Switch Control Complexity of Diminished Three-Quarter Crossbar Switches

Pavel Selin† and Hitoshi Obara†,

†Akita University, Akita, Japan

Summary
The three-quarter crossbar switch (TQ-XBS) has been known to have about $3N^2/4$ crosspoints, where N is the switch size, and shows a property intermediate between the crossbar switch (XBS) and the triangular switch (TAS). Its original configuration has an extra pair of input and output ports to make the analysis of the switch control complexity difficult and some major properties of TQ-XBS have been left open. In this paper, we consider a diminished TQ-XBS that has not the extra ports as a first step to examine performance of TQ-XBSs. We made a comprehensive study of the switch control complexity and the worst case scenario for rearrangement in the diminished TQ-XBS. It is shown that the switch control complexity for setting up a new connection is $O(N)$ at most. It is also shown that the identical permutation ensures truly a worst case scenario for rearrangement and the maximum number of rearrangements for the worst case remains two regardless of N. The complexity of the rearrangement process is also given by $O(N)$. Additionally, it is pointed out for the first time that the diminished TQ-XBS shows a better performance than the TAS regarding a newly defined figure of merit.
Key words:
Three-quarter crossbar switch, Rearrangeably nonblocking, Switch control complexity, Rearrangement.

1. Introduction

The crossbar switch (XBS) consisting of 2x2 basic switch elements (BSEs) offers a most practical design option in the optical domain. It is wide-sense nonblocking and requires quite a simple switch control, while it has as many as N^2 BSEs, where N is the switch size [1,2]. It has been a research focus to reduce the number of BSEs in XBSs, especially for the application to optical networks, because each individual optical BSE, typically implemented with an optical directional coupler, occupies space, consumes power, and costs a lot, among others [3]. In early stages of the research, a number of switches with less than N^2 BSEs have been developed from the XBS [4,5]. The most basic family is the triangular switch (TAS) [6,7]. The TAS was created from the XBS by subtracting about a half of BSEs within a triangular area. This design principle corresponds to the characteristics of the XBS: All the BSEs in a column of the XBS are devoted to an output port for switching a call to it, and thereby the numbers of calls to be switched decreases one by one as the calls go through the columns. Although the design principle of TASs is well established, there remain several basic questions. For example, an NxN XBS has N^2 BSEs and wide-sense nonblocking (i.e. no rearrangements are required under a certain switch control algorithm), while an NxN TAS has about a half of N^2 BSEs and is rearrangeably nonblocking. In fact, the number of rerarrangements required for the TAS reaches the maximum possible value of N-2 for a worst case scenario. Such an observation led us directly to a question of how we can get an intermediate performance with respect to rearrangements between these two extremes. In our preliminary work [8], we have introduced the three-quarter crossbar switch (TQ-XBS) as an answer to the question. It is rearrangeably nonblocking and its performance was examined mainly from the number of rearrangements. However, its basic properties such as switch control complexity for setting up a connection and the validity of a worst case scenario assumed for rearrangement have been left open. The aim of this paper is to address these issues. The paper is organized as follows. We begin with an outline of TQ-XBSs in Section 2. In this paper we consider a diminished TQ-XBS as a first step toward comprehending the basic properties of TQ-XBSs. In Section 3 we analyze its switch control complexity for setting up a connection and discuss its rearrangement process in detail. The paper concludes in Section 4.

2. Outline of TQ-XBSs

2.1 Switch configuration

Let us begin with a conventional XBS with a switch size of N as shown in Fig. 1(a). We assumed N=2^n (n is an integer and n≥2) throughout the paper for convenience. Variables i and j represent input and output port numbers, where 0≤i≤N-1 and 0≤j≤N-1. BSE(i, j) denotes the BSE on the cross point of the i-th row and the j-th column. A BSE has two connection states; bar and cross as shown in Figs. 1(b) and 1(c). Generally, all the BSEs are initially set to the cross state [9], which we assumed as the default state in this paper. When the i-th input corresponds to the j-th

Manuscript received October 5, 2012
Manuscript revised October 20, 2012

output under the default state, only the BSE(i, j) will be set to the bar state. As a result, the connection from i to j takes a rectangular route with one turn as shown by a dashed line in Fig. 1(a). When releasing the connection, the same BSE(i, j) will be set back to the cross state. Therefore, the switch control for XBSs is fairly simple with complexity of $O(1)$. In this paper we define R(i) and C(j) as the number of BSEs with the bar state in the i-th row and the j-th column, respectively. It is readily shown that both $R(i) \leq 1$ and $C(j) \leq 1$ hold for any i and j in conventional XBSs. We refer to this property as property-1.

