Evaluate Area for Very Large Integrated Digital Systems Based on Bandwidth Variation

Afshin Shaabany [1], Fatemeh Jamshidi [1]

[1] Islamic Azad University, Fars Science and Research Branch, Shiraz, Iran
afshinshy@yahoo.com, Fjamshidi59@yahoo. com

Abstract: In this paper, Network on Chip is used as an alternate approach for very large integrated digital systems (System on chip) that is based on bus communications and IP interconnections. This approach has solved some problems like scalability that buses encounter them. One of the basic steps in this approach is correct simulation of NoC implementation; moreover, simulation design operability and perform ability require its synthesizability. Designing and implementation of NoC communication are presented in this work. Finally, bandwidth variation effect on area requirements is evaluated, and area requirements changing due to these alternations will be discussed and explained.
[Afshin Shaabany, Fatemeh Jamshidi. Evaluate Area for Very Large Integrated Digital Systems Based on Bandwidth Variation. Journal of American Science 2011;7(1):163-169]. (ISSN: 1545-1003). http://www.jofamericanscience.org.

Keywords: Network on Chip, IP interconnection, bandwidth variation effect, scalability, perform ability.

1. Introduction

Power and performance are two essential features which in corresponded with each other, produce main concerns in design and implementation. Nowadays, very large integrated digital systems [Benini (2005), Chen (2003), Pende (2005), Eisley (2004)] (Systems on Chip) may contain different components such as processor, input- output units and different types of memories. Likewise, each of these components may include different specifications such as variable bandwidth, buses and different communication protocols. Generally, bus is utilized for interconnecting the processing elements of System on Chip (SoC). However by increasing the number of processing elements, the bus itself is transmuted into a bottleneck. To obviate this difficulty, the idea of Network on Chip (NoC) has been introduced [Chiu(2000)].

This network can be modeled as a graph wherein nodes, processing elements and edges are the connection links of the processing elements. In this article, design and implementation of a NoC router are presented. In the second section of this article, the utilized routing algorithm is briefly analyzed. In implementation, XY routing algorithm is utilized [Holsmark (2006), Xiaohu (2007)]. In the third section, the wormhole switching which is used in implementation is reviewed [Duato (1993), Hsh (1992)]. In the forth of this article, the utilized traffic pattern is briefly explained. In the fifth section which considers being the main body of this article, handshaking communication mechanism is introduced and analyzed. In this section, the structure of information packets, router function and different states of the router are analyzed. Furthermore, the experimental results of implementation and synthesis of this routing are presented in the final section of this article. In this implementation, handshaking communication protocol is utilized to interconnect different processing elements.

2. The Utilized Routing Algorithm

The utilized topology for implementation is an $n \times n$ regular two dimensional mesh. A sample of this topology is shown in Figure 1.

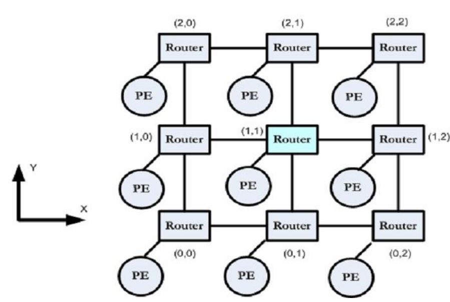

Figure 1. A regular 3×3 mesh topology

The elements which are shown in rectangles represent NoC routers and those which are shown in circles represent the processing elements of this network. By the use of communication links and routers, these processing elements which are connected to each other communication information. Routers are named based on their position in coordinate system. Router ports are also named based on their geographical direction.

However, as it is shown in Figure 1. the number of the ports connected to each other is different due to its position in topology. For example, the router which is placed in the northeast of topology in 2×2

coordinates ([2, 2]), possesses 3 ports and the router in the center of the topology in 1×1 coordinate ([1, 1]), has 5 ports.

