ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of Reconfigurable Router for NOC Applications Using Buffer Resizing Techniques

Nandini Sultanpure
M.Tech (VLSI Design and Embedded System),
Dept of Electronics and Communication Engineering,
Lingaraj Appa Engineering College, Bidar, Karnataka,
Visvesvaraya Technological University.

Prashant Bachanna
M.Tech (VLSI Design and Embedded System),
Dept of Electronics and Communication Engineering,
Lingaraj Appa Engineering College, Bidar, Karnataka,
Visvesvaraya Technological University.

Abstract:

An FPGA based, 3x3 router using buffer resizing technique for 1D and 2D network on chip architecture has been designed and implemented. These routers are reconfigurable and they are capable of transmitting both 1D and 2D packets throughout the network. The latency in transmitting packets is decreases with increasing the depth of the buffers; this consumes more area in the routers. The reconfigurable router designed here uses "Buffer Resizing Technique" which reduces the depth of the static buffers present in the router. This reduces the area of the router without increasing the latency in the network.The proposed design is implemented in SPARTAN3 FPGA by using Xilinx ISE 13.4 and simulated in modelsim 6.3f.

Keywords:
Network on chip, Buffer resizing technique, FSM, FPGA, LUT.

I.INTRODUCTION:

Moore's Law continues to reduce the device feature size and increase system performance. Hence, we observe a trend of increasingly complicated architectures, enhanced clock rates and on chip logic density. One emerging architecture is Network-On-Chip (NoC). This new design is quickly replacing older design strategies from the bus architectures of micro-electronic chips. NoCs have been instrumental in achieving high bandwidth; low latency and scalable multi-core inter communication architectures in order to keep up with Moore's law.But NoCs still face some limitations. Firstly, in NoC architectures, numbers of cores share the workload of a task. Data transmission from one core to different core is very frequent and therefore, much emphasis needs to be done on area. Power utilization limits how many cores can be placed on single chip and can be utilized efficiently at the same time.

Thus, reducing energy utilization per logic operation is becoming very much important to keep power dissipation within the limit. Secondly, since all the cores are connected through an interconnect fabric, be it bus, ring or mesh. The interconnect topologies play a major role in the performance of the network. Therefore, it is clear that router energy utilization and interconnect fabric topologies are the two leading factors that limit the performance of NoCs. Several researches have been made to improve the NoC performance through communication and power. For instance, suggests that the performance can only be achieved if the NoCs moves away from a homogeneous fabric and become more to a heterogeneous hierarchical network. It also effects data locality to reduce interconnect usage and power utilization. The reason behind this concept is that if the data has to travel over several hops, then there will be more amount of power utilization by the routers and interconnects.

Thus, this technique does not always apply to existing approaches. For instance, if there is an added hierarchy of interconnect which has a direct access to processing elements within close path, then there will be considerable reduction of power utilization if data has to move to one of the intra cluster processing elements.The use of Field Programmable Gate Array (FPGA) as enabling technology for the NoC is becoming very much popular due to the flexibility provided by the FPGAs in terms of re-configurability of the design, underlying the networks, associated bandwidths and last but not least in design validation. With the evolution of the semiconductor processing technology it has been possible to visualize FPGA devices comprising the millions of LUTs and will have number of soft cores and other macro blocks such as memory, co-processor's etc. Once again the use of NoCs to configure a heterogeneous design within such FPGAs will reduce design and validation cycles and enable more complex architecture's implementations on this platform.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The major contributions of this paper are: a) design of a cross platform packet switch NoC router, b) a novel pipeline architecture for single cycle latency of data traversal per hop resulting in low latency and power dissipation, c) ability to use router in a globally asynchronous, locally synchronous (GALS) NoC, d) analysis of latency, power dissipation and FPGA utilization by subjecting several network topologies to various traffic test conditions post design synthesis and emulation. Packet switch routers traditionally provide low latency and the flexibility in terms of packet configuration and data transfer size. Using such an interface enables the construction of both 1-D and 2-D NoC architectures with high scalability.The purpose of switch allocation for a set of input's/output's port is made dynamically by the round robin priority logic located at each allocator of the arbiter block and three state finite state machine (FSM) of each output port. The design is well-known in the sense that under zero load latency, the incoming data is located on the output buffer in just 1 clock cycle by parallelizing operations of look ahead logic for the downstream router's output channel and the arbitration for the present router. Hence, the inter hop delay and dynamic power dissipation are substantially reduced and presents a significant advantage over and other synchronous NoC routers.

