International Journal of Advance Engineering and Research Development (IJAERD)
Volume 1,Issue 3, April 2014, e-ISSN: 2348 - 4470 , print-ISSN:2348-6406

FPGA Implementation of Network-on-Chip Router Architecture for Multicore-SoC Communication Paradigm

K.Subbulakshmi[1], K.Balamurugan[2]

[1]PG Scholar, Department of ECE, Einstein College of Engineering,lakssubbu91@gmail.com
[2] Associate Professor, Department of ECE, Einstein College of Engineering,bala237115@gmail.com

Abstract— In the deep submicron era, the downscaling of silicon technology and the possibility of building multiprocessor System-on-Chip (MPSoCs) makes intrachip communication, a key challenge in the gigascale chip designing process. Performance and power of gigascale System-on-Chip(SoC) is mainly communication dominated. SoC communication architectures start facing scalability as well as modularity limitations and more advanced bus specifications are emerging to deal with these issues at the expense of silicon area and complexity. To overcome the scalability limitations, Network-on-Chip paradigm is currently viewed as a innovative approach to provide a high performance, scalable and robust infrastructure for on-chip communication. This paper presents a Network-on-Chip router architecture for intrachip communication of SoC architectures. The router is designed using VHDL language and implemented on Virtex6 FPGA with the help of Integrated Software Environment ISE 14.5. The simulation and synthesis results are also presented.

Keywords- NoC, SoC , VHDL, Router, FPGA

I.INTRODUCTION

As indicated by the International Technology Roadmap for Semiconductors (ITRS), nanometer Systems-on-Chip (SoCs) will most likely not have an economic yield if all transistors must be functional [1] [2]. It is expected that Moore's law will continue to hold for another five to fifteen years where billion gates can be integrated in a chip. This capacity will allow integration of several tens to hundred resources like processor cores, DSP cores, and interface circuits (like Blue-tooth or Ethernet adapter), FPGA blocks, analog blocks, and memory blocks (any kind such as RAM, ROM and CAM). Thereby, it is possible to integrate more than one Processing Element (PE) in a SoC, being known as Multi-Processor System-on-Chip (MPSoC). MPSoCs have been widely used in high performance embedded systems, such as web servers, network processors, and parallel media processors. They combine the advantages of data processing parallelism of multi-processors and the high level integration of SoCs. The continuously increasing number of cores for such multi-billion transistor SoCs calls for a new communication architecture as traditional bus-based architectures are inherently nonscalable, making communication a bottleneck [3].
The Network-on-Chip (NoC) architecture paradigm, based on a modular packet-switched mechanism, can address many of the on-chip communication design issues such as performance limitations of long interconnects, and integration of high number of PE on a chip [4] [5].

II.NETWORK-ON-CHIP

2.1 The Advantages of On-Chip Networks

Energy efficiency, reliability, reusability, scalability, and flexibility are the most important benefits of NoC from other on-chip communication approaches.

2.1.1 Energy Efficiency

According to the International Technology Roadmap for Semiconductors (ITRS) [8] and Semiconductor Industry Association (SIA) [9] roadmaps, clock frequency and number of on-chip devices are increased. That is, much tighter power budgets for all system components are required. Based on the roadmaps, as computation and storage components benefit from device scaling, the energy for global communication does not scale down. Hence, communication-energy minimization will be a growing concern in future technologies. The on-chip networks aim to reduce this problem by scaling wires. This new model allows the decoupling of the PEs from the network. The need for global synchronization can thereby disappear. This new approach employs explicit parallelism, exhibits modularity to minimize the use of global wires, and utilizes locality for power minimization [10] [11]. Furthermore, network traffic control and monitoring can help in better managing the power consumed by networked computational resources. For instance, clock speed and voltage of end nodes can be varied according to available network bandwidth. The emphasis on energy minimization creates a sleuth of novel challenges that have not been addressed by traditional high-performance network designers [10] [11].

