© 2016 IJSRSET | Volume 2 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Design and Implementation of Cross Bar NoC Architecture

Anita N Pachange, Dr. S S Kerur
Department of Electronics and Communication Engineering, SDM College of engineering and technology, Dharwad, Karnataka, India

ABSTRACT

The aim of this paper is to give briefing of the concept of network-on-chip (NoC). NoC is an approach to design the communication subsystem between IP cores in a System on Chip (SoC). NoCs can span synchronous and asynchronous clock domains or use unclocked asynchronous logic. This NoC brings an effective improvement over conventional busses and cross bar switches. The power requirement of the SoC is high where as it can be reduced by the NoC architecture. NoC is an developing paper in the field of VLSI. Since the use of emerging NoC architecture in VLSI it reduces the size of the architecture due to the reduced amount of buses and transmission lines. ." In a NoC system, modules such as processor cores, memories and specialized IP blocks exchange data using a network as a "public transportation" sub-system for the information traffic. The wires in the links of the NoC are shared by many signals

Keywords : Network-on-Chip (NoC), synchronous and asynchronous clock, processor cores, memories and specialized IP blocks.

I. INTRODUCTION

To meet the growing computation-intensive applications and the needs of low-power, high-performance systems, the number of computing resources in single-chip has enormously increased, because current VLSI technology can support such an extensive integration of transistors. By adding many computing resources such as CPU, DSP, specific IPs, etc to build a system in System-on-Chip, its interconnection between each other becomes another challenging issue. The number of processor, memory and accelerator cores on systems on chip is rapidly increasing to support evolving standards and new applications. Computation and communication complexity is skyrocketing, and scalability-centric design paradigms are critically needed. NoCs have emerged as the best alternative to provide high performance in communication for futures SoCs with dozens of cores integrated on single silicon die. NoC is efficient communication architecture for SoC. It allows the integration of a huge number of computational blocks in only one integrated circuit. NoC allows the reuse of blocks, which provide a high degree of parallelism and it may present high improvement in performance. The increasing need of NoC has been driven by the demand for high bandwidth in communication. According to Moore's law, transistor integration doubles in approximately eighteen months, but as the delay of the global wires has increased; physical connection in an IC (Integrated Circuit) is the performance bottleneck. NoC can provide short interconnections, which can reduce the wire delays.

The traditional methods (end-to-end, bus-based and others) are not a good solution to integrate many IP cores in SoCs, because they do not scale well as more components are added to the system. Ad hoc solutions that use multiple buses, crossbar and dedicated wiring can achieve near optimal performance for a specific application but they are hard to reuse and have a high development and verification cost. The use of NoC to replace the traditional methods of interconnection has advantages for performance and modularity. With the use of a NoC, it was observed that electrical properties can be optimized, power consumption can be decreased by 10 times and speed increased by 3 times when compared to bus-based interconnections.

NoCs are in charge of interfacing IP cores to the communication infrastructure and the overall system.

They represent critical points in the design of a tracing the routs in NoC. In fact, the routing in NoC can cause errors that, directly affecting the correct transmission of data and control information, could be extremely hard to detect and recover the original data which can degrades the latency , therefore appropriate routing algorithms are used to transmit and receive the data without any losses with accurate routing. Which can increases the latency, throughput and reliability of the cross bar NoC architecture.

II. METHODS AND MATERIAL

1. Crossbar NOC Architecture

The crossbar (also called bus matrix) solution is known as one of the most effective communication architectures for modern high-performance embedded systems. To make it even more effective, several topology synthesis methods have been proposed. They mostly generate a crossbar network in a cascaded fashion under the assumption that each crossbar switch is fully-connected (i.e. each input has a connection to every output). This assumption often limits optimizing the area efficiency and/or performance of the network, due to the unnecessary connections inside the crossbar switches. Some existing methods marginally improve their synthesis results by eliminating the unnecessary connections after the synthesis step. Such post-processing approaches make sense, since considering partially-connected crossbar switches earlier in the synthesis flow can greatly increase the optimal topology search space, thereby increasing the runtime. However, the result from these post-processing techniques is typically far inferior to that from the exhaustive search. In this work, we tackle such limitations of previous methods by introducing a heuristic method based on iterative switch merging.