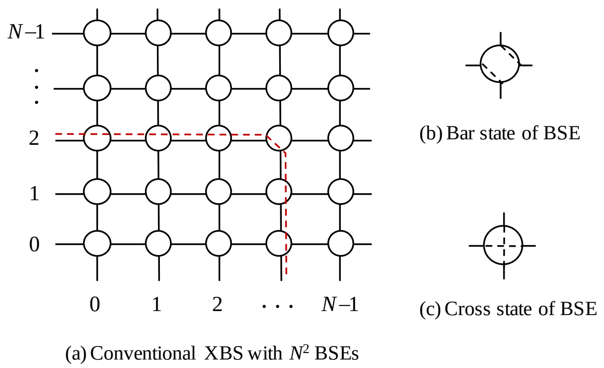


Fig. 1 Conventional crossbar switch composed of N^2 BSEs.

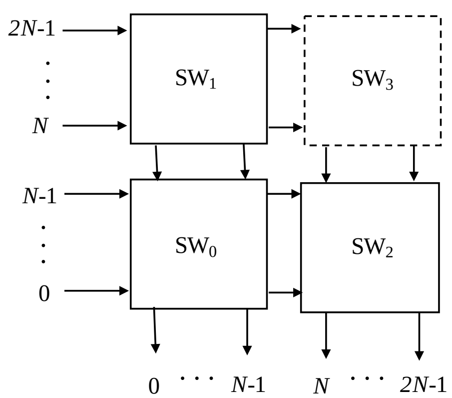


Fig. 2 Scaling the switch size of an XBS to double.

XBSs have idle links at the top row and at the rightmost column to scale the switch size. If we add an XBS denoted by SW_0 with three other XBSs of the same size (i.e. SW_1, SW_2, SW_3), we have an XBS of double size as shown in Fig. 2. We see that SW_1 and SW_2 are necessary to scale the number of input and output ports. However, is SW_3 absolutely essential for building up a nonblocking switch? Such a question led us to the three-quarter crossbar switch (TQ-XBS) from conventional XBSs [8]. Although the original TQ-XBS has an extra pair of input output ports at SW_1 and SW_2, we consider a diminished TQ-XBS (DTQ-XBS) without them as shown in Fig. 3 for the sake of simplicity. Input and output ports of the DTQ-XBS are divided into two groups according to their port positions. The first group includes those from 0 to $N/2$-1, and the second from $N/2$ to N-1 as shown in Fig. 3. We label them with (I_0, I_1) and (O_0, O_1). We assign switch components with labels SW_k (k=0, 1, 2). Each of SW_k is an XBS of a half size. We also assign internal links between the switch components with labels L_k (k=0, 1).

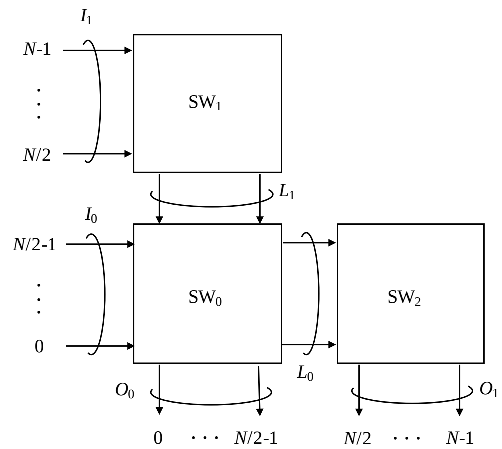
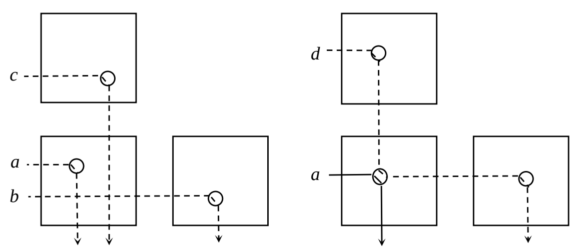


Fig. 3 Switch components of a diminished TQ-XBS.