2.1.2 Reliability

As the geometries of the transistors reach the physical limits of operation, it becomes increasingly difficult for the hardware components to achieve reliable operation. The variability in process manufacturing, issues of thermal hotspots and effects of various noise sources, such as power supply fluctuations, pose major challenges for the reliable operation of current and future NoC-based MPSoCs. NoCs are particularly suited for implementation of fault-tolerant techniques, due to their inherent parallelism and potential for reconfigurability. Fault-tolerant techniques can be implemented at different levels, from hardware redundancy to software-based error recovery schemes. Adaptive routing algorithms combined with error detection mechanisms show great promise in achieving fault-tolerant on-chip communication. If data is sent on an unreliable channel in packets, error detection and recovery is easier, because the effect of errors is contained by packet boundaries, and error recovery can be carried out on a packet- by-packet basis. Error correction can be achieved by using standard error correcting codes (ECC), whereas robust and fault-tolerant routing algorithms can route around faulty regions [12].

2.1.3 Reusability

PEs are usually obtained from internal sources or third parties, and integrated on a single chip. These reusable PEs may include embedded processors, memory blocks, interface blocks, analog blocks, and components that handle application specific processing functions. Corresponding software components are also provided in a reusable form and may include real-time operating systems and kernels, library functions, and device drivers. That is, PEs are reusable in nature if they conform to a common interface and synchronization mechanisms with the on-chip network. Using a standard interface such as AXI, OCP , and DTL [13], in on-chip networks facilitates the employment of reusable components. In fact, employing a standard interface does not change the way PEs re developed, since they will still be developed for a certain protocol. What changes is that a public domain protocol is used and accepted by the industry as a standard, like the PCI standard for microcomputer manufacturers. Accordingly, not only the PEs reusability becomes higher but also the design

time is reduced [16]. In addition, on-chip routers are generic in nature and the communication can be employed with any conforming PE.

2.1.4 Scalability

NoC platform is composed of on-chip routers and communication links that are basically distributed and independent. Each PE is added into the network along with a dedicated router having a unique address or coordinate in the network. The communication exploits the packet switching scheme while there is no central arbitration mechanism of the communication platform. Therefore, the performance in this communication architecture is not constrained or degraded by the addition of PEs. This is the essential characteristic of a scalable and modular architecture [1] [2] [3]. Indeed, on-chip interconnection network plays an important role in providing scalability to integrate hundreds or even thousands of processing elements in a single billion-transistor chip and alleviate design productivity gap. In fact, using data packets for communication, a high level of parallelism is achieved as all channels can be operated simultaneously. Thereby, on-chip network improves the scalability in comparison with previous communication structures such as shared buses or segmented buses.

2.1.5 Flexibility

Utilizing common buses between the communicating resources in SoCs will not give any flexibility since the needs of the communication have to be thought of every time a design is made. However, they suffer from low scalability [1]- [5]. NoC solves their shortcomings by implementing a communication network of routers and resources. NoC is a very flexible communication infrastructure allowing the same physical link to be shared by many different connections. As future SoC platforms are expected to contain hundreds of PEs,
NoC needs to support an even larger number of connections and many connections span a large number of routers. This leads the same SoC platform to be used in a wide range of different applications and thereby increases the production volume. As the same SoC platform is to be used for many different applications, the NoC must be able to support a wide range of bandwidth and Quality-of-Service (QoS) requirements. The requirements of the applications can be very different, and the NoC must therefore be very flexible.

III. NOC TOPOLOGY

The network topology is the study of the arrangement and connectivity of the routers. In other words, it defines the various channels and the connection pattern that are available for the data transfer across the network. Performance, cost, and scalability are the important factors in the selection of the appropriate topology. Shared-Bus, Crossbar, Butterfly Fat-Tree, Ring, Torus, and 2D-Mesh are the most popular topologies for on-chip interconnects which have been commercially used [2]. Direct networks have at least one PE attached to each router of the network so that routers may regularly spread between PEs. This helps to simplify the physical implementation. The shared-bus, ring, and 2D mesh/torus topologies are examples of direct networks, and provide tremendous improvement in performance, but at a cost of hardware overhead, typically increasing as the square of the number of PEs. All tree-based topologies where PEs are connected only to the leaf routers (e.g. the butterfly topology) as well as crossbar switch are indirect networks. The shared-bus topology is the simplest using a shared link common to all PEs where they compete for exclusive access to the bus. For communication intensive applications it is necessary to overcome the bandwidth limitations of the shared-bus topology and move to scalable networks. However, this topology scales very poorly as the number of PEs increases. A small modification to the shared-bus topology

