Low Power NoC Switch using Novel Adaptive Virtual Channels

Rabab Ezz-Eldin[1], **Magdy A. El-Moursy**[2] **and Amr M. Refaat**[3]

[1]**Electrical and Computer Engineering Department, Bani-suef University,
Bani-suef, Egypt, Electronics Research Institute,
Cairo, Egypt**

[2]**Mentor Graphics Corporation, Cairo, Egypt**

[3]**Electrical Engineering Department, Fayoum University,
Fayoum, Egypt**

Abstract

Adaptive Virtual Channel (AVC) is proposed as a novel technique to achieve low power NoC switch. Power supply gating is employed to reduce the power dissipation of NoC switch without degrading network performance. Hierarchical multiplexing tree is used to achieve efficient AVC. AVC could reduce both dynamic and leakage power of the switch. Hierarchical multiplexing tree decreases the area of the switch which reduces the dynamic power by 60%. Using the leakage power reduction technique, the average leakage power consumption of Adaptive Virtual Channels is reduced by up to 97%.

Keywords: *Virtual Channels, NoC, power gating, hierarchical multiplexing.*

1. Introduction

As the technology continuously scales down, the need for high performance, low power, as well as high throughput and reliable integrated circuits increases. Integrated Circuits are moving towards System on a Chip (SoC) which increases the circuit complexity. In recent years the complexity of interconnection architectures of the SoCs increased significantly. Network on Chip (NoC) was proposed as a solution for the interconnection problem. NoC is an on-chip network composed of processing cores connected by switches and communication channels [1]. Each physical channel can be split into several virtual channels using multiple parallel buffers. All virtual channels share the bandwidth of the physical channel. Different number of virtual channels is previously used to improve the network throughput. As the number of virtual channels increases, network throughput increases [2]. A tradeoff between power dissipation of the circuit and network throughput exits. In the previous network implementations, fixed number of virtual channels was used [3-5]. Achieving high throughput while reducing power dissipation is the objective of this paper.

Power consumption grows rapidly in NoC as interconnection complexity increases [6]. Reducing the power consumption becomes the first objective in NoC design. Power consumption should be minimized for reliability and cost-efficiency. Dynamic power and leakage power are the main components of power dissipation in NoC. Reducing leakage power is taking a lot of attention since it is dominating the power dissipation in today’s and tomorrow’s technologies. The main focus of this paper is to present a new Adaptive Virtual Channel (AVC) technique as a novel technique to reduce power dissipation of NoC switch. AVC allows efficient power gating to be employed to reduce power dissipation of the switch as shown in Figure 1. AVC is used to reduce the leakage power of a network switch. Hierarchical multiplexing tree is shown to be efficient in reducing not only the leakage power but also the dynamic power of the switch.

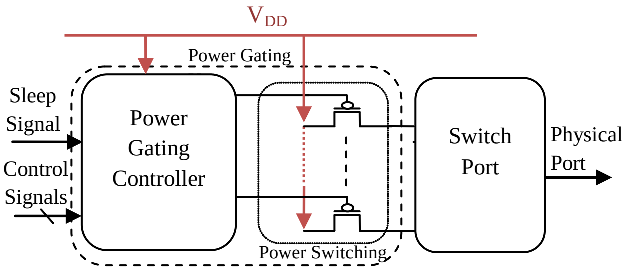

Figure 1: Block diagram of switch port with power gating

The paper is organized as follows: in section 2, AVC architecture is proposed. The power gating mechanism is presented in section 3. In section 4, simulation results are demonstrated. Conclusions are provided in section 5.

