Design of Low-Power and Low-Latency 256-Radix Crossbar Switch Using Hyper-X Network Topology

Seung-Heon Baek, Sung-Youb Jung, and Jaeha Kim

***Abstract*—This paper presents the design of a low-power, low area 256-radix 16-bit crossbar switch employing a 2D Hyper-X network topology. The Hyper-X crossbar switch realizes the high radix of 256 by hierarchically combining a set of 4-radix sub-switches and applies three modifications to the basic Hyper-X topology in order to mitigate the adverse scaling of power consumption and propagation delay with the increasing radix. For instance, by restricting the directions in which signals can be routed, by restricting the ports to which signals can be connected, and by replacing the column-wise routes with diagonal routes, the fanout of each circuit node can be substantially reduced from 256 to 4~8. The proposed 256-radix, 16-bit crossbar switch is designed in a 65 nm CMOS and occupies the total area of 0.93×1.25 mm^2. The simulated worst-case delay and power dissipation are 641 ps and 13.01 W when operating at a 1.2 V supply and 1 GHz frequency. In comparison with the state-of-the-art designs, the proposed crossbar switch design achieves the best energy-delay efficiency of 2.203 cycle/ns·fJ·λ^2.**

***Index Terms*—Crossbar, hyper-X, low-power, switchcore, NoC**

I. INTRODUCTION

This paper presents a low-power, high-radix crossbar switch employing a 2D Hyper-X network topology. A crossbar switch is a circuit block that can route the data arriving at one of its input ports to an arbitrary output ports. It is a critical component for Network-on-Chips (NoC's), of which good example is a multi-core processor. Due to demand for high performance and diverse functionality of these multi-core processors, the need for high-radix crossbar switches is rising. Here, the radix of the crossbar switch refers to the number of its input ports or output ports that can be fully routed. However, the power and area consumptions as well as the propagation delay of a crossbar switch increase rapidly as the radix increases, posing a challenge to the design of such a high-radix crossbar switch. This paper presents a 256-radix, 16-bit crossbar switch that aims to mitigate the adverse scaling of power and delay by adopting a 2D Hyper-X network topology and introducing three novel modifications to it.

The key challenge with implementing a high-radix crossbar switch is that its power consumption and propagation delay degrade rapidly with the increasing radix N. For instance, for a basic matrix crossbar switch shown in Fig. 1, the delay and power scale as N and N^2, respectively. Each input port is connected to the inputs of N tri-state buffers in the same row and each output port is driven by the outputs of N tri-state buffers in the same column. The loadings on the input and output lines consist of the gate and junction capacitances of N tri-state buffers, respectively. Therefore, the loading on each node increases proportionally with radix N and the overall delay increases proportionally with N. On the

Manuscript received May. 12, 2014; accepted Nov. 17, 2014
A part of this work was presented in Korean Conference on Semiconductors, Seoul in Korea, Feb. 2014

Seoul National University
E-mail : shbaek@mics.snu.ac.kr

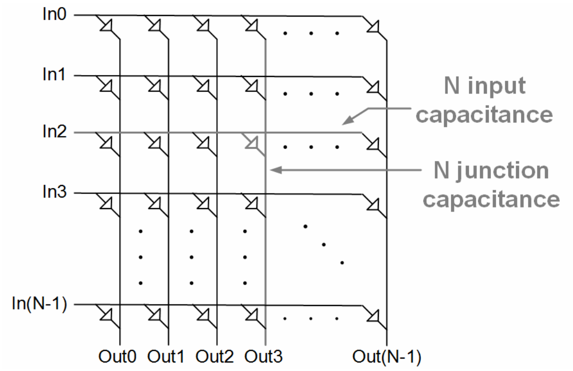


Fig. 1. A basic N-radix matrix crossbar switch.

other hand, the maximum power consumption is proportional to N^2 since there are total of N paths each consisting of the loading of N. The adverse scaling of delay and power makes the realization of high-radix crossbar switches difficult.