2.2 Assigning routes to connections

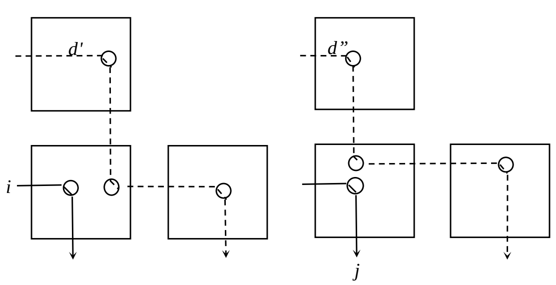
A connection between a pair of input and output ports of a switch will be established by assigning an appropriate route to it within the switch. There are three types of routes for TQ-XBSs [8], while two of them are enough for DTQ-XBSs. The first is the rectangular route that has been used for conventional XBSs. For example, connections with I_0 to O_0, I_0 to O_1, and I_1 to O_0 take rectangular routes like a, b, and c as shown in Fig. 4(a). It is obvious that connections with I_0 to O_0 take rectangular routes because SW_0 itself is an XBS with a switch size of $N/2$. We see that the set of SW_0 and SW_1 constitutes an asymmetric XBS with N inputs and $N/2$ outputs and provides rectangular routes. So does the set of SW_0 and SW_2. The second is a concatenated rectangular route, which is composed of two consecutive rectangular routes. Connections with I_1 to O_1 like d, take concatenated rectangular routes. The former part of a concatenated route looks like a conventional rectangular route between I_1 and L_1 as shown in Fig. 4 (b), while the latter part of it also does between L_0 and O_1. A BSE with the bar state in SW_0 on the way of a concatenated route is the termination point of the former rectangular route and is also the originating point of the latter rectangular route. From another viewpoint, we can see that a portion of a conventional rectangular route between I_1 and O_1, which would be provided over SW_1, SW_3, and SW_2 in Fig. 2, is changed from SW_3 to SW_0 to make up a concatenated route. This observation facilitates the development of a switch control algorithm for DTQ-XBSs, which will be discussed in detail in Section 3.1. Note that the type of connections will be readily identified with the combination of source input and destination output numbers.

It should be stressed that the property-1 holds within each of SW_0, SW_1, and SW_2 in Fig. 3 and thus $R(i) \le 2$ and $C(j) \le 2$ hold as a whole. We refer to this property as property-2. In SW_0, there are possible alternative routes for a concatenated route *d*, e.g. *d*' and *d*" as shown in Figs. 5(a) and 5(b), where $R(i)=3$ or $C(j)=3$. Such routes, however, are prohibited in DTQ-XBSs because blocking can occur. Consequently, if all the inputs of I_1 correspond to O_1 in DTQ-XBSs for a worst case scenario (e.g. identical permutation [10], where input *i* corresponds to output *j*, $j=i$), each BSE with the bar state in SW_0 can be shared by two connections from I_0 and I_1. This means that certain coordination for setting up concatenated connections is required and, eventually, makes the DTQ-XBS rearrangeably nonblocking. These aspects, which were not concerned in the previous work [8], will be discussed in detail in Section 3.2.



(a) Rectangular routes (b) Concatenated rectangular route

Fig. 4 Two kinds of routes for DTQ-XBSs.



(a) Concatenated route with $R(i)=3$ (b) Concatenated route with $C(j)=3$

Fig. 5 Possible alternative routes for concatenated routes.

3. Basic properties of diminished TQ-XBSs

3.1 Switch control for assigning routes to connections

Consider rectangular routes in an XBS under the identical permutation shown in Fig. 6, where BSEs with the bar state align diagonally and the property-1 holds. The identical permutation constitutes a worst case scenario for assigning routes to connections in DTQ-XBSs, because it yields the maximal number of concatenated routes between I_1 and O_1: $N/2$ connections through the removed area must divert their rectangular routes to concatenated routes as shown in Fig. 6. The routes of the other $N/2$ rectangular routes remain unchanged like that from $i=0$ to $j=0$. This means that the rectangular routes with I_0 to O_0, I_0 to O_1, and I_1 to O_0 are given priority over concatenated routes. These rectangular routes are set up and released with $O(1)$ complexity, just the same as conventional XBSs. In the worst case scenario, each column in the left-hand half of SW_1 accepts only one connection at most in order to minimize the number of rearrangements. In other words, the diverted routes are distributed evenly over the columns in the left-hand half of SW_1. We refer to this property as property-3.

