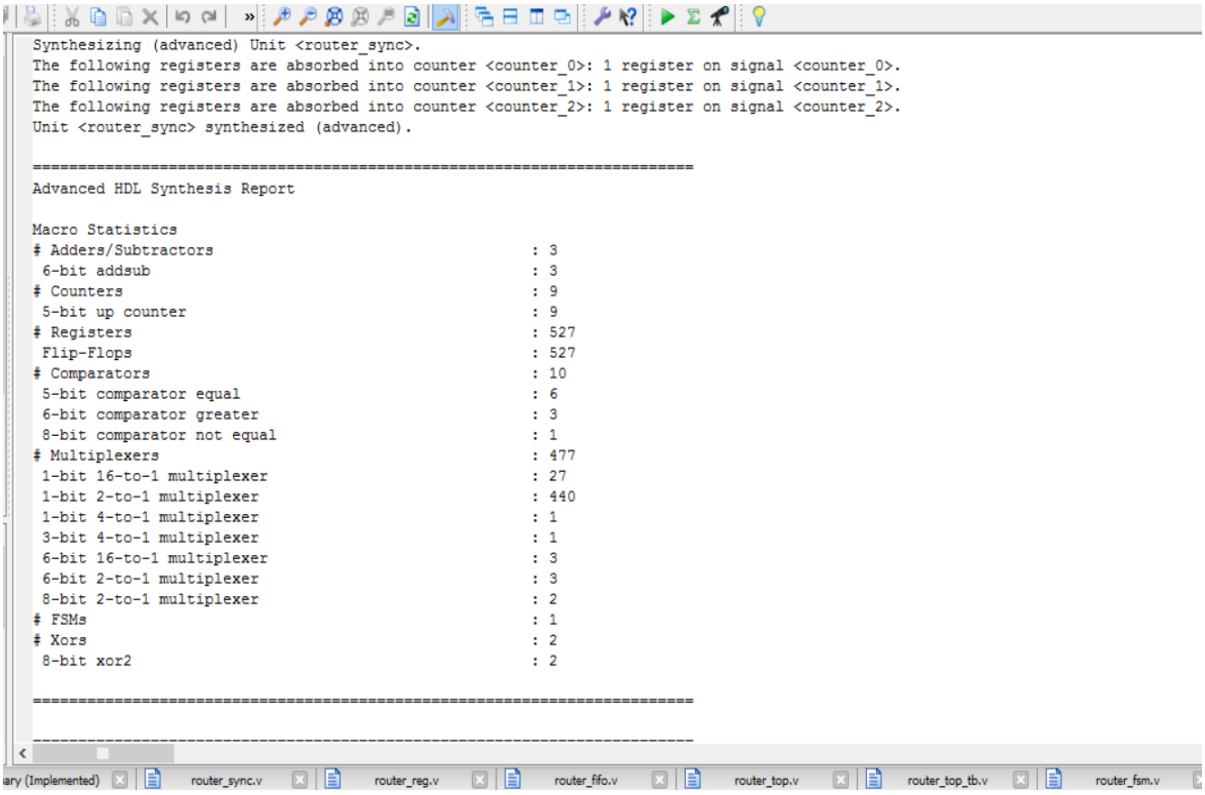
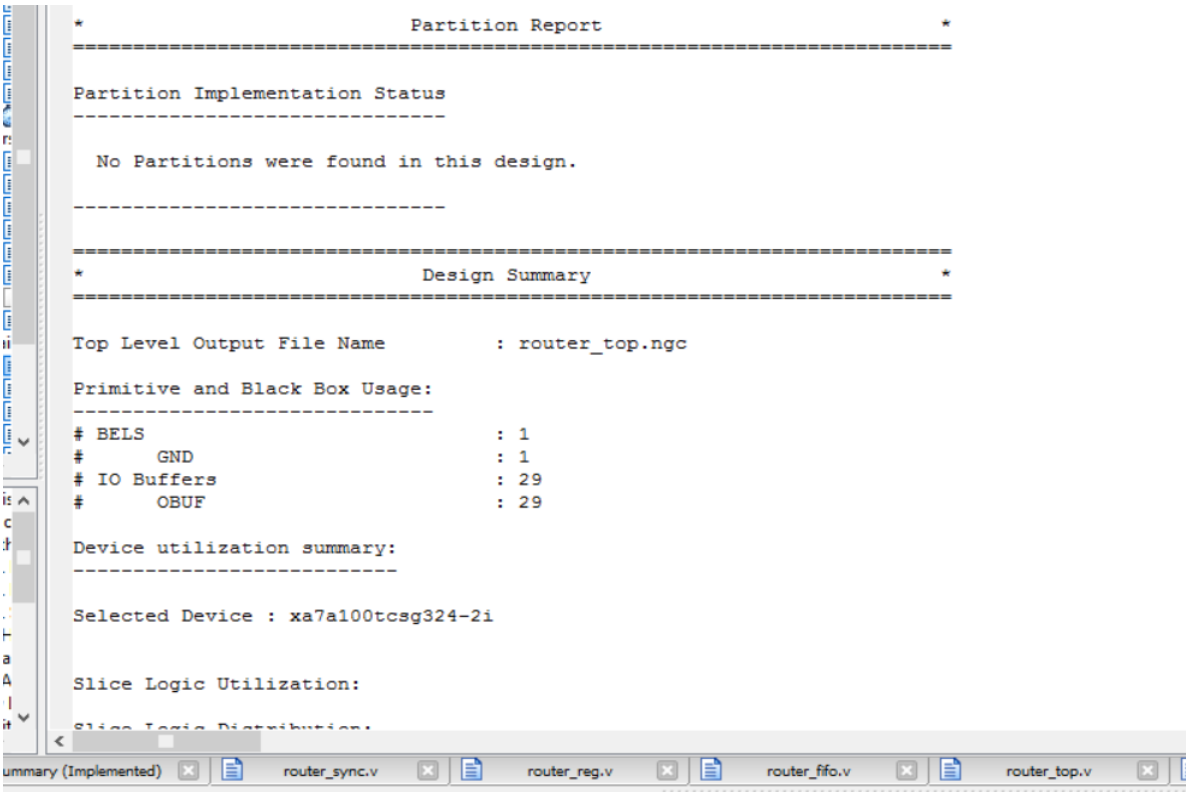


