IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 8, Issue 3 (Nov. - Dec. 2013), PP 57-62
www.iosrjournals.org

An Area Optimized Robust Router Design Implementation

* Chilka Prasada Rao[1], S.Suman [2]
[1] *PG Student (M. Tech), Dept. of ECE, Chirala Engineering College, Chirala, A.P, India.*
[2] *Assistant Professor, Dept. of ECE, Chirala Engineering College, Chirala, A.P, India.*

Abstract: *The router is a " Network Router" has a one input port from which the packet enters. It has three output ports where the packet is driven out. Packet contains 3 parts. They are Header, data and frame check sequence. Packet width is 8 bits and the length of the packet can be between 1 bytes to 63 bytes. Destination address(DA) of the packet is of 8 bits. The switch drives the packet to respective ports based on this destination address of the packets. Each output port has 8-bit unique port address. If the destination address of the packet matches the port address, then switch drives the packet to the output port, Length of the data is of 8 bits. In this paper the Xilinx ISE EDA Tool is used for synthesis and Modelsim is used for simulation.*
Keywords: *Network-on-Chip, Simulation Router, FIFO, FSM, Register blocks.*

I. Introduction

For most home users, they may want to set-up a LAN (local Area Network) or WLAN (wireless LAN) and connect all computers to the Internet without having to pay a full broadband subscription service to their ISP for each computer on the network. In many instances, an ISP will allow you to use a router and connect multiple computers to a single Internet connection and pay a nominal fee for each additional computer sharing the connection. This is when home users will want to look at smaller routers, often called broadband routers that enable two or more computers to share an Internet connection. Within a business or organization,you may need to connect multiple computers to the Internet, but also want to connect multiple private networks Not all routers are created equal since their job will differ slightly from network to network. Additionally, you may look at a piece of hardware and not even realize it is a router. What defines a router is not its shape, color, size or manufacturer, but its job function of routing data packets between computers. A cable modem which routes data between your PC and your ISP can be considered a router. In its most basic form, a router could simply be one of two computers running the Windows 98 (or higher) operating system connected together using ICS (Internet Connection Sharing). In this scenario, the computer that is connected to the Internet is acting as the router for the second computer to obtain its Internet connection. Going a step up from ICS, we have a category of hardware routers that are used to perform the same basic task as ICS, albeit with more features and functions. Often called broadband or Internet connection sharing routers, these routers allow you to share one Internet connection computers.

Broadband or ICS routers will look a bit different depending on the manufacturer or brand, but wired routers are generally a small box-shaped hardware device with ports on the front or back into which you plug each computer, along with a port to plug in your broadband modem. These connection ports allow the router to do its job of routing the data packets between each of the computers and the data going to and from the Internet. Depending on the type of modem and Internet connection you have, you could also choose a router with phone or fax machine ports. A wired Ethernet broadband router will typically have a built-in Ethernet switch to allow for expansion. These routers also support NAT (network address translation), which allows all of your computers to share a single IP address on the Internet. Internet connection sharing routers will also provide users with much needed features such as an SPI firewall or serve as a a DHCP Server.

This paper provides specifications for the Router is a packet based protocol. Router drives the incoming packet which comes from the input port to output ports based on the address contained in the packet. The router is a " Network Router" has a one input port from which the packet enters. It has three output ports where the packet is driven out. Packet contains 3 parts. They are Header, data and frame check sequence. Packet width is 8 bits and the length of the packet can be between 1 bytes to 63 bytes. Packet header contains three fields DAand length.

Destination address(DA) of the packet is of 8 bits. The switch drives the packet to respective ports based on this destination address of the packets. Each output port has 8- bit unique port address. If the destination address of the packet matches the port address, then switch drives the packet to the output port, Length of the data is of 8 bits and from 0 to 63.

Length is measured in terms of bytes. Data should be in terms of bytes and can take anything. Frame check sequence contains the security check of the packet. It is calculated over the header and data.