Next, we shall consider searching process of concatenated routes in detail. We exemplify the process with an input port i_s ($N/2 \le i_s \le N-1$), which corresponds to j_d ($N/2 \le j_d \le N-1$). The first process by the switch controller is to inspect the number of BSEs with the bar state in the first column (i.e. $j=0$), $C(0)$. Note that the controller keeps a record of $R(i)$ and $C(j)$ as well as the location of BSEs with the bar state for each row and column. If $C(0)=0$ (i.e. output port 0 is left unconnected), the controller searches a row i_r, where $R(i_r)=0$ holds, between $i=0$ and $i=N/2-1$. Then, set $BSE(i_s, 0)$, $BSE(i_r, 0)$, and $BSE(i_r, j_d)$ to the bar state. If $C(0)=1$ (i.e. output port 0 is already connected), the controller inspect the location of the BSE with the bar state through the recorded table. If the BSE stands in SW_1, the controller skips the process to the next *j*, because no concatenated route can be set up in this case. If not, assume the location of the BSE is given by i_t, $0 \le i_t \le N/2-1$. Then, set $BSE(i_s, 0)$ and $BSE(i_t, j_d)$ to the bar state (n.b., $BSE(i_t, 0)$ has been already set to the bar state). If $C(0)=2$ (i.e. a concatenated route is already provided), the controller skips current *j* and continues the above process for succeeding *j*. The complexity of this switch control is given by $O(N)$. When an existing call is released, the BSEs set to the bar state on the setup phase will be toggled to the cross state in $O(1)$ time. Note that those BSEs can be specified instantaneously through the recorded table because each column and row of SW_k (k=0, 1, 2) has only a single BSE with the bar state due to the property-2. As a consequence, the switch control complexity for setting up and releasing a connection in DTQ-XBSs is given by $O(N)$ at most.

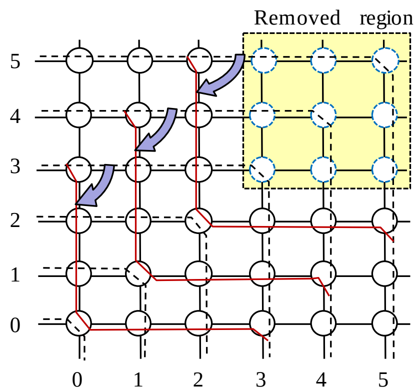


Fig. 6 Diverting routes in a worst case scenario.

3.2 Switch control for rearrangement

The property-2 implies that the DTQ-XBS is rearrangeably nonblocking and the number of rearrangements becomes two at most. We can confirm these properties as follows. Assume the identical permutation again as shown in Fig. 7(a), where solid and dashed lines denote busy and idle edges. In Fig. 7(a), suppose i_2 and j_2 are idle, an existing connection of i_1 to j_1 is released, and a new connection request of i_1 to j_2 is issued. Although both i_1 and j_2 are idle, blocking occurs due to the existing connections q. The switch controller detects the blocking through the symptoms that both C(j_2) and R(i_1) becomes three if BSE(i_1, j_2) is set to the bar state in order to provide the rectangular route with i_1 to j_2. The controller also knows that there are two existing connections (p and q) to be rearranged and begins the following rearrangement process. Firstly, p and q are released; BSEs associated with p and q in SW_0, SW_1, and SW_2 are set back to the cross state, and it takes $O(1)$ time for this process. Secondly, the BSE(i_1, j_2) is set to the bar state because the rectangular route has a top priority. It also takes $O(1)$ time. Thirdly, p and q are re-established through the same process as the provisioning of connections described in Section 3.1, for which it takes $O(N)$ time. The blocking disappears as shown in Fig. 7(b). The positions of BSEs with the bar state in SW_1 remain the same in the example given in Fig. 6. However, they can be changed in general. As a consequence it takes $O(N)$ time for the rearrangement process and the number of rearranged connections becomes two at most.

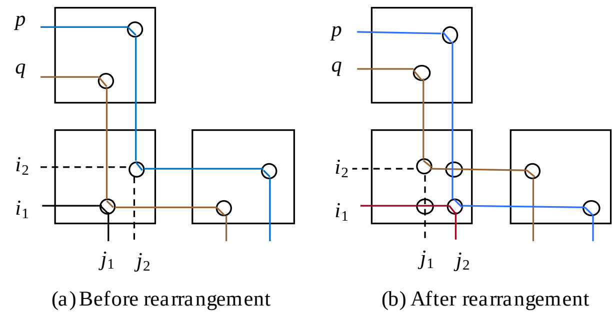


Fig. 7 Rearrangement of a concatenated route for a worst case scenario.