Hierarchical, network-within-network architectures like 2D Hyper-X topology [4] can mitigate this scaling of power and delay. In these architectures, the network is built in a hierarchical fashion by composing a set of small sub-networks. For instance, if a full N-radix crossbar can be realized with M^2 of N/M-radix sub-switches, the delay and power of the entire crossbar can be less than those of the basic matrix crossbar.

Hyper-X network topology is one of the network-within-network topologies. A regular Hyper-X topology is a multi-dimensional, directed network where the routers in each dimension are fully connected [3]. In this paper, a 2D Hyper-X topology is adopted.

In addition, this paper proposes three modifications to the basic Hyper-X topology that can further enhance the power and delay performance of the crossbar switch. First, the directions in which the data can be routed are restricted. Second, the ports to which the data can be connected are restricted based on their originating external input ports. Third, the column-wise routes are replaced with diagonal routes. These modifications substantially reduce the fanout of each circuit node from the maximum of 256 to 4~8.

The rest of the paper is organized as follows. First, Section II describes the proposed Hyper-X crossbar switch architecture with the additional three modifications. Then, Section III discusses the detailed circuit implementation of each sub-switch, which is composed of an input driver stage, upper and lower mux stages, and scan chain modules. Section IV describes the floorplan of the crossbar switch physical design with a purpose of estimating the wire loads realistically. Based on the floorplan, Section V presents the simulation results of the proposed crossbar switch and compares its energy-delay efficiency with the other designs reported in literature.

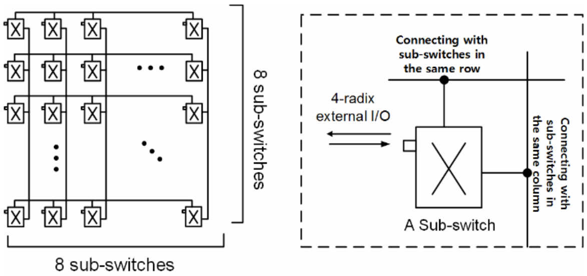

8 sub-switches
8 sub-switches
Connecting with sub-switches in the same row
4-radix external I/O
A Sub-switch
Connecting with sub-switches in the same column

Fig. 2. A 256-radix crossbar switch employing a basic 2D Hyper-X topology.

II. Hyper-X Crossbar Switch Architecture

The proposed 256-radix crossbar switch adopts a 2D Hyper-X topology using 4-radix sub-switches as unit elements. Fig. 2 illustrates the topology of a basic 2D Hyper-X crossbar switch and architecture of the sub-switch. It is organized as an 8-by-8 matrix of 16-bit, 4-radix sub-switches. The radix of 4 is chosen as a compromise between the propagation delay which favors a low radix and complexity which favors a high radix. This parameter can be resized with design. Each sub-switch has four external input ports and four external output ports, therefore making the total of 256 input ports and 256 output ports of the 256-radix crossbar. And each sub-switch has 4 internal outputs and 60 internal inputs connected to sub-switches in the same row or column. Within the matrix, each sub-switch has internal, bidirectional I/O connections with all the sub-switches on the same row or column. Therefore, the data entering into one of the sub-switches can reach any other sub-switch in the matrix and leave the crossbar with at most two hops.

To further improve its power and delay performances, three novel modifications are proposed that aim to reduce the number of loads that each sub-switch needs to drive. The load capacitance of each sub-switch's internal output is proportional to the number of sub-switches it can be

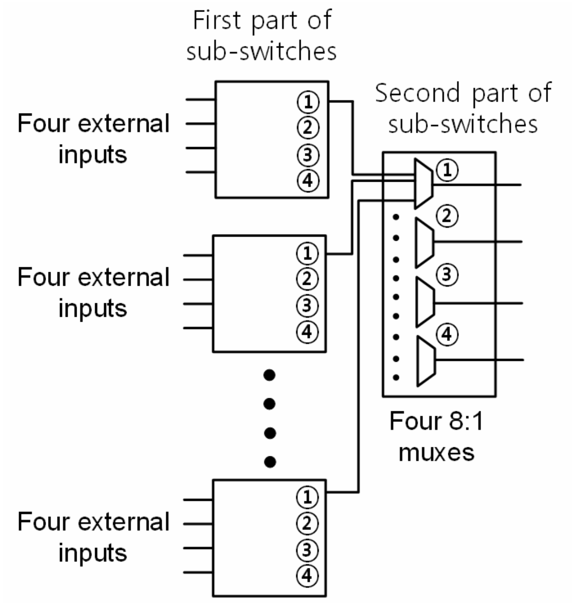


Fig. 3. Illustration on the external input ordering and restriction on their connections with the sub-swithces.

