Forecasting-based Dynamic Virtual Channels Allocation for Power Optimization of Network-on-Chips

Amir-Mohammad Rahmani, Masoud Daneshtalab, Ali Afzali-Kusha, Saeed Safari, Masoud Pedram[†]

Nanoelectronics Center of Excellence
School of Electrical and Computer Engineering
University of Tehran
{am.rahmani, m.daneshtalab}@ece.ut.ac.ir,
{afzali, saeed}@ut.ac.ir

[†]*Department of Electrical Engineering-Systems*
University of Southern California
Los Angeles, CA 90089
pedram@usc.edu

Abstract

In this paper, we present a dynamic power management technique for optimizing the use of virtual channels in network on chips. The technique which is called dynamic virtual channels allocation (DVCA) makes use of the traffic conditions and past buffer utilization to dynamically forecast the number of virtual channels that should be active. In this technique, for low (high) traffic loads, a small (large) number of VCs are allocated to the corresponding input channel. This provides us with the ability to reduce the power consumption of the router while maintaining the data communication rate. To assess the efficacy of the proposed method, the network on chip has been simulated using several traffic profiles. The simulation results show that up to 35% reduction in the buffer power consumption and up to 20% savings in the overall router power consumption may be achieved. Finally, the area and power overheads of the technique are negligible.

1. Introduction

Reducing feature sizes into the nanoscale regime and the trend towards integrating more functionality onto a single chip led to the rise of the System-on-Chip (SoC) paradigm which could have area, power, and delay problems [1][2][3][4]. The architecture used for the data communication in these systems is one of the components strongly affecting the area, power, and delay as three critical design parameters. Networks on Chip (NoCs) were proposed as a solution for the SoC interconnect power and delay problem. Among different components of routers, buffers consume a large amount of dynamic power which increases rapidly as the packet flow throughput increases [5][6]. Increasing the buffer size improves the performance of the interconnection network significantly at the price of a higher power consumption and, hence, the buffer size should be optimized [6]. One of ways to reduce the buffer size is to use the wormhole routing [7]. The latency of data communication in NoCs is one of the key design parameters which should be minimized. One of the ways to minimize the latency is to use virtual channels (VCs) which provide virtual communication path between routers as the main elements for the data communication in NoCs. Virtual Channel (VC) [8] decouples buffer resources from transmission resources. This decoupling allows active message to pass blocked messages using the available network bandwidth that would otherwise be left idle. In addition to avoiding deadlock situations [9], virtual channels increase network throughput by up to 40% over a wormhole router without VCs and reduce the dependence of throughput on the depth of the network [8]. The use of VCs, which increases the communication throughput, increases the power consumption of NoCs. The power consumption is also a crucial parameter in NoCs which should be minimized. To optimize the power, one ought to employ power efficient designs for routers.

In this paper, a dynamic power management technique for reducing the power consumption of the NoCs with virtual channels is proposed. The technique optimizes the number of active VCs for the router input channel based on the traffic condition and past link utilization. The rest of the paper is organized as follows. Section 2 presents the switch structure in NoCs while Section 3 describes the proposed forecasting-based dynamic virtual channels allocation architecture while the simulation results are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Switch Structure

In this work, we make use of a switch whose main structure is based on *RASoC* switch [10]. We have made some modifications to its buffering, routing and flow control parts and added support for Virtual Channels (VCs) based on [8] was needed. The switch contains two generic modules, namely, input channel and output channel parts. In this section, we describe the details of the switch.

2.1 Communication Model

The switch utilizes a handshake mechanism for its communication model. Switch communicates with its neighbor switches or cores by sending and receiving request and response messages. Each link includes two unidirectional channels in opposite direction to each other used to transmit data, framing and flow control signals. In addition to n bits for the data, there are two bits used for the packet framing which are bop and eop. The bop (begin-of-packet) is set only at the packet header, and eop (end-of-packet) is set just in the last payload word, which is also the packet trailer. Therefore variable packet length was supported.