II. Designing The router

The communication on network on chip is carried out by means of router, so for implementing better NOC, the router should be efficiently design. This router supports three parallel connections at the same time. It uses store and forward type of flow control and FSM Controller deterministic routing which improves the performance of router. The switching mechanism used here is packet switching which is generally used on network on chip.

In packet switching the data the data transfers in the form of packets between cooperating routers and independent routing decision is taken. The store and forward flow mechanism is best because it does not reserve channels and thus does not lead to idle physical channels. The arbiter is of rotating priority scheme so that every channel once get chance to transfer its data. In this router both input and output buffering is used so that congestion can be avoided at both sides.

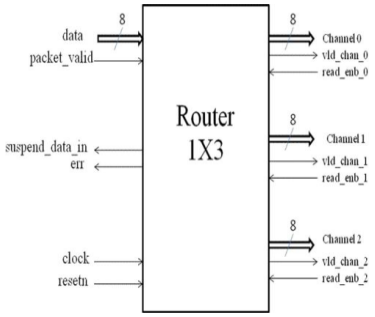

Figure 1 A Three Port Router

Packet Format:

Packet contains 3 parts. They are Header, payload and parity. Packet width is 8 bits and the length of the packet can be between 1 bytes to 63 bytes.

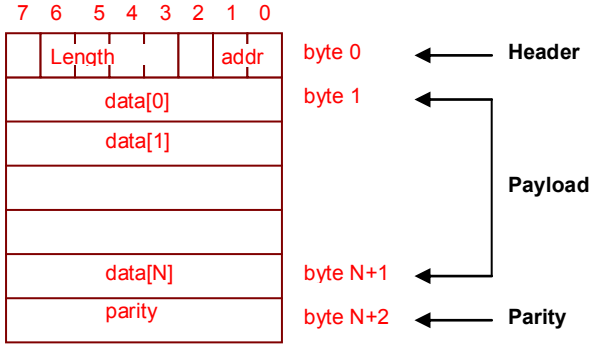

Figure 2 Data Packet Format

Packet - Header:

Packet header contains two fields DA and length.

DA: Destination address of the packet is of 2 bits. The router drives the packet to respective ports based on this destination address of the packets. Each output port has 2-bit unique port address. If the destination address of the packet matches the port address, then router drives the packet to the output port. The address "3" is invalid.

Length: Length of the data is of 6 bits and from 1 to 63. It specifies the number of data bytes. A packet can have a minimum data size of 1 byte and a maximum size of 63 bytes.

If Length = 1, it means data length is 1 bytes

If Length = 2, it means data length is 2 bytes

If Length = 63, it means data length is 63 bytes

Packet - Payload:

Data: Data should be in terms of bytes and can take anything.

Packet - Parity:

Parity: This field contains the security check of the packet. It should be a byte of even, bitwise parity, calculated over the header and data bytes of the packet.

III. Design Considerations

The clock signal is provided by the master to provide synchronization. The clock signal controls when data can change and when it is valid for reading. Since ROUTER is synchronous, it has a clock pulse along with

the data. RS-232 and other asynchronous protocols do not use a clock pulse, but the data must be timed very accurately.
Since ROUTER has a clock signal, the clock can vary without disrupting the data. The data rate will simply change along with the changes in the clock rate. As compared with its counterpart I2C, ROUTER is more suited for data stream applications. Communication between IP's,
The characteristics of the DUT input protocol are as follows:
1 All input signals are active high and are synchronized to the falling edge of the *clock*. This is because the DUV router is sensitive to the rising edge of *clock*. Therefore, driving input signals on the falling edge ensures adequate setup and hold time, but the signals can also be driven on the rising edge of the clock.
2 The *packet_valid* signal has to be asserted on the same *clock* as when the first byte of a packet (the header byte), is driven onto the *data* bus.
3 Since the header byte contains the address, this tells the router to which output channel the packet should be routed (*data_out_0*, *data_out_1*, or *data_out_2*).
4 Each subsequent byte of data should be driven on the *data* bus with each new rising/falling *clock*.
5 After the last payload byte has been driven, on the next rising/falling *clock*, the *packet_valid* signal must be deasserted , and the packet parity byte should be driven. This signals packet completion.
6 The input *data* bus value cannot change while the *suspend_data* signal is active (indicating a FIFO overflow). The packet driver should not send any more bytes and should hold the value on the *data* bus. The width of *suspend_data* signal assertion should not exceed 100 cycles.
The *err* signal asserts when a packet with bad parity is detected in the router, within 1 to 10 cycles of packet completion.