Figure 4.10 – HDL Synthesis Report



```
Synthesizing (advanced) Unit <router_sync>.
The following registers are absorbed into counter <counter_0>: 1 register on signal <counter_0>.
The following registers are absorbed into counter <counter_1>: 1 register on signal <counter_1>.
The following registers are absorbed into counter <counter_2>: 1 register on signal <counter_2>.
Unit <router_sync> synthesized (advanced).

=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Adders/Subtractors                                   : 3
 6-bit addsub                                         : 3
# Counters                                             : 9
 5-bit up counter                                     : 9
# Registers                                            : 527
 Flip-Flops                                           : 527
# Comparators                                          : 10
 5-bit comparator equal                               : 6
 6-bit comparator greater                             : 3
 8-bit comparator not equal                           : 1
# Multiplexers                                         : 477
 1-bit 16-to-1 multiplexer                            : 27
 1-bit 2-to-1 multiplexer                             : 440
 1-bit 4-to-1 multiplexer                             : 1
 3-bit 4-to-1 multiplexer                             : 1
 6-bit 16-to-1 multiplexer                            : 3
 6-bit 2-to-1 multiplexer                             : 3
 8-bit 2-to-1 multiplexer                             : 2
# FSMs                                                 : 1
# Xors                                                 : 2
 8-bit xor2                                           : 2

=========================================================================

-------------------------------------------------------------------------
```


Figure 4.11 – Advanced HDL Synthesis Report



```
*                            Partition Report                           *
=========================================================================

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

-------------------------------

=========================================================================
*                             Design Summary                            *
=========================================================================

Top Level Output File Name         : router_top.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 1
#      GND                         : 1
# IO Buffers                       : 29
#      OBUF                        : 29

Device utilization summary:
---------------------------

Selected Device : xa7a100tcsg324-2i


Slice Logic Utilization:
```


Figure 4.12 – Design Summary

4.5 Tools Used:

- Xilinx ISE (Integrated Synthesis Environment) 14.7 Design Suite.