II.BACKGROUND

A.FPGA BASED ROUTER DESIGN

An FPGA based, single cycle, low latency router design that can be reconfigured to 1-D and 2-D network on chip architectures is proposed. The design is highly scalable and exploits the features provided by any standard FPGA platform and can be easily ported to an ASIC or any other FPGA platform. Due to the highly interconnect-centric nature of NoCs, the built-in resources of an FPGA in terms of routing channels and on chip logic are ideal and providea well-utilized platform for the router design.The packet length for this design is extremely flexible and each packet transferred from one router to another is divided into a header flit, a series of body flits concluded by a tail flit as shown in figure 1. The header flit contains all the information a router needs to configure its crossbar in order to send the flit to the desired downstream router and set up the channel for following body flits. When a tail flit passes through a router, it indicates the end of the packet and de-allocates all resources for the current router.

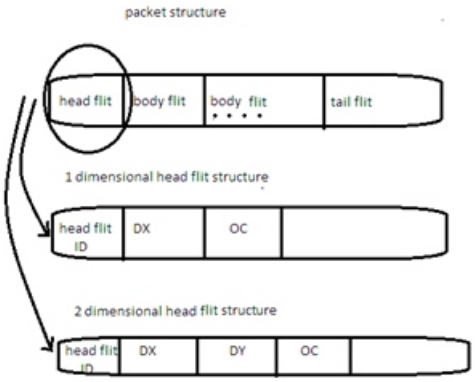

Fig. 1. Router packet structure

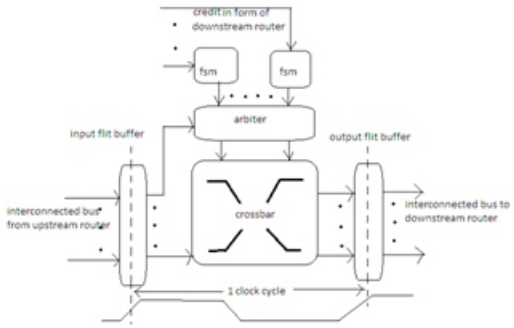

Fig. 2. Architectural block diagram of router

Figure 2 shows the block diagram of architecture of the router. The block diagram includes:
A.Input /Output Ports
B.Arbiter
C.Crossbar Switch

A.Input & output ports:

The input port has look-ahead logic to determine the forwarding path of data for the downstream router. Upon receiving a header flit, the logic quickly checks if the destination address matches the current router's address, if so then the data is sent to the core port and it never requests an output port from the arbiter. Otherwise, dimensional order routing is used to compute the forwarding path. This logic computes the new output channel (oc) value and replaces the current oc value in the header flit for the downstream router. The incoming header flit has the 2 bit encoded oc information which is the enumerated specific output channel to be used by the current router in flit transfer. The output ports follows a three-state FSM having the idle, active and wait credit states.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The output port is available to be associated to an input port only during an idle state. Once a flit is received from the input port, the output port moves into the active state and is ready to transfer data to a downstream router. The transmission however depends upon the status of the credit buffer. The credit buffer keeps account of available buffer locations at the downstream routers. This method is also known as the credit based flow control. The credit buffers counter decrements when a flit leaves the output port and the counter increments when it receives a 'credit in' signal from the downstream router. The credit in signal is generated only when a flit leaves the downstream router. This flow control mechanism alleviates congestion issues associated with the NoC system. Once the tail flit leaves the output port, the state machine goes back to the idle state as a downstream buffer indicates an open space for any future incoming flit.

B.Arbiter:

This block works in conjunction with a round robin priority encoder and the credit based flow control fsm. Due to this design choice, a very efficient way of congestion control and avoiding resource starvation was deployed. The arbiter has four allocators, one for each input-output port pair. The arbiter may receive resource allocation request from all four input ports simultaneously, and hence the priority logic is essentially processing multiple requests and once an output channel is matched to an input port, the selection logic holds the allocator active till a tail flit is processed through the crossbar.

C.Crossbar switch:

The switch is implemented using four 4x1 multiplexers (mux). The output of each mux is registered and goes straight to its corresponding output port. The crossbar is controlled by one of the select signals from the arbiter block.