2.2 Switching

The switch uses the wormhole packet switching approach [11] where messages are sent by means of packets composed of flits. A flit (*fl*ow control un*it*) which is equal to the physical channel word (or phit – *ph*ysical un*it*) has *n*+2 bits. It is the smallest unit over which the flow control is performed.

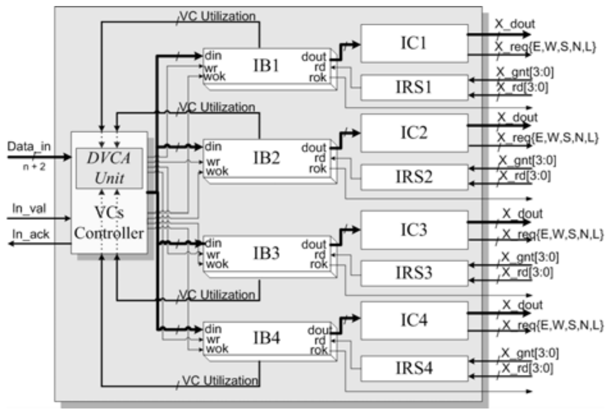

Fig. 1. Input channel architecture.

2.3 Routing and arbitration

The proposed switch supports different deterministic or adaptive routing algorithms such as XY [10], DyXY [12], and Odd-Even [13] used in the 2-D mesh topology. In addition, exhaustive round-robin [14] and priority-based [15] arbitration schemes are supported by the switch. Note that the switch supports locking mechanism required for the wormhole packet switching.

2.4 Flow control and VCs management

Since the handshaking mechanism is used for the communication, when a sender puts a data on the link, it activates the *val* (valid) signal. When the receiver receives the data, it activates the *ack* (acknowledge) signal. In our VC management approach, after the reception of each packet, *VC Controller* unit allocates one of the free VCs to this packet and locks this VC until the packet leave the VC.

2.5 Input channel and Output channel modules

The input channel module shown in Fig. 1 consists of four important units which are *VC Controller, IB* (Input Buffer), *IC* (Input Controller), and *IRS* (Input Read Switch). In this figure, four VCs are used for each input channel. The VC Controller exploits handshaking protocols for the flow control, allocation/deallocation of each VC to the input flow, and DVCA (Dynamic Virtual Channels Allocation) mechanism which will be in Section 3. The IB block is a p × (n+2)-bit FIFO buffer which is responsible to store the flits of the incoming packets while they can not be forwarded to an output channel. The number of the VCs is p. The IC block performs the routing function while IRS is responsible to deliver the read signal form the output channel to its connected IB.

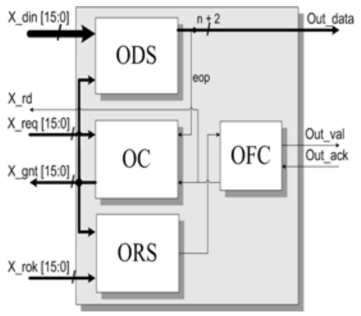

Fig. 2. Output channel architecture.

The output channel architecture of the proposed switch which is similar to RASoC switch [10] is depicted Fig. 2. It is composed of four blocks which are *OC* (Output Controller), *ODS* (Output Data Switch), *ORS* (Output Read Switch), and *OFC* (Output Flow Controller). The OC block runs the arbitration algorithm to select one of the requests sent by the VCs. Then, it activates the grant line of the selected request which induces proper switching of the ODS and ORS blocks. They connect the x_din and x_rok signals of the selected input channel to the external output channel interface. ODS and ORS blocks respectively connect the x_din and x_rok signals of the selected input channel to the external output channel interface However, before being connected to out_val, the x_rok signal pass through the OFC block.

3. Forecasting-based Dynamic Virtual Channels Allocation (DVCA) Architecture

Buffers are the single largest power consumer for a typical switch in an on-chip network [17] such as the Alpha 21364 router [17] the input buffers contribute 46-61% of the total power in a switch. This provides the motivation for us to analyze and optimize the power consumption of VCs without degradation in performance in the context of input channel switches for an on-chip network. In this work, we propose to use a dynamic power management (DPM) technique to dynamically determine the number of active VCs. The DPM technique has the objective of minimizing the power consumption with a minimal impact on the throughput. It also provides us with flexibility of adjusting the trade-off between the power and performance. The technique is based on a distributed forecasting-based policy, where each router input port predicts its future communication workload and required virtual channels based on the analysis of the prior traffic.