2. Adaptive Virtual Channel Architecture

Adaptive number of virtual channels is achieved in the proposed technique to multiplex the input channels to network switch. The characteristics of the network traffic are used as indicator to enable/disable the appropriate number of virtual channels. The number of available virtual channels is divided into power-of-two sets of configurable virtual channels. Each set could be configured as active or in-active. AVC multiplexing tree where the virtual channels are located at the leaves of the tree and the physical port is located at the root ($level_n$) is illustrated Figure 2. The tree is developed as a binary tree to optimize circuit implementation. Non-binary tree would complicate circuit implementation with limited flexibility of activating arbitrary number of virtual channels. The number of virtual channels equals 2^p where $p = m + n$. The number of connected virtual channels to a single cell (set of VC) is 2^m. n is the number of multiplexing levels. The maximum number of active sets in the tree is 2^n. m and n are positive integer numbers where $m \geq 1$ and $n \geq 0$. Each set is connected to one multiplexing cell located at the first multiplexing level. Every two cells in a low level of the tree are connected to one cell in the upper level. The total number of cells in the tree is given by

$$k = 2(2^n - 1) \tag{1}$$

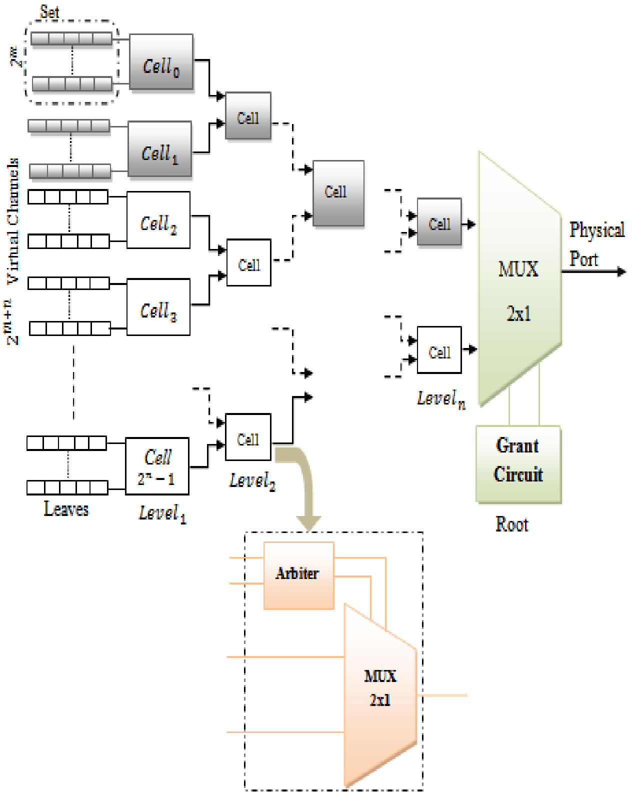

Figure 2: Adaptive Virtual Channel multiplexing tree structure

The root consists of one multiplexer 2x1 and one grant circuit 2x2 as shown in Figure 2. Every cell (in all levels expect $level_1$) of the tree consists of one arbiter 2x2 and one multiplexer 2x1. Cells in $level_1$ contains one multiplexer 2^m x1 and one arbiter 2^m x 2^m. At the root, only one virtual channel is granted the physical port. m and n introduce a single degree of freedom in designing the switch. The tree structure can be created with p different implementation options. m and n defines a tradeoff between circuit delay and configurability. For n equals zero, the tree contains only the root. Therefore, no multiplexing tree is required. All virtual channels operate simultaneously. Eliminating the multiplexing hierarchy reduces the circuit delay. However, all virtual channels are included in the root. The flexibility of configuring the virtual channels is minimized and no saving in power is possible.

The delay of the tree structure increases with increasing the number of multiplexing levels n. However, the area of the switching circuitry decreases as n increases since the hardware implementation is optimized. The flexibility of activating/deactivating the virtual channel sets increases as n increases which allow saving in power components.

The virtual channels are activated in groups, according to the network traffic. Multiplexing tree activation is highlighted in Figure 2 for m =1 and for light traffic (*i.e.* only two sets are activated). All upstream cells connected to the active virtual channel sets are, accordingly, activated. In-active virtual channels are power gated to reduce the leakage power dissipation as described in section 3. n should be maximized to maximize the circuit configurability, minimize circuit area, and maximize the power saving as describe in section 4.

