

# A SYSTEMATIC ROUTING-ALGORITHMS FOR NOC HARDWARE

Venkata Sridhar.T<sup>1</sup>, Dr. G. Chenchu Krishnaiah<sup>2</sup>

<sup>1</sup>Research Scholar, VTU, Belagavi,Karnataka.

<sup>2</sup>Professor, GKCE,Sullurpet, Nellore

<sup>1</sup>venkatasridhar.thatiparthi@gmail.com,

<sup>2</sup>krishna.rakesh4@gmail.com

**Abstract— Networks on Chip (NoC) has been widely discussed for its smart structure and high performance. Routing algorithms significantly influence design cost and system performance of NoC. In this paper, a new hardware method called Final- Destination-Tag (FDT) is proposed to improve the original Destination-Tag (DT) method for implementing different routing algorithms. Compared with the DT-method, the proposed FDT method could reduce the header size of the packet. We evaluate NoC with this proposed-method in terms of circuit resource, average latency, max latency, average throughput and power consumption. The results indicate that the proposed method is effective in increasing throughput and reducing circuit resource, latency and power consumption for NoC.**

## I. INTRODUCTION

With the development of silicon technology and continuous growth of the integration in one chip, the limitation of interconnects for designing the chip has attracted much attention [1]. And this problem becomes seriously important especially in multiprocessor System-on-Chip (MPSoC) [2][3]. As an effective solution for this problem, NoC has been widely researched and discussed in these years for solving complex on-chip communication problems [4][5]. Routing algorithm significantly influences the performance and design cost [6]. Therefore, the way how to implement a routing algorithm onto NoC becomes more important. In this work, a new hardware method called Final- Destination-Tag (FDT) is proposed for

implementing routing algorithm onto NoC. Compared with the conventional Destination-Tag (DT) method, the proposed FDT method is effective in reducing the header size of the packet. NoC architectures with FDT and DT methods are implemented by using FPGA to evaluate the circuit resource of the proposed method. The experimental results show that the proposed FDT method could reduce the circuit resource compared than the former one. The simulation results by NoC simulator also indicate that the proposed method provides low average latency, low max latency, high average throughput and low power consumption.

## II. RELATED LITERATURE

There are many kinds of routing algorithms, but few of them can be implemented by a hardware method [7][8].

The well-known hardware implementation method is DT which was introduced in [9] and has been used in a lot of NoC research, such as Intel 80-Tile NoC [10], 18-Tile ANN (Artificial Neural Network) NoC [11], and so on. In the DT method, the bits of the destination address which is stored in header of the packet are used to select the output port of each router. The bit number of the destination address is decided by the port number of the router and the hop number from source node to destination node. For example, one 5-port router needs 3 bits destination address to select its output port. If there are 9 hops from source node to destination node, it needs  $9 \text{ hops} \times 3 \text{ bits} =$

27 bits in total. It means that the total bit number of the destination address becomes larger proportional to the network size. Thus, the header of the packet will also become larger.

If only the address of the final destination node is stored in header, the packet size becomes much smaller. For this development, each router should be redesigned to suit to this method. The output port will be selected referring to the address of the final destination node. This selection method needs to follow the same rules similar to the existing routing algorithms. In general, smaller packet size will cause lower latency [12], and the throughput will increase and the power consumption will be reduced. Therefore, our work focuses on reducing header size of the packet to achieve high performance and low power consumption.

**III. IMPLEMENTATION OF PROPOSED FDT METHOD FOR RA**

A router model with DT consists of input registers, MUXs, switch allocators, shifters and output registers as shown in Fig.

1. This router model is improved to suit the proposed FDT method. The packet format, router architecture, and different routing algorithms implementation of the FDT method are introduced in this section.

**A. Packet Format of the Router Architecture with the Proposed FDT Method**

The packet format of FDT method is different from that of DT in header of the packet. The header of DT includes PT (phit type) and DA (Destination node Address). In DT method, 3-bit is used to address 5 ports of each router. The DA needs to contain some 3-bit destination address for each hop. The header of the proposed FDT method includes PT (phit type) and FDA (Final Destination node Address). FDA consists of x-axis address and y-axis address of the final destination node. Assume that 4x4 NoC with 5-port routers transmits one packet from node (1, 1) to node (3, 2). With FDT method, FDA of the header is 4 bits of "1110". DA of the header is 4 bits x 3 hops = 12 bits, while the proposed FDA has less number of bits in header of the packet.