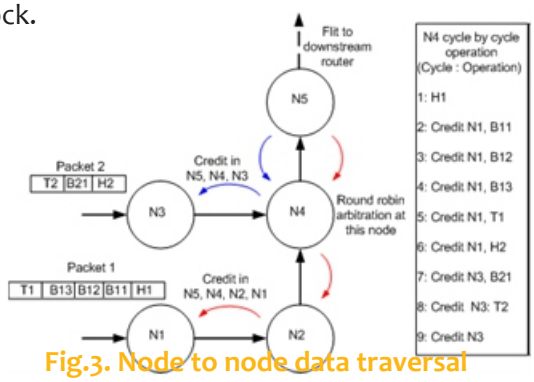

Fig.3. Node to node data traversal

The ability to reduce the zero load latency per hop to just 1 clock cycle is a significant improvement over and which take a least 2 cycles and other contemporary NoC routers which require 7-8 cycles. The design of the router primarily comprises of combinational logic implemented via FPGA primitive LUTs. The biggest advantage of such a design practice is the minimal impact on the critical path time budget. The LUTs contribute to very little logic delay and improve the FPGA utilization percentage and power dissipation as they are implemented with maximum efficiency by the synthesis tools. Due to these combined features, an 8x8 mesh was architected as an attempt to fully utilize the target FPGA and still leave room for scaling it up further.

B.Specification:

Software Requirement Specification:

- Operating System: Windows XP with SP2
- Synthesis Tool: Xilinx 13.4 ISE
- Simulation Tool: Modelsim6.3f.

Hardware Requirement specification:

- Minimum Intel Pentium IV Processor
- Primary memory: 2 GB RAM,
- Spartan III FPGA
- Xilinx Spartan III-EFPGA development board
- JTAG cable, Power supply

Here Spartan 3 tools will use to work on this project, it has the operating speed of 100MHZ on board, and it also has soft processor Micro Blaze. And I working on tools like Xilinx ISE 13.4 and also work on simulation software ModelSim6.3c.

III.PROJECT METHODOLOGY:

Below flow graph shows the method or procedure of the project, and each block present in the flow graph is briefly explained below:

Specification: Here Spartan 3 device will use to work on this project, it has the operating speed of 100MHZ on board, and it also has soft processor Micro Blaze. And I work on tools like Xilinx ISE 12.2 and also work on simulation software ModelSim6.3c.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Block Diagram: In this architecture for Reconfigurable Router for NoC Applications on FPGA.

Logic Design: In this project logic design will be use to design architecture for Reconfigurable Router for NoC Applications on FPGA.

Verilog: To work on this project Verilog language will use for the implementation of architecture for Reconfigurable Router for NoC Applications on FPGA in Xilinx ISE 13.4.

Synthesis: After combining, testing has to do, i.e whether the program is working properly or not, if yes it will continue with next process else it has to rewrite or correct. After correction again test the program, if its successfully working then it will be implement.

Implementation: All the process till testing will implement in this step.

Physical Dumping: In this step implementation of the project will dump into the FPGA.

Physical Testing: Finally the project will be test in the FPGA kit.

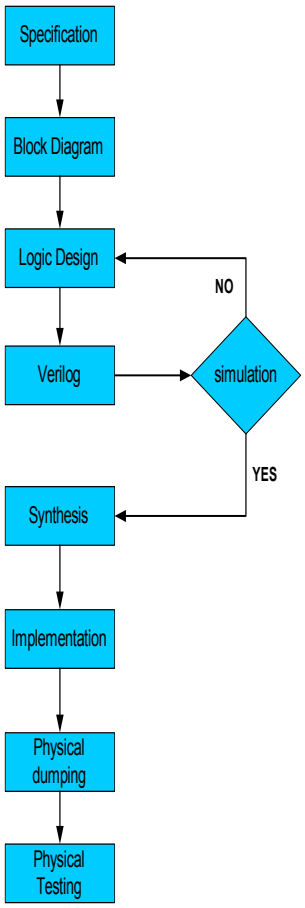

Fig4: FPGA Data Flow

IV.PROPOSED STRUCTURE:

A buffer is a region of a physical memory storage device and is also used to temporarily store the data while it is being moved from one place to another place. The data is stored in a buffer as it is retrieved from an input device or just before it is sent to an output device.NOC have a relative area and delay overhead compared to the buses. These can be improved in application specific systems where heterogeneous communication infrastructures provide high bandwidth in a localized fashion and reduce underutilized resources. One approach for improving area and/or delay of NoCs is to consider dynamic adaptation of the resources at runtime. Routers need buffers, routing tables, a switching circuit and arbiters within it. Thus, they occupy more area than a bus based networksThe efficiency of the interconnected network can be improved if the runtime changes.