3.3 Transformation of DTQ-XBSs

It has been known that a part of a TQ-XBS can be replaced with a smaller TQ-XBS in a recursive manner [8]. A similar transformation can be applied to DTQ-XBSs. At the first replacement (or k=1), both SW_1 and SW_2 in Fig. 8 can be replaced with a DTQ-XBS of switch size $N/2$ (or two $N/4$ x $N/4$ regions are removed). At the second replacement (i.e. k=2), four $N/8$ x $N/8$ regions will be removed, and so on. In this paper, we discuss the number of rearrangements in a simple and visible way different from the previous work. We shall employ a new worst case scenario that each half of I_0 and I_1 correspond to each half of O_1 and O_0 (i.e. partially identical permutation). In this worst case scenario, a full number of connections pass through the one-quarter area removed in SW_1 and SW_2. These connections have to divert their routes from the removed region to the left half of SW_0 and SW_1 as shown in Fig. 8. Each column in the left half of SW_0 and SW_1 have to accept at least one diverted connection due to the property-3. Let us focus on the leftmost column (i.e. j=0) in the DTQ-XBS shown in Fig. 8. After the first replacement, C(0) increases from two to four and the number of rearrangements becomes three (or C(0)-1) at most. This result can be easily seen through an example, in which i=N-1 corresponds to j=0.

The number of BSEs with the bar state at the leftmost column after the k-th replacement ($0 \le k \le n-2$), $C_k(0)$, is given by

$$C_k(0) = 2^{k+1}. \tag{1}$$

Note that k=0 corresponds to the original DTQ-XBS. The number of rearrangements, A_k, is given by

$$A_k = \begin{cases} 2 & \text{for } k = 0, \\ 2^{k+1} - 1 & \text{for } 1 \le k \le n-2, \\ 2^{k+1} - 2 & \text{for } k = n-1. \end{cases} \tag{2}$$

Recall that we assumed N=2^n. When k=n-1, a TAS with switch size of N will be obtained and its number of rearrangements is limited to N-2 because at least two idle ports are required for rearrangement.
With the recursive replacement of DTQ-XBSs, the total number of BSEs after the k-th (0≤k≤n-1) replacement, B_k, is given by

$$B_k = \frac{N^2}{2}\left(1+\frac{1}{2^{k+1}}\right). \tag{3}$$

In (3), B_0=$3N^2/4$ and B_{n-1}=$N(N+1)/2$ denote the number of BSEs in the original DTQ-XBS and that in the TAS, respectively. Figure 9 shows how A_k and B_k depend on the degree of recursion k for N=512 (or n=9). Note that k=0 and k=8 correspond to the original DTQ-XBS and the TAS, respectively. B_k decreases quickly at small k to saturate at $3N^2/4$, while A_k increases steeply at large k to be limited to N-2. In other words, smaller A_k results in more B_k and vice versa. Here we can see a clear trade-off relation in performance between the number of BSEs and rearrangements [11]. Now we define the ratio of the decreased number of BSEs (i.e. B_0 - B_k) to the number of rearrangements (i.e. A_k) as a new figure of merit f_k in order to assess the overall performance of the transformed DTQ-XBSs. Figure 10 shows how f_k depends on k. We see that f_k keeps a constant value of $N^2/8$ for k=0 and 1, while it falls quickly around k=2 or 3. We see a substantial decrease in the number of BSEs can be achieved with a small number of rearrangements. As a consequence, it is interesting to note that the TQ-XBS is promising among the family of switches derived from XBSs (e.g. TAS) although it has more than the number of BSEs necessary for the TAS.

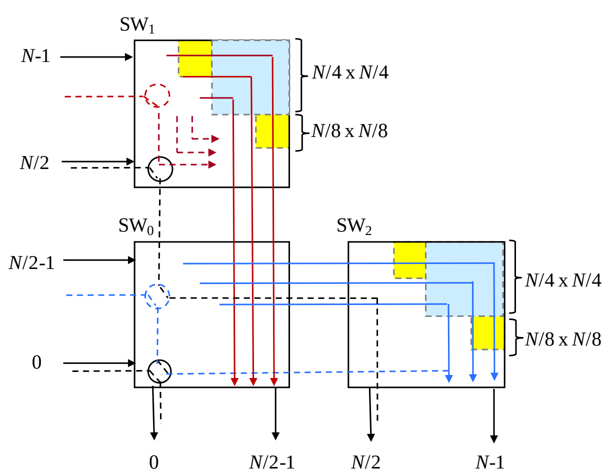


Fig. 8 Recursive replacement of DTQ-XBS.