Chapter 5

Conclusion

- The Router 1X3 is implemented and verified successfully. During the design part we came across many coding errors and successfully got rid of those errors.
- During verification we try to incorporate every scenario packets with payload length-
  - 8, 16, 14, 17 bytes.
  - Random byte
- Other scenarios like-
  - Destination network busy
  - Bad/corrupt packet
- These packet lengths were derived from the testbench in order to assess the design's robustness. All the scenarios were considered, inputs were driven through testbench and the Router was able to perform under all conditions successfully.

References

[1]. Palnitkar, Samir. Verilog HDL: a guide to digital design and synthesis. Pearson Education India, 2003.

[2]. Spear, Chris. System Verilog for verification: a guide to learning the testbench language features. Springer Science & Business Media, 2008.

[3]. Arnab Banerjee, Pascal T. Wolkotte, Robert D. Mullins, Simon W. Moore and Gerard J. M. Smit, "An Energy and Performance Exploration of Network-on-Chip Architectures", IEEE Transactions on Very Large Scale Integration Systems, Vol. 17, No. 3, pp. 319-329, 2009.

[4]. Cota,E., de Morais Amory, A., & Lubaszewski, M. S. (2012). NoC Basics. In Reliability, Availability and Serviceability of Networks-on-Chip (pp.11-24).Springer US.

[5]. Swapna, S., Swain, A. K., & Mahapatra, K. K. (2012, December). Design and analysis of f5 port router for network on chip. In Microelectronics and Electronics (PrimeA$ia), 2012 Asia Pacific Conference on Postgraduate Research in (pp. 51-55). IEEE.

[6]. Choudhari, E. M., & Dakhole, P. K. (2014, April). Design and verification of five port router for network on chip. In Communications and Signal Processing (ICCSP), 2014 International Conference on (pp. 607-611). IEEE [6] Gopal N., Router 1x3 RTL Design Journal of Engineering.

turnitin

Similarity Report ID: oid:27535:17723633

11% Overall Similarity

Top sources found in the following databases:

- 10% Internet database
- Crossref database
- 7% Submitted Works database
- 4% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**docplayer.net** Internet	5%
2	**irjmets.com** Internet	<1%
3	**electronicstopper.blogspot.com** Internet	<1%
4	**Texas A & M University, Kingville on 2018-11-14** Submitted works	<1%
5	**Aki Vamsi Krishna, E. Prabhu. "Chapter 2 Hardware Trojan Detection at...** Crossref	<1%
6	**doczz.net** Internet	<1%
7	**Higher Education Commission Pakistan on 2016-06-21** Submitted works	<1%
8	**docplayer.nl** Internet	<1%

Lokesh Gautam

turnitin **Similarity Report ID:** oid:27535:17723633

9	South Thames College on 2017-02-24 Submitted works	<1%
10	doi.nrct.go.th Internet	<1%
11	Visvesvaraya National Institute of Technology on 2020-12-22 Submitted works	<1%
12	Swansea Metropolitan University on 2012-03-07 Submitted works	<1%
13	Bridgepoint Education on 2021-11-08 Submitted works	<1%
14	K. R. Kashwan, G. Selvaraj. "Implementation and performance analyses... Crossref	<1%
15	Sharp, D.. "Component based product line development of avionics sof... Crossref	<1%
16	lib.dr.iastate.edu Internet	<1%
17	University of South Florida on 2016-02-13 Submitted works	<1%
18	Pathfinder Enterprises on 2012-11-05 Submitted works	<1%