Power Comparison of an Asynchronous and Synchronous Network on Chip Router

Pooria M.Yaghini, Ashkan Eghbal, S.A. Asghari, and H. Pedram
Amirkabir University of Technology
{pooria.yaghini, ashkanxy, seyyed_asghari , and pedram}@aut.ac.ir

Abstract

This paper presents an asynchronous and a synchronous NoC router architecture. The asynchronous scheme is implemented by the help of CSP-Verilog language and the synchronous one is designed employing VHDL language. Their designs are similar except the extra links which are in charge of handshaking processes in asynchronous architecture. According to the experimental results the transition counts of buffer, and switch components in synchronous router are almost 82% and 60% of asynchronous one, respectively. On the other hand, the transition counting of routing unit in asynchronous NoC router is nearly 73% of synchronous one. Power consumption of them are evaluated according to the obtained transition counting. Based on the comparison the power consumption of buffer and switch components are almost same due to their similar structure. However, the power consumption of routing unit component in asynchronous design is lower than synchronous one.

1. Introduction

Shrinking feature sizes in VLSI technologies accompanies the advantage of employing billions of transistors on a single chip. Emerging paradigms utilizing these vast on-chip resources are system-on-chip (SoC) and multi-core architectures. In the future a SoC architecture is expected to consist of tens of computing cores operating in the multi-gigahertz range. The various cores would require a communication medium that can support simultaneous high bandwidth data transfers with low latencies. The Network-on-Chip (NoC) infrastructure has been considered to accommodate better modularity, scalability, and higher bandwidth compared to bus-based infrastructure.

Implementing synchronous NoC router has been pondered on most of previous efforts. They have been deigned to be utilized in 2D mesh, torus, fat tree, and hierarchical topologies [1][2][3]. On the other hand, asynchronous router designs are not considered as much as synchronous ones. Synchronous routers and supporting asynchronous interconnect are presented in [4][5].

The synchronous NoC router implementation contains major problems such modularity and design reuse, Electro Magnetic Interference, worst case performance, clock power performance and clock skew [6][7]. However, asynchronous schemes are more complex and include some extra controlling links and components. They provide more scalability and lower power consumption features. They also can be utilized as a separated blocks with multi-clock mechanism [7][8][9][10] or clock less structure [6][7].

This article presents an asynchronous and a synchronous NOC router separately to compare their transition counts and consequently their power consumptions. The selected signals for comparison in both designs have similar functionality. The synchronous router is implemented in VHDL language and the asynchronous one is simulated in CSP-Verilog (communicating sequential processes) language. CSP-Verilog is a well-known language for description of concurrent systems which is accepted as a good description language for asynchronous systems [11]. Both of the architectures contain three main components of a paradigm NoC router. The main difference between them is their components correlations. The synchronous router applies a global clock to synchronize different components with each other. However, the asynchronous one utilize handshake signaling to control the interaction of different components. The comparison of asynchronous and synchronous NoC router power consumption in this experiment is based on transition count methods.

Power estimation can be done at various levels of abstraction. There is a trade-off between the accuracy of the power estimation and the level of details at which the circuit is analyzed or simulated. Simulation time increases by going into more details in the level of abstraction. Process of analyzing the power in the transistor level is extensive and the simulation time is very long. The power consumed in CMOS digital circuits, in the order of importance consists of three main parts: *Dynamic* power *short circuit* power and *static* power. Approximately 90% of power is consumed as dynamic and short circuit currents. So, neglecting the static power the power consumption of a circuit with no activity is almost

Proceedings of the 14th International CSI Computer Conference (CSICC'09)
978-1-4244-4262-1/09/$25.00 ©2009 IEEE

