Novel Design of Three-Dimensional Crossbar for Future Network on Chip based on Post-Silicon Devices

Kumiko Nomura
Keiko Abe, Shinobu Fujita
Corporate R&D Center, Toshiba Corporation,
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
Email: {kumiko.nomura, keiko2.abe, shinobu.fujita}@toshiba.co.jp

André DeHon
Department of Computer Science,
California Institute of Technology,
Pasadena CA91125, USA
E-mail: andre@cs.caltech.edu

Abstract. **We present a novel 3D crossbar for future Network-on –a-Chip implementations. We introduce a routing algorithm for the 3D crossbar circuit and detail two specific 3D crossbar topologies. We evaluate the defect tolerance of the 3D crossbar and quantify the number of extra layers required to support arbitrary permutations as a function of the defect rate. Further, we estimate the circuit performance and advantages of the 3D crossbar circuit based on post-silicon devices.**

Keywords; three-dimensional crossbar, post-Si device, network-on-a-chip

1. Introduction

The performance of recent microchips tends to be dominated by communication time in the network rather than the performance of logic blocks and memory blocks. The design of the on chip network (*e.g.* Network-on-a-Chip (NoC)) will be one of the most important aspects of microchip design. The space for the NoC is, however, limited in microchips based on conventional 2D CMOS technology. This limits the wiring density and design options available for the NoC. A new three-dimensional (3D) circuit design based on non-classical devices such as carbon nanotube (CNT) mechanical switches, CNT transistors and nanowire transistors has been proposed by Dr. Fujita *et al.* [2] (See Fig. 1). Such new 3D circuit designs can provide ultra-wide bandwidth beyond conventional 3D circuits using Si wafers, since density of vertical interconnections between different layers can be increased significantly [2]. One of the most effective applications of the new 3D circuit design is the 3D crossbar for NoC design due to such ultra-wide bandwidth. However, 3D crossbar design is, itself, potentially complex.

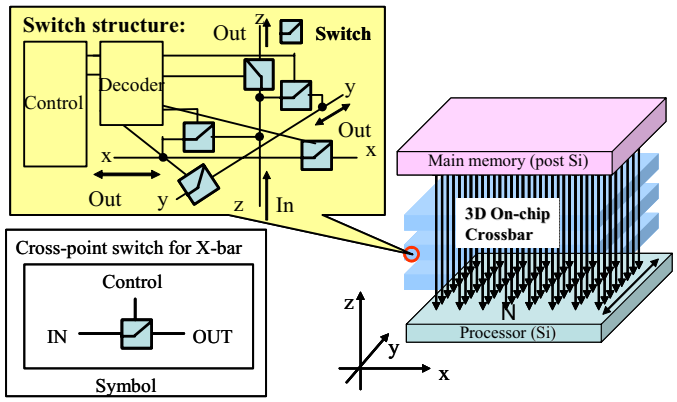

Fig . 1 Crossbar-bus circuit and switch structure in the crossbar-bus

Although there have been many concrete designs for 2D crossbar bus, most of them are locally optimized for each circuit and cannot be applied to 3D circuit design. In this paper, we describe general design for a 3D crossbar NoC using new 3D circuits and explore tradeoffs, which arise optimizing the crossbar design. Such optimized 3D on-chip crossbars may prove beneficial in many parts of a microchip design where data transfer is the performance limiting bottleneck.
The contributions of this paper include the following points:

- Elaborations and optimization of 3-stage 3D crossbar switching architecture.
- Adaptations of Leighton's Mesh Routing Algorithm to perform routing for a 3-stage, spatial 3D crossbar.
- Development and evaluation of crossbar routing algorithm that tolerantes open-crosspoint defect.
- Estimation of 3-stage crossbar performance using both scaled CMOS and post-silicon switching devices.

2. Organization for 3D on-chip crossbar

We design the 3D crossbar to route any permutation between a set of $N \times N$ I/O terminals on one layer of the 3D microchip and a second set of $N \times N$ I/O terminals on another layer. Fig. 1 shows an example where one set of IO terminals is a dense memory and the other is the datapath of a processor.
For routing in the 3D on-chip crossbar, the x-wire, y-wire, and z-wire are set in x-axis, y-axis, and z-axis direction, respectively. In this paper, the circuit consists of some crossbar-switch layer sets (CLSs) sandwiched by a top I/O layer and bottom I/O layer. One CLS consists of a layer of crossbar-switch matrix array (switch layer) and a few wire layers. We explore the implications of using both CMOS switches and post-silicon devices to implement the crossbar switching.