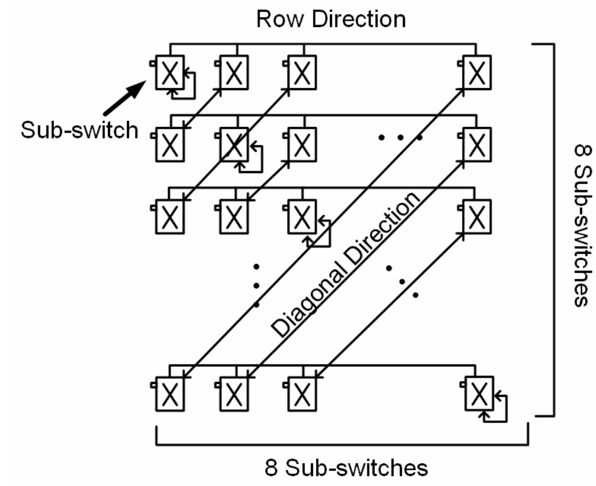


Fig. 4. Replacing column-wise connections with diagonal connections to lower the fanout.

possibly connected to, adversely affecting the delay as well as the power dissipation of the driver. Therefore, one way to lower the power and delay of the crossbar switch is to reduce the number of possible paths between the sub-switches.

The first modification restricts the degree of freedom in routing the data. For instance, when the data arriving at one external input port needs to reach an external output port located at a different position of the sub-switch matrix, the data must be routed first in the row-wise direction and then in the column-wise direction. In other words, the paths that route the data in the column-wise direction first and row-wise direction next are removed. With this modification, the number of possible paths originating from one sub-switch output node is reduced from 60 to 32, without limiting the routing freedom of the overall crossbar switch.

The second modification restricts the way each sub-switch connects to another sub-switches located in the same row, as illustrated in Fig. 3. This modification is achieved by dividing each sub-switch into two parts. In the first part, four external inputs are rearranged and numbered from 1 to 4. In the second part, there are four muxes numbered also from 1 to 4. And each mux determines to receive which data within the eight same numbered data from the first part of all switches in the same row. With this modification, each internal output port is routed only to the corresponding output ports of the first and second parts. This modification reduces the number of possible paths in the row-wise direction originating from each output of first and second part from 32 to 4 and 8, respectively.

The third modification replaces the column-wise connections among the sub-switches with diagonal connections as shown in Fig. 4. The number of possible paths for a diagonal hop is 4, which is smaller than the number of possible paths for a column-direction hop of 32. But with a diagonal hop, a hop to any random sub-switch in matrix can't be completely achieved in only two or three hops because the two data routes may have to use the same diagonal path. Therefore, at least four redundant diagonal paths are required. The order of hops in each routing is row first, diagonal second, row third and diagonal last. As a result, a column directional hop is replaced with a diagonal hop and a row directional hop. With this modification, the number of possible paths in the column-wise direction is reduced from 32 to 4 and 8 for the diagonal and row-wise direction hop, respectively.

Fig. 5 illustrates the organization of the sub-switch, supporting the Hyper-X crossbar switch with the described three modifications. Each sub-switch receives 4 external inputs and produces 4 external outputs. The overall crossbar switch contains 64 sub-switches, therefore receiving 256 external inputs and producing 256 outputs in total. The sub-switch is organized as two parts: upper and lower sub-switches. The upper one handles the row-wise propagations before the diagonal