3. POWER GATING MECHANISM

In order to reduce power dissipation, virtual channels are deactivated when the network traffic is light. The hierarchical multiplexing structure is exploited to configure the virtual channel sets to active/in-active mode according to network traffic. Power supply gating is employed to deactivate the cells and the virtual channels connected to them. Each cell has one power gating switch. Deactivating the virtual channel sets reduces the leakage power dissipation of the whole switch without degrading the network throughput since it is applied according to traffic characteristics. Power management is performed using a power gating controller in addition to the sleep transistors. In section 3.1, the sleep controller unit is described. A mechanism to size the sleep transistor of a multiplexing cell is presented in section 3.2.

3.1 The Sleep Controller

A power gating control unit is used to control activating the virtual channels. The controller manages the sleep transistor of each cell. The sleep controller unit has two inputs and one output signal. A *cntrl_sleep* signal is used to enable/disable the switch port. This signal turns off the whole switch. The *nt* signal (*n* bits) indicates the status of the network traffic.

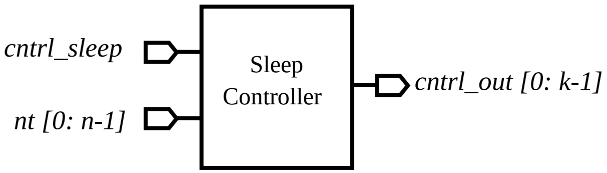

Figure 3: The sleep controller block diagram

It is required to find a general canonical representation of the truth table for the first level of the *cntrl_out* of the sleep controller. Taking into consideration that the number of bits for inputs and outputs change depending on the traffic heaviness and the number of controlled cells, cells are indexed as shown in Figure 2. Therefore, the upper cell in the $level_1$ of the tree has the lowest index of zero. The index increases going from top to bottom. The index of the bottom cell in $level_1$ is $(2^n - 1)$ as shown in Figure 2. The equation is algebraically expressed in a sum of minterms form

$$cntrl\_out_x(nt_0, nt_1, .. nt_{n-1}) = \sum(0,1, \dots \dots, x-1)$$

$$\text{where} \quad 1 \leq x \leq 2^n - 1, \qquad (2)$$

where x is the index of the cell. This equation produces (2^n *−1*) columns of the output truth table, each column controls one cell in the $level_1$. At $x = 0$, the column of $cntrl\_out_0$ equals zeroes which means always activate $Cell_0$ regardless the traffic heaviness. Turning on a child cell in $level_1$ requires turning on all parents cells of this child. Therefore, the truth table of every cell in $level_1$ which has even index is the same truth table of all parent cells in the same path from $level_1$ to the root.

The output signal *cntrl_out* has *k* bits depending on the number of cells in the switch port as shown in Figure 3. The *cntrl_out* signal is used to manage sleep transistors according to the required number of virtual channels to be activated. Depending on the value of the *cntrl_out* signal, some sleep transistors are switched ON to activate its connected cells and the other sleep transistors are switched OFF to ensure that the connected cells are deactivated.

When *cntrl_sleep* signal is 1, all the bits of the *cntrl_out* are 1 and hence all the sleep transistors will be switched OFF. Thereby the switch port is forced to turn off regardless of the value of the *nt*. This increases the power saving of the port as shown in section 4. When *cntrl_sleep* signal is 0, the sleep controller calculates the value of the *cntrl_out* signal according to input signal *nt* which activates certain number of virtual channels. Activating virtual channels depends on the traffic heaviness which can take different levels. The number of traffic heaviness levels depends on granularity of activating the virtual channels which equals 2^n. For example, for number of virtual channels of eight and for *m=1*, there are four levels of traffic heaviness (*n=2*), “Very Heavy”, ”Heavy”, ”Light”, “Very Light”. For *m=2*, there are only two levels of traffic heaviness, “Heavy” and ”Light”. With very heavy traffic profile, all virtual channels are activated by switching ON all cells.