A system running a set of applications can benefit from the runtime reconfiguration of the topology and of the routers to improve performance, area and power consumption considering a particular data communication pattern. Customizing the number of ports, the size of the buffers, the switching technique, the routing algorithm and the switch matrix are possible runtime changes that can be considered in a dynamically adaptive NoC. One way to overcome the relative area and delays associated with the Networks-on-chip is by using the buffer resize technique.

Buffer resizing technique can be used to improve latency or minimize the area and delay of a router. For heavy loaded networks the buffer size increases which will decrease blocking of packets. Since, buffers can be emptied faster because the following buffers in the path have higher probability of not being full. The average latency grows faster with increasing injection rate for NoCs with smaller buffers. The latency decreases rapidly when the depth of most utilized buffers are doubled. There are mainly different methods in implementing buffer resizing technique. Firstly, in this a channel may borrow some buffer units from its neighbor. This method uses extra multiplexers so that any buffer unit can be assigned to a neighbor. If this method is implemented on FPGA, then the buffers have to be implemented as registers, which can be very expensive in terms of resources. So this solution is does not work for FPGA.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Secondly, a centralized buffer structure is proposed in this method, which dynamically allocates buffer resources based on traffic requirements. Each and every buffer is divided into slots which are implemented as registers. Slots are than linked by one linked list. This centralized buffer management approach dynamically allocates buffer slots to different packets according to the traffic needs. Again the problem occurred with this method. The problem is that, they are quite expensive in terms of resources when implemented on FPGA.

And the third method of buffer resizing technique which greatly reduces the area and latency of the network compared to the previous two methods. Here we use the use of floating buffers that can be assigned to any output port to increase their buffering size. With extra buffers it is possible to reduce the size of the fixed buffers to a minimum (e.g., 16 words) and then dynamically compensate for the lack of buffering by assigning the floating buffers to the output ports. The router will have a structure similar to the static router, except that the adaptive router includes a few extra modules to control the floating buffers and to dynamically put them in the data path of the router.

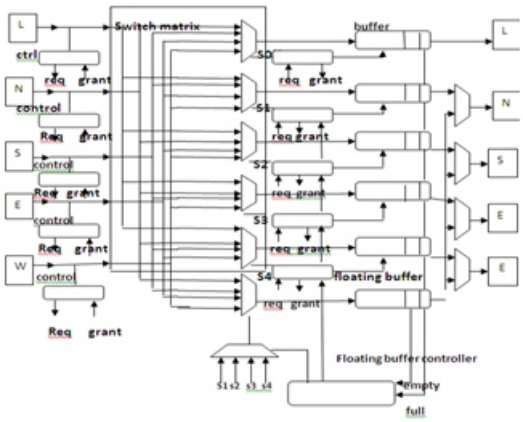

Fig.5. Architecture of an adaptive router with buffer resize.

Figure 5 shows an architecture having a single floating buffer that can be associated with any output port, except with the local port. This port has not been considered since we assume the processing element connected to the local port is unable to collect data simultaneously from two inputs. The arbiter associated with an output port receives the requests from the input ports and grants access to its buffer. Case the static buffer is full and the floating buffer is assigned to it then it grants access to the floating FIFO. If the floating FIFO gets full then it returns to the static FIFO.

The floating buffer is controlled by the floating buffer controller. The floating buffer can be reassigned only if it is empty and the controller assigns it to a port having a full fixed buffer for at least five cycles. For a fair assignment, all eligible ports for assignment (those with full static buffers) are chosen in a round-robin manner. Using more floating buffers we can improve the performance from 4% to 7%.

V.RESULTS:

This project mainly focuses on dynamic buffer resizing technique for NoC architecture. All communication system needs the buffers to store the data in it either at input side or at output side. As the buffer size start increasing, the area increases as well hence, buffer size is limited, which will have the impact on the performance. Here we are using two types of buffers namely,

- Static buffers &
- Dynamic or floating buffers.

All these approaches aim to increase the throughput and minimize the latency which, by means of virtual channel flow control. In statically allocated buffer architecture, size and organization are design time constraints and thus, do not perform optimally for all traffic conditions. In addition, statically allocated virtual channel consumes a waste of area and significant leakage power. However, dynamic buffer allocation scheme claims that buffer utilization can be increased using dynamic virtual channels. Static buffers are fixed buffers where as dynamic buffers varies with the application.

