Design of 3x3 router using buffer resizing technique for 1d and 2d NoC architectures

Vivek Raj .K[1], Prasad Kumar B M[2], Shashi Raj .K[3]
[1]Department of ECE, SJCIT, Chickballapur-562101, Karnataka, India.
[2]Assistant Professor, Department of ECE, SJCIT, Chickballapur-562101, Karnataka, India
[3]Assistant Professor, Department of ECE, DSCE, Bangalore-560078, Karnataka, India.

Abstract- **An FPGA based, Reconfigurable 3x3 router for both 1-D and 2-D network on chip architectures design 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. These routers are reconfigurable and they are capable of transmitting both 1D and 2D packets throughout the network, this reduces the power consumption when communication pattern changes. In general these routers occupy most of the area in the network, which however reduces the performance and increases the latency. The latency in transmitting packets is decreases with increasing the depth of the buffers; this consumes more area in the routers. The reconfigurable router proposed 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 results prove the proposed design is robust and cost effective. Each router consumes a mere 3.689W power in the worst case while achieving high clock rates of 344.270MHz easily on the target FPGA device post design synthesis and emulation.**

Key words: **Network on chip, buffer resize, FPGA, router, data packets, FIFOs**

1. INTRODUCTION

First of all, in NoC architectures, several cores share the workload of a task. Data transfer from one core to another is very frequent and therefore, much emphasis needs to be placed on this area. Power consumption limits how many cores can be placed on a single chip and be utilized efficiently at the same time. Thus, reducing energy consumed per logic operation is becoming more and more important to keep power dissipation within limit. Secondly, since all cores are connected via an interconnect fabric, be it bus, ring or mesh, the interconnect topology plays a Major role in the performance of the network. Henceforth, it is clear that router energy consumption and interconnect fabric topology are the two leading factors that limit the performance of NoCs.

NoCs provide much higher bandwidth than buses but have higher area and delay. Routers need buffers, routing tables, a switching circuit and arbiters. So, they occupy more area than a bus based network. Also, direct bus connections are faster than pipelined connections through one or more routers since these introduce a delay due to packing, routing, switching and buffering.Since Several researches has been made in the field of router design , In this paper, we analyze the proposed reconfigurable router for NOC architectures and some proposed dynamic buffer resize techniques used to design routers for FPGA and then propose and analyze a more efficient buffer resize technique for reconfigurable 3x3 router. section II includes related work, in section III, we analyze the proposed system. Section IV shows the results and, finally, section V concludes the paper.

ISSN: 2278 – 7798

II.RELATED WORK

As semiconductor technology enable the integration of increasing numbers of IP blocks in a single SoC, Interconnect infrastructures, such as buses, switches, and networks on chips (NoCs), combine the IPs into a working SoC Due to the interplay between increasing chip capacity and complex applications, System-on-Chip (SoC) development is confronted by severe challenges, such as managing deep submicron effects, scaling communication architectures and bridging the productivity gap. Network-on-Chip (NoC) has been a rapidly developed concept in recent years to tackle the crisis with focus on network-based communication. NoC problems spread in the whole SoC spectrum ranging from specification, design, implementation to validation, from design methodology to tool support. Paper [1] describes, using on-chip interconnection networks in place of ad-hoc global wiring structures the top level wires on a chip and facilitates modular design. With this approach, system modules (processors, memories, peripherals, etc...) communicate by sending packets to one another over the network. The structured network wiring gives well-controlled electrical parameters that eliminate timing iterations and enable the use of high-performance circuits to reduce latency and increase bandwidth. This paper introduces the concept of on-chip networks.NoC-based interconnect achieves better scalability than traditional solutions in terms of the number of Modules on a chip. Recent analysis [2] shows that the power and area of a NoC based on a grid of routers connected by short wires scales linearly with the chip density (number of modules), whereas common solutions, namely buses, segmented buses, and point to point links, exhibit super-linear growth in both dynamic power and area. In case of a bus, the increase in the number of modules not only increases the wire length, but also introduces a major waste of energy due to the inherent broadcast nature of buses. Paper [3] proposes a NOC platform, consisting of architecture and design methodology, which scales from a few dozens to several hundred or even thousands of resources. Are source may be a processor core, a DSP core, an FPGA block, a dedicated HW block, a mixed signal block, or a memory block of any kind such as RAM, ROM or CAM. The proposed NOC platform would effectively separate the specification of inter-task communication from the implementation of that communication; separate the design, implementation and verification of individual tasks from the rest of the application (a precondition for task reuse); separate the development, optimization and verification of the individual resource from the network infrastructure. We argue that the consequent separation of different concerns is a way to develop high-performance, cost-effective products while boosting design productivity. Paper [4], describes the architecture's main attractions of packet-switched NoC systems, while addressing the problem of hop-by-hop propagation latency. Each pipeline stage is optimized as such that the zero-load packet propagation latency of the proposed NoC is only two cycles per hop including the router pipeline and link traversal. This we believe represents a significant enhancement over state-of-the-art FPGA designs. Key contributions include (a) the definition of a highly scalable router architecture capable of supporting various network topologies on FPGA (1D, 2D and 3D) (b) the architectural optimization of the router such that two cycles per hop can be achieved, (c) a detailed analysis of the proposed architecture in terms of scalability, hardware cost (area), operation speed (critical path) and power dissipation and (d) demonstrating the feasibility of the proposed router in an real-world on-board design environment.