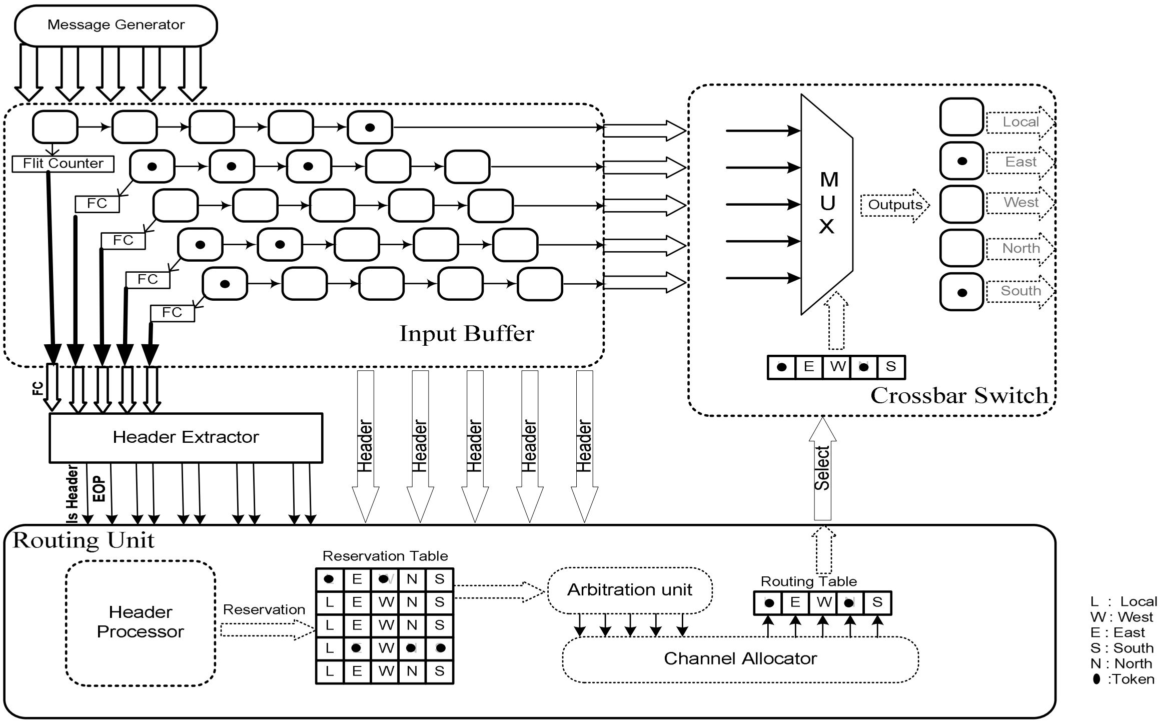

Figure-1. An Asynchronous NoC router architecture

zero. In other words, the circuit consumes power whenever transitions occur. In CMOS circuits, the dominant factor in power dissipation is the dynamic power consumed in the charging and discharging of the capacitive loads [12].

The rest of this paper is organized as follows: section 2 introduces the proposal asynchronous architecture and its specifications. The synchronous router architecture is illustrated in part 3. Portion 4 provide the experimental results of both NoC router types. The fifth section concludes this article.

2. An Asynchronous Design Structure of NoC router

Figure-1 indicates the architecture of the asynchronous NoC router which is applied in this experiment. This router is implemented with five bidirectional ports which are suitable to be employed in both Torus and Mesh topologies. There is no clock in this design and all of the data transmissions are put into action by the help of handshaking signals. A four-phased handshake protocol has been employed in this router. The router consists of three main modules which are input buffer, crossbar switch, and routing unit. Input buffer is responsible to store temporarily incoming flits from adjacent routers whether it has free space. The capacity of input buffer in all five ports is five flits (one buffer is dedicated per port). Each input buffer contains a counter to determine the number of received flits in current packet. This counter is accessed by a sub-component called header extractor which distinguishes the header flit among others in a packet. Header processor is another sub-component of routing unit which is responsible to process the incoming packet's header.

Routing unit is the main component to implement XY routing algorithm in this design. Simple XY algorithm cannot the extra links in torus topology. Routing process is implemented in this component once a new packet has been received. Each input channel can reserve one of output ports after routing process has been accomplished. An arbitration unit is utilized to dedicate the output port with round robin algorithm if there is more than one request for a special output channel. In other words, by the help of arbitration sub component output ports are distributed fairly through input ports. As soon as routing table set the switch links, the input port is connected to appropriate output with the help of asynchronous method. The last component which is crossbar switch is in charge of connecting all input ports to each of output ones. The control signals of routing unit select one of output port for an incoming header flit.