The granularity of activating the virtual channels is 2^m. For *m=1*, binary multiplexing tree is used and two virtual channels are activated at a time. For *m=2*, four virtual channels are activated simultaneously. As *m* increases, the depth of the multiplexing tree decreases reducing the area overhead and the critical path delay. On the other hand, for small *m*, larger power saving is achieved since power gating could be applied with higher granularity. Hierarchical switching increases the flexibility of activating the virtual channels making the switch more adaptive to the changes in the traffic characteristics. Accordingly, reducing *m* increases the power saving.

3.2 Circuit Implementation of Power Switching

The power switching block consists of *k* sleep transistors. In our architecture, the sleep transistors are implemented by PMOS transistors to gate the power supply path to ground. Sleep transistor acts as a switch to turn-off the supply voltage during the sleep mode. On the other hand, sleep transistors in the active mode are ON and hence the value of the virtual supply node is v_{DD}. Sizing the sleep transistor affects both circuit performance as well as the efficiency of power saving.

A tradeoff in sizing the sleep transistor exists. During the active mode, the sleep transistor impedes the flow of the supply current requiring the transistor to be up sized to keep circuit performance. On the other hand, sizing up the sleep transistor reduces its ability to mitigate the leakage current and power. In addition, the power gating control circuit dissipates more dynamic power with larger sleep transistor.

The traffic heaviness signal *nt* is assumed to arrive to the target switch one clock cycle before the actual cycle at which the signal is needed to activate the cells. This assumption allows only one clock cycle to switch the cell from sleep-to-active mode. The switching time from sleep-to-active TSA must be less than or equal to the critical path delay t_d of the cell circuitry.

$$TSA \leq t_d \tag{3}$$

The cell is considered active when its virtual v_{DD} node reaches 90% of v_{DD}. Sizing sleep transistor and its implication on the circuit performance is demonstrated in section 4.

4. SIMULATION RESULTS

The proposed architecture is implemented using the ADS tools. 45nm technology is used with supply voltage of 1V. A switch port with eight virtual channels is considered. The tradeoff between the TSA switching time, the critical path delay of the cell, and the leakage power reduction is presented in Figure 4. Leakage power increases with sizing up the sleep transistor. To increase power saving, the sleep transistor needs to be sized down. On the other hand, TSA could not be larger than t_d since only one clock cycle is needed to switch the cell from sleep-to-active mode. The intersection point on t_d and TSA curves is used as the optimum size for high performance and low power switch design. Based on that, the width of sleep transistor is determined to be 0.35µm for the target technology.

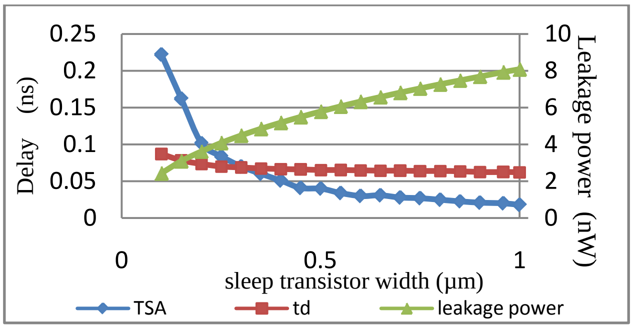

Figure 4: Sleep mode leakage power, critical path delay and TSA for different sleep transistor widths

4.1 Depth of the multiplexing tree

For eight virtual channels, there are p =3 implementation options

$$\begin{cases} option\ A: & m = 3, & n = 0\,, & k = 0 \\ option\ B: & m = 2, & n = 1, & k = 2 \\ option\ C: & m = 1, & n = 2, & k = 6 \end{cases} \tag{4}$$

For option *A*, The tree structure consists of only the root. For option *B*, the available virtual channels are divided into two sets using four virtual channels per set. For option *C*, the available virtual channels are divided into four sets using binary multiplexing tree and two multiplexing levels. There are a total of six cells in the tree where at least two can be, simultaneously, activated.