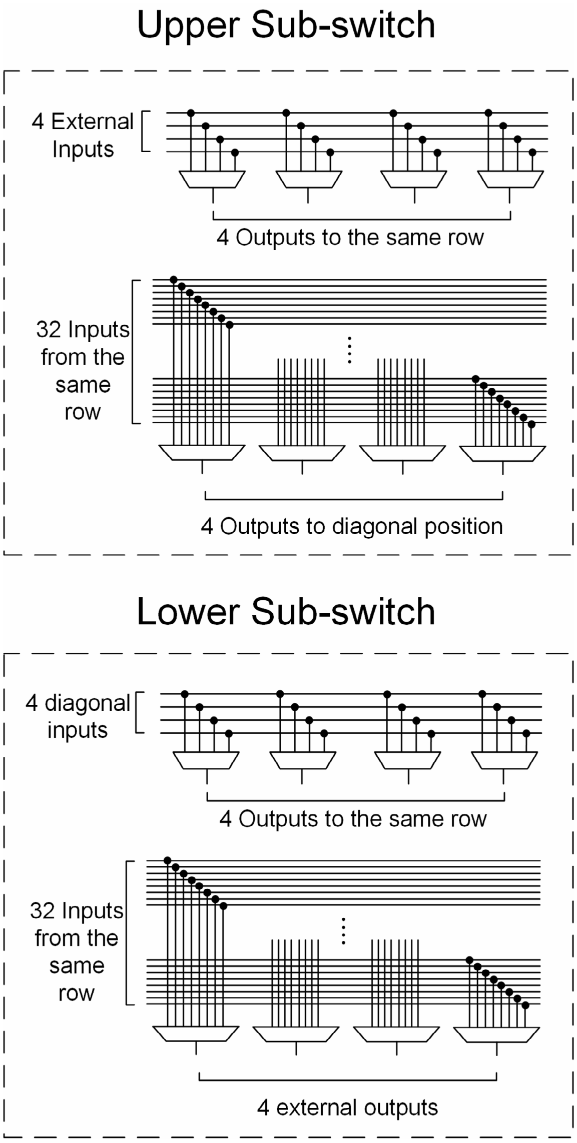

Upper Sub-switch
4 External Inputs
4 Outputs to the same row
32 Inputs from the same row
4 Outputs to diagonal position
Lower Sub-switch
4 diagonal inputs
4 Outputs to the same row
32 Inputs from the same row
4 external outputs

Fig. 5. The proposed micro-architecture of each sub-switch composing the Hyper-X crossbar switch.

routes and the lower one handles the row-wise propagations after the diagonal routes. Both the upper and lower sub-switches are made of four 4:1 muxes and four 8:1 muxes. All the multiplexers are 16-bit wide. The 4:1 muxes route the signals coming from the external inputs or diagonal routes in the row-wise direction. And the 8:1 muxes route the signals in the diagonal direction.

To avoid the possible reduction in throughputs due to path confilcts, the number of available tracks between the sub-switches is doubled. Since the proposed crossbar routes each input signal to an output port using a total of 4 hops, the resulting path diversity remains the same with the case of a matrix crossbar.

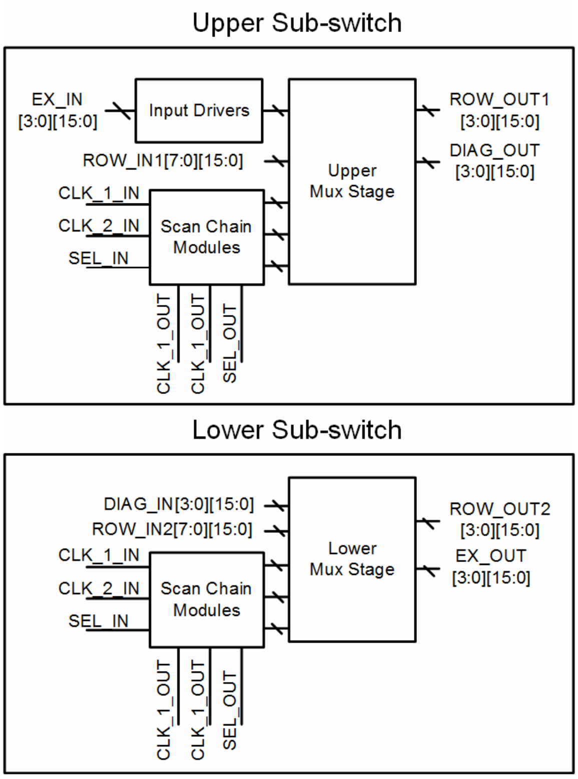


Fig. 6. The sub-switch organization.