3.1 Communication Traffic Characterization

The characteristics of the communication traffic in an input channel may be captured using several potential network parameters. Different traffic parameters such as link utilization, input buffer utilization, and input buffer age have been proposed for simple input channels (without VCs) [18]. None of these parameters (indicators) alone may correctly represent the communication traffic in VC-based input channels. Thus, we need a combination of these parameters to explore suitable indicators that are useful for predicting the network load in VC-based input channels. In this work, we use the link utilization and the virtual channel utilization as explained here.

This *Link Utilization* parameter is defined as

$$LU = \frac{\sum_{t=1}^{H} A(t)}{N}, \quad 0 \le LU \le 1 \qquad (1)$$

where $A(t)$ is equal to 1, if the traffic passes through the link i in cycle t and 0 otherwise, and N is the number of clock cycles, which is sampled within a history window with the size of H defined in clock cycles. The link utilization is a direct measure of the traffic workload.

Regarding this parameter, it should be noted that when the network is lightly loaded or highly congested, the link utilization is low [18]. At low traffic loads, the link utilization is low due to the fact that the flit arrival rate is low. When the network traffic increases, the flit arrival rate between the adjacent routers and the link utilization of each link increases. When the network traffic approaches the congestion point, the number of free buffer spaces in the upstream router will become limited. This causes the link utilization to start to diminish. This observation reveals that the link utilization alone will not be sufficient for assessing the network traffic. The forecasting-based DVCA policy, therefore, requires more information for making a right decision. In this work, we use the utilization of each virtual channel to complement the link utilization indicator for the proposed forecasting policy.

As mentioned earlier, after receiving each packet the VC Controller allocates one available VC to the corresponding received packet and locks the VC (L = 1). When the packet completely leaves the router, the controller releases the VC (L = 0). The virtual channel utilization tracks how many locks of VC in the router input channel occurs.

Let us denote the number of cycles that each packet uses a VC if it is sent without interrupt by s. $L(s)$ is set when the corresponding VC is occupied during s cycles, n is the number of VC per input channel and H is the window size. We denote the virtual channel utilization by VCU and define it as the lock rate of each VC through H cycles obtained from

$$VCU = \frac{\sum_{s=1}^{H} L(s)}{H}, \quad 0 \le VCU \le 1 \qquad (2)$$

Also, the overall VCU, denoted by *OVCU*, is defined as the sum of VCUs of each input channel obtained from

$$OVCU = \frac{\sum_{i=1}^{n} VCU}{n}, \quad 0 \le OVCU \le 1 \qquad (3)$$

Table 1 shows a simple example of calculating the virtual channel utilization. In this example, H is equal to 5s, the number of virtual channels per input channel is four (n = 4), the numbers in the *VCx* columns are the packet number that uses a given VC at cycle i, and the numbers in *Lx* columns show the locking status of *VCx* at cycle i. The VCU of each VC in the window is given in the last shaded row. For this input channel, the OVCU is equal to 15/20 or 3/4 and LU is equal to 7/20.

Table 1. An example of calculating VCU

cycle	*VC1*	*VC2*	*VC3*	*VC4*	*L1*	*L2*	*L3*	*L4*
1	1	2	-	-	1	1	0	0
2	1	3	4	-	1	1	1	0
3	1	3	5	-	1	1	1	0
4	6	3	5	7	1	1	1	1
5	-	3	5	7	0	1	1	1
Total	2	2	2	1	**4**	**5**	**4**	**2**

We use the link utilization as the primary traffic indicator, while the virtual channel utilization is used as a litmus test for detecting the network congestion. Next, we will show the usage of these indicators for DVCA policy.