The required area to implement the three options, including the area of the sleep controller and sleep transistors, is shown in Figure 5. The area of multiplexing tree of option *C* is less than the area of the multiplexing tree of option *A* and *B*. As compared to option *A*, the area decreases in option *B* by 50.93% and by 60.11% for option *C*. The overhead in the input gate capacitance (sleep controller and sleep transistors) in option *C* is 6.6% of the total port capacitance. In option *B*, the overhead is only 3.61%.

The hierarchical tree implementation has two-fold effect in reducing power dissipation of the switch. The dynamic power is reduced for the reduction in the input gate capacitance of the switch. With hierarchical multiplexing, dynamic power could decrease by up to 60%. In addition, the leakage power is decreased with light traffic since power gating is more efficient.

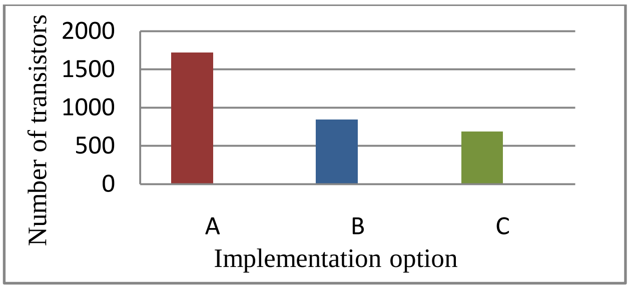

Figure 5: The area of switch port for different number of virtual channel per one set

On the other hand, the hierarchical tree structure increases the critical path delay of the circuit reducing the maximum operation frequency. The maximum operation frequency and leakage power for the three implementation options are listed in Table 1. The maximum operation frequency and leakage power decrease with increasing the number of levels. The leakage power for option *C* reduces by 87.12 % as compare to the leakage power of option *A*. A pipeline stage could be used to maintain the operating frequency but latency of switching would increase.

Table 1. Maximum operating frequency and leakage power for three implementation options with different hierarchical depths

m	***Maximum operation frequency***		***Maximum leakage power***	
	(GHz)	***Reduction (%)***	***(nW)***	***Reduction (%)***
3	18.99	-	2821.11	-
2	12.18	35.86	616.85	78.13
1	8.44	55.76	363.32	87.12

4.2 Power dissipation of AVC

For total number of virtual channels of eight (*p* =3), *m =1* and *n*=2. There are four levels of traffic heaviness listed in Table 2.The network traffic is used to control the number of active virtual channels. Since *n* =2, two, four, six or eight virtual channels could be simultaneously activated depending on the traffic of the network.

The reduction in leakage power dissipation is reported in Table 2 for different network traffic characteristics. The power saving increases as the number of active virtual channels decreases. Power saving could reach up to 81% when only two virtual channels are simultaneously activated. When no virtual channels are activated, power saving increases up to 97%. Adaptive virtual channel with hierarchical multiplexing tree significantly decreases the power consumption of the switch.

Table 2. the leakage power saving with different number of virtual channels

Number of Virtual channel	***6***	***4***	***2***	***0***
Traffic heaviness	Heavy	Light	Very Light	No traffic
Power saving (%)	17.3	34.1	80.2	96.8

5. CONCLUSIONS

Adaptive Virtual Channel is proposed as an efficient novel technique to reduce power dissipation of NoC switch. AVC uses hierarchical multiplexing tree and power gating mechanism to reduce both dynamic and leakage power dissipation of the switch. The virtual channels are activated based on the network traffic. The area of switch port reduces with increasing hierarchical levels which decreases the dynamic power by up to 60%. Power saving increases with decreases the number of active virtual channels. The reduction in leakage power dissipation could reach 81% when only two virtual channels are activated simultaneously using AVC. At in-active mode of the switch port, power saving could increase up to 97%.