2.1 Simple Connection Model (SC-model)

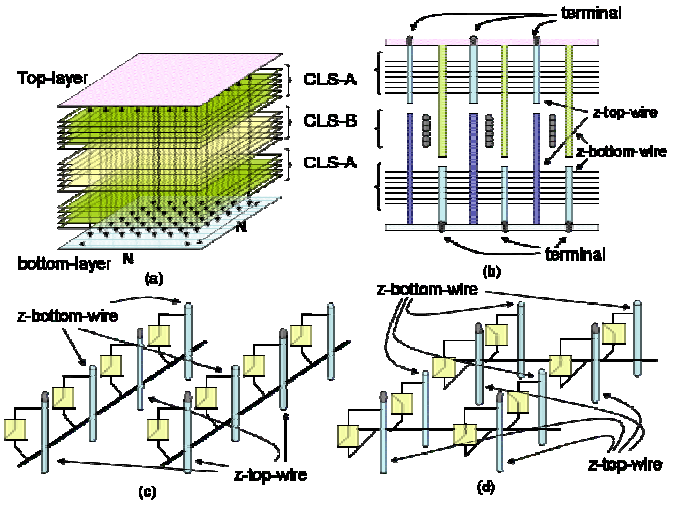

Fig. 2 SC-model: (a) outline of the SC-model, (b) outline of the side viewing of the SC-model, (c) CLS-A of the 2x2 circuit, (d) CLS-B of the 2x2 circuit

In this model, 3D crossbar circuit consists of $3N$ CLSs. The characteristic of this circuit is that the position of the terminals in the top-layer differs from the position of the terminals in the bottom-layer (See Fig. 2(b)). A circuit constructed like this contains fewer switches than a circuit which has the top layer and bottom layer terminals in the same position. Moreover, in this model each routing passes fewer switches than a circuit having same position terminals. Each terminal is connected to the z-wire for transferring a bit from the terminal to the terminal on the other side. This circuit consists of $3N$ sets of CLS having $N \times 2N$ z-wires. The $3N$ sets of CLS are obtained by stacking N layers of A type-CLS, N layers of B type-CLS, and N layers of A type-CLS. From now on, A type-CLS and B type-CLS are called CLS-A and CLS-B, respectively. The CLS-A has one wire-layer consisting of N y-wires and one switch layer having $2N^2$ switches. The switches in the switch layer connect y-wires and z-wires, which are in the same x-coordinate. The CLS-B has one wire layer consisting of N x-wires and one switch layer having $2N^2$ switches. The switches in the switch layer connect x-wires and z-wires, which are in the same y-coordinate. Fig. 2(c) and Fig. 2(d) show the relation of the connection of two wires in CLS-A and CLS-B for 2×2 terminal circuit, respectively. Z-top-wires which connect to terminal in the top layer are cut off between CLS-A and CLS-B. Z-bottom-wires which connect to terminal in the top layer are cut off between CLS-B and CLS-A. Fig. 2(b) shows the z-wire situation, viewed from the side of the circuit.

2.2 Flexible Connection Model

Next we introduce the flexible connection model, which is an improvement on the SC model and consists of fewer CLSs and switches than those of the SC model. In this model, the switch and wire CLS layer contains the same function as three CLSs in the SC model; in particular, each CLS layer in the FC model acts as two CLSs-A and one CLS-B from the SC-model. This model consists of N sets of CLS, $5N^2$ switches, and three types of N z-wires. The terminal position is also different between top-layer and bottom-layer. Fig. 3(a) shows the location of top-layer terminals and bottom-layer terminals. The z-wires connecting to terminals in the top-layer and the bottom-layer are called top-z-wire and bottom-z-wire, respectively. The remaining N z-wires are called aid-z-wires. In this model, each CLS consists of two types of wire layers and one switch-layer. One wire layer is constructed of only N x-wires and the other is constructed of only $2N$ y-wires. The switches are used for connecting two wires in the same CLS as follows.

- Top-z-wire and y-wire at the same x coordinate.