3.2 Forecasting-based DVCA Policy

In the proposed technique, the DVCA unit uses the LU and OVCU parameters for measuring the past communication traffic. Based on this, the communication traffic of the next period, the number of required active VCs is adjusted. To reduce the area and power overheads of the proposed unit, we should simplify the forecasting equation. For this, we combine the two measures using a simple weighted equation given by

$$CT = LU + W \times (OVCU - OVCU_{\min}),$$
$$0 \le W \le 1, 0 \le CT \le 1 \qquad (4)$$

where W is forecasting weight, CT is the communication traffic parameter, $OVCU_{\min}$ is the sum of all $VCU_{\min}$ for each input channel in each history interval, and $VCU_{\min}$ is the smallest possible lock rate for each VC. $VCU_{\min}$ occurs when there are no stalls for the packets to leave the corresponding VC and, hence, $\text{OVCU}_{\min}$ is equal to LU. In this equation, we set W to 0.5 which simplify Eq. (4) to a straightforward average equation (because $OVCU_{min} = LU$). As this equation implies, the communication traffic is a function of LU and the network load. The network load is proportional to the extra locks of VCs multiplied by the coefficient of *W*. The latter will be added to the link utilization when the congestion happens in VCs.

To make the forecasting formula reliable, we use an exponential smoothing function. This is simple and popular forecasting formula which is used in programming and inventories control science and defined as [19]:

$$CT_{predict} = CT_{past} + \alpha \times (CT_{actual} - CT_{past}) \quad \text{or}$$
$$CT_{predict} = \alpha \times CT_{actual} + (1 - \alpha) \times CT_{past} \quad (5)$$

where α is the forecasting weight $CT_{predict}$ is the predicted communication traffic, CT_{actual} is the actual communication traffic in the current history period, and CT_{past} is the predicted communication traffic in the previous history period (H). The accuracy of the prediction is a strong function of α and hence its value should be selected such that the error may be minimized.

The network traffic profile has short-term and long-term fluctuations. The proposed technique in this work filters out the short-term traffic fluctuations adapting the number of active VCs based on long-term traffic transitions. Based on the prediction, the controller decides to increase, decrease, or keep the same the number of active VCs. The pseudo-code of our proposed Forecasting-based DVCA policy for the case of four virtual channels per input channel is shown in Algorithm 1.

3.3 Hardware Implementation

Figure 3 shows the hardware realization of the proposed forecasting-based DVCA policy which relies on the local link and buffer information. Since the communication overhead of relying on global information is avoided, a simple hardware implementation may be used. To measure the link utilization (LU), a counter at each input port gathers the total number of packets that are passed from the link in each history interval. Similarly, there is a counter for each VC calculating the number of occurred locks (VCU). For computing the OVCU, an adder block is used to sum up the VCUs in each input channel. The *CT Calculator* carries out CT using Eq. (4). The *Forecasting Unit* uses the calculated CT and previous predicted CT from the previous interval (CT_{past}) to predict the new CT ($CT_{predict}$) for the next H period intervals. A register stores the $CT_{predict}$ to be used as the CT_{past} in the next interval.

To reduce the hardware overhead, we set α to 12/16, which is very close to the optimal values for this parameter for the traffic profiles used in this work. We implemented the division and multiplication operations by right and left shifts, respectively.

The Decision Logic unit determines the number of required active virtual channels (Required_VCs) using $CT_{predict}$, CT_{past} and required number of the virtual channels for the previous *H* period. Based on the Required_VCs value, the number of virtual channels in each input channel may be changed. The change in the number of active VCs is performed via clock gating technique. Finally, note that we simplify the division and multiplication operations by setting *H* and *H*×*n* values as power-of-two. For example, we set both *n* and *H* to 4.