is the ring topology where every PE has exactly two neighbors. In this topology, messages hop along intermediate PEs until they arrive at the final destination.

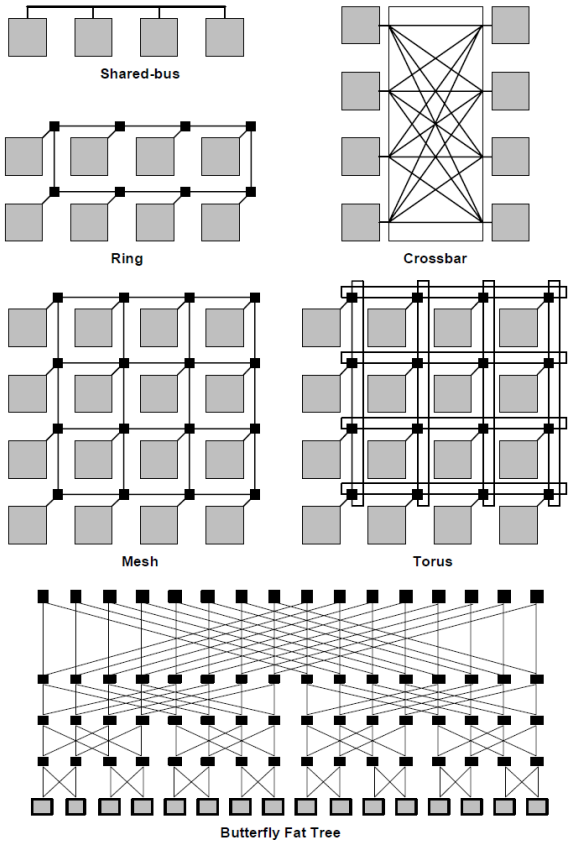

Fig.1 NoC topologies

The crossbar topology is a fully connected one which allows every PE to directly communicate with any other PE. Hence, each topology has its own advantages and disadvantages. The fat-tree topologies suffer from the fact that the number of routers exceeds the number of PEs, when the amount of PEs increases. This incurs an important network overhead. For the on-chip interconnects the network overhead is more critical than for the off-chip networks, and the design scalability is more essential. Because of the simple connection and easy routing provided by adjacency, mesh and torus networks are widely used in multiprocessor architectures. Both torus and mesh topologies are fully scalable. Although torus provides a better performance, the regularity, better utilization of links, and lower network overhead are some of the preferences for mesh. That is, the mesh topology is more economic scheme since the routers on the borders are smaller[5].

3.1 Switching Mechanism

The switching mechanism determines how messages traverse a route in a network. The goal is to effectively share the network resources among messages traversing the network.

Basically, circuit switching and packet switching form the two extremes of switching mechanisms.

In circuit switching a connection from a source to a destination is established prior to the transmission of data and exclusively reserved until the message is completely transferred, i.e. as in telephone networks that set up a circuit through possibly many routers for each call. This mechanism has low delay and guaranteed bandwidths, but suffers from channel utilization, low throughput, and long initialization time to setup a connection.