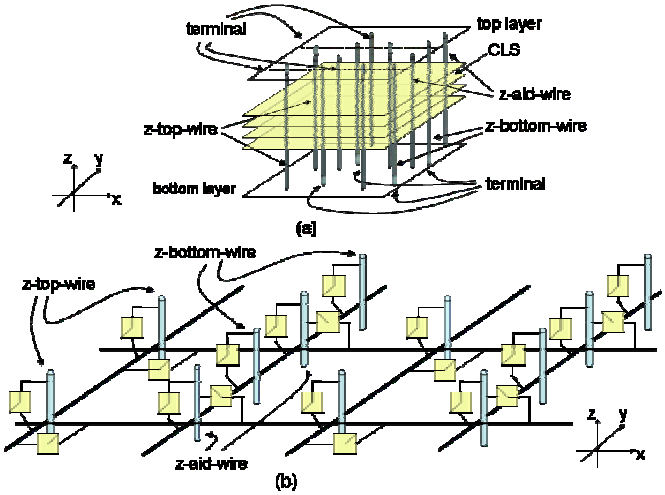

Fig. 3 Flexible model: (a) outline of the flexible model, (b) CLS of 2x2 circuit

- Y-wire and x-wire which are crossing.
- X-wire and aid-z-wire at the same y coordinate.
- Y-wire and aid-z-wire at the same x coordinate.
- Bottom-z-wire and y-wire at the same x coordinate.

Fig. 3(b) shows a relation of switches and wires in one CLS of terminals circuit.

2.3 Trade-off between the number of CLS and the number of wires

Considering fabrication cost of 3D crossbar, it is preferable that 3D on-chip crossbar routing be able to connect any I/O terminal on the top-layer to any I/O terminal on the bottom-layer using the least number of CLSs. We assume that all the routing required in the SC-model and Flexible model are decided, respectively. The number of CLSs in the crossbar circuit is related to the number of wires in one wire layer. When we can place more wire in a physical CLS, the terminal location is changed from original two models. Fig. 4 shows the location of the top layer terminals in 4×4 I/O terminal circuit. One case has two wires in one wire layer of CLS and the other has four wires in one wire layer of CLS. Fig. 5 shows the relation between the number of CLSs and the number of wires for 8×8 terminals circuit. It can be seen that as the number of wires in a wire layer is increased, the number of CLSs can be decreased. Since the limitation of number of wires is determined by fabrication process and circuits are used for the crossbar, the minimum number of the CLS is determined using Fig. 5. It can also be seen that the minimum number of CLS based on the flexible connection

model is one third of that based on the simple connection model.

3. Fundamental Routing Algorithm for 3D Crossbar

The algorithm for routing our 3D crossbar is an adaptation of Leighton's Offline Mesh Routing algorithm (OMR) [3]. Leighton shows that a mesh can route any permutation in three one-dimensional permutation steps. That is, with proper assignment of the intermediate rows, we can perform any 2D mesh permutation by performing a column permutation, a row permutation, and a final column permutation (See Theorem 1.16 in [3]).

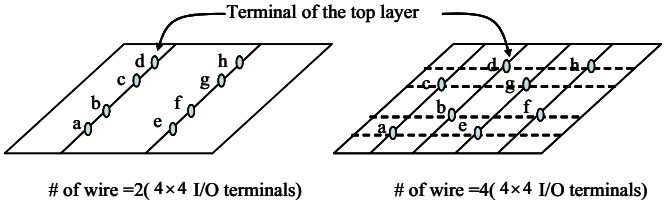

Fig. 4 the location of the top layer terminals for 4×4 I/O terminals

In our 3D crossbar, we also want to perform a mesh permutation. Here, the permutation is a spatial one between the terminals on two different layers of the microchip rather than a temporal one inside a nearest-neighbor mesh. We effectively use a $N \times N$ 2D crossbar to perform each of the three one-dimensional permutations in Leighton's algorithm. As a result, our 3D crossbar uses $3 \times N \times (2N \times N) = 6 \times N^3$ switches to route the $N \times N$ permutation in 3 steps. This should be contrasted with the N^4 switches it would take to build a $(N \times N) \times (N \times N)$ crossbar to perform the permutation in a single switching step and the $3N$ steps it would take if we used only the $O(N^2)$ hardware assumed in Leighton's original mesh. The 3-stage switching arrangement described can be viewed as an N-ary Benes network [8] which can be built in 3-stage. There is, of course, a whole design space of Benes networks between the N-ary and a 2-ary case. A complete elaboration of this design space is beyond the scope of this paper and will be considered in future work.