Algorithm 1 Forecasting-based DVCA

```
$CT_{actual} = LU + (W \times (OVCU - LU))$
$CT_{predict} = CT_{past} + \alpha \times (CT_{actual} - CT_{past})$
**if** ($CT_{predict} > CT_{past}$) **then**
  **if** ($required\_VCs = 1$ **and** $CT_{predict} > (\frac{H \times (1) - 1}{H \times n})$ ) **then**
      $required\_VCs = required\_VCs + 1$
  **else if** ($required\_VCs = 2$ **and** $CT_{predict} > (\frac{H \times (2) - 1}{H \times n})$ ) **then**
      $required\_VCs = required\_VCs + 1$
      …
      …
  **else if** ($required\_VCs = n - 1$ **and** $CT_{predict} > \frac{H \times (n-1) - 1}{H \times n}$ )**then**
      $required\_VCs = required\_VCs + 1$
  **end if**
**else if** ($CT_{predict} < CT_{past}$) **then**
  **if** ($required\_VCs = n$ **and** $CT_{predict} < \frac{n-1}{n}$) **then**
      $required\_VCs = required\_VCs - 1$
  **else if** ($required\_VCs = n-1$ **and** $CT_{predict} < \frac{n-2}{n}$) **then**
      $required\_VCs = required\_VCs - 1$
      …
      …
  **else if** ($required\_VCs = 2$ **and** $CT_{predict} > \frac{1}{n}$ ) **then**
      $required\_VCs = required\_VCs - 1$
  **end if**
**end if**
$CT_{past} = CT_{predict}$
```

4. Results and Discussion

To assess the efficiency of the proposed technique to, we have compared NoCs with the forecasting-based DVCA and conventional virtual channel controllers. The comparison is performed in terms of power and latency for different traffic profiles. We used VHDL to develop six switches based the XY routing algorithm with 2, 4 and 8 virtual channels based on the conventional (non-DVCA) and DVCA and. They are labeled as XY-2VCs, XY-2VCs-DVCA, XY-4VCs, XY-4VCs-DVCA, XY-8VCs and XY-8VCs-DVCA, respectively. The simulations were carried out for a 5×5 mesh NoC using these six switch models. Also, we set *W*, α, and *H* to 1/2, 12/16 (75%), and 4, respectively. The performance of the network is evaluated using latency curves as a function of the packet injection rate (i.e., the number of packets injected to the network per cycle). The packet latency is defined as the duration from the time when the first flit is created at the source core to the time when the last flit is delivered to the destination core. For each simulation, the packet latencies are averaged over 250,000 packets. Latencies are not collected for the first 30,000 cycles to allow the network to stabilize. It is assumed that the packets have a fixed length of five flits, the buffer size of each virtual channel is five flits, and the data width is set to 32 bits. To perform simulations, we used *uniform* and *transpose* traffic patterns [20]. In the uniform traffic pattern, a core sends a packet to any other cores with an equal probability while in the transpose traffic pattern, a core at the position (i, j) only send packets to the core at the position ($5 - j$, $5 - i$).

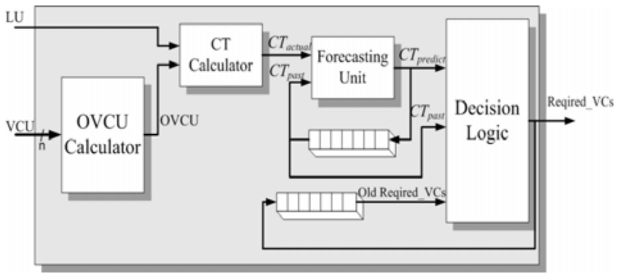

Fig.3. The hardware implementation of the forecasting-based DVCA policy.

The NoC performances for the two virtual channel management schemes under uniform and transpose traffic are given in Fig. 4 and Fig. 5. As observed from the figure, each pair of DVCA and non-DVCA with 2, 4, and 8 VCs have almost the same performance at low traffic loads. As the traffic load increases, the packet latency rises dramatically due to the network congestion. The results show that the conventional VC controller performs slightly better than the DVCA VC controller. This originates mainly from the prediction error. The power consumptions of the routers which are computed by Synopsis Power Compiler for a 0.13μm standard CMOS technology are presented in Fig. 6 and Fig. 7. As seen from the figure, the average power consumptions of the switches with the DVCA VC of the switches with the DVCA VC controller are considerably lower than the corresponding conventional switches at low traffic loads where some of the VCs may be clock-gated for saving the power. As the traffic load increases the difference between the average power consumptions of the switches with the same number of the VCs decreases till they eventually become almost the same at the congestion injection rate. The power saving is achieved at the price of slightly higher latency for the NoC with the DVCA VC controller. As we summed up the power consumption at each injection rate the power dissipation is reduced up to 35% in the buffer power consumption and up to 20% savings in the overall router power consumption. Finally, to determine the area and power overheads of the proposed technique, we synthesized the DVCA and conventional switches using Synopsys Design Compiler. Note that the controller is not on the critical path of the router and, hence, its delay overhead does not need to be considered. The synthesis results which are obtained for a 0.13μm standard CMOS technology are given in Table 2 and Table 3. The figures given in these tables reveal, the area and power overheads of the proposed forecast-based DVCA VC controller are negligible. Note that in estimation of power overhead, both dynamic and leakage power was considered in high traffic loads (worst-case).

