A 256-Radix Crossbar Switch Using Mux-Matrix-Mux Folded-Clos Topology

Sung-Joon Lee and Jaeha Kim

Abstract—This paper describes a high-radix crossbar switch design with low latency and power dissipation for Network-on-Chip (NoC) applications. The reduction in latency and power is achieved by employing a folded-clos topology, implementing the switch organized as three stages of low-radix switches connected in cascade. In addition, to facilitate the uniform placement of wires among the sub-switch stages, this paper proposes a Mux-Matrix-Mux structure, which implements the first and third switch stages as multiplexer-based crossbars and the second stage as a matrix-type crossbar. The proposed 256radix, 8-bit crossbar switch designed in a 65nm CMOS has the simulated power dissipation of 1.92-W and worst-case propagation delay of 0.991-ns while operating at 1.2-V supply and 500-MHz frequency. Compared with the state-of-the-art designs in literature, the proposed crossbar switch achieves the best energy-delay-area efficiency of 0.73-fJ/cycle·ns· λ^2 .

Index Terms—High-radix crossbar switch, networkon-chip, folded-clos, mux-matrix-mux architecture, digital integrated circuits

I. INTRODUCTION

A high-bandwidth, high-radix crossbar is a critical component to sustain the performance growth of Network-on-Chip's (NoCs) with the Moore's law. NoCs

E-mail : sjlee@mics.snu.ac.kr, jaeha@snu.ac.kr

such as multi-core processors and FPGAs [1, 2] aim to mitigate the adverse scaling of on-chip interconnects and design complexity by exploiting a regular structure [3]. For these systems, the functionality and performance of the system improves as the more elements can be integrated on a single die increases. However, as the number of elements increases with the CMOS technology scaling, crossbar switches with low radix, low bandwidth, and high power consumption can quickly become the bottleneck to the overall NoC's performance and power dissipation. It is particularly the case since the power and delay of a crossbar switch can increase rapidly as its radix increases. To address this challenge, this paper presents a high-radix crossbar switch design which significantly improves the power dissipation and latency performance.

The major issue of implementing a high-radix crossbar switch is that its propagation delay and power dissipation rapidly increases with its radix. For example, in the case of the basic matrix crossbar switch shown in Fig. 1, its delay increases with the rise of the radix k, while the power increases as k^2 . It is primarily because the loading of each input port includes the gate capacitance of k tri-state buffers and the loading of each output port includes the junction capacitance of k tri-state buffers. Hence, the delay and power of each path scale with k. On the other hand, since the crossbar switch has k paths that can be simultaneously active, the overall power consumption scales as k^2 . Thus, this increase in delay and power must be mitigated when designing a high-radix crossbar switch.

To deal with the increase in power and delay, the previous crossbar switches described in [4, 5] used partially-activated signal paths by employing selectable

Manuscript received May. 12, 2014; accepted Nov. 4, 2014 Department of Electrical and Computer Engineering and Interuniversity Semiconductor Research Center at Seoul National University, Seoul, Korea



Fig. 1. A basic k-radix matrix crossbar switch.

repeaters and multiple hierarchies in the design. Such a technique can reduce the impact of power and delay scalings, but the area of crossbar switch increases due to the larger number of gates required. For instance, in case of [6], the crossbar switch is divided into three pipelined stages to increase the delay by 1.34-ns, but the power and area of crossbar switch increased by 5.5-W and 16-mm², respectively.

This paper adopts a folded-clos topology [7] in order to design a high-radix crossbar switch with reduced power and delay. The folded-clos topology is one of the network-within-network topologies, constructing a large network by putting together a set of small sub-networks in a hierarchical fashion. When such a network-withinnetwork topology is used, the delay and power can improve since they are small multiples of the delay and power of the smaller-radix sub-switch, respectively. Therefore, if the multiplying factor is less than the scaling rate with the radix, the overall power and delay can be reduced. In this paper, a folded-clos topology is investigated as a way to explore this possibility of reducing the power and delay.

