AREA-EFFICIENT DESIGN OF SCHEDULER FOR ROUTING NODE OF NETWORK-ON-CHIP

Rehan Maroofi,[1] V. N. Nitnaware,[2] and Dr. S. S. Limaye[3]

[1]Department of Electronics, Ramdeobaba Kamla Nehru College of Engg, Nagpur, India
maroofir@rknec.edu
[2]Department of EDT, Ramdeobaba Kamla Nehru College of Engg, Nagpur, India
nitnawarevn@rknec.edu
[3]Jhulelal Institute of Technology, Nagpur, India
ss_limaye@gmail.com

ABSTRACT

Traditional System-on-Chip (SoC) design employed shared buses for data transfer among various subsystems. As SoCs become more complex involving a larger number of subsystems, traditional bus-based architecture is giving way to a new paradigm for on-chip communication. This paradigm is called Network-on-Chip (NoC). A communication network of point-to-point links and routing switches is used to facilitate communication between subsystems. The routing switch proposed in this paper consists of four components, namely the input ports, output ports, switching fabric, and scheduler. The scheduler design is described in this paper. The function of the scheduler is to arbitrate between requests by data packets for use of the switching fabric. The scheduler uses an improved round robin based arbitration algorithm. Due to the symmetric structure of the scheduler, an area-efficient design is proposed by folding the scheduler onto itself, thereby reducing its area roughly by 50%.

KEYWORDS

Network-on-Chip, System-on-Chip, On-chip routing switch, Scheduler, iSLIP, Synthesis.

1. INTRODUCTION

The current trend in technology has allowed an ever increasing number of circuits to be placed on a wafer of silicon. This has lead to the emergence of complex Systems-on-Chip (SoC) where entire systems consisting of analog as well as digital components are being implemented on a single chip. Traditionally, shared busses were used for communication between the different components in an SoC [1]. In a shared bus architecture, a common communication link is shared between components is a time-division fashion, resulting in a communication latency that increases with the number of components sharing the bus.

A switched interconnect providing more than one parallel point to point link is a more efficient option offering higher performance [2]. A natural progression of the switched interconnect is to employ a network design based on a number of small switching components inside an SoC [3]. Packet switching can be used in SoCs with an arbitrarily large number of components (resources) [4]. ASIC [5] and FPGA [6] implementations of packet switched networks on chip have been demonstrated to be viable solutions for the SoC interconnect problem. Hence, as stated in [7], packet switched NoCs are the clear solution to the problem of complex SoC interconnect design.

The key research problems in the design of NoCs include but are not limited to topology, channel width, buffer size, floorplan, routing, switching, scheduling, and IP mapping [8]. Additionally,

DOI : 10.5121/vlsic.2011.2309

[9] lists research issues to be application modeling and optimization, NoC communication architecture analysis and optimization, NoC communication architecture evaluation, and NoC design validation and synthesis. In this regard, the current work is related to NoC communication architecture.

The components of the NoC include the network adapter, the routing node, and the network links [13]. The routing node in turn consists of four major components: the input ports, the scheduler, the crossbar switch, and the output ports.

Packet switching is the predominant mode of routing in NoCs [10]. X-Y routing is the simplest mechanism [9], while adaptive routing provides better throughput and fault tolerance by allowing alternate paths depending on congestion and runtime faults [11]. With respect to signal integrity in deep submicron technology, [12] have proposed a coding scheme to reduce crosstalk in SoCs.

The remaining sections of the paper as divided as follows. Section 2 discusses the implementation details of the proposed design, beginning with an overview of the archictecture of an NoC. The components of an NoC are described along with their function. The existing scheduler architecture and proposed scheduler architecture are presented. The scheduler building blocks are described, showing the nature of modifications in the proposed design. Finally, section 3 contains a summary of the results and conclusions of this work.

2. IMPLEMENTATION

2.1. Network-on-chip architecture

As more complex SoCs begin to emerge, the task of communication between the subsystems of an SoC cannot be adequately handled by bus based communication architectures. NoCs are proposed to be a viable alternative to bus based architectures for communication within SoCs. A generic 2D mesh NoC architecture is shown in figure 1 [4].