Packet switching is an alternative mechanism where data is not transmitted on a predefined circuit. A message can be divided into packets which share channels with other packets. Each packet consists of a header which contains routing and control information, data payload, and possibly a tail. The data payload follows the channel reserved by header while the tail releases the channel reservation. Packets are individually and independently routed through the network, and at the destination the packets are assembled into the original message. If a message is divided into several packets, the order of packets at arrival PE must be the same as departure. Therefore, in-order delivery is an essential part that should be supported by on-chip networks. The packet switching mechanism improves channel utilization and network throughput. In the packet switching domain, buffered flow control defines the mechanism that deals with the allocation of channels and buffers for the packets traversing between source and destination. The flow control mechanism is necessary when two or more packets compete to use the same channel, at the same time. Commonly three different buffered flow control strategies are used: store-and-forward, virtual cut through, and wormhole. When these mechanisms are implemented in on-chip networks, they have different performance metrics along with different requirements on hardware resources[10].

3.2 Flow Control Mechanisms

3.2.1 Store-and-Forward

The store-and-forward mechanism is the simplest flow control mechanism. In this approach, each router along the path stores the entire packet in the buffer and then, the packet is forwarded to a selected neighboring router if the chosen neighboring router has enough empty buffering space available to hold the whole packet. This mechanism requires a large amount of buffering space (at least the size of the largest packet) in each router of the network, which can increase the implementation cost dramatically. On top of that, network latency increases significantly because a packet cannot be forwarded to the next router until the whole packet is received and stored in the current router. Consequently, the store-and-forward approach is impractical in large-scale Networks-on-Chip.

3.2.2 Virtual Cut-Through

The virtual cut-through mechanism was proposed to address the large network latency problem in the store-and-forward strategy by reducing the packet delays at each routing stage. In this approach, one packet can be forwarded to the next stage before its entirety is received by the current route which reduces the store-and-forward delays. However, when the next stage router is not available, similar to the store-and-forward, the virtual cut through approach also requires a large buffering space at each router to store the whole packet.

3.2.3 Wormhole

In this mechanism, a packet is divided into smaller segments called FLITs (FLow control digIT). Then, the flits are routed through the network one after another, in a pipelined fashion. The first flit in a packet (header) reserves the channel of each router, the body (payload) flits will then follow the reserved channel, and the tail flit will later release the channel reservation. The wormhole mechanism does not require the complete packet to be stored in the router while waiting for the header flit to route to the next stages. One packet may occupy several intermediate routers at the same time. That is, the wormhole approach is similar to the virtual cut-through, but here the channel and buffer allocation is done on a flit-basis rather than packet-basis. Accordingly, the wormhole approach requires much less buffer space, thus, enabling small, compact and fast router designs. Because of these advantages, the wormhole mechanism is an ideal flow control candidate for on-chip networks[4].

3.3 Routing Algorithms

A routing algorithm determines a path for a packet to reach its destination. It must be decided within each intermediate router which output channels must be selected for the incoming messages. There are various types of routing algorithms differentiated according to their key characteristics. In accordance with the place where the routing decision is made they may be grouped as centralized, source, and distributed routing algorithms. If an algorithm is centralized the path is chosen by a centralized controller, if it is source routed then the route is determined by the source router prior to sending a packet, in distributed algorithms the path is chosen in a distributed manner at the intermediate routers. According to the way how they choose a path routing algorithms are broadly classified as deterministic and adaptive algorithms.

Deterministic algorithms do not take into account network conditions when they take a decision that is why they always supply the same path from source to destination. But, it is not the case for adaptive ones in which network load, traffic conditions, information about available output channels are always taken into consideration. Every algorithm has different impact on the network. Routing algorithms use a variety of metrics that affect the calculation of the optimal path for a message. Many properties of the interconnection network depend on the routing algorithm used because the complexity of an individual router has a significant impact on the complexity of the entire network. For example, if the routing algorithm is too complicated it will require extra hardware to realize the routing logic, moreover it may take much more time to make a decision about the direction where the message should be sent to. It will in turn lead to increase of packet latency. Deadlock, livelock and starvation freedom are also among those properties. This property shows the ability to guarantee that packets will not block or wander across the network forever or permanently stop and never reach its destination.