The key trick in Leighton's OMR algorithm is the assignment of the intermediate rows in order to guarantee that everything assigned to each row is destined for a distinct column. Leighton observes that this problem can be cast as a series of perfect matching problems on a bipartite graph, and Hall's theorem [7] proves the existence of a solution for each of the perfect matching problems. The bipartite graph is constructed by placing a node in the source set for each start column and a node on the sink set for each destination column. For each one-to-one link from (s_x, s_y) to (d_x, d_y) in the permutation, we add an edge between src_x and dst_x. Since we are routing a permutation, this means each node in the source and destination sets will each have exactly $r = N$ edges, making it an *r*-regular bipartite graph. This regularity is sufficient to guarantee that Hall's theorem applies and can provide a matching (Corollary 1.18 in [3]). A match tells us a set of source-column to sink-column exchanges which can occur concurrently (*i.e.* every exchange in the set has distinct source and sink columns); we assign everything in the match to the same row-route crossbar. If we remove all of these assigned routes from the bipartite graph, we have an $(r-1)$-regular graph, in which we are also guaranteed to find a perfect match. Consequently, we can continue finding perfect matches, assigning match sets to wire, and removing edges until every edge is assigned to a crossbar. With this assignment, we are now guaranteed that:

- the routes in each source column are assigned to a distinct row crossbar
- the routes in each row crossbar route from a distinct source column to a distinct destination column
- the routes in each sink column are assigned to a distinct row crossbar

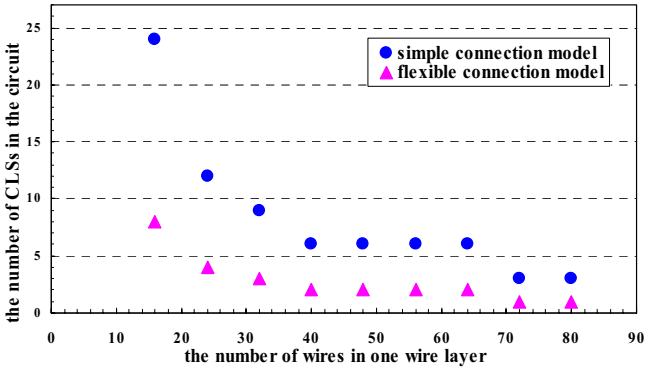

Fig. 5 Number of CLSs vs. number of wires per layer for an 8×8 3D crossbar in the SC- and flexible- connection model

Consequently, we guarantee each of the three 1D permutations required to route the entire 2D permutation can be performed.

We can summarize the algorithm for assigning links to intermediate crossbars as follows:

1. Create a bipartite graph, G, with two sets (src, dest) each with N nodes; for each link between (s_x, s_y) and (d_x, d_y) in the original 2D permutation, add an edge between src_x and dst_x
2. Set $c = N$
3. Find a perfect matching, M, in G
4. Assign the routes associated with all edges in M to crossbar c
5. Remove all edges in M from G
6. $c = c - 1$
7. if $c > 0$, go to step 3.

The most complex step in this algorithm is the bipartite matching in step 3 which is executed N times. Using the Hopcraft-Karp algorithm [6], a bipartite match can be performed in $O(\sqrt{|V|}\,|E|)$ time. Here $|V| = 2N$, and $|E| = cN$ on the iteration performing an assignment to crossbar c. The total runtime at step 3 then is:

$$\sum_{c=1}^{c=N} O(\sqrt{N} \times cN) = O\left(N^{1.5} \times \left(\sum_{c=1}^{c=N} c\right)\right)$$

The sum simplifies to $O(N^2)$, making the entire result $O(N^{3.5})$. Since this is the dominate work in the algorithm, the total algorithm runs in $O(N^{3.5})$ time.

4. Tolerance to Defects of Switches

In our defect model, we assume that only switches in the crossbar may be defective. In particular, we envision using nanoscale, post-silicon switches such as CNT switches and nanowires for the crosspoints. Consequently, we expect the switches to have a much higher defect rate than conventional lithography. In contrast, the wires in this design remain lithographic wires which we expect to be much less error prone. Nonetheless, since these are crossbars, wire sparing is easily handled simply by providing spare wires. A free-switch is a non-defective switch which can be set both to on-state and off-state. A defective switch is a switch which can be set only to off-state. We assume opens or breaks in these nanoscale devices are much more likely than shorts. Further, the high defect-tolerance demonstrated with our algorithm suggests that it is reasonable to tune manufacturing to avoid shorts even at the expense of increasing open defects. Now we introduce the defect tolerant circuit for SC-model with this defect model.