The bits length of FDA is dependent to network size. The largest NoC by (a+b) bits FDA (a bits for x-axis and b bits y-axis) is  $2^a \times 2^b$  NoC.

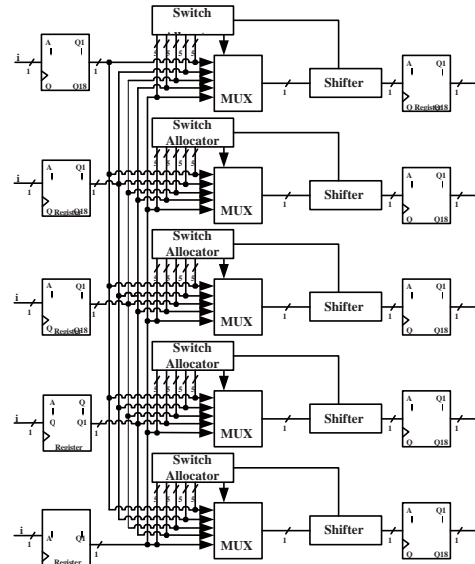


Fig. 1. Router model with DT method.

**B. Router Architecture with FDT Method**

The router model with proposed FDT method is shown in Fig. 2. It has 5 input ports and 5 output ports that are connected to one processing element (PE) and four adjacent routers. It consists of input registers, MUXs, switch allocators, and output registers. i0 is input from PE. i1, i2, i3 and i4 are input phits from four adjacent routers. When phits arrive at an input register of this router, then they are transmitted to five 5-1 MUXs to choose the output port. Each MUX is controlled by a switch allocator. Switch allocator0 is used to choose the input phits for output port PE, and switch allocator1, 2, 3 and 4 for output port East, North, West, and South, respectively. Compared with the original router model with DT method, shifters are removed, and the switch allocators are improved to suit FDT method.

The key part of the router design is that each switch allocator controls one MUX to select the proper output port. It consists of decoder, arbiter and hold logic, as shown in Fig.

3. Every switch allocator has the same architecture except the function. In decoder, 6-bit of PT and FDA of each input phit are decoded. PT is used for partition the phits type and FDA for choosing output port. The FDA should be compared with the function of each switch allocator. *fij* denotes function in switch allocator j for phit from port i. A function of each switch allocator will be described in detail in the next subsection.

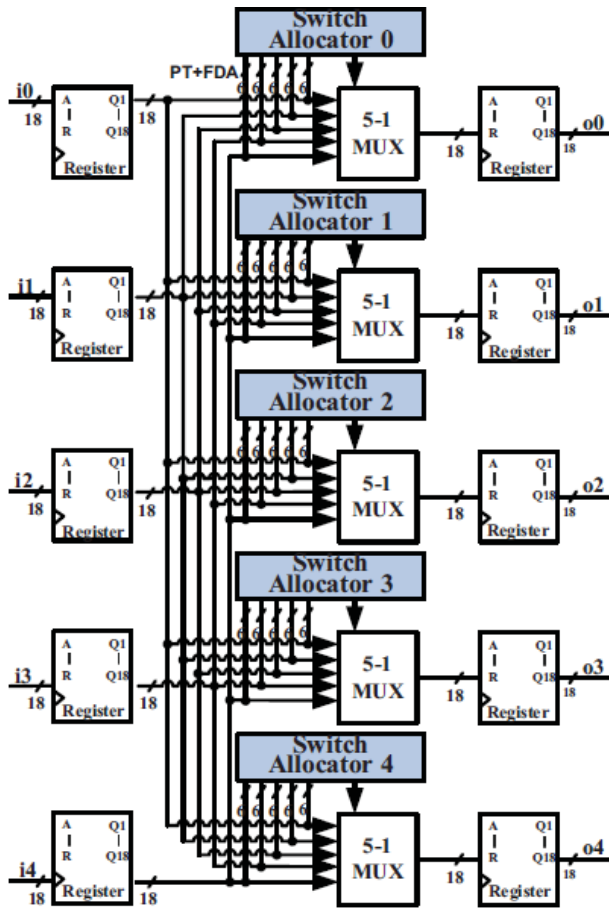


Fig. 2. Block diagram of the proposed FDT router

### C.FDT Method for Implementing Different Routing Algo- rithms

As shown in Fig 3, 4 bits FDA needs to be compared with the function to select the output port. The x-axis address of FDA is compared with the x-axis address of local router and also the y-axis address. If the input phit is satisfied with the function, the input will be selected by this output port. We use  $xFDA$  and  $yFDA$  to denote x-axis and y-axis address of FDA, use  $xlocal$  and  $ylocal$  to denote x-axis and y-axis address of a local router, and  $f_{ij}$  means function in switch allocator  $j$  for phit from port  $i$ . For X-Y routing algorithm, all functions  $f_{ij}$  in the switch allocator  $j$  are shown as follows:

#### I. X-Y routing algorithm

• Allocator0  $\Rightarrow$  PE (Allocator0 will select input phits for output port of PE)

$$f_{i0}: (xFDA = xlocal) \wedge (yFDA = ylocal), i = 0, 1, \dots, 4;$$

• Allocator1  $\Rightarrow$  East

$$f_{i1}: (xFDA > xlocal), i = 0, 1, \dots, 4;$$

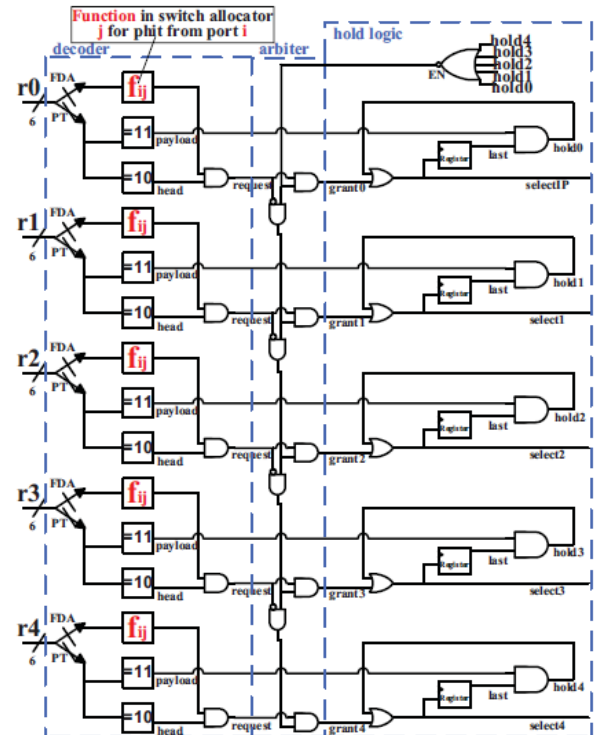


Fig. 3. Architecture of the switch allocator  $j$

• Allocator2  $\Rightarrow$  North

$$f_{i2}: (xFDA = xlocal) \wedge (yFDA < ylocal), i = 0, 1, \dots, 4;$$

• Allocator3  $\Rightarrow$  West

$$f_{i3}: (xFDA < xlocal), i = 0, 1, \dots, 4;$$

• Allocator4  $\Rightarrow$  South

$$f_{i4}: (xFDA = xlocal) \wedge (yFDA > ylocal), i = 0, 1, \dots, 4.$$

Other common routing algorithms, such as West-First (W-F) and North-Last (N-L) could be implemented into NoC with the same method. We just need to change the functions of switch allocators as follows:

#### II. W-F routing algorithm

• Allocator0  $\Rightarrow$  PE

$$f_{i0}: (xFDA = xlocal) \wedge (yFDA = ylocal), i = 0, 1, \dots, 4;$$

• Allocator1  $\Rightarrow$  East

$$f_{01}, f_{11} \text{ and } f_{31}: (xFDA > xlocal) \wedge (yFDA = ylocal); f_{21}: (xFDA > xlocal) \wedge (yFDA \geq ylocal)$$

$$f_{41}: (xFDA > xlocal) \wedge (yFDA \leq ylocal).$$

- Allocator2 $\Rightarrow$  North  
 $f_{02}, f_{12}, f_{22}$  and  $f_{32}: (xFDA \geq xlocal) \wedge (yFDA < ylocal)$ ;  $f_{42}: (xFDA = xlocal) \wedge (yFDA < ylocal)$ ;
- Allocator3 $\Rightarrow$  West  
 $f_{i3}: (xFDA < xlocal)$ ;
- Allocator4 $\Rightarrow$  South  
 $f_{04}, f_{14}, f_{34}$  and  $f_{44}: (xFDA \geq xlocal) \wedge (yFDA > ylocal)$ ;

TABLE I

COMPARISONS OF FPGA RESOURCE (NUMBER OF ALUTS).