Deadlock: Deadlock is one of the situations that can postpone packet delivery indefinitely. It happens when a packet is requesting a resource that is held by another packet while holding the resource that is requested by other packet. There is a cyclic dependency between channels. Thus the packet may be blocked forever. Deadlock is the most difficult problem to solve. There are three strategies that can cope with deadlock: deadlock prevention, deadlock avoidance and deadlock recovery.

Livelock: Livelock usually happens in adaptive routing schemes. It happens when a packet is running forever in circular motion around its destination, because the channels that are required to reach the destination are occupied by other packets.In order to remove livelock several techniques have been proposed such as minimal path, restricted non-minimal path, probabilistic avoidance.

Starvation: Starvation may happen when a resource that was requested by a packet is always granted to other packets.Starvation can be avoided by using correct resource assignment scheme.

3.3.1 Deterministic Routing Algorithms

Deterministic algorithms should be progressive and profitable, which means that the header should move forward reserving a new channel at each routing operation, under condition that the supplied channel always brings the packet closer to the destination. Thus deterministic routing algorithms use greedy algorithms, always choosing the shortest path.

The most popular deterministic algorithm is known as *dimension-order routing*. It is based on the idea that some topologies can be decomposed into several orthogonal dimensions, i.e. hyper cubes, meshes and tori. The distance between two nodes in these topologies is computed as the sum of the offsets in all dimensions. The algorithm reduces one of these offsets in each routing step. The offset of the current dimension must be equal to zero before the algorithm considers the offset of the next dimension.

Dimension–order routing is usually used for meshes and hypercubes. In 2D mesh it is called XY- or YX-routing depending on the dimension in which a packet travels first. The algorithm is deadlock-free for *n*-dimensional hypercubes and meshes, as their channel dependency graph (CDG) is acyclic. CDG is a directed graph where channels are represented by vertices and edges are pairs of channels connected by a routing function. However, the CDG for some topologies has cycles. In order to remove cycles, physical channels may be split into virtual channels. Most commercially available parallel machines usually use distributed deterministic routing as it is simple and fast. But distributed deterministic routing assumes that the traffic is uniform. In case of non uniform traffic the performance of distributed deterministic routing in terms of latency and throughput is very poor [11].

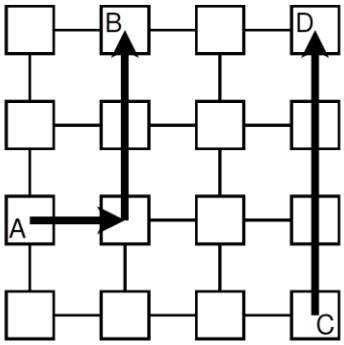

Fig.2 XY dimension order routing example

Fig.2 shows an network. Under XY routing, travel along the X axis occurs first. A packet traveling from A to B must first make one hop to the right before taking two hops upward. All packets and flits from A to B will take this route. A packet wants to go from C to D. In this case no travel in the X dimension is required. Packets from C can immediately travel upward along the Y axis until they reach D.

