

# A New Scheme to Realize Crosstalk-free Permutations in Optical MINs with Vertical Stacking<sup>1</sup>

Xiaohong Jiang, Hong Shen, Md. Mamun-ur-Rashid Khandker and Susumu Horiguchi  
Graduate School of Information Science,  
Japan Advanced Institute of Science and Technology,  
JAIST, Tatsunokuchi, ISHIKAWA 923-1292, JAPAN  
Email: { jiang, shen, khandker, hori }@jaist.ac.jp

## Abstract

Vertical stacking is an alternative for constructing nonblocking multistage interconnection networks (MINs). In this paper, we study the crosstalk-free permutation in rearrangeable, self-routing Banyan-type optical MINs built on vertical stacking and propose a new scheme for realizing permutations in this class of optical MINs crosstalk-free. The basic idea of the new scheme is to classify permutations into permutation classes such that all permutations in one class share the same crosstalk-free decomposition pattern. By running the Euler-Split based crosstalk-free decomposition only once for a permutation class and applying the obtained crosstalk-free decomposition pattern to all permutations in the class, crosstalk-free decomposition of permutations can be realized in a more efficient way. We show that the number of permutations in a permutation class is huge, enabling the average time complexity of the new scheme to realize a crosstalk-free permutation in an  $N$  by  $N$  network to be reduced to  $O(N)$  from previously  $O(N \log N)$ .

## 1. Introduction

A basic element of optical switching networks is a directional-coupler (DC) with similar function of  $2 \times 2$  switching element (SE). DC-based optical switching networks can switch signals at the very high speed, and such networks are also capable of switching signals with multiple wavelengths. Crosstalk is a major shortcoming of DC, which occurs between two signals carried in the two waveguides of the coupler [1][2]. By ensuring that only one signal passes through a switch at a time, the first order crosstalk can be eliminated and this provides a cost-effective solution to the crosstalk problem. Due to the stringent bit-error rate requirement of optical transmission facilities, elimination of crosstalk in a DC-based switching system has been widely studied [1,7,10,11,12,13, 14, 15].

Banyan [3] or its topologically equivalent (e.g. baseline, omega) networks [5, 8] are a class of attractive switching networks because they are fast in switch setting (self-routing) and also have a small number of switches between an input-output pair. These characteristics make Banyan-type network an ideal network structure for constructing DC-based optical switching networks. Banyan-type networks have a unique path between an input-output pair, and this makes them blocking networks. Vertical stacking [6] is a novel scheme for constructing nonblocking network as illustrated in Fig.1.

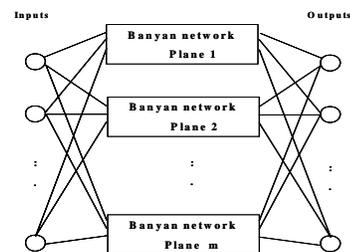


Fig.1 Creating non-blocking network based on the vertical stacking scheme

The rearrangeable nonblocking optical networks are attractive because the cost and signal degradation of a rearrangeable nonblocking optical network are much lower than its strictly nonblocking and wide-sense nonblocking counterparts. Based on the vertical-stacking scheme, the condition for a banyan-type network to be rearrangeable nonblocking and free of crosstalk in SEs has been determined in [9,12]. In this paper, we look into the crosstalk-free permutation in rearrangeable nonblocking Banyan-type optical MINs built on the vertical stacking technique. A scheme was proposed in [17] to realize crosstalk-free permutation in this type of rearrangeable nonblocking optical MINs. The basic idea of this scheme is to first decomposed a permutation into multiple crosstalk-free partial permutations which

<sup>1</sup> This work is support by Telecommunications Advancement Organization of Japan.

requires  $O(M \log N)$  time, and then to realize them crosstalk-free in different planes (stacked copies) of the MIN simultaneously. In this paper, we present a new scheme for realizing permutations in this class of vertically stacked optical MINs crosstalk-free. Since inherent similarities among permutations in the sense of crosstalk-free decomposition are fully utilized in the new scheme, crosstalk-free permutation can be realized with an average time complexity of  $O(N)$ .