For n- dimensional mesh topologies in NoCs, dimension order routing produces deadlock- free routing algorithms. These algorithms are very popular, like XY routing (for 2- D mesh). The routing algorithm which is used in this design is a version of XY algorithm. This algorithm is deterministic algorithm in which packet takes routing in one dimension and it continues till this packet attains the desired coordinate in that dimension. Then routing is fulfilled in the same way. This method warrants no deadlock to occur [Duato (1993), Hsh (1992)]. In this algorithm, according to the coordinates of each router and destination address, routing takes place first in X direction and then in Y destruction and may not be able to adopt a substituting router. It is due to the fact that these types of algorithms adopt routing only based on the source- destination address of packets. Therefore, two packets with the same source and destination address necessarily cross the same route and do not consider the momentary traffic in the route.

3. The Utilized Switching

The need to buffer complete packet within a router can make it difficult to construct low area, compact and fast routers. In implementation, wormhole switching is used which is utilized in almost all of NoCs [Duato (1993)].

In wormhole switching, message packets are also pipelined through the network. A message packet is broken up into flits that the flit is the unit of message flow control. Therefore, input and output buffers at a router are typically large enough to store a few flits [Hsh (1992)].

As we said, in this switching, message packets are divided into equal smaller sections named as flit. Flits are concurrently transferred in the network. Therefore if 16- bit flits are ready to be transferred, 32 signals between two routers are considered to transfer the flits, 16 signals for sending and 16 signals for receiving. In this way, flits are transferred in parallel. Other switching techniques are not commonplace in NoCs usages. For instance, circuit switching technique due to its low performance contradicts with power and performance parameters. Similarly packet switching as a result of its big buffers requirement shows the same contradiction.

4. The Utilized Traffic Pattern

The traffic model is one of the important parameters in evaluating the latency time of interconnection networks. These models are produced according to the application programs which are run on the machine. In different applications, different models are used. Traffic models are defined according to three parameters [Hsh (1992)]: a) The entrance time to networks b) Message length and c) Address distribution type.

The uniform traffic model is the simplest traffic model which used in most of evaluations (and this paper implementation). In this model, each node sends message to the other nodes in network with equal probability. For example in a 6×6 mesh topology, each nodes sends message to the other nodes with the probability of %2.85. All source or destination nodes are selected with equal probability. The selection of source and destination nodes for each message will be independent from other messages [Hsh (1992)].

5. Asynchronous Communication Mechanism

For making interaction between routers, handshaking communication protocol is utilized in case the data is put on the line; the existence of the data is informed to the next router. Next router takes the data from the line and transmits its confirmation to the sender router. So in addition to the flits sending and receiving channels, TX, ACK- TX, RX and ACK-RX signals are required. TX pin is the output and whenever the data is ready in the output port, this pin equals to one and waits for ACK- TX to be equaled to one. Likewise each input port after finding the RX input pin to be one, reads the data on this port and equals the ACK- RX output pin to one.

5.1 The structure of information packets

In each communication standards, the communication payload contains a series of control fields. These fields can be put in the main frame as the redundant fields in order to increase the controllability, fault tolerance, security and some other issues like these. In our intercommunication protocol, flits are used to structuralize. A flit structure is considered in the way that the first bit shows the flit to be the header- trailer or the data. When the first bit equals one, this flit is a header or trailer. In this case, the 2^{nd} bit determines which one is the header and which one is the trailer. This representation is shown in Table 1.

Table 1. The defined protocol that characterize the flit type

First bit	Information type	Second bit	Information type
0	Data	*	Data
1	Header/Trailer	0	Trailer
		1	Header

5.2 Routing function

Each router by receiving the header flit from input, accomplish routing and updates routing Tables according to its source and destination addresses based on XY algorithm.

Henceforth, all of the flits take routing based on the Tables till receiving the final flit (trailer). Routing Tables conclude two Tables: routing Table and output Table. The first Table represents the out port for each input and the second represents the state of each out port (busy or free). In Figure 2. you can see a NoC central router in mesh topology. The central router has 5 I/O port. The local port is utilized to connect the correspondent circle to the processing element (IP block) and other ports are for connecting to other routers.

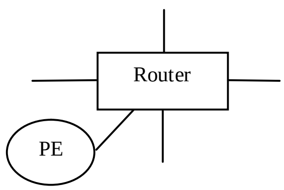

Figure 2. Central NoC router in mesh topology with its ports

The main point here is that the correspondent circle with this routing should have the same interface to be able to use this routing.