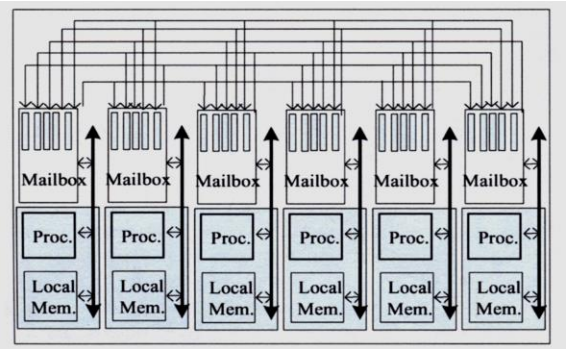

Figure 1: Crossbar NoC architecture

Figure1 shows the crossbar NOC architecture. The computation nodes are shown in the shadow part, which are composed of processing unit (Proc.) and local memory (Local Memory). In this architecture, there are two kinds of communication channels. One is the local bus between the processor and local memory, the other is the full connection crossbar network on chip architecture [1]. Through a receiver/ transmitter called mailbox and links between them, the computation nodes can communicate with each other

The major part of the communication backbone is the mailbox, and it consists of data-path, write control logic and read control logic modules. The data-path is major part of messages storing, and the other two modules are responsible for controlling of the data-path with message transfer. Two procedures of write and read are defined as follows.

The write is defines from the transfer initiators point of view, and the message data should be stored into the mailbox before a transfer starts from an initiator. That is to say, the mailbox belongs to initiators and they have full control to write their mailboxes.

The Read is defined as the message reading out of the FIFOs after one initiator has stored the messages in the data-path of the mailbox. It will not be activated until the initiator sends a request to the destination that the message is ready to be received. If the request is acknowledged, message data will be read out of the mailbox to the destination node's local memory. When this read procedure is finished, the memory space in data-path is automatically released [1].

2. Implementation

A. Designing of round robin algorithm (without priority)

Round-robin arbiter operates on the principle that a request which was just served should have the lowest priority on the next round of arbitration. Depending upon the control logic arbiter generates select lines for multiplexer based crossbar and read or write signal for FIFO buffer. Contention resolution is an important task of arbiter. If two or more resources are sending data to one destination at same time then there is contention for

destination. This contention can be resolved by assigning priorities to the resources based on different scheduling algorithms.

The router block in each processor receives data from neighboring processors (east, north, west, and south) and then sends data to the processor core or neighboring processors. Since communication links are not always available due to slow data processing speeds or link congestion, buffers are inserted at each input edge. The communication logic includes four buffers, multiplexer's, and there is some control logic to support the communication flow control.

B. Designing of round robin algoritm (with priority):

In Round Robin Arbitration scheme when all input port are requesting for same output port the Round Robin Arbiter checks the destinations address of each port. If the destination address is same then Round Robin Arbiter use arbitration algorithm. In this algorithm Round Robin Arbiter gives the Priority to input request [8]. Accordingly grant signal is generated and source packet is transmitted to destination through crossbar. In a NoC Router if more than one input is request for the same output the arbiter is used. Here we are explaining the behavior of Round Robin Arbiter in NoC Architecture. The Round Robin Arbiter operates on the principle that a request which was just served should have a lowest priority on the next round of arbitration. It keeps the updated status of all the ports and knows which ports are free and which ports are communicating with each other. Packets with the same priority and destined for the same output port are scheduled with a Round-Robin Arbiter. In the Round Robin Arbitration scheme, this may be granted that all input request are treated fairly. Hence they had proposed arbiter is suitable for NoC design

C. Designing of longest priority algorithm:

In this algorithm the request which is waiting from a longer time in the ready queue will be served first. Instead of assigning the priority manually as in the round robin algorithm based on the longest wait the requests will be served which increases the latency , efficiency and it provides the services for all the jobs which are waiting in a queue from longer time.

D. Designing of wormhole algorithm:

In wormhole switching, packets are divided into small and equal sized flits (flow control digit or flow control unit). A first flit of a packet is routed similarly as packets in the virtual cut-through routing. After first flit, the route is reserved to route the remaining flits of the packet. This route is called wormhole. Wormhole mode requires less memory than the two other modes because only one flit has to be stored at once. Also the latency is smaller and a risk of deadlock is larger. The risk can be reduced by multiplexing several virtual ports to one physical port, so the possibility of traffic congestion and blocking decreases. The local information of the router is considered because non-local information may increase the design complexity.