## 2. Rearrangeable nonblocking networks under crosstalk-free constraint

Ideally, we are interested in designing a network without any crosstalk. For convenience, we use the notation  $B(N, p)$  to refer to an  $N \times N$  MIN that consists of  $p$  vertically stacked copies of Banyan-type networks. We have the following result concerning the rearrangeable nonblocking conditions for a  $B(N, p)$  network [9,12].

**Theorem 1:** Under the constraint of crosstalk-free, a  $B(N, p)$  network is rearrangeable nonblocking if the following is true

$$p \geq 2^{\lfloor (n+1)/2 \rfloor} \quad (1)$$

Hereafter, we will use  $RB(N)$  to refer to the rearrangeable nonblocking network  $B(N, p)$  which consists of  $2^{\lfloor (n+1)/2 \rfloor}$  copies (planes) of Banyan-type network. Note that under the constraint of crosstalk-free, the complexity of a rearrangeable Banyan-type optical MIN is much lower than its strictly nonblocking counterpart [10] and its wide-sense nonblocking counterpart [16].

## 3. Permutation and Crosstalk-Free Partial Permutation (CFPP)

A permutation is a full one-to-one mapping between the network inputs and outputs. For an  $N \times N$  Banyan-type network, suppose input  $x_i$  is mapped to output  $y_i$ , where  $x_i = i$  and  $y_i \in \{0, 1, \dots, N-1\}$  for  $i = 0, 1, \dots, N-1$ . We denote this permutation as:

$$\begin{pmatrix} x_0, x_1, \dots, x_{N-1} \\ y_0, y_1, \dots, y_{N-1} \end{pmatrix} \quad (2)$$

In addition, we call a one-to-one mapping between  $N_1$  inputs and  $N_1$  outputs in the network ( $N_1 < N$ ) a partial permutation.

**Definition 1** A partial permutation is called Crosstalk Free Partial Permutation (CFPP) to an optical MIN if the partial permutation is crosstalk-free realizable to the optical MIN.

**Example 1** The decomposition of a permutation into CFPPs.

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 8 & 2 & 0 & 12 & 4 & 13 & 3 & 11 & 9 & 1 & 6 & 7 & 5 & 10 & 15 & 14 \end{pmatrix} \Rightarrow \quad (3)$$

$$\begin{pmatrix} 1 & 7 & 10 & 14 \\ 2 & 11 & 6 & 15 \end{pmatrix} \circ \begin{pmatrix} 2 & 5 & 8 & 12 \\ 0 & 13 & 9 & 5 \end{pmatrix} \circ \begin{pmatrix} 0 & 6 & 11 & 15 \\ 8 & 3 & 7 & 14 \end{pmatrix} \circ \begin{pmatrix} 3 & 4 & 9 & 13 \\ 12 & 4 & 1 & 10 \end{pmatrix}$$

## 4. Crosstalk-free permutation in $RB(N)$ network

The above result indicates that all permutations can be realized crosstalk-free in a  $RB(N)$ . In [17], the following Lemma concerning the crosstalk-free property of an optical MIN was presented.

**Lemma 1** For an  $N \times N$  Banyan-type network and an integer  $i$  ( $0 \leq i \leq \lfloor (n+1)/2 \rfloor - 1$ ), define sets:

$$I_j^{(i)} = \{u_{2^i-j}, u_{2^i-j+1}, \dots, u_{2^i-j+2^i-1}\}, O_j^{(i)} = \{v_{2^i-j}, v_{2^i-j+1}, \dots, v_{2^i-j+2^i-1}\}, 0 \leq j \leq \frac{N}{2^{i+1}} - 1$$

where  $u_0, u_1, \dots, u_{N/2-1}$  are the  $N/2$  inputs switches and  $v_0, v_1, \dots, v_{N/2-1}$  are the  $N/2$  outputs switches. For the two inputs (outputs) of any two one-pair mappings in the network, if their corresponding two input (output) switches belong to two different input (output) sets defined above, the two mappings will be crosstalk-free in the first (last)  $i+1$  stages of the network.