Router Output Protocol:
The characteristics of the output protocol are as follows:
1 All output signals are active high and can be synchronized to the rising/falling edge of the *clock*. Thus, the packet receiver will drive sample data at the rising/falling edge of the *clock*. The router will drive and sample data at the rising edge of *clock*.
2 Each output port *data_out_X* (*data_out_0*, *data_out_1*, *data_out_2*) is internally buffered by a FIFO of 1 byte width and 16 location depth.
3 The router asserts the *vld_out_X (vlld_out_0*, *vld_out_1* or *vld_out_2*) signal when valid data appears on the *vld_out_X* (*data_out_0*, *data_out_1* or *data_out_2*) output bus. This is a signal to the packet receiver that valid data is available on a particular router.
4 The packet receiver will then wait until it has enough space to hold the bytes of the packet and then respond with the assertion of the *read_enb_X* (*read_enb_0*, *read_enb1* or *read_enb_2*) signal that is an input to the router.
5 The *read_enb_X* (*read_enb0*, *read_enb_1* or *read_enb_2*) input signal can be asserted on the rising/falling *clock* edge in which data are read from the *data_out_X* (*data_out_0*, *data_out_1* or *data_out_2*) bus.
6 As long as the *read_enb_X* (*read_enb_0*, *read_enb_1* or *read_enb_2*) signal remains active, the *data_out_X* (*data_out_0*, *data_out_1* or *data_out_2*) bus drives a valid packet byte on each rising *clock* edge.
7 The packet receiver cannot request the router to suspend data transmission in the middle of the packet. Therefore, the packet receiver must assert the *read_enb_X* (*read_enb_0*, *read_enb_1* or *read_enb_2*) signal only after it ensures that there is adequate space to hold the entire packet.
8 The *read_enb_X* (*read_enb_0*, *read_enb_1* or *read_enb_2*) must be asserted within 30 *clock* cycles of the *vld_out_X* (*vld_out_0*, *vld_out_1* or *vld_out_2*) being asserted. Otherwise, there is too much congestion in the packet receiver.
9 The DUV *data_out_X* (*data_out_0*, *data_out_1* or *data_out_2*) bus must not be tri-stated (high Z) when the DUV signal *vld_out_X* (*vld_out_0*, *vld_out_1*or *vld_out_2*) is asserted (high) and the input signal *read_enb_X* (*read_enb_0*, *read_enb_1* or *read_enb_2*) is also asserted high.

ROUTER MODULES

The fsm_router block provides the control signals to the fifo, and router_reg module.The router_reg module contains the status, data and parity registers for the router_1x3.These registers are latched to new status or input data through the control signals provided by the fsm_router.There are 3 fifo for each output port, which stores the data coming from input port based on the control signals provided by fsm_router module. The ff_sync module provides synchronization between fsm_router module and 3 fifo s , So that single input port can faithfully communicate with 3 output ports.

FIFO BLOCK

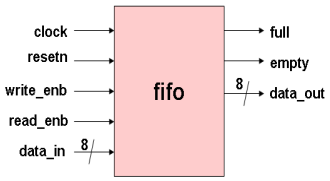

Figure 3 FIFO Block

There are 3 fifos used in the router design. Each fifo is of 8 bit width and 16 bit depth.The fifo works on system clock. It has synchronous input signal reset.If **resetn** is low then full =0, empty = 1 and data_out = 0

Write operation:

The data from input **data_in** is sampled at rising edge of the clock when input **write_enb** is high and fifo is not full.