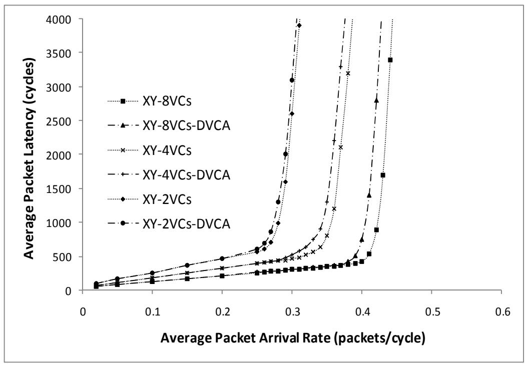

Fig. 4. Average latency under transpose traffic.

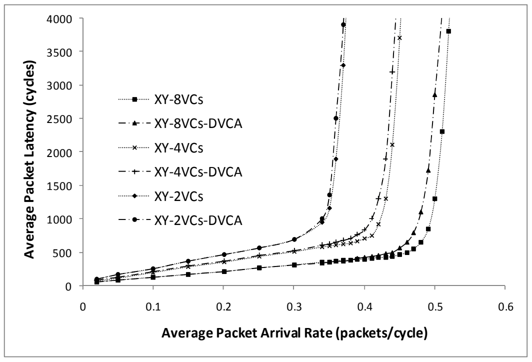

Fig. 5. Average latency under uniform traffic.

5. Conclusion

In this paper, we introduced a forecasting-based dynamic power management technique for controlling the number of active virtual channels (VCs). The approach makes use of the link and VC utilizations in predicting the communication traffic. Based on the predicted traffic, the number of the active virtual channels may be increased, decreased, or remain the same. The clock-gating power management technique is used to activate/deactivate the VCs. To determine the efficacy of the proposed technique NoCs with conventional and DVCA VC controller for 2, 4, and 8 VCs were simulated. The simulation results which performed for the uniform and transpose traffic profiles showed considerable power savings with a minimum impact on the latency for the proposed technique. The technique was implemented using a simple hardware to make the power and hardware overheads very small.

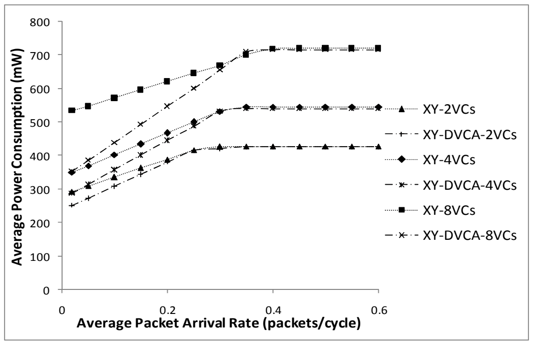

Fig. 6. Average power consumption under transpose traffic.

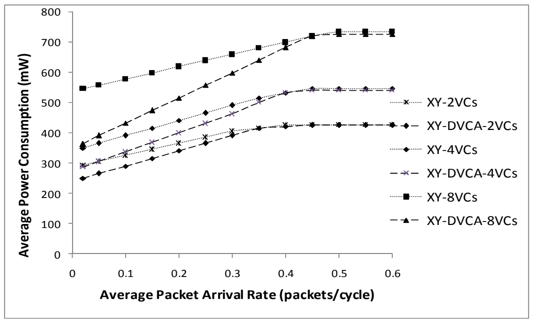

Fig. 7. Average power consumption under uniform traffic.