Router type and network size	Conventional	Proposed
3-port router	130	86
4-port router	235	147
5-port router	445	197
3x3 NoC	1937	1108
4x4 NoC	4079	2236
8x8 NoC	21858	10498

$f_{24}: (xFDA = xlocal) \wedge (yFDA > ylocal)$ .

### III. N-L routing algorithm

- Allocator0 $\Rightarrow$  PE  
 $f_{i0}: (xFDA = xlocal) \wedge (yFDA = ylocal), i = 0, 1, \dots, 4$ ;
- Allocator1 $\Rightarrow$  East  
 $f_{21}: xFDA > xlocal$ ;  
 $f_{01}, f_{11}, f_{31}$  and  $f_{41}: (xFDA > xlocal) \wedge (yFDA \leq ylocal)$ ;
- Allocator2 $\Rightarrow$  North  
 $f_{i2}: (xFDA = xlocal) \wedge (yFDA < ylocal)$ ;
- Allocator3 $\Rightarrow$  West  
 $f_{23}: xFDA < xlocal$ ;  
 $f_{03}, f_{13}, f_{33}$  and  $f_{43}: (xFDA < xlocal) \wedge (yFDA \leq ylocal)$ ;
- Allocator4 $\Rightarrow$  South  
 $f_{04}, f_{12}, f_{34}$  and  $f_{44}: (xFDA \leq xlocal) \wedge (yFDA > ylocal)$ ;  $f_{24}: (xFDA = xlocal) \wedge (yFDA > ylocal)$ .

## IV. IMPLEMENTATIONS AND RESULTS

In this section, FDT method is compared with traditional DT method and evaluated in terms of circuit resource, average latency, max latency, average throughput and power consumption under several router types and different NoC sizes. Achieved best results.

### A. Circuit Resource Evaluation

With DT method, different routing algorithms are implemented into the conventional router by sending the different packets, where the router architecture is not changed. With FDT method, different routing algorithms are implemented into the developed router by revising the function part of the switch allocators. The proposed routers with X-Y routing algorithm are used as the example. It is designed by Verilog HDL language and implemented on Altera Stratix II EP2S30F672C3. The FPGA resource comparison of router architecture is shown in Table I. These two methods are also compared in different sizes of NoCs as shown in Table I.

From Table I, we observe that the proposed method could reduce the number of ALUTs for different type of routers, especially for the router with more ports. The reduction of 5-port router with proposed method could achieve 55.7%. We also observe that more reduction appears in large NoC with our proposed method. The reduction could achieve 52.0%, it is due to the large number of 5-port router in larger size of NoC.



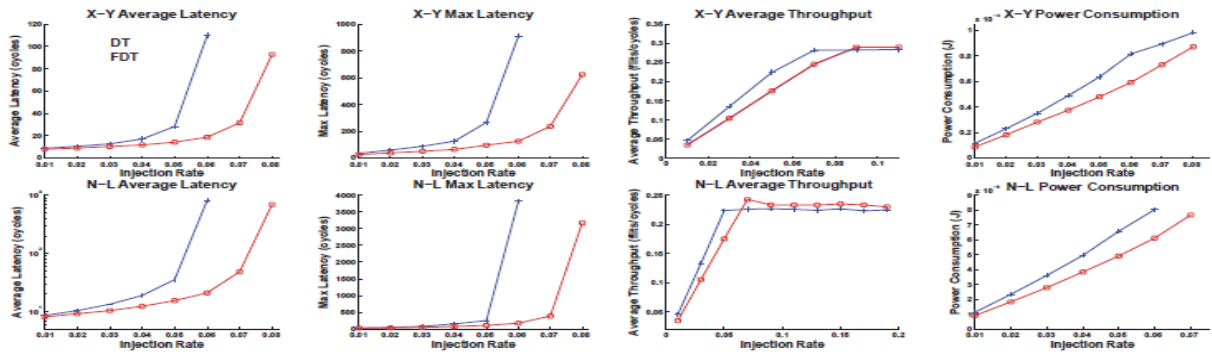


Fig. 4. Comparison of the FDT method and DT method with X-Y and N-L routing algorithms in 4x4 NoC.

TABLE II

COMPARISON OF AVERAGE HEADER SIZE.

NoC size	Proposed (bit)	Traditional (bit)
3x3	6	10.49
4x4	6	12.5
5x5	8	14.72
6x6	8	16.2
7x7	8	19.2
8x8	8	21.75

A. Performance and Power Consumption Evaluation by NoC Simulator

1) Simulation Environment and Experiment Setup: NOXIM NoC simulator [13] is used for evaluation, which is an extensible and modular simulator using system C.

