Area Efficient Design of Routing Node for Network-on-Chip

Rehan Maroofi[1], Vilas Nitnaware[2] and Shyam Limaye[3]

[1] Department of Electronics Engineering, Ramdeobaba College of Engineering, Nagpur, Maharashtra 440013, India

[2] Department of Electronic Design Technology, Ramdeobaba College of Engineering, Nagpur, Maharashtra 440013, India

[3] Principal, Jhulelal Institute of Technology, Nagpur, Maharashtra, India

Abstract

Network-on-Chip (NoC) is a paradigm proposed to satisfy the communication demands of future Systems-on-Chip (SoC). The main components of an NoC are the network adapters, routing nodes, and network interconnect links. Reducing area and power consumption has higher priority in the case of on-chip networks compared to conventional off-chip networks. This paper presents an area efficient design for the routing node component of an NoC. The area efficiency is obtained by applying the concept of a pipelined design as well as the use of custom IP (intellectual property) cores.

Keywords: *Network-on-Chip (NoC), router, crossbar switch, virtual output queuing, ASIC (application specific integrated circuit) implementation.*

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]. The components of the NoC include the network adapter, the routing node, and the network links [9]. The routing node in turn consists of four major components: the input ports, the scheduler, the crossbar switch, and the output ports.

2. Implementation

A block diagram of the routing node is given in figure 1. The input block accepts incoming packets and stores them in virtual output queues [10], implemented as linked lists in hardware. The scheduler processes these packet queues to determine a suitable match between input blocks and output blocks over the crossbar switch. The algorithm used by the scheduler is a modified round robin arbitration algorithm, which is a maximal size matching algorithm that provides good throughput under uniform traffic [11]. The crossbar switch is a synthesizable mux based design [12]. The input and output blocks make use of a credit-debit flow control mechanism to transmit data to and from the SoC resources [13].

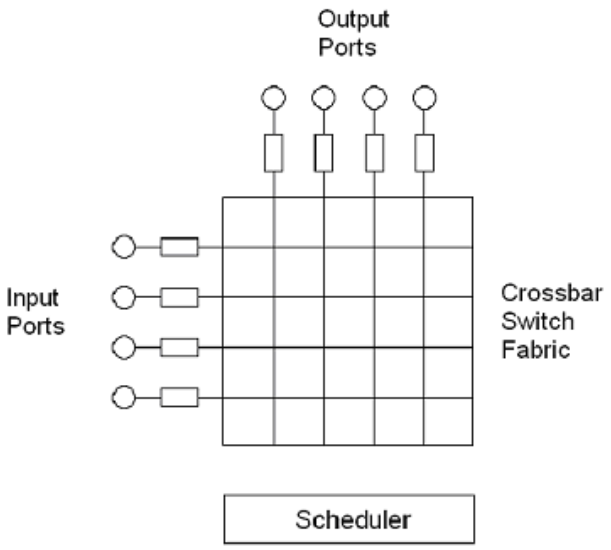

Figure 1: Routing Node Block Diagram

A more detailed diagram of the routing node is given in figure 2, showing the input blocks, the scheduler, the crossbar switch, and the output blocks.

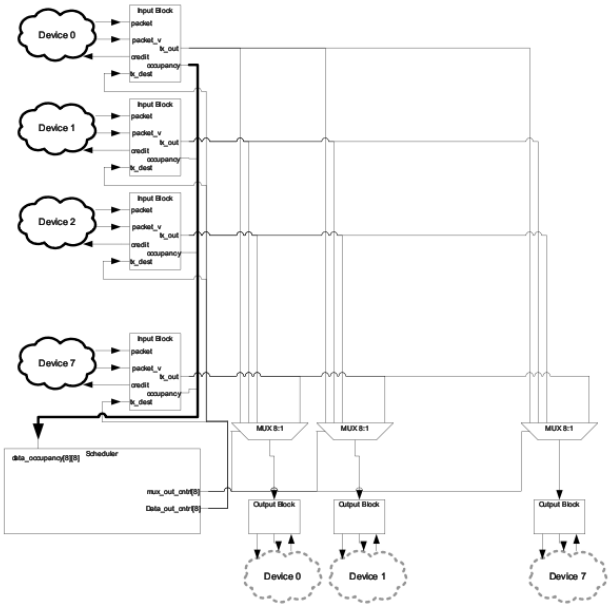

Figure 2: Detailed Diagram of Routing Node

The input block consists of five components. They are the packet array, the linked list array, the destination head array, the destination tail array, the freelist fifo, and a shift register. A block diagram of the input block is given in figure 3. The packet array is used to store incoming packets. The packets are stored in virtual output queues [10]. The virtual output queues are maintained in linked lists. The linked list functionality is implemented using the linked list array, the destination head array, the destination tail array, and the freelist fifo. The shift register is used to hold the outgoing packet for transmission over the crossbar. The input block memory elements were first implemented using synthesizable flip flops available in the SAED90nm standard digital library. Next, to optimize the design for area efficiency, the memory elements were implemented using custom SRAM based on the SAED90nm process. The SRAM based design uses fewer transistors, but the capacitance on each bit line is higher compared to the flip flop implementation [14].