In addition, this paper presents the novel circuit implementation and physical floorplanning to reduce the impacts of the on-chip wires on power and delay. For instance, a Mux-Matrix-Mux structure is proposed to reduce the wire lengths between the folded-clos stages. Also, an internal hierarchy is introduced in the second-



Fig. 2. A 16-radix folded-clos crossbar switch using 4-radix sub-switches as unit elements.

stage matrix sub-switch.

The rest of the paper is organized as follows. Section II describes the architecture of the proposed folded-clos crossbar switch and Section III discusses the circuit implementation of the proposed Mux-Matrix-Mux crossbar switch design. Section IV describes the overall floorplan of the crossbar switch's physical design and presents the simulation results of the crossbar switch designed in a 65 nm CMOS technology. Section V concludes the paper.

II. PROPOSED FOLDED-CLOS CROSSBAR Switch Architecture

The proposed k-radix crossbar switch adopting the folded-clos topology is shown in Fig. 2, in the case of k is 16. The k-radix folded-clos crossbar switch has a hierarchical architecture consisting of three stages, where each stage is organized with \sqrt{k} sub-switches with radix- \sqrt{k} . The connections between the stages are made by connecting each sub-switch of the second stage to all the sub-switches in the first and the third stage. In the case of k is 16, 16-radix folded-clos crossbar switch is organized using 4-radix sub-switches, while each stage has four sub-switches.

Adopting the folded-clos topology allows to mitigate the rapid scaling of power and delay with the increasing radix k. Assuming that all the sub-switches in the foldedclos architecture are implemented as a basic matrix switch, the delay and power of a single sub-switch scales with \sqrt{k} and $(\sqrt{k})^2 = k$, respectively. Since each input signal to the folded-clos crossbar switch passes through three sub-switches to reach the output, the delay of the folded-clos crossbar switch increases as $3 \cdot \sqrt{k}$. In case of the power, the total number of the sub-switches in the folded-clos topology is $3 \cdot \sqrt{k}$, and therefore the power consumption of the folded-clos crossbar switch increases as $3 \cdot \sqrt{k} \cdot k = 3 \cdot k^{\frac{3}{2}}$. Thus, it is expected that the delay and power of the high-radix crossbar switch improve when the folded-clos topology is adopted, since the scaling factors of the power and delay of the foldedclos crossbar switch is smaller than those of the basic matrix switch, which are *k* and *k*², respectively.

However, a naive realization of the folded-clos topology in Fig. 2 can result in long and unequal wire lengths which are not amenable to IC implementation. The main problem lies with the interconnect wires between the different stages. For instance, the output signals from the sub-switch of the first stage shown in Fig. 2 have to reach all the sub-switches in the second stage. Therefore, the worst-case distance of the wire is as long as the vertical dimension of the overall crossbar. More importantly, since the number of wires traveling a long vertical distance between the stages is large, large area must be dedicated for the interconnect wires, further increasing their lengths. For instance, in case of a 256radix 8-bit folded-clos crossbar switch, the number of parallel, vertical wire tracks required between the stages is $\sqrt{256} \cdot 8 \cdot (\sqrt{256} - 1) = 1920$, which takes substantial amount of area, incurring power and delay penalties and resulting in non-uniform lengths.

This paper solves this problem by using different types of switches for each level of the folded-clos crossbar, making the worst-case wires short and uniformly distributed. Recall that the problem with the homogeneous folded-clos crossbar stems from the fact that the large number of parallel wire tracks is required between the stages. To address this, the proposed structure in Fig. 3 implements the second stage as a matrix-type crossbar switch making the wire tracks uniformly distributed. The wire tracks in the second stage in Fig. 3 are shorter than the worst-case wire distance in



Fig. 3. The proposed 16-radix mux-matrix-mux folded-clos based crossbar switch.

the homogeneous folded-clos crossbar since the wires in the second-stage sub-switches can be shared with the output wire tracks of the first stage and input wire tracks of the third-stage sub-switches.