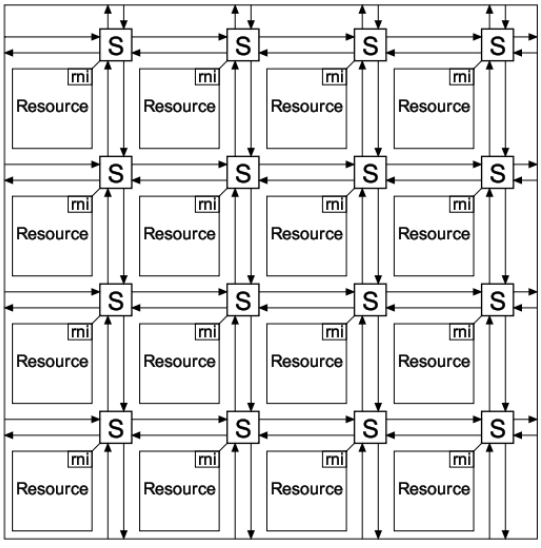


Fig. 1: SoC based on NoC

The SoC consists of a number of resources. These resources communicate with each other using an NoC. The NoC consists of the switches and point-to-point links. The internal structure of the NoC switch is given in figure 2 [10].

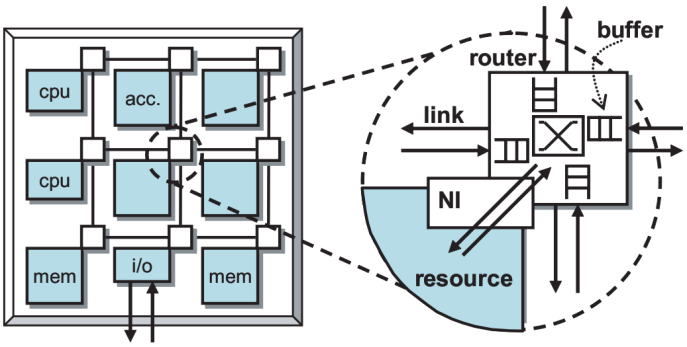


Fig. 2: NoC Switch Schematic

Each switch consists of four distinct components. A set of input blocks are connected to incoming packet lines. These input blocks contain buffers to queue the incoming packets so that they can be stored until they are ready to be transferred over the shared crossbar matrix. A set of output blocks are connected to the outgoing packet lines. No buffers are required in the output blocks as there is no possibility of conflict due to the absence of any shared resources at the outputs. The central crossbar matrix provides a direct link between each pair of input and output blocks. Finally, a scheduler is required to perform arbitration in to enable fair access to the common crossbar fabric for all incoming packets.

2.2. Scheduler Architecture

The function of the scheduler is to arbitrate between requests from input blocks to output blocks. The arbitration scheme is based on a maximal size approach proposed in reference [14]. It is a variant of round robin matching, which is shown to prevent starvation under uniform traffic [14]. A scheduling decision is arrived at in three steps: (a) Requests are sent from the input blocks to the output grant generation arbiters. (b) Grant signals are generated by the output grant arbiters and sent to input accept arbiters. (c) Accept signals are generated by the input accept arbiters and these signals represent the final scheduling decision. This decision is sent back to the input blocks as well as the crossbar switch to enable transfer of packets from the input blocks to the output blocks. The top level diagram of the sheduler is shown in figure 3.