The paper [6] has presented a novel NoC reconfigurable framework that can reconfigure the NoC topology at run-time, as well as enabling path reconfiguration and express lines creation/removal, while introducing an overhead on average of 10% of an initial static NoC design.Paper [7] provides ReNoC architecture that enables the network topology to be reconfigured using energy-efficient topology switches. The architecture was evaluated by mapping an application to a static 2D mesh topology as well as ReNoC architecture in two different topology configurations. The power consumption was decreased by 56% when configuring an application specific topology, compared to the static 2D mesh topology. The topology switches increased the area of the NoC architecture with 10%, and only contributed with 5% of the power consumption in the application-specific topology. The evaluation shows that the ReNoC architecture enables application-specific topologies to be configured with little overhead and indicates that the architecture has great potential for future SoC platforms.With an increasing

ISSN: 2278 – 7798

trend to implement Network-on-Chip (NoC)-based Multi-Processor Systems-on-Chips (MPSoCs), NoCs need to have guaranteed services and be dynamically reconfigurable. Many current NoCs consume too much area and cannot support dynamic reconfiguration. In paper [8], present an area-efficient Spatial Division Multiplexing (SDM)-based NoC. We replaced area consuming 32-bit to M-bit serializers with 32-bit to 1-bit serializers in the network interface and incur almost no loss in performance. They also restrict flexibility in the router to achieve further area reduction. A separate area-efficient control network, with an overhead of 3.9% of the total area of the NoC, is developed to support dynamic reconfiguration. In paper [9] they proposed a reconfigurable MPNoC architecture in which both the network and the processing nodes are configured. The flow allows for each component to be tested separately prior to testing the entire design. This allows for quick design iterations of the system. They presented a design flow that can be used to generate network-based MPSoCs quickly. The application determines the architecture and the communication requirements of the system. The design of computation and communication infrastructure is decoupled. IP blocks (processing nodes) and the network are generated separately. Later the two are customized to the application requirements.

In the buffer resize technique in [10] a channel may borrow some buffer units from its neighbour. The implementation uses extra multiplexers so that any buffer unit can be assigned to a neighbour. Paper also shows that the router with adaptive buffers of size 4 achieves the same performance of a static router with buffers of size 9, with a decrease of 6% in the area. They also proposed a same router that adapts itself to provide appropriate buffer depth for each channel to sustain the performance with minimum power dissipation. The paper [11] proposes a dynamic buffer resize technique, in which a number of buffer blocks are dynamically reassigned on-demand. The overhead of the adaptive router compared to that of the static router is 500 LUTs.With such number of LUTs it is possible to have 12 FIFOs of depth 16, which reduces considerably the efficiency of the solution. In [12],a centralized buffer structure is proposed, which dynamically allocates buffer resources based on traffic requirements. Each buffer is divided into slots 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.

III. FPGA BASED NOC DESIGN

The proposed design main NOC contains mainly 9 routers each acts as a 2d NOC structure. each contains data in, N,S,E,W are inputs and S0,S1,S2,S3,S4 are select lines for each data and write0,write1,write2,write3,write4 are write data and read0 ,read1,read2,read3,read4 are read data and D_out, N_out, S_out, E_out,W_out are acts as output as shown in the figure 3.1(a).

The detailed architecture mainly contains reconfigure 3x3 routers. Each router design uses buffer resizes techniques to improve the efficiency. The inputs for Main NOC are d_in1, d_in2, d_in3, d_in4, d_in5, d_in6, d_in7, d_in8 are acts as a packets each contains 8 bit data. Totally 9 -N (North), 9- S (south),9- E(east), 9-W(west) inputs are feeded to the main NOC. and S0,S1,S2,S3,S4 are select lines 9times for each data and write0, write1, write2, write3, write4 are write data 9 times and read0 ,read1,read2,read3,read4 are read data9 times and D_out, N_out, S_out, E_out,W_out -9 times are acts as outputs in main NOC.