III. CIRCUIT IMPLEMENTATION

The Mux-Matrix-Mux structure is implemented by splitting the second-stage sub-switches and placing their gates at the cross-points between the output wire tracks of the first-stage and input wire tracks of the third-stage, as shown in Fig. 4. In a 16-radix folded-clos design, the *n*-th switch in the second stage receives the *n*-th outputs from the first stage switches and drives the *n*-th inputs of the third stage switches where n is an integer ranging from 1 to 4 as shown in Fig. 4(a). Exploiting this regularity, when the first stage is placed vertically and the third stage horizontally, the n-th switch in second stage can be split and its gates be placed in the cross-points between the output wire tracks of the first-stage switches and input wire tracks of the third-stage switches, as shown in Fig. 4(c). This design principle is extended to a 256-radix, 8-bit Mux-Matrix-Mux crossbar switch with chains of inverters inserted to improve the delay. The rest of the section describes in more detail the implementation of each stage of the 256-radix 8-bit Mux-Matrix-Mux crossbar switch.

As previously mentioned, the switches in the first and



Fig. 4. Illustration of the switch distribution in the second-stage sub-switch (a) Highlighting one switch in the second switch, (b) The matrix-type implementation of the switch using tri-state buffers to make interconnect wires short and uniform, (c) Placement of these.

third stages are designed as 16-radix 8-bit multiplexerbased switches. To place them in the Mux-Matrix-Mux configuration shown in Fig. 3, the switches in the first and third stages must be able to route the signals first horizontally and then vertically. It can be realized by placing the first-stage 8-bit multiplexers horizontally, routing their wire track vertically. Similarly, the third stage multiplexers are placed vertically and their wire tracks are routed horizontally as shown in Fig. 5.

The 16:1 multiplexers in the 16-radix multiplexerbased switches are implemented as a single stage rather than multiple stages since the additional stages can degrade the delay performance and power dissipation and increase the total area of the proposed crossbar switch. For instance, while it is possible to implement the 16:1 multiplexer as two stages of 4:1 multiplexers, it causes



Fig. 5. The multiplexer-based implementation of the first and third stage 16-radix switches.



Fig. 6. Proposed second stage matrix switch.

significant increase in the wire lengths in the secondstage matrix and the overall area.

On the other hand, the second-stage switches are designed as sixteen 16-radix 8-bit matrix switches in order to distribute the interconnect wires uniformly between the first-stage and third-stage switches as shown in Fig. 6. In addition, the second-stage matrix switches are implemented with an internal hierarchy in order to further reduce the delay and power dissipation. The second stage matrix has more available spaces than the



Fig. 7. Layout example of (a) the 256-radix 1-bit basic matrix switch, (b) the proposed second stage matrix switch.



Fig. 8. The circuit schematics of (a) a transparent D-latch, (b) two-phase latch pair for the switch configuration.

full-matrix crossbar, and they can be used by the additional gates to implement the hierarchical structures as shown in Fig. 7. The proposed design has 4 segments of 4 tri-state buffers at each of the input bits and 2 segments of 8 tri-state buffers at each of the output bits. Each signal passes through a total of three tri-state buffers in the second stage.

The path selection is programmed using a scan chain which is made of a series of two-phase latches shown in Fig. 8, assuming the configuration of crossbar switch doesn't have to be change frequently. This is the case when the crossbar switch is employed in NoC using static routing algorithm [8]. The *data_out* signal of the two-phase latch drives the select signal of each tri-state buffer. The scan chain can consume larger area than other configuration memories, such as SRAMs, its area should be considered into the floorplan.

IV. SIMULATION RESULTS

The presented 256-radix, 8-bit crossbar switch is designed in a 65 nm CMOS technology. The floorplan of the proposed design is shown in Fig. 8. The first-stage switch is placed vertically on the left side, and the third-stage switch is placed horizontally on the bottom. The second-stage matrix switch is placed in the center, providing the routing paths between the first and second stages. The overall crossbar switch places the input ports on the left and output ports on the bottom.