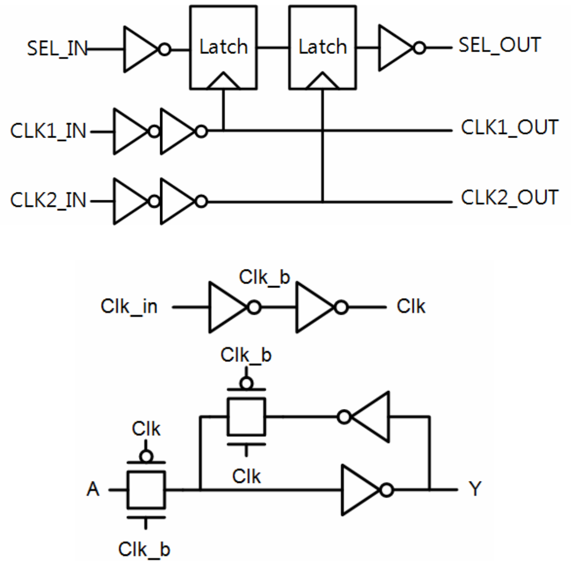


Fig. 7. Schematics of the scan-chain module.

III. Circuit Implementation

Fig. 6 illustrates the more complete implementation of the sub-switch including the input driver stages and scan-chain modules for configuration. Each input signal is buffered by three inverters in series before driving the

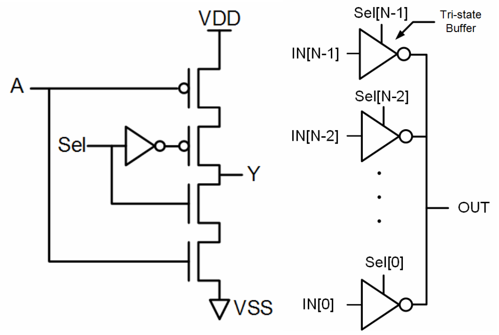


Fig. 8. Schematics of the tri-state buffer and N-to-1 mux used in the sub-switches.

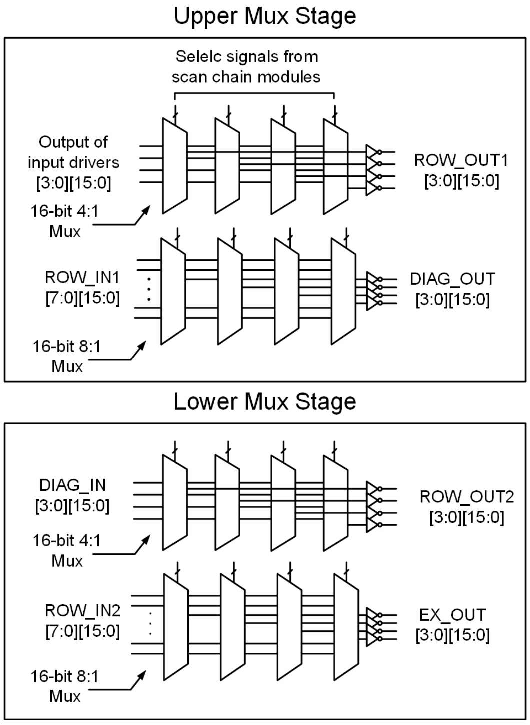


Fig. 9. The details of the upper and lower mux stages.

internal circuits. The select signals for the muxes are provided by the scan chain modules.

The scan chain is constructed by connecting scan chain modules serially. Each scan-chain module is composed of two latches and six inverters and a latch is composed of two transmission gates and four inverters. The schematics for the latch and scan-chain module are shown in Fig. 7.

Finally, the multiplexers are implemented with CMOS tri-state buffers as shown in Fig. 8. Each of the upper or lower mux stage is composed of four 4:1 16-bit muxes and four 8:1 16-bit muxes and the inverters at each output nodes serve as buffers (Fig. 9).