Table 2. Area overhead of the DVCA unit

Component	Area (μm^2)	Overhead (%)
DVCA Unit for 2 VCs	9296.81	2.78
DVCA Unit for 4 VCs	11918.68	1.99
DVCA Unit for 8 VCs	19863.11	1.09

Table 3. Power overhead of the DVCA unit

Component	Power (mW)	Overhead (%)
DVCA Unit for 2 VCs	4.86	1.7
DVCA Unit for 4 VCs	5.27	1.23
DVCA Unit for 8 VCs	8.55	0.89

Acknowledgement

The authors wish to acknowledge the financial support by Iran Telecommunication Research Center (ITRC) during the course of this project.

References

[1] L. Benini and G. D. Micheli, "Networks on chips: A new SoC paradigm," *IEEE Computer*, vol. 35, pp. 70–78, January, 2002.

[2] W.J. Dally et al., "Route Packets, Not Wires: On-Chip Interconnection Networks," *in Proc. of the DAC Conference*, pp.684-689, 2001.

[3] S. Heo and K. Asanovic, "Replacing global wires with an onchip network: a power analysis," *in Proc. of the ISLPED Conference*, pp. 369-374, 2005.

[4] R. Kumar, V. Zyuban, and D. M. Tullsen, "Interconnections in multi-core architectures: understanding mechanisms, overheads and scaling," *in Proc. of ISCA Conference*, pp. 408-419, 2005.

[5] W. Hangsheng, L. S. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," *in Proc. of the MICRO Conference*, pp. 105-116, 2003.

[6] T. T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," *in Proc. of DAC*, pp. 524-529, 2002.

[7] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. *IEEE Computer*, 26:62{76, Feb. 1993.

[8] W. J. Dally, "Virtual-channel flow control," *in Proc. of the ISCA*, pp. 60-68, 1990.

[9] W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on Computers*, pp. 547-553, 1987.

[10] C.A. Zeferino, M.E. Kreutz, A.A. Susin, "RASoC: A Router Soft-Core *for* Networks-on-Chip," *DATE*, Feb. 2004, pp. 198- 203.

[11] W. J. Dally and C. L. Seitz, "The torus routing chip," *Journal of Distributed Computing*, vol. 1(3), pp. 187-196, 1986.

[12] M. Li, Q.-A. Zeng, and W.-B. Jone, "DyXY – a proximity congestion-aware deadlock-free dynamic routing method for network on chip," *in Proc. of the DAC Conference*, July 2006, pp. 849–852.

[13] G. M. Chiu, "The odd-even turn model for adaptive routing," *IEEE Trans. on Parallel and Distributed Systems*, 11:729 – 738, July 2000.

[14] E.S .Shin, V.J. Mooney, G.F. Riley, "Round-robin Arbiter Design and Generation," *in Proc. of International Symposium on System Synthesis*, 2002, pp. 243-248.

[15] A. Bystrov, D. J. Kinniment, and A. Yakovlev, "Priority Arbiters," *in Proc. of the ASYNC Conference,* 2000, pp. 128-137.

[16] W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on Computers*, vol. C-36(5), pp. 547-553, 1987.

[17] S. Banerjee, and N. Dutt, "FIFO Power Optimization for OnChip Networks," *in Proc. of the 14th GLSVLSI Conference,* 2004, pp. 187-191.

[18] L. Shang, L. S. Peh, and Jha. N.K., "Dynamic Voltage Scaling with Links for Power Optimization of interconnection Networks," *in Proc. of the 19th HPCA Conference*, Feb. 2003, pp. 91-102.

[19] R .J. Tersine, *Principles of Inventory & Material Management*, Fourth Edition, Prentice Hall PTR, August 1994.

[20] M. Rezazad, H. Sarbazi-azad, "The Effect of Virtual Channel Organization on the Performance of Interconnection Networks," *in Proc. of the 19th IPDPS Conference*, April 2005, pp. 264-271.