To make a realistic estimation on the wire parasitics, the physical layout dimensions of the crossbar switch are estimated based on the detailed floorplan shown in Fig. 9. First, the area of the individual cell blocks is estimated using the following empirical equation [9]:

Area in
$$\lambda^2 = \sum_i 6 \cdot W_i \cdot L_i + 360 \cdot \# of \ all \ transistors$$
(1)

In (1), the first term estimates the area occupied by the transistors and the second term estimates the area occupied by the wires. Here, λ is equal to 35 nm for a 65 nm CMOS technology. Second, assuming that the wires have 4- λ width and 4- λ spacing, we can estimate the width of the wire tracks running vertically and horizontally. Using the area estimates of the cells and wire tracks, we can determine the total area of each stage, and estimate the length of wire tracks. The wires under special considerations are the input wire tracks to the first-stage multiplexer of which length is equal to the sum of the horizontal and vertical dimensions of the stage, and the output wire tracks of which length is equal to the horizontal dimension of the stage. The critical wires in the matrix stage are the input wire tracks to the matrix stage, of which length is equal to the horizontal



(b)

Fig. 9. (a) Floorplan of the first and third mux stage, (b) Overall floorplan of proposed architecture.

length of the stage, and the output wire tracks of which length is a half of the vertical length of the stage, and finally the wire tracks in the internal hierarchy of the matrix stage of which length is one quarter of the horizontal length and a half of the vertical length of the stage.

For the purpose of simulating the worst-case delay and power, we can compose the critical path of the designed crossbar switch as shown in Fig. 10, based on the



Fig. 10. Schematic of the critical-path of proposed architecture.

floorplan shown in Fig. 9. The first and third stages of the critical path consist of a chain of inverters, input wire parasitic loads, 16:1 multiplexers, and output parasitic loads. The second stage of the critical path includes another chain of inverters, tri-state buffers, and parasitic loads between the tri-state buffers due to the wire tracks in the internal hierarchy.

The power dissipation and propagation delay of the designed crossbar switch operating at a 1.2-V supply and 500-MHz frequency can be estimated from the critical path circuit shown in Fig. 10. Table 1 compares the achieved results with those of the state-of-the-art crossbar switches in literature, where λ is 45 nm, 35 nm, and 22 nm in 90 nm, 65 nm, and 45 nm CMOS technology, respectively.

To enable a comparison between the state-of-the-art crossbar switches, a normalized metric is suggested. Due to the increase in the radix and bit-width, delay, power, and the area of the crossbar switch increases as described in Section I. As the radix k of the basic matrix crossbar switch increases, the number of gates increases with k for a single critical path. The delay and energy per cycle for a critical path increases with k. For the bit-width b

	Process (CMOS)	Radix	Bit	Pipelined	Delay (ns)	Power (W)	Input Freq. (MHz)	Area (λ^2)	FOM ($fJ^{-1}/cycle^{-1} \cdot ns^{-1} \cdot \lambda^{-2}$)
[4]	45 nm	64	8	No	0.744	0.119	2000	1.65 · 10 ⁸	$6.02 \cdot 10^8$
[5]	45 nm	64	8	No	0.55	0.081	1000	7.41·10 ⁷	133·10 ⁹
[6]	90 nm	128	32	Yes	1.34	5.5	750	7.90·10 ⁹	$4.64 \cdot 10^{8}$
This work	65 nm	256	8	No	0.991	1.92	500	8.64 · 10 ⁸	1.37 • 109

Table 1. Crossbar switch performance summary

of the basic matrix crossbar switch, as *b* increases, the parasitic capacitance of the critical path increases due to increase of area. Since delay increases with the fan-out, which is the ratio between the input and output capacitance of a gate, the scaling effect of delay due to bit-width is negligible, while energy per cycle increases with *b* since it is proportional to the switching capacitance which includes parasitic capacitance. In case of area, the total number of gates increases as k^2 and b^2 , hence the area of the basic matrix crossbar switch increases as k^2 and b^2 . Therefore, we have defined a figure-of-merit (FOM) based on the scaling effect of the basic matrix switch, as shown below:

$$FOM = \left(\frac{Delay}{Radix} \cdot \frac{Energy / Cycle}{Radix \cdot bit} \cdot \frac{Area}{Radix^2 \cdot bit^2}\right)^{-1}$$
(2)

The result shows that the figure-of-merit of proposed design is 1.09, 1.03, and 2.95 times better than the crossbar switches presented in [4-6], respectively, and achieves 3.05, 1.37 and 2.29 times smaller normalized area, which is the area divided by $Radix^2 \cdot bit^2$. On the other hand, The normalized energy per cycle, which is energy per cycle divided by radix and bit, is 0.25, 0.34, and 0.47 worse than the switches presented in [4-6], respectively. This is because the proposed crossbar switch is more optimized to delay rather than the power. As a result, normalized delay, which is the delay divided by radix, is 3.00, 2.21, and 2.70 better than the crossbar switches presented in [4-6], respectively.