E. Implemtaion of algorithm in 9 node NoC archritecture

As shown in the Figure2 resources are placed on the slots formed by the switches. The maximal resource area is defined by the maximal synchronous area of a technology. The resources perform their own computational functionalities, and are equipped with Resource-Network-Interfaces (RNIs) to communicate with each other by routing packets instead of dedicated wires. Although Network-on-Chip is considered as an inevitable solution for future SoCs [4], the design challenges with NoCs rise from the architecture, design methodology, programming model etc. All these difficulties stem from the stringent SoC evaluation criteria: performance, power and cost, as well as time-to-market (productivity) and time-in-market (programmability) pressures.

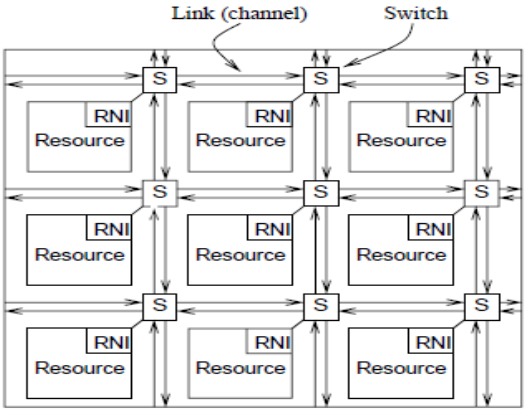

Figure 2: core cross bar NoC

The data-path is major part of messages storing, and the other two modules are responsible for controlling of the data-path with message transfer. Two procedures of write and read are defined as follows. The write is defines from the transfer initiators point of view, and the message data should be stored into the mailbox before a transfer starts from an initiator. That is to say, the mailbox belongs to initiators and they have full control to write their mailboxes. The Read is defined as the message reading out of the FIFOs after one initiator has stored the messages in the data-path of the mailbox will not be activated until the initiator sends a request to the destination that the message is ready to be received. If the request is acknowledged, message data will be read out of the mailbox to the destination node's local memory. When this read procedure is finished, the memory space in data-path is automatically released. FIFO architecture is used in the data-path module to store messages, which greatly simplifies the design complexity.

III. RESULTS AND DISCUSSION

A. Simulation results for round robin algorithm

Figure 3 shows the simulation result for round robin algorithm it has four input data as north_in, east_in, west_in, sounth_in and four output paths as north_out, east_out, west_out, sounth_out. Based on the request and busy signal the input data will be served to output path in a cyclic way.

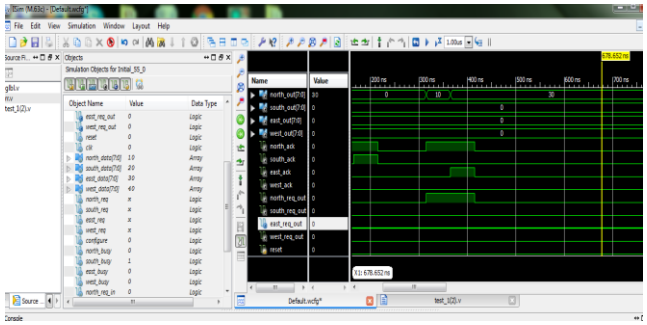

Figure 3. Simulation results for round robin algorithm

B. Simulation results for Longest priority:

In this algorithm the counters will be used to check the waiting time of the request. The request which will be waiting from longer time that request will be served first as shown in figure4.