Read Operation:

The data is read from output **data_out** at rising edge of the clock, when **read_enb** is high and fifo is not empty.Read and Write operation can be done simultaneously.**Full** – it indicates that all the locations inside fifo has been written.**Empty** – it indicates that all the locations of fifo are empty.

FF_SYNC BLOCK

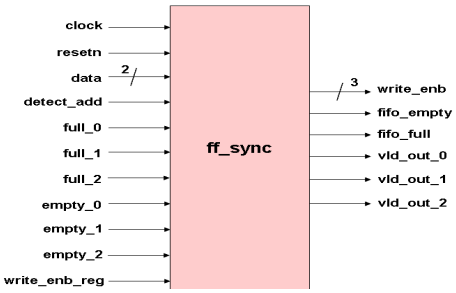

Figure 4 FF Sync Block

This module provides synchronization between fsm and fifo modules. It provides faithful communication between single input port and three output ports.It will detect the address of channel and will latch it till packet_valid is asserted, address and write_enb_sel will be used for latching the incoming data into the fifo of that particular channel.

A **fifo_full** output signal is generated, when the present fifo is full, and **fifo_empty** output signal is generated by the present fifo when it is empty.

If **data** = 00 then **fifo_empty** = **empty_0** and **fifo_full** = **full_0**
If **data** = 01 then **fifo_empty** = **empty_1** and **fifo_full** = **full_1**
If **data** = 10 then **fifo_empty** = **empty_2** and **fifo_full** = **full_2**
Else **fifo_empty** = 0 and fifo_full = 1.

The output **vld_out** signal is generated when **empty** of present fifo goes low, that means present fifo is ready to read.

vld_out_0 = ~empty_0
vld_out_1 = ~empty_1
vld_out_2 = ~empty_2

The **write_enb_reg** signal which comes from the fsm is used to generate **write_enb** signal for the present fifo which is selected by present address.

ROUTER_REG BLOCK

This module contains status, data and parity registers required by router. All the registers in this module are latched on rising edge of the ***clock***. Data registers latches the data from data input based on state and status control signals, and this latched data is sent to the fifo for storage. Apart from it, data is also latched into the parity registers for parity calculation and it is compared with the parity byte of the packet. An ***error*** signal is generated if packet parity is not equal to the calculated parity. If **resetn** is low then output (**dout, err, parity_done and low_packet_valid**) are low.

The output **parity_done** is high

- when the input **ld_state** is high and (**fifo-full and packet_valid**) is low

- or when the input **laf_state** and output **low_packet_valid** both are high and the previous value of **parity_done** is low. It is reseted to low value by **reset_int_reg** signal.

The output low_packet_valid is high

- when the input ld_state is high and packet_valid is low.
- It is reseted to low by reset_int_reg signal.

First data byte i.e., header is latched inside the internal register first_byte when ***detect_add*** and ***packet_valid*** signals are high, So that it can be latched to output ***dout*** when ***lfd_state*** signal goes high.

Then the input data i.e., payload is latched to output ***dout*** if ***ld_state*** signal is high and ***fifo_full*** is low.

Then the input data i.e., parity is latched to output dout if ***lp_state*** signal is high and ***fifo_full*** is low.

The input data is latched to internal register full_state_byte when ***ld_state*** and ***fifo_full*** are high; this full_state_byte data is latched inside the output ***dout*** when ***laf_state*** goes high.

Internal parity register stores the parity calculated for packet data, when packet is transmitted fully, the internal calculated parity is compared with parity byte of the packet. An ***error*** signal is generated if packet parity is not equal to the calculated parity.

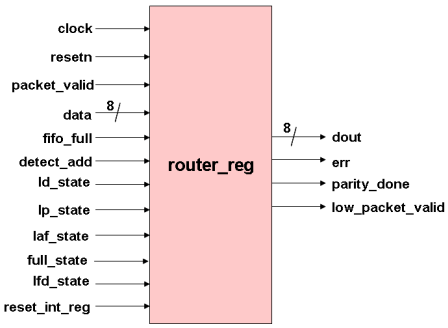

Figure 5Router Register Block

FSM BLOCK

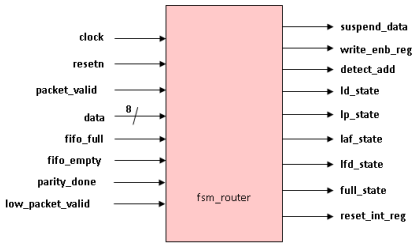

Figure 6 FSM Router Block

The 'fsm_router' module is the controller circuit for the router. This module generates all the control signals when new packet is sent to router. These control signals are used by other modules to send data at output, writing data into the FIFO.