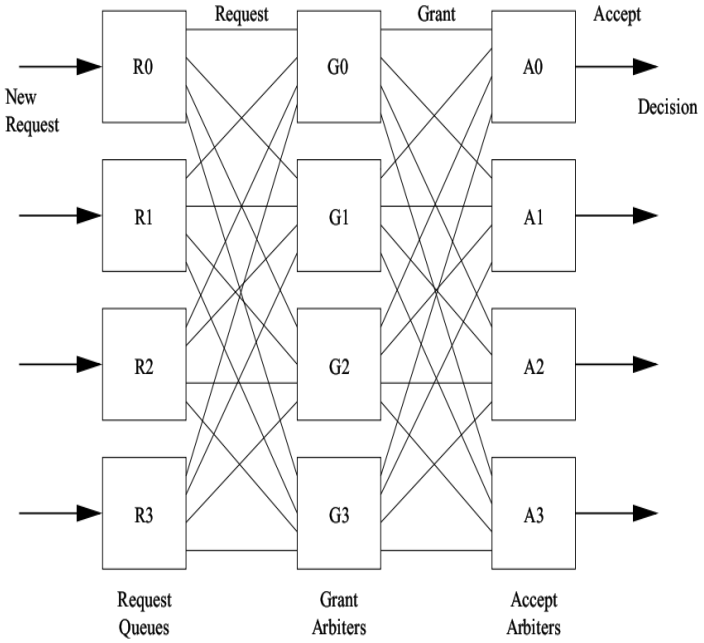


Figure 3: Scheduler Block Diagram

Each arbiter consists of a programmable priority encoder and a pointer to hold the value of the previously granted and accepted request. Figure 4 depicts the arbiter schematic. The programmable priority encoder generates a grant signal in response to a request signal based on a simple round robin scheme. Figure 5 depicts the programmable priority encoder schematic. The programmable priority encoder is realized using a hybrid design composed of two simple priority encoders. The programmable priority encoder design is based on thermometer encoding [15]. This design occupies less area compared to more classic programmable priority encoder designs.

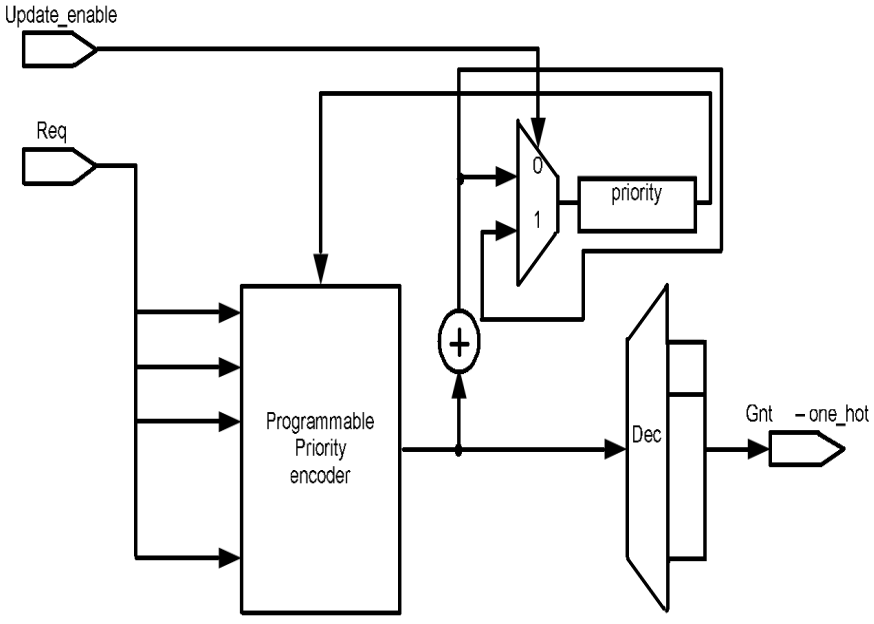


Figure 4: Arbiter Schematic

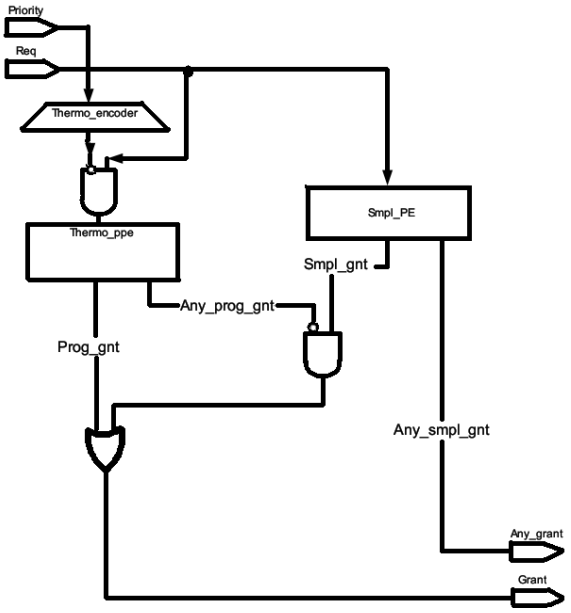


Figure 5: Programmable Priority Encoder

The scheduler was modified using a folding approach [16] due to the regular structure and placement of the arbiters. The modified scheduler is depicted in figure 6. Each arbiter in the modified scheduler now has to generate both grant and accept signals in a time multiplexed fashion. The arbiter is modified to hold both grant and accept pointers for successive time slots. The modified arbiter is depicted in figure 7. The programmable priority encoder of the modified arbiter remains unchanged from the single pointer arbiter design.