The following algorithm was also developed in [17] to actually decompose a permutation of set  $\{0, 1, \dots, N-1\}$  into  $2^{\lfloor (n+1)/2 \rfloor}$  CFPPs for a  $RB(N)$  network.

### Algorithm 1: CFPP decomposition algorithm:

**Initiate:**  $i = 0$  and take the permutation as the 0-level partial permutation.

**Step 1:** If  $i = \lfloor (n+1)/2 \rfloor$ , exit.

**Step 2:** For each  $i$ -level partial permutation, do steps 3-4.

**Step 3:** Construct a bipartite graph  $G = (V_1, V_2; E)$  for the  $i$ -level partial permutation. The vertex sets of  $G$  are defined by:

$$V_1 = \{A_0^{(i)}, A_1^{(i)}, \dots, A_{\frac{N}{2^{i+1}}-1}^{(i)}\}, V_2 = \{A_0^{(i+1)}, A_1^{(i+1)}, \dots, A_{\frac{N}{2^{i+1}}-1}^{(i+1)}\}$$

Here  $A_j^{(i)} = \{u_{2^i-j}, u_{2^i-j+1}, \dots, u_{2^i-j+2^i-1}\}$ ,  $A_j^{(i+1)} = \{v_{2^i-j}, v_{2^i-j+1}, \dots, v_{2^i-j+2^i-1}\}$

for  $0 \leq j \leq \frac{N}{2^{i+1}} - 1$ , and  $u_0, u_1, \dots, u_{N/2-1}$  are the  $N/2$  inputs switches and  $v_0, v_1, \dots, v_{N/2-1}$  are the  $N/2$  outputs switches.

The edge set  $E$  is defined as: for any one-pair mapping  $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$  in the  $i$ -level partial permutation, if the input switch

corresponding to  $x_i$  belongs to  $A_j^{(i)}$  and the outputs switch corresponding to  $y_i$  belongs to  $A_{j_2}^{(i+1)}$ , then there is an edge between vertex  $A_j^{(i)}$  and vertex  $A_{j_2}^{(i+1)}$  in  $E$ .

**Step 4:** Find the Euler tour.

Since any vertex in each connected component of  $G$  has degree 2, we know from graph theory [4] that there exists an Euler tour which traverses each edge of the component exactly once. Then for each connected component of  $G$ , start from any vertex in  $V_1$  in it, traverse through an unvisited edge to the neighboring vertex  $V_2$ , back and forth until return to the starting vertex. During the traversing, a visited edge is will be placed into set  $E_1$  if the traverse direction on this edge is from  $V_1$  to  $V_2$ ; and placed into set  $E_2$  if the direction is opposite. It is easy to see that the set of all edges in  $E_1$  is a perfect matching of the bipartite graph  $G$ , so is the set of edges in  $E_2$ .

**Step 5:** Take all one-pair mappings corresponding to the edges in  $E_1$ , to form one  $(i+1)$ -level partial permutation corresponding to the  $i$ -level partial permutation; let the remaining one-pair mappings, corresponding to the edges in  $E_2$ , form another  $(i+1)$ -level partial permutation corresponding to the  $i$ -level partial permutation.

**Step 6:**  $i \leftarrow i + 1$ . Go to Step 1.

It is clear that after running the decomposition algorithm for a permutation in an  $N \times N$  Banyan-type MIN, the permutation will be decomposed into  $2^{\lfloor (n+1)/2 \rfloor}$  partial permutations that eliminate the crosstalk in all stages of the network as guaranteed by Lemma 1. By

realizing each of these CFPPs in a single plane of a RB(N) network, the full permutation can be realized crosstalk-free in a single pass based on the parallel message transmission. The Steps 2-5 take  $O(N)$  steps and these steps repeat  $O(\log_2 N)$  times, the time complexity of the decomposition algorithm is therefore  $O(M\log_2 N)$ . Since the RB(N) network consists of multi-copies of Banyan-type networks which are self-routing, so the routing complexity is  $O(\log_2 N)$ .