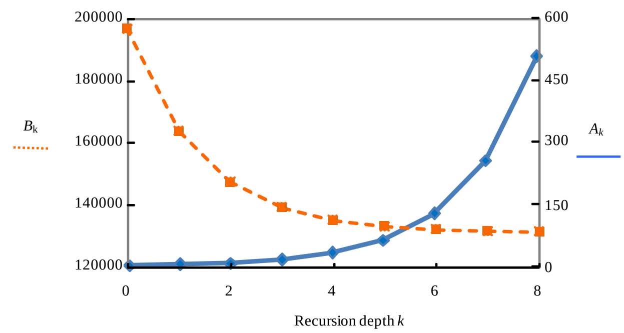


Fig. 9 Trade-off relation in performance between A_k and B_k.

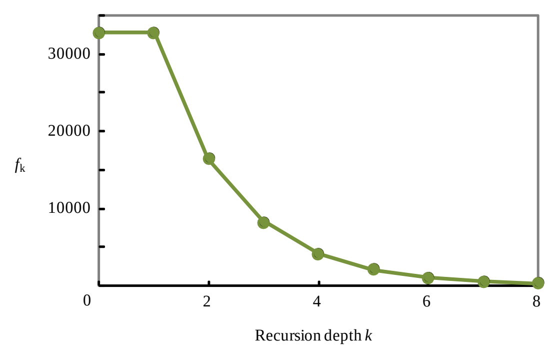


Fig. 10 Figure of merit f_k vs. recursion depth k.

4. Conclusions

The basic properties of the diminished three-quarter crossbar switch (DTQ-XBS) were investigated in detail as a first step to comprehend the properties of three-quarter crossbar switches. It is shown that its switch control complexity for setting up a connection is given by $O(N)$ if a set of switch control data, e.g. R(i) and C(j), is designated and utilized in a appropriate manner. The validity of a worst case scenario for rearrangement assumed in prior works was examined and the identical permutation should be adopted as a new worst case scenario for DTQ-XBSs. It is shown that a partially identical permutation facilitates the derivation of the number of rearrangements for a series of transformed DTQ-XBSs. A new figure of merit, i.e. f_k, was introduced to quantify the performance of transformed DTQ-XBSs, and it is pointed out that transformed DTQ-XBSs with a small degree of replacement show a better performance than the triangular switch.

References

[1] H. S. Hinton, “Photonic switching fabrics,” IEEE Commumications Magazine, vol. 28, no. 4, 1990, 71-89.
[2] G. I. Papadimitriou, C. Papazouglou and A. S. Pomportsis, “Optical switching: switch fabrics, techniques, and

architectures," IEEE Journal of Lightwave Technology, vol. 21, no. 2, 2003, 384-405.

[3] J. E. Midwinter, "Photonic swutching technology: component characteristics versus network requirements," IEEE Journal of Lightwave Technology, vol. 6, no. 10, 1988, 1512-1519.

[4] W. H. Kautz, K. N. Levitt and A. Waksman, "Cellular interconnection arrays," IEEE Trans. on Computers, vol. C-17, no. 5, 1968., 443-451

[5] A. Y. Oruc and D. Prakash, "Routing algorithms for cellular interconnection arrays", IEEE Trans. on Computers, vol. C-33, no. 10, 1984, 939-942.

[6] A. E. Joel, "On permutation switching networks," Bell System Technical Journal, vol. 47, no. 5, 1968, 813-822.

[7] J. C. Majithia, R. Bhar and Y. Tamboi, "Investigation of a setup procedure for triangular permutation arrays," Electronics Letters, vol. 12, issue 1, 1976, 9-10.

[8] H. Obara, M. Sakata and Y. Nemoto, "Bridging crossbar and triangular switches," Proceedings of International Workshop on High Performance and Highly Survivable Routers and Networks, Sendai, Japan, 13-14 March 2008, 31-36.

[9] W. Kabacinski, "Nonblocking electrical and photonic switching fabrics," Springer, New York, 2005.

[10] V. E. Benes, "Mathematical Theory of Connecting Networks and Telephone Traffic," Academic Press, New York, 1965.

[11] M. J. Marcus, "The theory of connecting networks and their complexity; A review," Proceedings of IEEE, vol. 65, no. 9, 1977, 1263-1271.