Routing function feature takes the charge of routing based on routing algorithm and selection function feature under takes the responsibility of choosing out port in competition circumstances based on the defined priority mechanism. In our designing, mechanisms is implemented by the software in the manner that it gives priority to input port and whatever an input port has a higher priority. It selects its desired output port faster. However, we should consider that competition circumstance only take place when in one moment, there is a request from two input port for one output port.

Our fulfilled designing is implemented by the use of VHDL hardware describing language. In order to router implementation, one entity is designed for whole routing. In code segment of Figure 3. size and type of I/O port are shown.

```
Entity router is
Port(
Clock: in std_lolgic;
Reset: in std_logic;
Data_in: in arrayPortsRegisters;
Rx: in PortsRegisters;
Ack_rx: out PortsRegisters;
Data_out: out arrayPortsRegisters;
Tx: out PortsRegisters;
Ack_tx: in PortsRegisters);
End router;
```

Types of array Ports Registers and Ports Registers signals are defined in one packet. In order to implement, we defined a machine of definite state for input which you can see in Figure 4.

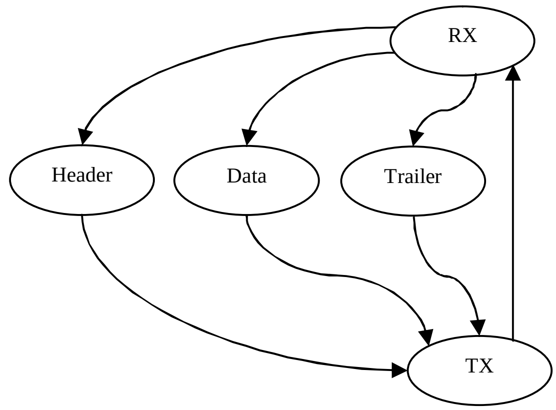

Figure 4. Finite State Machine for flit and router status analyze

5.2.1 Received state

In this state, the routing await for its RX base to be one. In case this happens, firstly the data in Data-in is need and then the correctness of this data is examined. In case of being correct, ACK- RX equal one. Then the next state is defined according to the header/trailer bit.

5.2.2 Header received state

In this state, the appropriate output port is defined based on the source and destination addresses and out port Table. Then routing Table and out port Table are updated. Finally we alter routing state to transmit state.

5.2.3 Trailer received state

In this state, after the destination port is determined by the routing Table, this Table of out port Table is updated. In order to do this, the home correspondent with the input is equaled to NO PORT and also the output port state in out port Table is equaled to free.

5.2.4 Data received state

In this state, after finding the output port by routing Table, the received flit is put in the output port.

5.2.5 Transmit state

In this state, after placing the flit in the output port and equaling the desired output port TX base to one, we wait for receiving ACK- TX and after it's receiving, we equal TX to zero and turn back to the received state.

6. Experimental results

All of the designs which are already presented for NoC, can be used in case they are synthesized. One of the parameters that challenges NoC design synthesizing is the area requirement. For example, many of the presented designs could not be synthesized on the ASIC platform. Table 2 shows the comparison between this article's designed router and other routers. This Table compares some parameters such as topologies, routing algorithms, flit sizes, synthesizability and implementation. As it is obvious from this Table, many of the routers are not synthesized and implemented on ASIC infrastructure. Our router is synthesized and implemented on FPGA as well as ASIC. TSMC 65n is used for ASIC and Spartan 3E is utilized for FPGA.

In order to test the router, a test bench is designed that can send packets from input ports in a uniform traffic pattern and save the output packets in output ports. In the best situation, the Receive state duration, Header- Received, Trailer- Received and Data- Received are one clock cycle. The Transmit state duration is two clock cycles.

Table 3 shows the area requirement for synthesizing the 8 bit designed router on Spartan 3E.

Utilizing percentage of Spartan 3E resources by the 8 bit router is shown in Table 4.

Table 5 shows the area requirement for synthesizing the 16 bit designed router on Spartan 3E.