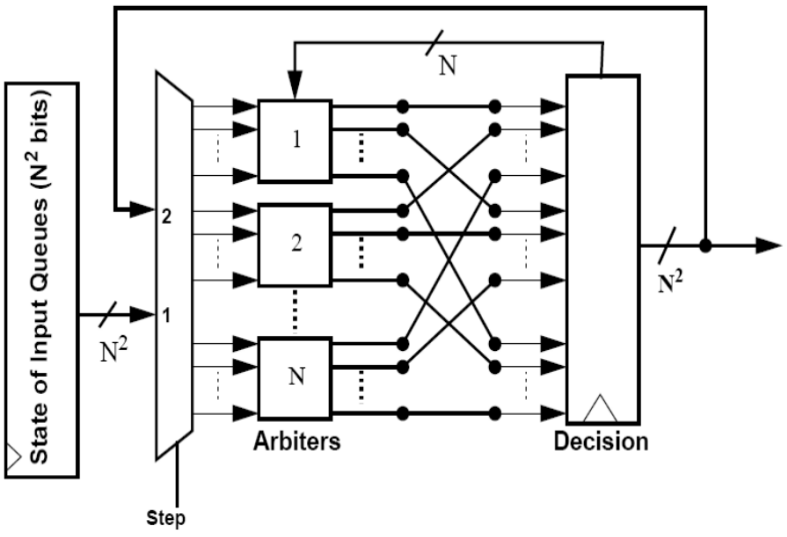


Figure 6: Modified Scheduler

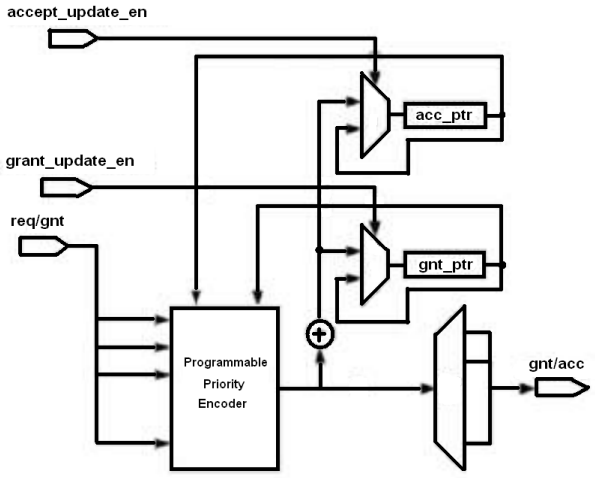


Figure 7: Modified Arbiter

3. RESULTS AND CONCLUSIONS

3.1 Results

For the original scheduler and modified scheduler, the RTL was synthesized using the Synopsys 90 nm Education Design Kit. The results of synthesis are given in table 1.

Parameter	Original Scheduler	Modified Scheduler
Area	18366 sq um	12342 sq um
Power	3.28 mW	3.86 mW
Operating Frequency	250 MHz	250 Mhz

3.2 Conclusion

Network-on-Chips (NoC) is an emerging solution to the communication issue of future generation Systems-on-Chip (SoC). Silicon area in onchip networks is of higher concern compared to offchip networks. The three major components of an NoC architecture are the communication links, routing nodes, and network interfaces. The routing node in turn is composed of input blocks, output blocks, a switching fabric, and a scheduler. In this paper we have targeted the scheduler design for area optimization. The folding concept has been used to improve the area efficiency of the scheduler. Due to the symmetric structure of the scheduler, the scheduler is folded onto itself to reduce area. Implementation of the design has shown that area requirement for the scheduler is improved by approximately 30%. However, the folding results in an additional timing requirement of two extra clock cycles per scheduling decision when compared to the timing of the original design.