V. CONCLUSIONS

This paper presented the design of a low-power and low-latency 256-radix crossbar switch employing a foldedclos topology. Its hierarchical architecture and Mux-Matrix-Mux heterogeneous implementation effectively mitigated the adverse effects of the interconnect parasitics on the switch's power dissipation and latency as the radix increases. The designed 256-radix, 8-bit crossbar switch operating at 1.2V and 500-MHz in 65 nm CMOS demonstrates the 0.991-ns latency and 1.92-W power, and achieves the best figure-of-merit among the previouslyreported state-of-the-art designs in [4-6].

ACKNOWLEDGMENTS

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1301-08.

REFERENCES

- T. Wu, et al., "A 2Gb/s 256*256 CMOS Crossbar Switch Fabric Core Design using Pipelined MUX," *in proc. International Symposium on Circuits and Systems (ISCAS)*, pp. 568-572, May 2002.
- [2] K. Choi and W. Adams, "VLSI Implementation of a 256x256 Crossbar Interconnection Network," *in proc. International Parallel Processing Symposium* (*IPPS*), pp. 289-293, Mar. 1992.
- [3] I. Shamim, "Energy Efficient Links and Routers for Multi-Processor Computer Systems," *Master's Thesis, Massachusetts Institute of Technology*, pp. 1-96, Sep. 2009.
- [4] D. Song and J. Kim, "A Low-Power High-Radix Switch Fabric Based on Low-Swing Signaling and Partially-Activated Input Lines," *in proc. VLSI Design, Automation, and Test (VLSI-DAT)*, pp. 1-4, Apr. 2013.
- [5] J. Ryoo, et al., "Design of Low-Power High-Radix Switch Fabric with Partially-Activated Input and Output Lines," *International SoC Design Conf.* (ISOCC), pp. 227-230, Nov. 2012.
- [6] G. Passas, et al., "Crossbar NoCs Are Scalable Beyond 100 Nodes," *IEEE Trans. Computer-aided Design of Integrated Circuits and Systems*, pp. 573-585, Apr. 2012.
- [7] J. Ahn, et al., "Network within a Network Approach to Create a Scalable High-Radix Router Microarchitecture," *in proc. High Performance Computer Architecture (HPCA)*, pp. 1-12, Feb. 2012.
- [8] A. Agarwal, et al., "Survey of Network on Chip (NoC) Architectures & Contributions," *Journal of Engineering, Computing and Architecture*, pp. 21-27, 2009.
- [9] F. Moraes, et al., "Estimation of layout densities for CMOS digital circuits," *International Workshop on Power and Timing Modeling Optimization and Simulation (PATMOS)*, pp. 61-71, Oct. 1998.



Sung-Joon Lee received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, Korea in 2014. He is currently pursuing towards a M.S. degree at the same university. His

research interests include low-power mixed signal systems and their design methodologies.



Jaeha Kim is currently an Assistant Professor at Seoul National University (SNU), Seoul, Korea and his research interests include low-power mixedsignal circuits and their design methodologies. He received the B.S. degree in electrical engineering from

Seoul National University in 1997, and received the M.S. and Ph.D. degrees in electrical engineering from Stanford University in 1999 and 2003, respectively. Prior to joining SNU, Prof. Kim was with Stanford University, Stanford, CA, U.S.A. as Acting Assistant Professor, with Rambus, Inc., Sunnyvale, CA, U.S.A. as Principal Engineer, and with Inter-university Semiconductor Research Center (ISRC) at SNU as Post-doctoral Researcher. Prof. Kim is the recipient of the Takuo Sugano Award for Outstanding Far-east Paper at 2005 International Solid-State Circuit Conference (ISSCC).