IV. Results and Conclusion

In this paper, a reduced area (NO OF LUT'S)of a three port router is presented .The proposed router structure functionality is implemented in Verilog HDL and proven that this architecture consumes less resources in terms of no of LUT'S ,slices and no of IO Buffers . In this paper the Xilinx ISE EDA Tool is used for synthesis and Modelsim is used for simulation. The data which can be sent through the router is reached the destination with 9.375ns latency. In future there is a chance to estimate the power consumption also. The simulation result of the design is shown in fig. 7 and The RTL schematic is shown in Fig 8. The Device utilization table is shown under Table-1.

Table 1 Device Utilization Table

Logic Utilization	Proposed Utilization	Existed Utilization	Available
Number of Slices	134	1287	4656
Number of Slice Flip Flops	126	1203	9312
Number of 4 input LUTs	253	724	9312
Number of bonded IOBs	43	148	232
Number of GCLKs	3	7	24

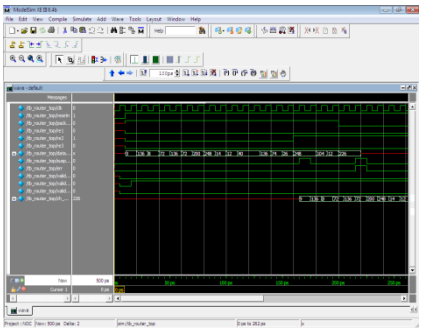

Figure 7 Simulation Result of the Designed Router

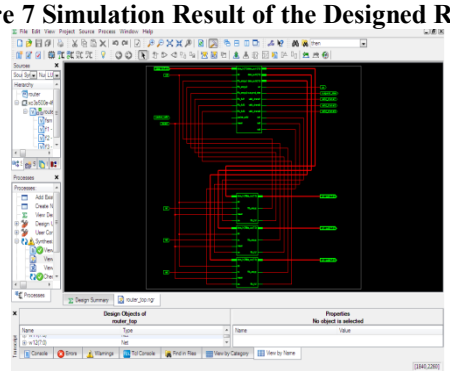

Figure 8 RTL Schematic of the Design

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality and presentation of this paper.

References:

[1]. P. Wolkotte, P. Holzenspies, and G. Smit, "Fast, Accurate and Detailed NoC Simulations", NOCS, 2007.
[2]. "D. Chiou, "MEMOCODE 2011 Hardware/Software CoDesign Contest",https://ramp.ece.utexas.edu/redmine/ attachments/ DesignContest.pdf
[3]. Xilinx, "ML605 Hardware User Guide", http://www.xilinx.com/support/documentation /boards and kits/ug534.pdf.
[4]. Xilinx, "LogiCORE IP Processor Local Bus (PLB) v4.6", http://www.xilinx.com/ support/documentation/ip documentation/ plb v46.pdf.
[5]. M. Pellauer, M. Adler, M. Kinsy, A. Parashar, and J. Emer,"HAsim: FPGA-Based High-Detail Multicore Simulation Using Time-Division Multiplexing", HPCA, 2011.
[6]. Bluespec Inc, http://www.bluespec.com

Authors Profile:

CHILKA PRASADA RAO is Pursuing his M. Tech from Chirala Engineering College, Chirala in the department of Electronics & Communications Engineering (ECE) with specialization in VLSI & Embedded Systems

S.Suman is working as an Assistant Professor in the department of Electronics & Communication Engineering in Chirala Engineering College,Chirala. He has 5 years of teaching and two years industrial experience.