The noticeable difference for these two methods is header size. As described in III-A, the average bit number of destination address of FDT method is decided by the NoC size. While the destination address of DT method is decided by the number of hops between each router. We assume each router has the same probability to send the packet to every routers in the whole network. Thus, the average hops number from one router to another router for different NoC whose size are 3x3, 4x4, 5x5, 6x6, 7x7, and 8x8 are 2.83, 3.5, 4.238, 4.734, 5.734, and 6.25 respectively. And each hop will cost 3-bit. This bit number added to 2-bit of flit type is the header size. The average header size of these two method in different network size is shown in Table II. The payload size in this experiment is set as 1~4 flit.

Our proposed architecture is evaluated by using different size of NoC from 3x3 to 8x8. Each size of NoC is used for running of ten experiments

with five for FDT method with different routing algorithms and five for the DT method. 10000 random packets are transmitted at random in different packet injection rate, and the buffer size is set as 4.

2) Simulation Results: FDT method and DT methods with different routing algorithms are compared in terms of average latency, max latency, throughput, and power consumption. The comparison results in 4x4 NoC and 8x8 NoC with X-Y, N-L are shown in Fig. 4 and Fig. 5. The comparison results with W-F routing algorithms is omitted here, because those results are similar with N-L routing algorithms. The F-A and N-A are also omitted here. The details of comparison results among all routing algorithms will be presented in another place.

B. In Fig. 4, the first row is the result for X-Y routing algorithm, the second for N-L. For different routing algorithm, FDT method could achieve low average latency, low max latency, and low power consumption. For X-Y and N-L routing algorithm, the higher throughput of the proposed method is obtained in high injection rate. The proposed method could not achieve higher throughput for N-F and F-A routing algorithm. Some special inflexion point of the injection rate is chosen to evaluate these two methods in average latency and power consumption. When the injection rate of these routing algorithms X-Y and N-L is 0.05, the average latency with the

proposed FDT method are reduced by 49.56% and 54.88% respectively.

The results in Fig. 5 are similar to Fig. 4, but FDT method is more effective. When the injection rate of these two routing algorithms is 0.022, the average latency are reduced by 69.17% and 61.50% with the proposed FDT method.

The latency  $T$  for transmitting  $L$  bits packet from source to destination in NoC with wormhole routers is expressed by

$$T = (L/BW + R) * H,$$

Where  $BW$  is the link bandwidth in bits per cycle;  $R$  is the routing delay per hop;  $H$  is the number of hops from the source node to the destination node [14].

In our work, FDT method can really reduce the packet size, and the packet size direct affects the latency and makes it lower.

The bigger the NoC size is, the more the proposed method could reduce the packet size. Furthermore, the reduction of latency is remarkable. The power consumption in packet transmission is decided by the routing algorithm and the number of flits. For the same routing algorithm and same injection rate, the number of packets is the same. But the proposed method has the smaller number of flits in one packet, thus the power consumption in packet transmission is smaller than the traditional method.

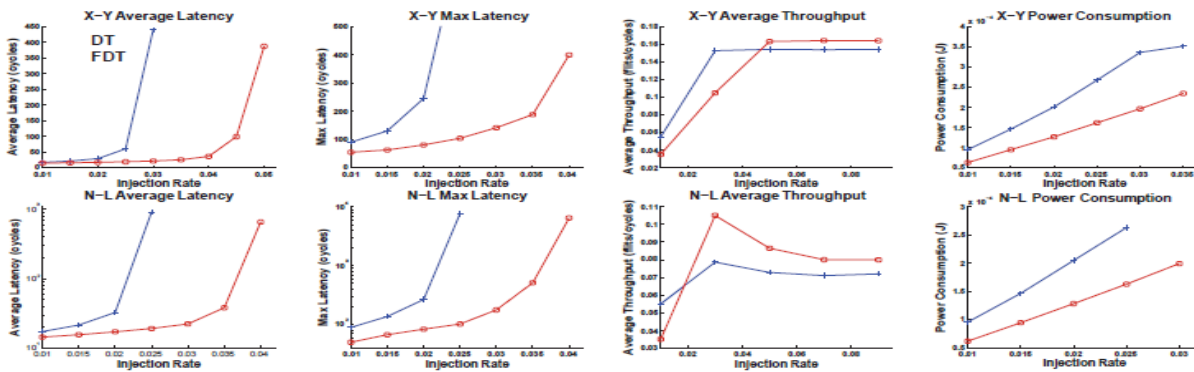


Fig. 5. Comparison of the FDT method and DT method with X-Y and N-L routing algorithms in 8x8 NoC.