4.1 Defect tolerant strategy

Here the defect-SC-model has extra CLSs for redundancy in order to avoid all defects of the circuit. Fig. 6(b) shows the defect-SC-model. For the defective crossbar circuit, redundant ΔL layers are added to each CLS. The crossbar circuit thereby consists of $N+\Delta L$ sets for each CLSs; $N+\Delta L$ sets of B-type CLSs, and $N+\Delta L$ sets of A-type CLSs. Based on this model, the minimum number of ΔL layer was estimated. The routing algorithm for the defect-SC-model is basically the same as the SC-model shown in Section 3. The different point of these two models is the way to assign routes to crossbars in each step. In the defect-free algorithm, a crossbar is assigned uniquely. In the algorithm for defect-SC-model, we are using greedy allocation strategy which is similar to the greedy method in [5]. A crossbar, which has two free-switches, has to be assigned to route in each step. At first, from upper CLS we search the unused crossbar which is in the same position for each CLS. Then we check the two switches which must be set to the on-state. If theses two switches are free switches, then the algorithm select the crossbar to assign the route. If not, the algorithm searches another crossbar of the other CLS. Fig. 6(b) shows an example of defect routing for 2×2 terminals circuit. As the switch of the crossbar of second CLS is defective, the target route selects the crossbar of the third CLS.

4.2 Analysis of redundancy

We compute the number of redundant layers, ΔL, needed for routing in practice. Let R be the set of routing requests and $r_i \in R$ be a routing request for $0 \le i \le N \times N - 1$. Let α be the fault probability of each crossbar switch. The wire must have two free-switches for use as a route. Thus the probability of successfully using the wire is $(1-\alpha)^2$. In the algorithm, the set of wires that is used by the routing request r_i is of size $N+\Delta L-i+1$. Therefore the probability of successfully assigning a wire to r_i is $1-(1-(1-\alpha)^2)^{N+\Delta L-i+1}$. Hence the probability of successfully performing all the required routing is:

$$\prod_{i=1}^{N}[1-(1-(1-\alpha)^2)^{N+\Delta L-i+1}].$$

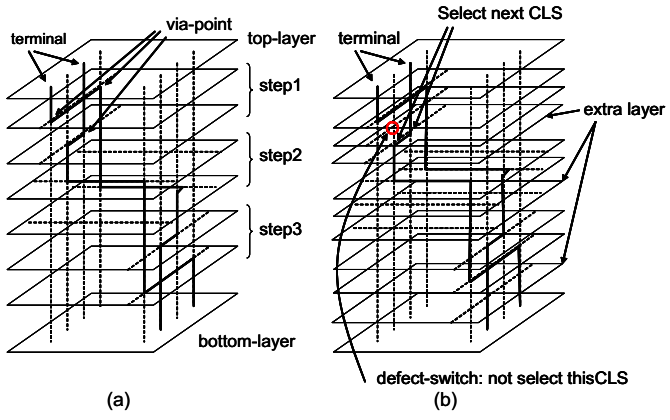

Fig. 6 Example of the routing for SC-model: (a) routing of the non-defect algorithm, (b) routing of the defect algorithm

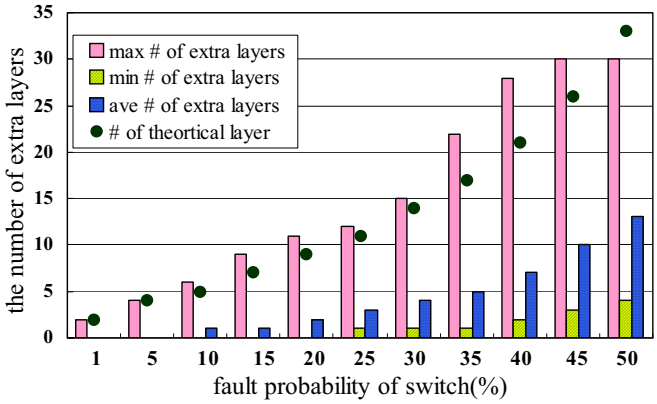

Fig. 7 The number of extra CLSs versus the fault probability of switch for 8x8 circuit