IV. Layout Floorplanning

In this paper, the power and delay performances of the proposed Hyper-X crossbar switch is assessed mainly by simulation, using realistic estimation on wire parasitic loads. To estimate the wire parasitic load, the floorplan of the entire crossbar switch should be determined. The lengths of wires in the critical path can be estimated by visualizing the critical path on floorplan. For determining the size of the floorplan, the areas of sub-switches were estimated first using the following empirical formula [9]:

$$\text{Area}\left(\lambda^2\right) = \left(\Sigma_{all\ transistors} 6WL\right) + \left(360 \cdot \#of\ all\ transistors\right) \quad (1)$$

The areas of the upper and lower sub-switches can be estimated using (1) based on the total number of transistors and the sum of their sizes. The first term estimates the area occupied by the transistors and the second term estimates the area occupied by the wires connecting the transistors [5]. Here, 1λ is roughly a half of the transistor's minimum length. In a CMOS 65 nm technology chosen, 1λ is equal to 35 nm. The overall sizes of the upper and lower sub-switches are estimated to be 93.5 μm×77.75 μm and 140 μm×77.75 μm, respectively.

To reduce the wire length in the diagonal directions, each row of the Hyper-X matrix is separated into upper and lower rectangular row blocks. And these row blocks are placed as 4×4 matrix as in Fig. 10. The upper or lower row block contains all the upper or lower sub-switches in the row of the same number, respectively. The routings shown on the floorplan can be categorized into two types: the diagonal hops between an upper row block and a lower row block and the row-wise hops between the blocks on the same row.

The diagonal hops are realized by routing 16 row blocks as shown in the right part of Fig. 10. Eight diagonal outputs of each upper row-block are connected to the horizontal lines of the same row number, respectively. Sixteen 4-radix horizontal lines are connected to sixteen 4-radix vertical lines of the same order. Eight 4-radix vertical lines connect 4-radix diagonal directional outputs from the upper blocks to the lower sub-switch of the same column number in the lower row blocks.

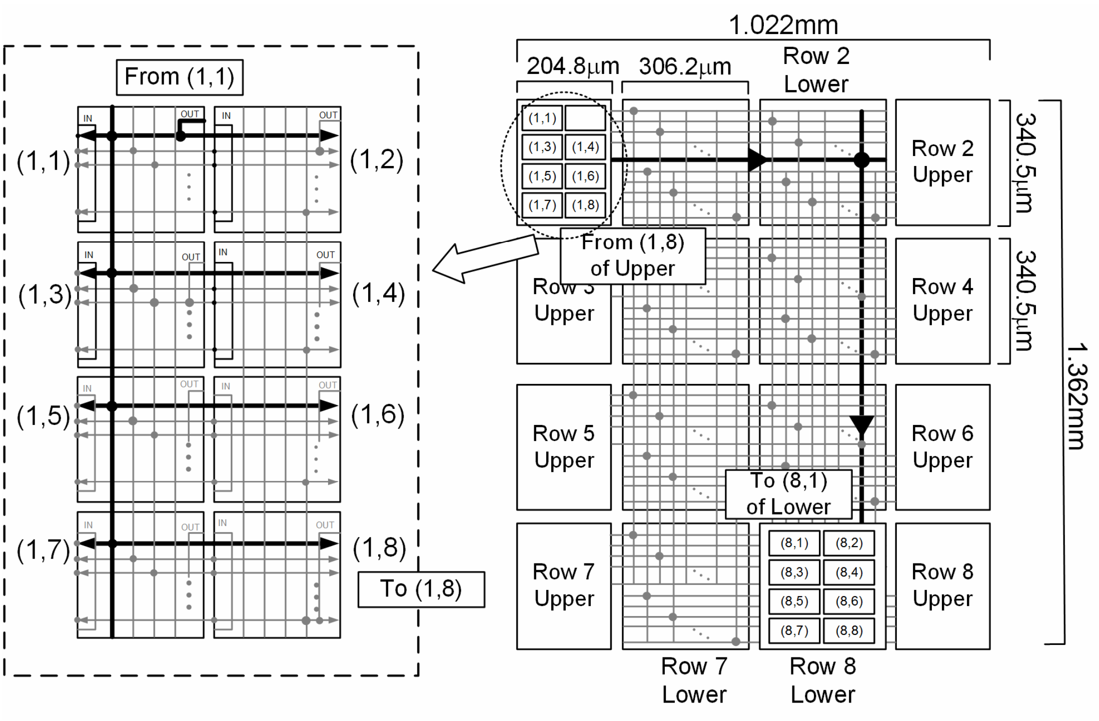


Fig. 10. The estimated floorplan of the proposed Hyper-X crossbar switch.