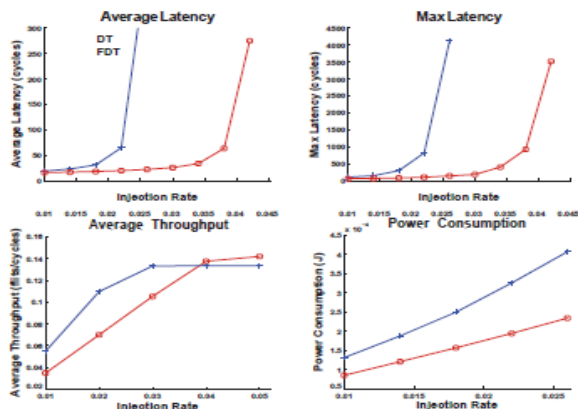


Fig. 6. Comparison result of the FDT method and DT method in 8x10 Intel 80-Tile NoC.

**V. ACTUAL APPLICATIONS**

FDT method is implemented for one actual applications to evaluate the performance. The application is Intel 80-Tile NoC [10] which 80 5-port routers are connected in mesh

topology. Each hop of this NoC with DT method needs 3-bit destination address. And the average number of hop is 6.925. Totally, the average header size is 20.775 bits. Using FDT method, the average header size is 9 bits. FDT method is implemented for this NoC architecture with X-Y routing algorithm, and compared with the original DT method in architecture level. The comparison result is shown in Fig. 6. We can see that the average latency, max latency and power consumption are reduced due to the proposed FDT method. Besides the higher throughput appears when the injection rate is larger than 0.04

**VI. CONCLUSION**

In this work, a new FDT hardware routing algorithm implementation method for high performance NoC architecture is proposed to replace the traditional DT method based NoC. The switch allocators for each port of a router

are improved and some redundant parts are reduced. The packet format is also changed to suit for the proposed method. Simulation results show that the NoC architecture with the proposed FDT method is effective in reducing circuit resource, average latency, max latency, power consumption and increasing average throughput. It is due to reducing in header size of the packet.

## REFERENCES

- [1] M. Reshadi, A. Khademzadeh, and A. Reza, "Elixir: A new bandwidth-constrained mapping for networks-on-chip," *IEICE Electronics Express*, vol. 7, no. 2, 2010.
- [2] L. Benini and G. D. Micheli, *Networks On Chips: Technology and tools*. Morgan Kaufmann, 2005.
- [3] J. Nurmi, H. Tenhunen, J. Isoaho, and A. Jantsch, *Interconnect-Centric Design for Advanced SOC and NOC*. Kluwer Academic Publisher, 2004.
- [4] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Surveys*, vol. 38, no. 1, 2006.
- [5] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proc. of the 38th Design Automation Conference (DAC)*, June 2001.
- [6] J. Kim, D. Park, T. Theochar, N. Vijaykrishnan, and C. R. Das, "A low latency router supporting adaptivity for on-chip interconnects," in *Proceedings of the 42nd Design Automation Conference (DAC)*, 2005.
- [7] A. Jantsch and H. Tenhunen, *Networks on Chip*. Kluwer Academic Publishers, 2003.
- [8] R. Marculescu and P. Bogdan, "The chip is the network: Toward a science of network-on-chip design," *Foundations and Trends in Electronic Design Automation*, pp. pp. 371–461, Mar. 2009.
- [9] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, 2004.
- [10] S. Vangal and et al., "An 80-tile 1.28tflops network-on-chip in 65nm cmos," in *Proc. ISSCC'07*, Feb 2007, pp. 5–7.
- [11] Y. P. Dong, C. Li, K. Kumai, Y. H. Li, Y. Wang, and T. Watanabe, "A new flexible network on chip architecture for mapping complex feedforward neural network," *Journal of Signal Processing (JSP)*, vol. 13, no. 6, pp. 453–462, Nov. 2009.
- [12] J. Duato, S. Yalamanchili, and N. Lionel, *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann, 2002.
- [13] F. Fazzino, M. Palesi, and D. Patti, "Noxim noc simulator," in <http://sourceforge.net/projects/nox>.
- [14] Z. Lu, A. Jantsch, and I. Sander, "Feasibility analysis of messages for on-chip networks using wormhole routing," in *Proceedings of the Asian*