By implementing the wormhole switching all of the remaining flits of a packet follow their header and they are blocked whether their header is blocked on its way toward destination. Eliminating the clock skew problem, developing modularity, lower power consumption, and applying average delay instead of worst case delay, are some of asynchronous design benefits. The modularity of it makes it adaptable to the newer technology, and less vulnerability to changes in voltage and other environmental

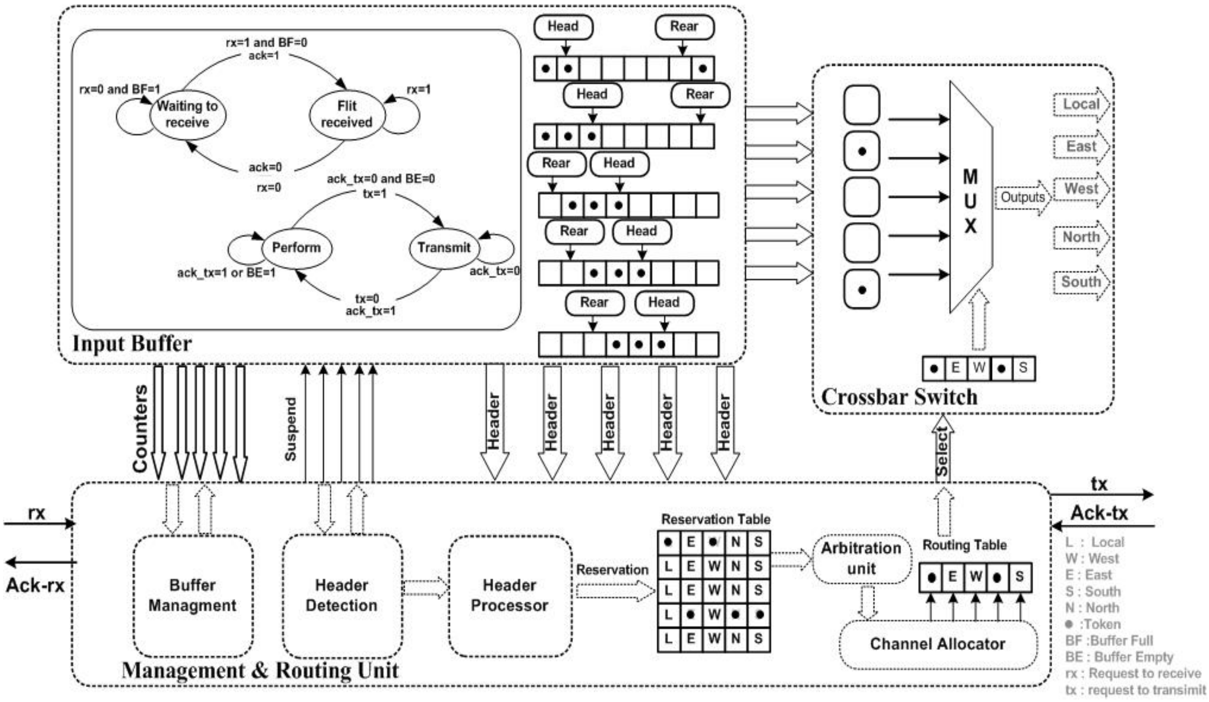

Figure- 2. A synchronous NoC router architecture

parameter such as temperature. The router is simulated with CSP-Verilog (communicating sequential processes) language as an asynchronous one. CSP-verilog is a well-known language for describing of concurrent systems which is accepted as a good description language for asynchronous systems. A Circuit in CSP is described as the composition of distinct processes that run in parallel and communicate with each other on channels by message passing. CSP-Verilog is an extension of the standard Verilog language which supports asynchronous communication as the hardware description language for all levels of abstractions except the net list which uses standard Verilog. CSP-Verilog is equipped with READ and WRITE programming language interface (PLI**)** macros to emulate CSP language communication actions on the channels.