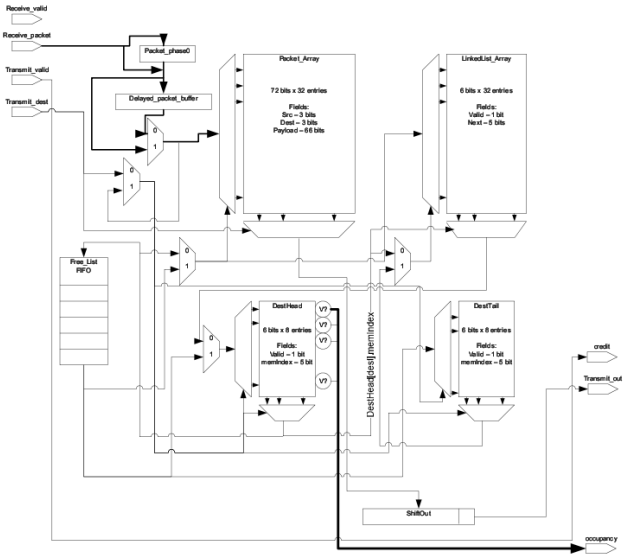

Figure 3: Input Block

The scheduler (figure 4) is implemented using a modified form of the round robin arbitration scheme called iSLIP. This was chosen because it offered a balance between performance and ease of implementation [11].

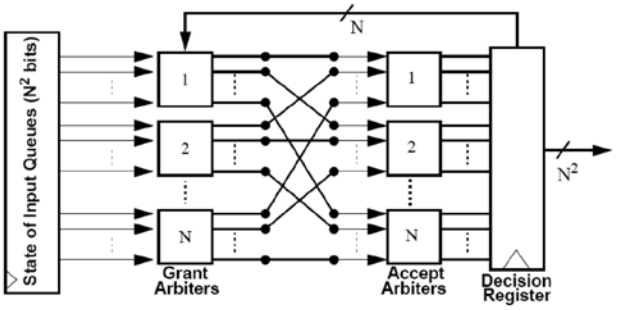

Figure 4: iSLIP Scheduler

The scheduler was modified using folding [15] to reduce the number of arbiters required to half the original number.

The block diagram of the folded scheduler is shown in figure 5.

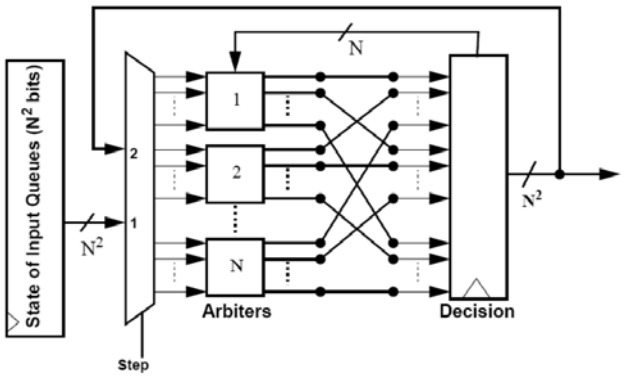

Figure 5: Folded Scheduler

3. Results

The implementation results for the input block are presented in table 1, while the implementation results for the scheduler are presented in table 2. The input block results show the saving in area as a result of using compact custom SRAM in place of flip flops for storage. Similarly, the scheduler results show the reduction in area due to folding. However, the number of clock cycles required to arrive at a scheduling decision increase due to the fact that each arbiter performs two functions in a time division multiplexed fashion.

Table 1: Input Block Results

Parameter	Flip-Flop Design	SRAM Design
Area	136438 sq units	42409 sq units
Speed	50 MHz	200 MHz
Power	2.985 mW	1.436 mW

Table 2: Scheduler Results

Parameter	Original Scheduler	Folded Scheduler
Area	15494 sq units	12381 sq units
Speed	200 MHz	200 MHz
Power	750 uW	1085 uW

4. Conclusions

An area efficient implementation of a routing node for an NoC is proposed. Of the four components of the routing node, the input block and the scheduler have been modified to save area requirements. The use of custom SRAM in place of synthesizable flip flops in the input block has resulted in a saving of over 50% of the silicon area. At the same time, timing of the input block improved from 50 MHz to 200 MHz maximum operating frequency. Power saving is also approximately 50%. The disadvantage is longer implementation time and higher effort than a standard digital ASIC flow due to the presence of non-standard custom SRAM.

Folding of the scheduler resulted in an area saving of approximately 20%. However, more clock cycles were required per matching. This is because the same set of arbiters performed two separate functions in a time division multiplexed fashion. Power consumption also showed a slight increase due to the greater number of clock cycles required for each matching in the case of the folded scheduler.

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] T. Bjerregaard, and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip," ACM Computing Surveys, Vol. 38, March 2006.
[10] D. Divakaran et al, "A Survey on Core Switch Designs and Algorithms," Research Report RR-6942, INRIA, 2009.
[11] N. McKeown, "Scheduling Algorithms for Input Queued Cell Switches", Ph.D. thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, USA, 1995.
[12] E. Coen-Alfaro, "Crossbar Architectures for VLSI Systems: A Comparative Study," M.Sc. thesis, College of Graduate Studies, University of Idaho, USA, 2004.
[13] B. Attia et al, "A Modular Network Interface Adapter Design for OCP Compatible NoCs", International Journal of Network and Computer Security, Vol. 1, No. 2, Novermber 2009, pp. 101-109.
[14] S. Lee, "Implementation of an NoC," International Symposium on System-on-Chip (ISSOC2005), 2005.
[15] K. Parhi, "VLSI Digital Signal Processing Systems: Design and Implementation," John Wiley, 1999.

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, India.

Vilas Nitnaware is pursuing his Ph.D. from Nagpur University, India, in the field of "SoC Interconnect". He has been an assistant professor and coordinator at the Department of Electronic Design Technology for the last 10 years. His current interests inlcude digital logic design and em bedded systems. He earned his M.Tech. degree in the field of VLSI. He has developed embedded system products and conducted training sessions on microcontrollers and applications at various Institutes.

Shyam Limaye received his Ph.D. from Nagpur University, India, in the faculty of Electronics Engineering. C urrently he i s the Principal of Jhulelal Institute of Technology, Nagpur, India. He has 33 years of teaching experience. He has undertaken 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 for Ph.D. scholars at Nagpur University, India.