Utilizing percentage of Spartan 3E resources by the 16 bit router is shown in Table 6.

Table 2. Comparison between article's designed router and other router

NoC Routers	**Topology/ Routing**	**Flit Sizes**	**Implementation and synthesis**
Marescaux (2003)	2D torus (scalable)/ XY blocking, hopbased, deterministic	16 bits data + 3 bits control	FPGA VirtexII /virtexII Pro
Xpipes (Dall'Osso (2003))	Arbitrary (designtime)/ Source static (street sign)	32.64 or 128 bits	No
AEthereal- Rijpkema (2003)	2D mesh/ Source	32 bits	ASIC layout
Eclipse (Tortosa (2002))	2D sparse Hierarchical mesh/ NA	68 bit	No
Proteo (Saastamoinen (2002))	Bi-directional ring/ NA	Variable control and data sizes	ASIC layout CMOS 0.18um
SOCIN (Zeferino (2003))	2D mesh (scalable)/ XY source	n bits data + 4 bits control	No
Hermes (Pande (2003))	2D mesh (scalable)/ XY	8 bits data + 2 bits control	FPGA VirtexII
T- SoC (Grecu (2004))	Fat- tree/ Adaptive	38 bits maximum	
QNOC (Bolotin (2004))	2D mesh regular or irregular/ XY	16 bits data + 10 bits control	No
Our Design	**2D Mesh Regular**	**Variable Data And Control bits**	**ASIC (ASL05 and TSM13u) + FPGA (SPARTAN and Virtex)**

Table 3. Total required area for synthesis of 8 bit router on Spartan 3E

Cell	**Library**	**References**	**Total Area**
BUFGP	xis3e	1×1	1 BUFGP
FDCE	xis3e	1×30	30 Dffs or Latches
FDE	xis3e	1×141	141 Dffs or Latches
FDPE	xis3e	1×5	5 Dffs or Latches
IBUFG	xis3e	1×51	51 IBUFG
LUT2	xis3e	1×68	68 Function Generators

Table 4. Utilization percentage of SPAETAN 3E by 8 bit router

Resource	**Used**	**Avail**	**Utilization**
IOs	101	194	52.06%
Global Buffers	1	24	4.17%
Function Generators	548	21712	2.52%
CLB Slices	274	8672	3.16%
Dffs or Latches	176	22100	0.80%

Block RAMs	0	28	0.00%
Block Multipliers	0	28	0.00%
Block Multiplier Dffs	0	2016	0.00%

Table 5. Total required area for synthesis of 16 bit router on **SPARTAN 3E**

Cell	Library	References	Total Area
BUFGP	xis3e	1×1	1 BUFGP
FDCE	xis3e	1×30	30 Dffs or Latches
FDE	xis3e	1×221	221 Dffs or Latches
FDPE	xis3e	1×5	5 Dffs or Latches
IBUF	xis3e	1×91	91 BUF
LUT2	xis3e	1×132	132 Function Generators
LUT3	xis3e	1×174	174 Function Generators
LUT4	xis3e	1×518	518 Function Generators
MUXF5	xis3e	1×2	2 MUXF 5
OBUF	xis3e	1×90	90 OBUF

Table 6. Utilization percentage of SPAETAN 3E by 16 bit router

Resource	Used	Avail	Utilization
IOs	181	194	93.30%
Global Buffers	1	24	4.17%
Function Generators	824	21712	3.80%
CLB Slices	412	8672	4.75%
Dffs or Latches	256	22100	1.16%
Block RAMs	0	28	0.00%
Block Multipliers	0	28	0.00%
Block Multiplier Dffs	0	2016	0.00%

Figure 5. shows the comparison between synthesizing area requirements of the 8 and 16 bit routers on the Spartan 3E.

The designed router synthesizing process is also done on ASIC 65n platform.

Table 7. shows the area requirement for synthesizing the 8 bit designed router on TSMC 65n.

In the same way, Tables 8. and 9. show synthesizing area requirements of the 16 and 32 bit routers on TSMC 65n.