The row-directional hops are realized by routing eight upper or lower sub-switches in a row block as shown in the left part of Fig. 10. Eight outputs of the upper or lower sub-switches are connected to the horizontal lines of the same column. The eight horizontal lines are connected to the eight inputs of the 8:1 mux and the eight inputs are connected to the horizontal lines of the same order.

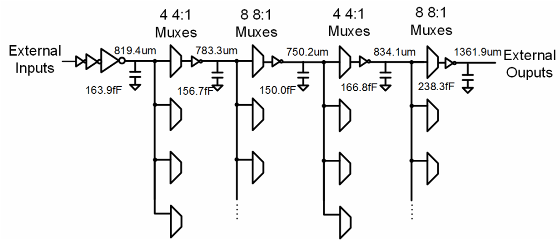


Fig. 11. The critical path of the crossbar switch.

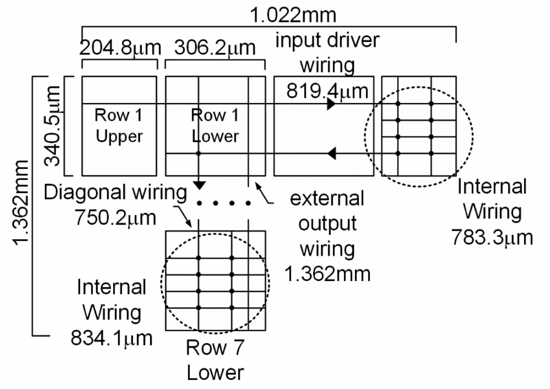


Fig. 12. The wire length estimation.

V. Simulation Results

The input-to-output propagation delay and power consumption of the proposed Hyper-X crossbar switch are estimated based on the critical path circuit shown in Fig. 11. The simulation is carried out assuming a 65nm CMOS process. The critical path is composed of two stages of four 4:1 muxes, two stages of eight 8:1 muxes, three inverters in the input driver, and four inverters in the mux stages. The lengths of the wires are estimated based on the floorplan, of which detail is shown in Fig. 12. The wires are modeled as capacitive loads, assuming the per-length capacitance of 0.2 fF/um.

Table 1 illustrates the improvements in delay and power contributed by each modification described in this work. It is noteworthy that each modification does not necessarily yield monotonic improvements both in delay and power. Nonetheless, the power-delay product is improved by all three modifications, resulting in the total 96.31% reduction in power and 83.4% reduction in delay

Table 1. Comparison of the performances of architectures with the proposed modifications

	Delay(ns)	Power(mW)	Delay×Power(ns×mW)
Matrix	3.859	85.98	331.8
Simple Hyper-X crossbar	26.26	9.830	258.1
Restricting order	9.285	2.884	26.78
Restricting order + Signal ordering	6.918	1.622	11.22
Restricting order + signal ordering + Transposing (Proposed Architecture)	0.641	3.176	2.036

Table 2. Comparison of the performances of the recently proposed high-radix crossbar switches

	Radix	Bit	Delay (ns)	Energy (fJ/cycle)	Area(λ^2)	FOM(cycle/ns·fJ·λ^2)
[5]	64	8	0.744	55.60	1.65E+08	1.262
[6]	64	8	0.55	79.10	7.41E+07	1.332
[7]	128	32	1.34	1790.37	7.90E+09	0.464
Hyper-X	256	16	0.641	3176	3.93E+09	2.203
Matrix	256	16	3.859	20991.21	4.11E+09	0.038

compared to the basic matrix crossbar.