Let Pc be the probability of successfully routing in the circuit. The following inequality gives a tighter lower bound on α:

$$\prod_{i=1}^{N}[1-(1-(1-\alpha)^2)^{N+\Delta L-i+1}] \ge Pc\,.$$

Notice that $1-(1-(1-\alpha)^2)^{N+\Delta L-i+1}$ is very small from $0<\alpha<1$. Thus we can get a following inequality.

$$\prod_{i=1}^{N}[1-(1-(1-\alpha)^2)^{N+\Delta L-i+1}] \ge (1-(1-(1-\alpha)^2)^{\Delta L+1})^N \ge Pc$$

From the inequality of the right hand side of the above, ΔL is shown by Pc and α as follows:

$$\Delta L \ge \left|\frac{\log(1-Pc^{1/N})}{\log(2\alpha-\alpha^2)}\right| - 1\,.$$

4.3 Computational Experimental

We implemented the defect tolerant algorithm in this section and calculated the number of extra layers for an 8×8 crossbar circuit and the fault probability α of a switch. For statistical purposes the same terminal circuit is routed 10^5 times for each α. Fig. 7 shows the maximum, minimum, and average of the number of extra CLS ΔL for the fault probability α. This shows that, on average, the circuit will require no extra layers when the defect probability is at or below 10%. We calculate ΔL from the inequality in Section 4.2 for a routing success rate Pc of 0.999. Fig. 7 shows the theoretical value of ΔL. From this result, the theoretical value ΔL reaches almost the same value of the experimental result. The maximum number of CLS layers is larger than the average case. Currently, we are using a greedy method for selecting wires. Consequently, as part of our future work, we expect to be able to reduce the

maximum number of layers required using a more aggressive algorithm.

5 Circuit Performance of 3D crossbar

We calculated latency of 2D and 3D crossbar buses based on sub-90 nm CMOS technology. Fig. 8(a) shows latencies associated with one crossbar switch and one interconnect between crossbar switches and their dependencies on interconnect length between crossbar switches. CMOS circuit for the crossbar switch is used as shown in Fig. 8(b). For 2D crossbar, since the maximum interconnect length is more than 1 mm, the latency of crossbar is dominated by interconnect delay. For 3D crossbar, since maximum interconnect length can be decreased to less than about 100 um by optimizing design, the latency of crossbar is dominated by delay of crossbar switch its self. The delay of crossbar switch is naturally reduced by decreasing feature size of transistor. The delay will be further reduced by replacing silicon-MOSFET by carbon-nanotube FET or Ge nanowire FET to less than half of those in Fig. 9[2]. Although the delay can be reduced, the advantage of 3D structure is that no repeaters are needed to reduce interconnect delay, since overhead of area and power associated with repeaters is increasing rapidly as the feature size of transistor is decreased.

Fig. 9 shows estimation for bandwidth of 2D and 3D crossbar. Frequency is the inverse of crossbar latency calculated from Fig. 8(a), where the crossbar is composed of five crossbar switches and interconnects between them. Here the footprint area of crossbar circuit is assumed to be 0.1 mm x 0.1 mm. Whereas bandwidth of conventional 2D crossbar based on silicon-CMOS is 100 Gbps to 1 Tbps, that of the 3D crossbar based on silicon-CMOS and wafer-bonding technology is 10 Tbps to 1 Tbps. Furthermore, the bandwidth of 3D crossbar based on post-silicon device is expected to be over 1 Pbps.

6 Conclusion

Emerging, post-silicon technologies open the way to densely integrated 3D switched interconnect. At the same time, these densely packed, nanoscale devices may be highly prone to defects. We have developed the detail design of a 3-stage, 3D crossbar NoC and shown that it can be routed, tolerating a high defect rate with modest overhead. The 3D design keeps wire short, enabling high bandwidth communications. NoC design using such an extremely large bandwidth can solve various bottlenecks of data transfer in future high-performance micro-chips.

Acknowledgments

The authors are grateful to Shinichi YASUDA and Dr. Tetsufumi TANAMOTO, Corporate R&D Center, Toshiba Corp., and Dr. Bipul C Paul and Dr. Masaoki OKAJIMA, Toshiba America Research, Inc., for helpful comments.

References

[1] S. Fujita, et al. "Three-dimensional Logic Architecture by Four-terminal Electrical Switches beyond Two-dimensional CMOS Architecture". *Proceedings of nanotechnology Conference. Anaheim*, NSTI. pp.213-216, 2005.