IV. RESULTS AND DISCUSSION

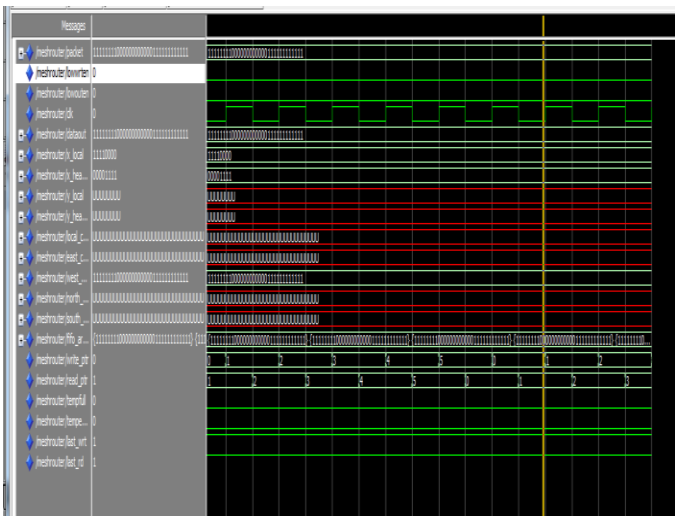

Fig.3 Output of Mesh topology router

The Fig.3 shows the simulation result of the Noc router. The router has five input and output ports. The input to the router is 64-bit data packet. The packet is routed through the mesh topology NoC router according to the source and destination addresses in the packet. If the value in x-coordinate of current node is greater than x-coordinate of the destination node then the packet goes to the east port, otherwise to west port. If the value in y-coordinate of current node is greater than y-coordinate of the destination node then the packet goes to the north port, otherwise to south port. When both values are equal then the packet goes to the local port.

TABLE. 1 SYNTHESIS RESULT OF NOC ROUTER

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	50	93120	0.05%
Number of Slice LUTs	64	46560	0.13%
Number of bonded IOBs	122	240	50%

V. CONCLUSION AND FUTURE WORK

A Network-on-Chip router design for Multicore-SoC platform is presented. The synthesis results shows that proposed architecture consumes only smaller area compared to previously reported architectures. The future work will explore the design of Wireless Network-on-Chip (WiNoC) router and to compare its performance with the traditional wired Noc router.

VI. REFERENCES

[1] L. Benini, G. De Micheli, "Networks on Chips: A New SoC Paradigm," IEEE Computer, pp.70-78, January 2002.
[2] A. Jantsch and H. Tenhunen, "Networks on Chip," Kluwer Academic Publishers, 2003.
[3] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," In Proc. of Design Automation Conference (DAC), pp. 684-689, June 2001.
[4] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Oberg, M. Millberg, and D. Lindqvist. "Network on chip: An architecture for billion transistor era," In Proc. of the IEEE Norchip Conf., pp. 120-124, November 2000.
[5] S. Kumar, A. Jantsch, M. Millberg, J. Oberg, J. Soininen, M. Forsell, K. Tiensyrj, and A.Hemani. "A network on chip architecture and design methodology," In Proc. Symposium on VLSI, pp. 117-124, April 2002.
[6] S. Vangal et al., "An 80-tile 1.28TFlops Network-on-Chip in 65nm CMOS," In Proceedings of ISSCC'07, pp. 98-100, 2007.
[7] Tilera. http://www.tilera.com, 2008.
[8] Semiconductor Association. The International Technology Roadmap for Semiconductors (ITRS).
[9] Semiconductor Association. Semiconductor Industry Association (SIA).
[10] G. D. Micheli, L. Benini, "Powering Networks on Chips: Energy-Efficient and Reliable Interconnect Design for SoCs," in Proceedings of the 14th international ieee symposium on Systems synthesis (ISSS '01), pp.33-38, 2001.
[11] F. Moraes , A. Mello, L. Moller, L. Ost, and N. Calazans, "A Low Area Overhead Packet switched Network on Chip: Architecture and Prototyping," in Proc. of International Conference on Very Large Scale Integration (VLSI-SoC), Germany, pp. 318-323, 2003.

[12] A. A. Chen, and J. H. Kim, "Planar-Adaptive routing: Low-cost adaptive networks for multiprocessors," In Proc. of 19th Ann Int'l Symp Computer Architecture, pp. 268–277, 1992.
[13] OCP International Partnership, Open Core Protocol Specification. 2.0 Release Candidate, 2003.
[14] Ankur Agarwal, Cyril Iskander, Ravi Shankar(2009) Survey of Network on Chip(NoC) Architectures & Contributions, Journal of Engineering, Computing and Architecture, Volume 3, Issue 1.
[15] D. Atienza, F. Angiolini, S. Murali, A. Pullini, L. Benini, G. De Micheli, "Network-On-Chip Design and Synthesis Outlook," Integration-The VLSI journal, vol. 41, no. 3, pp. 340-359,2008.
[16]L.Benini and D. Bertozzi, Network-on-chip architectures and design methods, IEEE proceedings DOI: 10.1049/ip-cdt:20045100.