Data communication is done by writing data to the ports and read it from the corresponding port on the other side of the channel. In order to send data, write macro is employed like: **'WRITE (Port name, value)'.** If the sender wants to write another data on that port, it would be suspended until the last data is read from that port with the following command **'READ (Port name, value)'.**

The receiver module also remains suspended until a data is written on its counterpart port. Read and write operation will be implemented by 4 phase handshake signaling [11].

Figure-3 shows a sample part of the asynchronous NoC router, employing both READ and WRITE macro operation. This code reads the three bit selector of multiplexer from the routing table to select one of its input ports and connect it to the proper output channel.

3. A Synchronous Design Structure of NoC router

A synchronous NoC router is designed and implemented in VHDL language. Figure-2 indicates the architecture of the synchronous NoC router which is applied in this experiment. It is implemented with five bidirectional ports which are apt to be employed in both mesh and torus topologies. It has been synthesized with the help of Leonardo and result in 38560 gates. The clock frequency of synthesized router is also reported 69.9MHz. Similar to the asynchronous scheme, synchronous NoC router consists of three main components including input buffer, management and routing unit, and crossbar switch. Two state diagrams show in the buffer component are responsible to take and transmit a flit in a single clock. The upper state diagram is triggered in positive edge of clock and the lower one is stimulated by negative edge of clock. Employing different edge of clock increases the performance of NoC router. Such feature increases the performance of implemented synchronous router.

The routing unit component is the central part which includes sub components such as buffer management, header processor, and arbitration unit and also channels allocator. The buffer management unit is in charge of controlling the buffer space to preclude overwritten flits on each other. The buffer is implemented as a circular queue to optimize the buffer efficiently.

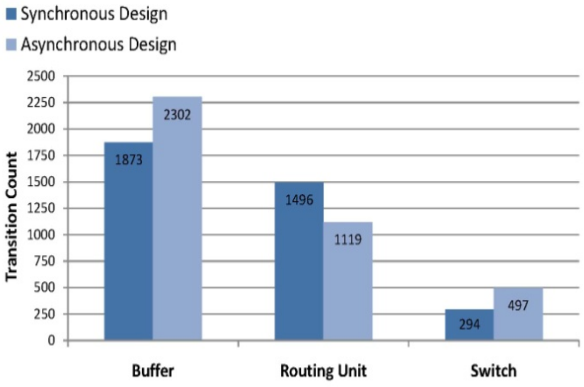

Figure-3. Power consumption of different components

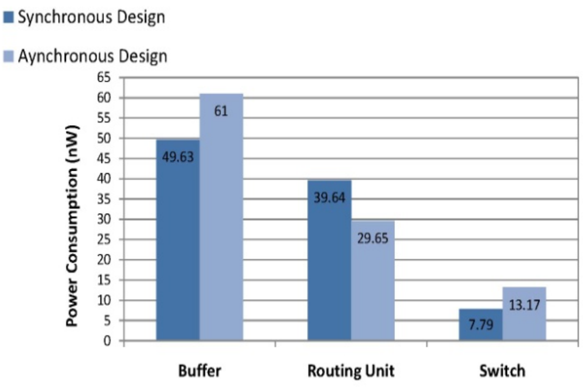

Figure-4. Transition count of different components

By the help of this feature the flits are written on the buffer as soon as a free space is found on it.

4. Experimental Results

Figure-3 and Figure-4 show the transition count and power consumption of different components in both architectures. According to the experimental studies, the transition counting of synchronous router are almost 82% of asynchronous one. The input buffer in asynchronous router is composed of succeeding buffers which acts like a shift register. In other words, as a new flit is received the previous ones are shifted toward the end of buffer.

In this scheme a flit must pass all the buffers which cause more transitions. On the other hands, this implementation reduces the complexity of the circuit. It also improves the performance of the router by utilizing pipeline property. However, the input buffer scheme in synchronous router is based on the random access memories. The required decoder for distinguishing buffer cells increases the circuit complexity. It also raises the circuit delay due to additional decoder delay.