[2] S. Fujita, et al. "Novel architecture based on floating gate CNT-NEMS switches and its application to 3D on-chip bus beyond CMOS architecture". *Proceedings of IEEE Nanotechnology Conference*, 2006.

[3] F. Thomson Leighton, "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, Inc., 1992.

[4] Davis, J. A., et al. "Interconnect Limits on Gigascale Integration (GSI) in the 21st century". Proc. IEEE, no. 89, vol. 3, pp.305-324, 2001.

[5] H. Naeimi and A. DeHon, "A Greedy Algorithm for Tolerating Defective Crosspoints in NanoPLA Design", Proc. of ICFPT, pp.49-56, 2005.

[6] J. E. Hopcroft and R. M. Karp, "An $n^{2.5}$ Algorithm for Maximum Matching in Bipartite Graphs", *SIAM Journal on Computing*, vol. 2, no. 44, pp.225-231, 1973.

[7] P. Hall. "On representative of subsets", *Journal of the London Mathematical Society,* vol. 10, no. 1, pp.26-30, 1935.

[8] V. Benes, "Mathematical Theory of Connection Networks and Telephone", Academic Press, 1965.

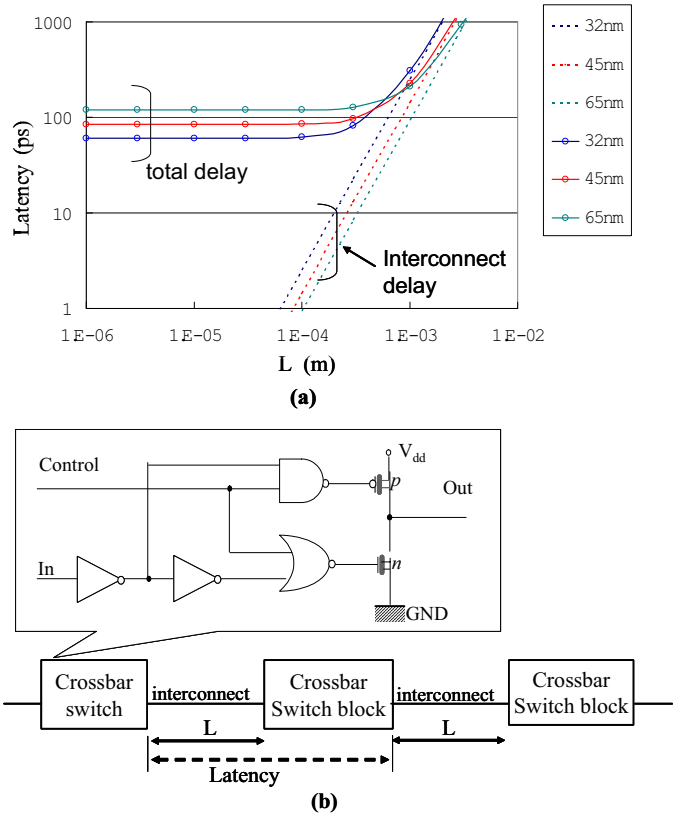

Fig 8 (a) the latency of a crossbar switch block with an interconnect between two switch blocks as a function of interconnect length L. The dashed line is a delay of crosspoint switch and interconnect and the solid line is a delay of interconnect. (b) crossbar switch model

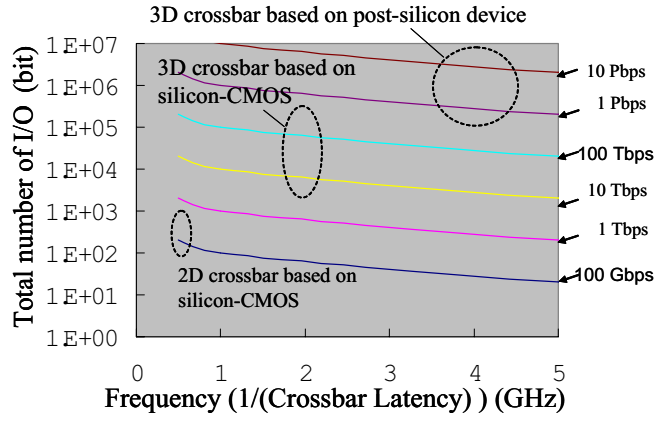

Fig 9 Comparison of band width for 2D and 3D crossbar based on Si-CMOS