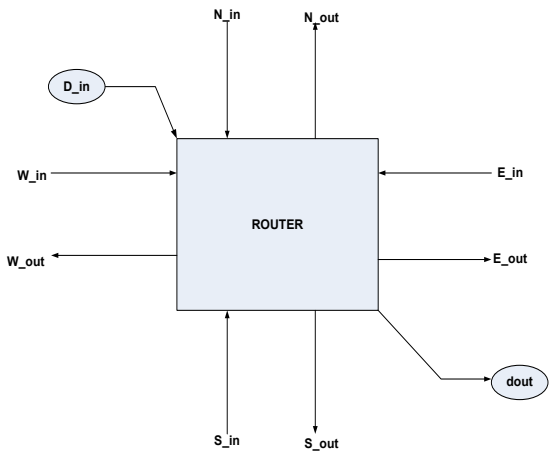

FIG 2.1(A): SINGLE ROUTER

In 1st router design, the din1, zero (N), N2(S) from 2nd router north output, zero (W) and E4 from router 4 east out is feeded to the router as a inputs respectively. According to the select lines used in buffer resize technique it will write the data using write 0, 1,2,3,4. And then it will read the data and

ISSN: 2278 – 7798

generate the outputs D_out1, N1, S1, W1, and E1 respectively. The same way the data communicate in each router according to the select line. And generates the outputs in different 8 routers. Each router design works according to the buffer resize techniques. The performance of a communication network is closely depended on the architectural model and design of the individual component of the network, base on the application requirements static And dynamic buffer size is varied to meet the design goals/constraints. In this section I attempt to capture architectural issues and highlight their implications.

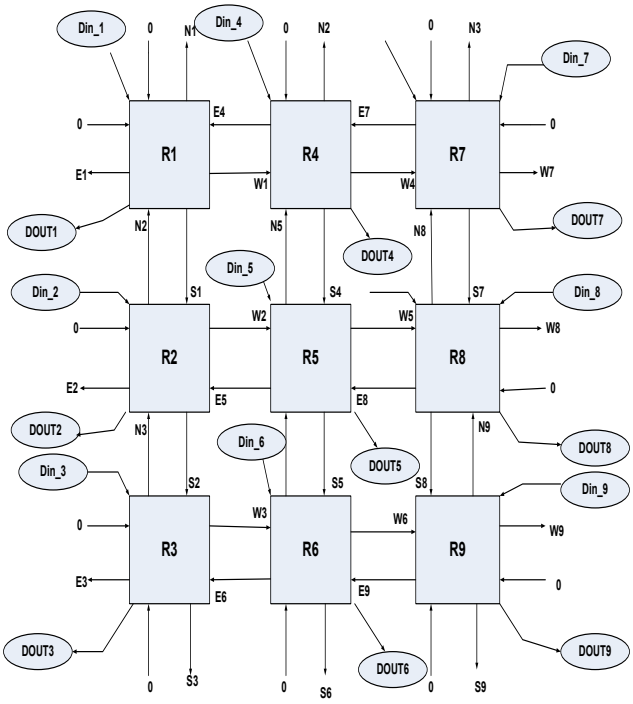

FIG 2.1(B): DETAILED ARCHITECTURE FOR MAIN NOC CONTAINS 3 X3 ROUTERS.

.

2.2. BUFFER RESIZES TECHNIQUE

A buffer is a region of a physical memory storage used to temporarily store data while it is being moved from one place to another. Typically, the data is stored in a buffer as it is retrieved from an input device (such as a microphone) or just before it is sent to an output device (such as speakers).

Networks-on-chip have a relative area and delay overhead compared to buses. These can be improved in application specific systems where heterogeneous communication infrastructures provide high bandwidth in a localized fashion and reduce underutilized resources. However, for general purpose architectures, design time techniques are not efficient. One approach for improving area and/or performance of NoCs for general purpose systems is to consider dynamic adaptation of the resources at runtime. NoCs provide much higher bandwidth than buses but have higher area and delay. Routers need buffers, routing tables, a switching circuit and arbiters. So, they occupy more area than a bus based network. Also, direct bus connections are faster than pipelined connections through one or more routers since these introduce a delay due to packing, routing, switching and buffering.