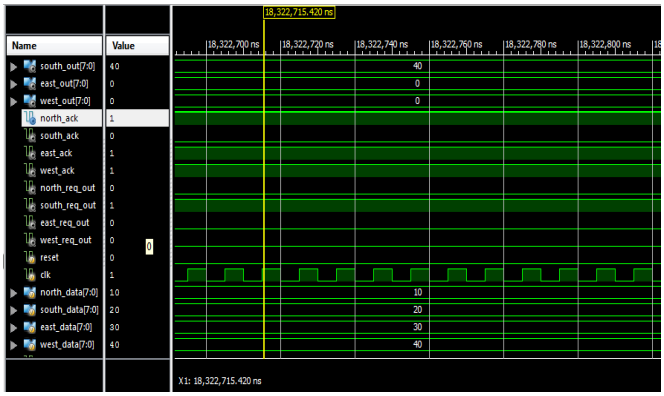

Figure 4. Simulation results for longest priority

C. Simulation results for Wormhole algorithm

In the wormhole flow control which is shown in the figure5, each packet is broken into small pieces called FLITs (flow control digits).Commonly, the first flits, called the header flits, holds information about this packet's route (for example, the destination address) and sets up the routing behavior for all subsequent flits associated with the packet. The name "wormhole" plays on the way packets are sent over the links: the address is so short that it can be translated before the message itself arrives. This allows the router to quickly set up the routing of the actual message.

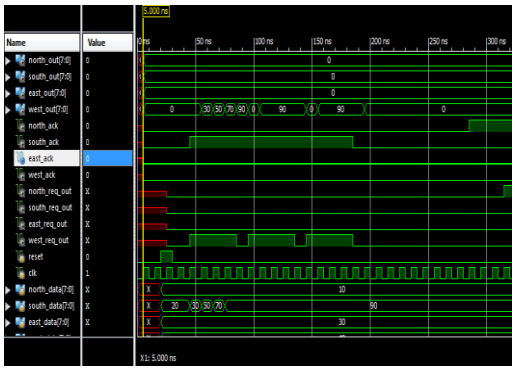

Figure 5. Simulation results for wormhole algorithm

D. Simulation results for 9 core cross bar for NoC:

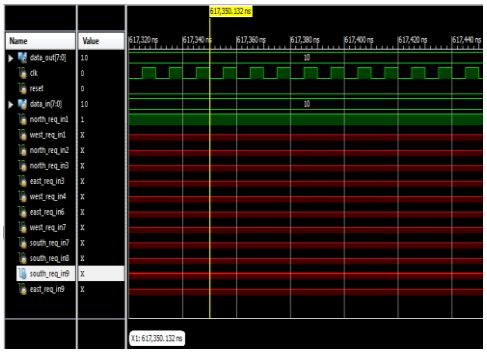

Figure 6. Simulation results for 9 core cross bar NoC

In this process all node / algorithms are processed with 9 nodes and establishment of path between one node to another node is achieved. The routing algorithm has to find the shortest and faster path to send the data as per request

IV. CONCLUSION

In theory, the idea unitary speedup value is 1, but it is hard to achieve for the sequential of a program affect this greatly. This paper evaluated the scalability of an in-house developed crossbar NoC in terms of area and performance on FPGA prototypes. The experiments show that the crossbar NoC scales well in performance, but it has too much area penalty as the processing number increases. Furthermore, the communication module's area overhead can be reduced a great deal with careful modification.

V. REFERENCES

[1] Du Gaoming1 Zhang Duoli1, Performance Evaluation of FPGA based Crossbar NoC Architecture, Institute of VLSI Design, Hefei University of Technology, Hefei 230009, P. R. China

[2] David Bafumba-Lokilo, Generic Crossbar Network on Chip for FPGA MPSoCs, Département de Génie Électrique, École Polytechnique de Montréal, CANADA

[3] Leandro Fiorin, Fault-tolerant Network Interfaces for Networks-on-Chip IEEE transactions on dependable and secure computing

[4] Eduard Fernandez-Alonso1, survey of NoC and Programming Models Proposals for MPSoC, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3, March 2012CAIAC, Campus UAB, 08193, Bellaterra, Spain

[5] https://en.wikipedia.org/wiki/Qualcomm_Snapdragon

[6] Using Wormhole Switching for Networks on Chip: Feasibility Analysis and Microarchitecture Adaptation by Zhonghai Lu Stockholm 2005

[7] Partial Connection-aware Topology Synthesis for On-Chip Cascaded Crossbar Network Minje Jun, Member, IEEE, Deumji Woo, and Eui-Young Chung, Member IEEE, Digital Object Indentifier 10.1109/TC.2010.211

[8] A Review on: Priority Based Arbiter for NoC Router International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue 1st International Conference on Advent Trends in Engineering, Science and Technology "ICATEST 2015", 08 March 2015