References

[1] Fernando Moraes, Ney Calazans, Aline Mello, Leandro Möller, Luciano Ost,”HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip”, Integration, the VLSI Journal, Vol.38 (1), October 2004, pp. 69-93.

[2] William J. Dally, “Virtual-Channel Flow Control”, IEEE Transactions on parallel and distributed systems, Vol. 3, No.2, March 1992, pp. 194-205.

[3] Partha Pratim Pande, Cristian Grecu, Michael Jones,Andre ´ Ivanov, Resve Saleh, “Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures”, IEEE Transactions on computers, Augusts 2005, pp. 1025-1040.

[4] Partha Pratim Pande, Cristian Grecu, André Ivanov, Res Saleh, “High-Throughput Switch-Based Interconnect for Future SoCs”, In Proceedings of the 3rd IEEE International workshop on SoC for real-time applications, July. 2003, pp. 304-310.

[5] Aline Mello, Leonel Tedesco, Ney Calazans, Fernando Moraes, “Virtual Channels in Networks on Chip: Implementation and Evaluation on Hermes NoC”, In Proceedings of the Integrated Circuits and Systems Design, September 2005, pp. 178 – 183.

[6] Srinivasan Murali, David Atienza, Paolo Meloni, Salvatore Carta, Luca Benini, Giovanni De Micheli, Luigi Raffo, “Synthesis of Predictable Networks-on-Chip-Based Interconnect Architectures for Chip Multiprocessors.”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 15, No. 8, August 2007, pp. 869-880.

Rabab Ezz-Eldien received the B.Sc. degree in Electronics and communications department with honors from the Faculty of Engineering, Fayoum University, Fayoum, Egypt, in 2004. She is currently working a research assistant in Electrical and Computer department at Bani-suef University. She is joined the M.Sc program in Fayoum university in 2009. Her areas of interest include Networks-on-Chip, Computer Architecture and Embedded System.

Magdy A. El-Moursy was born in Cairo, Egypt in 1974. He received the B.S. degree in electronics and communications engineering (with honors) and the Master's degree in computer networks from Cairo University, Cairo, Egypt, in 1996 and 2000, respectively, and the Master's and the Ph.D. degrees in electrical engineering in the area of high-performance VLSI/IC design from University of Rochester, Rochester, NY, USA, in 2002 and 2004, respectively. In summer of 2003, he was with STMicroelectronics, Advanced System Technology, San Diego, CA, USA. Between September 2004 and September 2006 he was a Senior Design Engineer at Portland Technology Development, Intel Corporation, Hillsboro, OR, USA. During September 2006 and February 2008 he was assistant professor in the Information Engineering and Technology Department of the German University in Cairo (GUC), Cairo, Egypt. Dr. El-Moursy is currently a Technical Lead in the Mentor Hardware Emulation Division, Mentor Graphics Corporation, Cairo, Egypt. His research interest is in Networks-on-Chip, interconnect design and related circuit level issues in high performance VLSI circuits, clock distribution network design, and low power design. He is the author of more than 30 papers, four book chapters, and one book in the fields of high speed and low power CMOS design techniques and high speed interconnect.

Amr M. Gody; Joined Cairo University, faculty of Engineering in 1986. He is earned BSc. in Electronics and communication engineering in 1991 with an honor degree. He is earned the M.Sc degree in Electronics and communication engineering in 1995 from Cairo University, faculty of Engineering. He is joined the PhD program in Cairo university in 1996. He is earned the PhD in 1999 in the field of speech signal processing. Amr is Associate professor in Fayoum University, Electrical engineering department and he is acting as head of Electrical Engineering since 2010 till now.