To enable a fair comparison among crossbar switches with different radixes, bit-widths, operating frequencies, and IC technologies, a new figure-of-merit (FoM) that mesures the energy-delay-area efficiency is proposed. The power consumption is proportional to the operating frequency, using energy per cycle is a better normalized quantity to compare switches operating at different frequencies. Also, the propagation delay increases proportionally with radix and energy consumed in one path of a crossbar increases proportionally with the radix and bit width. And the area consumption increases proportionally with the square of the product of the radix and bit width. Therefore, the delay per radix, energy per (radix × bit), and area per (radix × bit)^2 are the chosen normalized metrics to compare variety of crossbar switch architectures. A figure-of-merit (FoM) is defined as an inverse of the product among these metrics: delay/radix, energy/bit, and area/bit^2.

$$\mathrm{FoM} = \left(\frac{\mathrm{Energy} / \mathrm{cycle} \times \mathrm{Delay} \times \mathrm{Area}}{Radix^4 \times Bit^3} \right)^{-1} \qquad (2)$$

Table 1 lists the specifications and FoM values of the recently-published crossbar switches. The proposed Hyper-X architecture achieves the FoM of 2.203-cycle/ns·fJ·λ^2, which is best performance among the architectures listed in Table 2.

VI. Conclusion

In this paper, the use of a Hyper-X network topology, one of network-within-network topologies, was investigated as a way to improve the power and delay of a high-radix crossbar switch. In addition, the use of three modifications further improved the power and delay by reducing the maximum fanout of each circuit node. The designed 256-radix crossbar switch in 65 nm CMOS achieves the simulated delay of 0.641ns, power of 13.01-W, and area of 0.93 mm x 1.25mm. And proposed architecture has the figure-of-merit of 2.203-cycle/ns·fJ·λ^2 which is the best figure among the state-of-the-art crossbar switches recently reported.

Acknowledgment

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

References

[1] T. Wu, et al., “A 2Gbh 256*256 CMOS Crossbar Switch Fabric Core Design using Pipelined MUX,” *in Proc. International Symposium on Circuits and Systems (ISCAS)*, pp. II-568~571, May 2002.

[2] K. Choi, et al., “VLSI Implementation of a 256x256 Crossbar Interconnection Network,” *in*

Proc. Parallel Processing Symposium, pp. 289-293, Mar. 1992.

[3] 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.

[4] J. Ahn, et al., “HyperX Topology, Routing, and Packaging of Efficient Large-Scale Networks,” *in Proc. Conference on High Performance Computing Networking, Storage and Analysis,* pp. 14-20, Nov. 2009.

[5] D. Song, et al., “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. 22-24, Apr. 2013.

[6] J. Ryoo, et al., “A 25-FO4, 81-mW Radix-64 Crossbar Switch with Partially-Activated Input and Output Lines,” *International SoC Conf. (ISOCC)*, pp. 227-230, Nov. 2012.

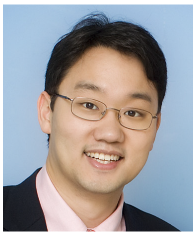
[7] G. Passas, et al., “Crossbar NoCs Are Scalable Beyond 100 Nodes,” *IEEE Trans. Computer-aided Design of Integrated Circuits and Systems*, pp. 573-585, Apr. 2012.

[8] G. Passas, et al., “A 128 × 128 × 24Gb/s Crossbar Interconnecting 128 Tiles in A Single Hop and Occupying 6% of Their Area,” *in Proc. ACM/IEEE International Symposium on Networks-on-Chip,* pp. 87-95, May 2010.

[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.

Seung-Heon Baek 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 and his research interests include reconfigurable and scalable systems of analog/mixed-signal systems and FPGA.

Sung-Youb Jung received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, Korea in 2013. He is currently pursuing towards a M.S. degree at the same university and his research interests include energy harvesting circuits of analog/mixed-signal systems and FPGA.



Jaeha Kim is currently an Assistant Professor at Seoul National University (SNU), Seoul, Korea and his research interests include low power mixed-signal 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 Postdoctoral Researcher. Prof. Kim is the recipient of the Takuo Sugano Award for Outstanding Far-east Paper at 2005 International Solid-State Circuit Conference (ISSCC).