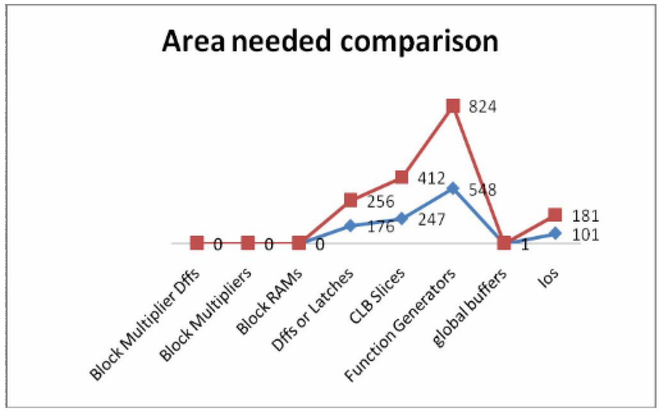

Figure 5. Area requirement comparison between 8 and 16 bit routers for synthesis on Spartan 3E

Table 7. Total required area for synthesis of 8 bit router on TSMC 65n.

Element	Library	Number of Element
Number of ports	umc165sp	108
Number of nets	umc165sp	6616
Number of cells	umc165sp	6554
Number of references	umc165sp	57
Combinational area	umc165sp	16953.120183
Non combinational Area	umc165sp	9302.039932
Net Interconnect area	umc165sp	3.157800
Total cell area	umc165sp	26255.160116
Total area	umc165sp	26258.317915

Table 8. Total required area for synthesis of 32 bit router on TSMC

Element	Library	Number of Element
Number of ports	umc165sp	188
Number of nets	umc165sp	8230
Number of cells	umc165sp	8128
Number of references	umc165sp	60
Combinational area	umc165sp	20388.600205
Non combinational Area	umc165sp	14990.039848
Net Interconnect area	umc165sp	4.176200

Total cell area	umc165sp	35378.640053
Total area	umc165sp	35382.816253

Table 9. Total required area for synthesis of 16 bit router on TSMC 65n.

Element	Library	Number of Element
Number of ports	umc165sp	348
Number of nets	umc165sp	11693
Number of cells	umc165sp	11511
Number of references	umc165sp	64
Combinational area	umc165sp	29047.680285
Non combinational Area	umc165sp	26366.039680
Net Interconnect area	umc165sp	6.276800
Total cell area	umc165sp	55413.719966
Total area	umc165sp	55419.996766

Figure 6. shows the comparison between synthesizing area requirements of the 8, 16 and 32 bit routers on the TSMC 65n.

Based on the presented statistics data, the following results are provided:

1. The effect of bandwidth variation on the area requirements is not linear.
2. The increase rate of area requirement proportion enhances by the bandwidth increase. As it was shown in this article, the area requirement increase proportion of 8 bit bandwidth to 16 bit was 1.34. However, this rate was 1.49 for 16 to 32 bandwidth increase.

Power consumption of implemented router has been analyzed. Results are shown in following Tables (Table 10, 11 and 12). These results belong to 8, 16 and 32 bit routers.

6. Conclusion

In this article not only we used an asynchronous communication mechanism based on handshaking to transfer information but also by using statistical data, we showed that this designed router occupies very little space.

Scalable design of this router leads to easy and efficient addition of new capabilities like 16-bit and 32-bit bandwidth. The resource utilization of this router is more efficient than similar implementation on FPGA and ASIC platforms.

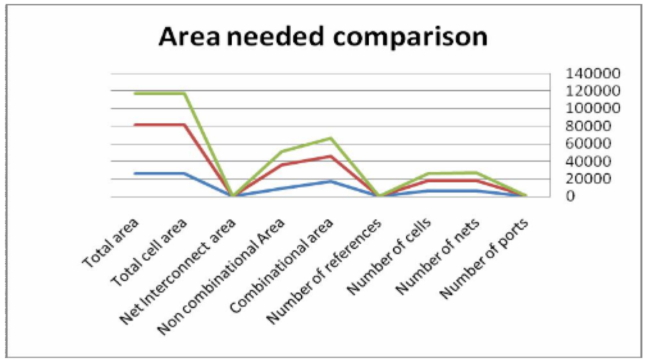

Figure 6. Total accumulated area requirement comparison among 8, 16 and 32 bit routers for synthesis on TSMC 65n (ASIC)