In routing unit of asynchronous scheme header flit is processed once it has been arrived. This process sets the routing table and activates the arbiter unit to select the appropriate output channel. There is same story in the synchronous implementation. The routing unit is employed just as the receiving flit is recognized as a header. So during data transmission no transition count is reported. Conversely, in synchronous design the router unit endures all input clocks without regarding to the type of receiving flit. In other words, the routing unit transition count doesn't reduce in this architecture as same as asynchronous implementation.

The structure of switch component is simpler than the two others. This fact results in lower power consumption of this component in both schemes. However, the switch employed in asynchronous architecture contains more transmission counts due to its handshaking signals.

5. Conclusions

The NoC router can be divided into two main parts including data flow and control flow. Data flow contains the buffer and switch components which are responsible to store and transmit the flits. Control flow which is located in routing unit is in charge of finding the appropriate output channel for each of incoming flits. Based on the experimental results the routing unit component in asynchronous design is more complex due to handshaking signals but its power consumption is lower in contrast to synchronous one. On the other hand, the buffer component and also switch components circuits' are almost similar. However, the power consumption of them in asynchronous scheme are higher due to employing handshaking signals.

References

[1] Wai Hong Ho, and T.M. Pinkston, "A Methodology for Designing Efficient On-Chip Interconnects on Well-Behaved Communication Patterns", *Proceedings of International Symposium on High-Performance Computer Architecture (HPCA'03)*, 2003, pp. 377-388.

[2] S. Kumar, A. Jantsch, J-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A network on Chip Architecture and Design Methodology", *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, 2002, pp. 105-112.

[3] Hu Jingcao, and R. Marculescu, "Exploiting the Routing Flexibility for Energy/Performance Aware Mapping of Regular NoC Architectures", *Design, Automation and Test in Europe Conference and Exhibition*, 2003, pp.683-693.

[4] N. Banerjee, P. Vellanki, and K.S. Chatha, "A Power and Performance Model for Network-on-Chip Architectures", *Proceedings on Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 1250-1255.

[5] P. Vellanki, N. Banerjee, and K.S. Chatha, " Quality-of-Service and Error Control Techniques for Network-on-Chip Architectures" *Proceedings of the 14th ACM*

Great Lakes symposium on VLSI (GLSVLSI), 2004 pp. 45-50.

[6] M. Amde, T. Felicijan, A. Efthymiou, D. Edwards and L. Lavagno, "Asynchronous on Chip Networks", *IEEE Proceedings on Computers and Digital Techniques*, 2005, pp.273-283

[7] B. Rajan, and R.K. Shyamasundar, "Multi clock ESTEREL: A reactive framework for asynchronous design", *Proceedings of 14th International Symposium on Parallel and Distributed Processing*, 2000, pp.201-210.

[8] M.W. Health, W.P. Burleson, and I.G. Harris, "Synchro-tokens: a deterministic GALS methodology for chip-level debug and test", *IEEE Transacions on Computers*, 2005, pp. 1532–1546.

[9] Y.K. Yun, and L.D. Dill, "Automatic synthesis of extended burst-mode circuits. I.(Specification and hazard-free implementations)", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1999, pp. 101–117.

[10] S. Ogg, E. Valli, C.D. Alessandro, A. Yakovlev, B. Al-Hashimi, and L. Benini, " Reducing Interconnect Cost in NoC through Serialized Asynchronous Links", *First International Symposium on Networks-on-Chip (NOCS'07)*, 2007, pp.219-219.

[11] A. Seifhashemi, H. Pedram, “Verilog HDL, Powered by PLI: a Suitable Framework for Describing and Modeling Asynchronous Circuits at All Levels of Abstraction”, *Proceedings of 40th DAC,* 2003, pp. 330-333.

[12] M. Salehi, K. Saleh, H. Kalantari, M. Naderi, and H.Pedram, "High level Energy Estimation of Template-Based QDI Asynchronous circuits Based on Transition Counting", *Proceedings of the 16th International Conference on Microelectronics (ICM)*, 2004, pp. 489-492.