The efficiency of the interconnection network can be improved if runtime changes are considered. A system running a set of applications can benefit from the runtime re configuration 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 can be used to improve latency or minimize the area of a router. For heavy loaded networks increasing the buffer size 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. For example, with 35% network loading and buffers with depth of 16 the average latency is about 128 cycles. Increasing the buffers to 32 words will improve the latency by about 33 % and increasing it to 64 words will improve the latency by about 40%. The latency decreases rapidly when the depth of most utilized buffers are doubled. The main disadvantage of this approach is the area over head associated with the buffers. Therefore, in the customization process of the router, the size of the FIFOs must be carefully chosen to avoid using

ISSN: 2278 – 7798

buffers deeper than what is needed to achieve the system requirements while optimizing area utilization. The efficiency of dynamic buffer resize depends on the cost of buffers in terms of occupied resources compared to that of other blocks of the router. This cost relation depends on the target technology, that is, buffers occupy more area Relative to the other blocks when implemented in standard-cell technology than when implemented in FPGA.

There are different methods in implementing buffer resize technique. In one of the buffer resize technique method a channel may borrow some buffer units from its neighbor. The implementation uses extra multiplexers so that any buffer unit can be assigned to a neighbor. By using this method the router with adaptive buffers of size 4 achieve the same performance of a static router with buffers of size 9, with a decrease of 6% in the area. But If implemented this method in FPGA, the buffers have to be implemented as registers, which are very expensive in terms of resources given that an implementation of buffers with size 4 or 9 are best implemented with SRL16 primitives using the same number of LUTs. Also, the solution is not scalable since the number of extra logic increases more than linearly with the size of the buffers .So this solution is worthless for FPGA.

In another method a centralized buffer structure is proposed, which dynamically allocates buffer resources based on traffic requirements. Each buffer is divided into slots 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. The problem with these solutions is that they are quite expensive in terms of resources when implemented in an FPGA.

The third method of buffer resizes technique which greatly reduces the area and latency, 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 most congested. 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 (fig2.2.1).The architecture shown in the figure has 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 assignment of the floating buffer is made by the floating buffer controller. Several policies can be Followed to assign the floating buffer. Here the floating buffer can be reassigned 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 adds an additional improvement to the performance from 4% to 7%.

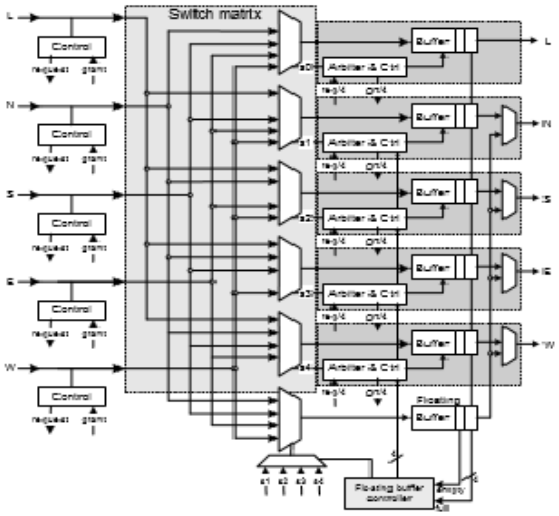

FIG.2.2. 1.ARCHITECTURE OF AN ADAPTIVE ROUTER WITH BUFFER RESIZE.

IV. RESULTS

This section discusses the results obtained by running several tests mentioned in the previous section. A set of experiments were conducted to estimate the area, utilization and maximum clock rates for each network topology. The data was obtained post synthesis of the design and also providing the user defined constraints for clock, reset and other external

ISSN: 2278 – 7798

user controlled inputs. Here Xilinx 13.4 ISE is used to design the proposed design and it is simulated using Modelsim6.3f. The language selected for the design is Verilog and implemented on FPGA. The table1 shows the device utilization summary for the 2D NOC architecture .it clearly shows the different resources used and there effective utilization rate in percentage. The timing report also shows the maximum frequency that can be achievable through the 2D network.

TABLE 1. DESIGN SUMMARY OF 2D- NOC ARCHITECTURE FOR ROUTER DESIGN

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	4298	301440	1%
Number of Slice LUTs	4846	150720	3%
Number of fully used LUT-FF pairs	2180	6964	31%
Number of bonded IOBs	166	720	23%
Number of BUFG/BUFGCTRLs	1	32	3%

TIMING SUMMARY:

Speed Grade: -2

Minimum period: 2.905ns (Maximum Frequency: 344.270MHz)

Minimum input arrival time before clock: 1.992ns

Maximum output required time after clock: 1.717ns

Maximum combinational path delay: No path found

Table 2 shows the design summary of 1D NoC architecture and its timing summary.