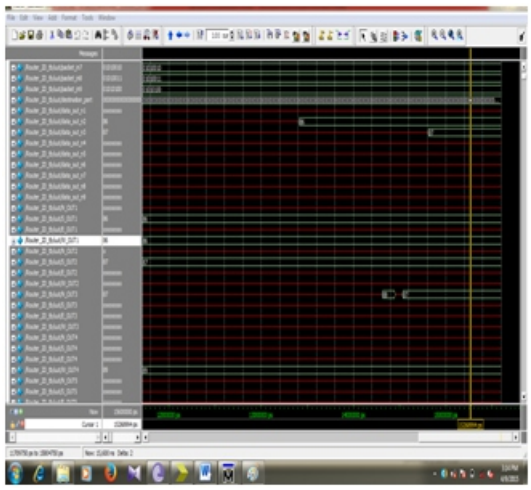

Fig 6.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

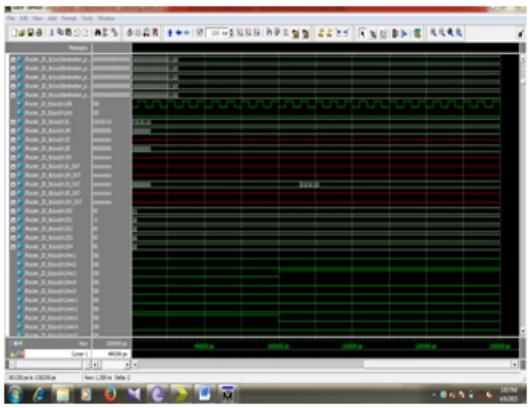

Fig 7.

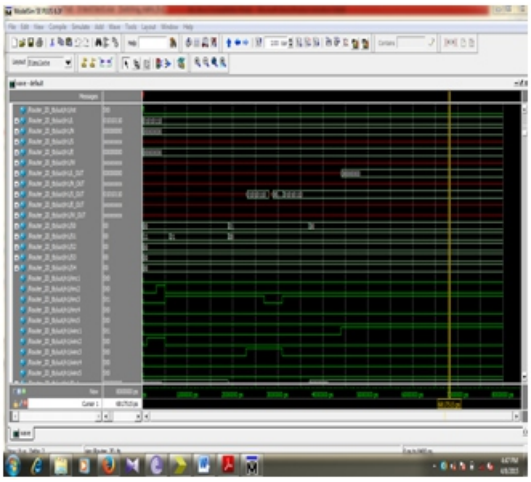

Fig 8.

VI.CONCLUSION:

This Project mainly predicts on adaptive buffer where we have many input nodes and output nodes with corresponding buffers. Buffer size is proportional to area hence here adaptability has a great significance over area as well as performance. We are planned to design dynamic architecture of router using adaptive buffer resize technique. So new concept and new router design in 2D architecture.

VII.FUTURE WORK:

In future, we can increase the router size like 4X4, 5X5 etc, and architecture to allow more data packets in 2D NOC.

REFERENCE:

[1] Ali Akoglu, Kathleen Melde, Janet Roveda Department of Electrical and Computer Engineering. University of Arizona Tucson, USA ," FPGA Based Single Cycle, Reconfigurable Router forNoC Applications", 978-1-4673-5762-3/13/$31.00 ©2013 IEEE.

[2] T. Bjerregard and S. Mahadevan, "A survey of research and practices on network-on-chip," ACM Computing Surveys (CSUR), vol. 38, no. 1, pp.1-51, 2006.

[3] S. Borkar, "Future of Interconnect – A Contrarian View", SLIP, 2010.

[4] Ye Lu, J. McCanny and S. Sezer, "Generic Low-Latency NoC Router Architecture for FPGA Computing Systems," Field Programmable Logic and Applications (FPL), pp.82-89, 2011.

[5] M. Andres, S. Borkaret. al.,"A 2.9 Tb/s 8W 64-core Circuit Switched Network on Chip in 45nm CMOS," ISSCC, pp. 182 – 185, 2008.

[6] N. Genko, D. Atienza, G. D. Micheli, J. M. Mendias, R. Hermida, and F.Catthoor, "A complete network-on-chip emulation framework," Design,Automation and Test in Europe, vol. 01, pp. 246–251, 2005.