Table 10. Total power consumption information for 8 bit router

	Switching Power(mW)	Internal Power(mW)	Leakage Power(pW)	Total Power(mW)
Router Power	0.157	0.979	5.63e + 08	1.698

Table 11. Total power consumption information for 16 bit router

	Switching Power(mW)	Internal Power(mW)	Leakage Power(pW)	Total Power(mW)
Router Power	0.154	1.484	7.32e + 08	2.370

Table 12. Total power consumption information for 32 bit router

	Switching Power(mW)	Internal Power(mW)	Leakage Power(pW)	Total Power(mW)
Router Power	0.201	2.677	1.15e + 09	4.027

Corresponding Author:
Afshin Shaabany, Islamic Azad University, Fars Science and Research Branch, Shiraz, Iran.
E-mail: afshinshy@yahoo.com

References

1. Benini L, Bertozzi D. Network- on- chip architectures and design methods. IEE 2005
2. Chen X, Peh LS. Leakage power modeling and optimization in interconnection networks. ISLPED'03, Seoul, Korea. August 2003:25-27.
3. Pende PP, Grecu C, Jones M, Ivanov A, Saleh R. Performance evaluation and design trade- offs for network- on-chip interconnect architectures, IEEE Transaction on Computers. 2005; 54 (8):1025-1040.
4. Eisley N, Peh LS. High- level power analysis for on- chip networks. CASES'04. Washington, DC, USA. September2004:22-25.
5. Chiu GM. The odd- even turn model for adaptive routing, IEEE Transactions on Parallel and Distributed Systems. 2000;11:729-38.
6. Holsmark R, Palasi M, Kumar S. Deadlock free routing algorithms for mesh topology NoC systems with regions. 9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools. 2006:696-703.
7. Xiaohu Zh, Yang C, Liwei W. A novel routing algorithm for network-on-Chip. International Conference on Wireless Communications, Networking and Mobile Computing. September 2007:21-25.
8. Duato J. A new theory of deadlock- free adaptive routing in wormhole network, IEEE Transaction On Parallel and Distributed Systems. 1993;4(12):1320-1331.
9. Hsh W. Performance issues in wire-limited hierarchical networks, PhD Thesis, University of Illinois- Urbana Champaign. 1992.
10. Marescaux T, Mignolet JY, Bartic A., Moffat W, Verkest D, Vernalde S, Lauwereins R. Networks on chip as hardware components of an OS for reconfigurable systems. Field-Programmable Logic and Applications Conference. September 2003.
11. Dall'Osso M, Biccari G, Giovannini L, Bertozzi D, Benini L, Xpipes. A latency insensitive parameterized network-on-chip architecture for multi-processor SoCs. International Conference on Computer Design. 2003:536-539.
12. Rijpkema E, Goossens K, Radulescu A. Trade offs in the design of a router with both guaranteed and best- effort services for networks on chip. Design, Automation and Test in Europe. March 2003:350-355.
13. Tortosa DS, Nurmi J. Proteo: A new approach to network- on- chip. IASTED International Conference on Communication Systems and Networks. September 2002.
14. Saastamoinen I, Alho M, Pirttimaki J, Nurmi J, Proteo Interconnect IPs for Networks- on- Chip. IP Based SoC Design Conference. October 2002.
15. Zeferino C, SoCIN AS. A parametric and scalable network- on- chip. 16th Symposium on Integrated Circuits and Systems Design. September 2003:169-174.
16. Pande P, Grecu C, Ivanov A, Saleh A. Design of a switch for network on chip applications. International Symposium on Circuits and Systems. May 2003:217-220.
17. Grecu C, Pande P, Ivanov A, Saleh R. A scalable communication- centric SoC interconnect architecture. International Symposium on Quality Electronic Design. 2004.
18. Bolotin E, Cidon I, Ginosar R, Kolodny A. QNoc: QoS architecture and design process for Network on Chip, The Journal of Systems Architecture, Special Issue on Networks on Chip. 2004.

5/11/2010