TABLE.2.DESIGN SUMMARY OF 1D- NOC ARCHITECTURE FOR ROUTER DESIGN

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	583	301440	0%
Number of Slice LUTs	609	150720	0%
Number of fully used LUT-FF pairs	274	918	29%
Number of bonded IOBs	102	720	14%
Number of BUFG/BUFGCTRLs	1	32	3%

TIMING SUMMARY:

Speed Grade: -2

Minimum period: 2.905ns (Maximum Frequency: 344.270MHz)

Minimum input arrival time before clock: 1.905ns

Maximum output required time after clock: 2.221ns

Maximum combinational path delay: No path found

Table 3 shows the number of LUT and FIFO used for the router design which doesn't uses buffer resize technique.

TABLE 3. DESIGN SUMMARY OF ROUTER DESIGN

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	110	301440	0%
Number of Slice LUTs	259	150720	0%
Number of fully used LUT-FF pairs	107	262	40%
Number of bonded IOBs	12	720	1%
Number of BUFG/BUFGCTRLs	1	32	3%

Table 4 shows the number of LUT and FIFO used for the router design which uses buffer resize technique.

TABLE 4. DESIGN SUMMARY OF FIFO FOR ROUTER DESIGN

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	79	301440	0%
Number of Slice LUTs	45	150720	0%
Number of fully used LUT-FF pairs	20	104	19%
Number of bonded IOBs	26	720	3%
Number of BUFG/BUFGCTRLs	1	32	3%

COMPARISON RESULTS

The table shows the comparison of both the Architecture and it proves our proposed design is better than the previous design. Interns of slices, LUT'S, Flip-flop pairs (i.e. AREA) and Frequency.

ISSN: 2278 – 7798

FPGA UTILIZATION	PROPOSED	PREVIOUS
Number of Slice Registers	0%	1%
Number of Slice LUTs	0%	4%
Number of fully used LUT-FF pairs	29%	40%
Freuency	344.27MHZ	325 MHZ

V. CONCLUSIONS

The reconfigurable 3x3 router is designed using adaptive buffer resize techniques and it is presented. The simulation model is implemented in the Modelsim simulation. Simulation results describe the principles and performance of the Main NoC and adaptive buffer resize of a NoC model. An evaluation is done based on the comparison of input size as well as buffer size. When the static buffers are filled the data will be routed to floating buffer , hence data is not lost as well as area is constrained, here the floating buffer plays a greater role in saving the data when static memory are filled. The obtained results show that adaptive buffer resize improves area as well as performance hence available resources are used effectively.

REFERENCES

1. William J. Dally and Brian Towles" Route Packets, Not Wires: On-Chip Interconnection Networks".

2 .E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. Cost considerations in network on chip.Integration,Special Issue on NoC, 38(1):19–42, Oct. 2004.

3.Shashi Kumar, Axel Jantsch, Juha-Pekka Soininen"A network on chip architecture and design methodology". Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSIi02) 0-7695-1486-3/02 © 2002.

4.Ye Lu, John McCanny, Sakir Sezer " Generic Low-Latency NoC Router Architecture for FPGA Computing Systems".

5. Jingcao Hu, Radu Marculescu "Energy and Performance Aware Mapping for Regular NoC Architectures".

6. Vincenzo Rana, David Atienza, Marco Domenico san- tambrogio Donatella Sciuto and Giovanni De Micheli "A Reconfigurable Network-on-Chip Architecture for Optimal Multi-Processor SoC Communication".

7. Mikkel B. Stensgaard and Jens Spars "ReNoC: A Network-on-Chip Architecture with Reconfigurable Topology".

8. Zhiyao Joseph Yang, Akash Kumar and Yajun Ha" An Area-efficient Dynamically Reconfigurable Spatial Division Multiplexing Network-on-Chip with Static Throughput Guarantee".

9. Ido Ovadia, Jos Huisken, Henk Corporaal,Jef van Meerbergen and Yajun Ha"Reconfigurable Multi-Processor Network-on-Chip on FPGA".

[10] Concatto,C.,Matos,D.,Carro,L.,Kastensmidt,F.,Susin, A., and Kreutz, M., "NoC Power Optimization Using a Reconfig- urable Router". In the IEEE Symposium on VLSI, 2009.

[11] Al Faruque, M.A., Ebi. T., Henkel J., "ROAdNoC: Runtime ObservabilityforanAdaptiveNetworkonChipArchitecture". In IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2008, 543-548.

[12] Wang, L., Zhang, J., Yang, X., and Wen, D., "Router with Centralized Buffer for Network-on-Chip". GLSVLSI, 2009.

ISSN: 2278 – 7798