REFERENCES

[1] M. Mitic, and M. Stojcev, "An Overview of On-Chip Busses", Facta Universitatis (Nis) Ser: Elec. Energ., Vol 19, No, 3, December 2006, pp. 405-428.

[2] D. Wiklund, and D. Liu, "Switched Interconnect for Systems-on-a-Chip Designs," in Proceedings of the IP 2000 System-on-Chip for a Connected World Conference, 2000, pp. 187-192.

[3] A. Brinkmann et al, "On-Chip Interconnects for Next Generation Systems-on-Chip," in Proceedings of the 15th Annual IEEE International ASIC/SOC Conference, 2002, pp. 212-215.

[4] S. Kumar et al, "A Network on Chip Architecture and Design Methodology," in Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'02), 0-7695-1486-3/02, 2002.

[5] T. Henriksson, D. Wiklund, and D. Liu, "VLSI Implementation of a Switch for On-Chip Networks," in Proceedings of the International Workshop on Design and Diagnostics of Electronic Circuits and Systems, Poznan, Poland, 2003.

[6] R. Holsmark, A. Johansson, and S. Kumar, "On Connecting Cores to Packet Switched On-Chip Networks: A Case Study with Microblaze Processor Cores," in Proceedings of the 7th IEEE Workshop DDECS-04, Slovakia, April 2004.

[7] S. Furber, "Future Trends in SoC Interconnect," in Proceedings of the International Symposium on System-on-Chip, 2005, pp. 183-186.

[8] U. Ogras, J. Hu, and R. Marculescu, "Key Research Problems in NoC Design: A Holisitic Perspective," in Proceedings of the CODES, ISSS, 2005, pp. 69-74.

[9] R. Marculescu, U. Ogras, L. Peh, N. Jerger, and Y. Hoskote, "Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 28, No. 1, January 2009.

[10] E. Salminen et al, "Survey of Network-on-Chip Proposals", White Paper, OCP-IP, 2008.

[11] W. Dally, and B. Towles, "Principles and Practices of Interconnection Networks," Morgan Kaufmann, 2004.

[12] J. Rao, and A. Tilak, "A Bus Encoding Scheme to Reduce Crosstalk Noise Effects in Systems-on-Chip," International Journal of VLSI and Communication, Vol. 2, No. 2, June 2011.

[13] T. Bjerregaard, and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip," ACM Computing Surveys, Vol. 38, March 2006.

[14] N. McKeown, "Scheduling Algorithms for Input Queued Switches", Ph.D. Thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1995.

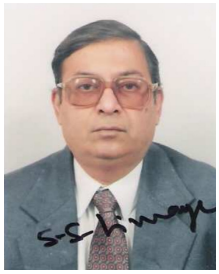
[15] P. Gupta and N. McKeown, "Designing and Implementing a Fast Crossbar Scheduler", Micro, IEEE, Vol. 19, No. 1, pp. 20-28, Jan/Feb 1999.

[16] K. Parhi, "VLSI Design Signal Processing Systems: Design and Implementation", John Wiley, 1999.

Authors

Rehan Maroofi is a post-graduate student in the Department of Electronics at Ramdeobaba Kamla Nehru College of Engineering in Nagpur, India, pursuing his M. Tech in VLSI. He has a B.E. in Electronic Design Technology from Nagpur University.

Prof. Vilas Nitnaware is pursuing PhD from R.S.T.M.Nagpur University, India, in the field of "SOC Interconnect". He is an Assistant Professor and working as a Coordinator for the Department of Electronic Design Technology for the last 10 years. He has been associated with microprocessors and microcontrollers related subjects right from the start of his career. His Masters was in the field of VLSI. He has undergone training on Synopsys "Design Compiler". He has developed some embedded system products and also delivered training on microcontrollers and their applications at various Institutes.



Dr. Shyam S. Limaye received his PhD from Nagpur University in the faculty of Electronics Engineering. Currently he is working with JIT as a Principal. He has 33 years of experience in teaching as well as in industry. He also carried out a number of consultancy projects at DCM Data products, Delhi, PSI Data Systems Bangalore, and Zen and Art New York. His areas of specialization include digital signal processing, VLSI design, LDPC codes, and CORDIC algorithms. He is a recognized supervisor and guide to PhD scholars at Nagpur University.