It is easy to verify that if we apply the above Algorithm to the permutation (3), the CFPPs obtained are just the four CFPPs given in example 1. The results in the Algorithm 1 show that any permutation of  $N$ -element set  $\{0,1,\dots, N-1\}$  can be decomposed into  $\sqrt{N}$  ( $\sqrt{2N}$ ) CFPPs if  $\log_2 N$  is even (odd) and each of these CFPPs consists of  $\sqrt{N}$  ( $\sqrt{N/2}$ ) mapping pairs. To avoid confusion, we will refer to this special kind of CFPPs of a permutation as the *Specified Crosstalk-Free Partial Permutation* (SCFPPs) of the permutation. It is easy to say that the CFPPs of a permutation obtained by running Algorithm 1 over the permutation are just the SCFPPs of the permutation.

## 5. A new scheme for crosstalk-free decomposition

As discussed in Section 4, the overall time complexity to realize a permutation crosstalk-free in a RB(N) network is dominated by the time complexity of the CFPP decomposition algorithm, which is  $O(M\log_2 N)$ . Thus, the performance of high speed optical MIN can be enhanced significantly if the time complexity of decomposition algorithm can be reduced. In this section, we present a new scheme for crosstalk-free decomposition. The basic idea of the scheme is to classify permutations into permutation classes such that all permutations in one class share the same crosstalk-free decomposition pattern. By running the Euler-Split based algorithm 1 only once for a permutation class and applying the obtained crosstalk-free decomposition patterns to all permutations in the class, crosstalk-free decomposition of permutations can be realized in a more efficient way.

### 5.1 Permutation pattern and permutation class

To explore the class of permutations that have similar crosstalk-free decompositions, we start with the following definition based on the crosstalk-free condition of an entire Banyan-type network as described in Lemma 1.

**Definition 2:** For a permutation of form (2), we set  $i = \lfloor (n+1)/2 \rfloor - 1$ . Then a specified undirected bipartite graph  $G^* = (V_1, V_2; E)$  can be constructed for the permutation as that of step 3 in Algorithm 1. We define the topology pattern (ignoring the details of mapping pairs) of the bipartite graph  $G^* = (V_1, V_2; E)$  to be the *permutation pattern* of the permutation. If we let the vertices of  $V_1$  and the vertices of  $V_2$  correspond to the rows and the columns of a matrix, respectively, then the permutation pattern can also be expressed as a matrix  $M_{PP} = (a_{jk})_{\frac{N}{2^{i+1}} \times \frac{N}{2^{i+1}}}$  with

its entry  $a_{jk}$  being the number of edge(s) from  $j$ -th vertex in  $V_1$  to  $k$ -th vertex in  $V_2$ . We define the matrix  $M_{PP}$  as the *permutation pattern matrix* (PPM) of the given permutation. Furthermore, we can also construct a matrix  $M_{DPP} = (b_{jk})_{\frac{N}{2^{i+1}} \times \frac{N}{2^{i+1}}}$  with its entry  $b_{jk}$  being the set

of the mapping pair(s) corresponding to the edge(s) from  $j$ -th vertex in  $V_1$  to  $k$ -th vertex in  $V_2$ . We call the matrix  $M_{DPP} = (b_{jk})_{\frac{N}{2^{i+1}} \times \frac{N}{2^{i+1}}}$  the *detailed permutation pattern matrix* (DPPM).

**Example 2.** The permutation pattern, PPM and DPPM of permutation (3) in Example 1.

According to the definition, the permutation pattern of permutation (3) is illustrated in Figure 2.

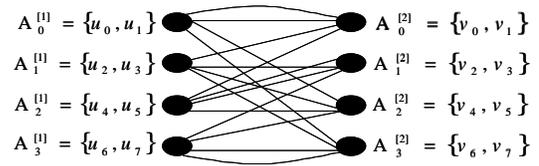


Fig.2 The permutation pattern of permutation (3)

The corresponding PPM of the permutation is:

$$M_{PP} = \begin{bmatrix} 2 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

If we number mapping pairs as  $e_j = \begin{pmatrix} x_j \\ y_j \end{pmatrix}$  for  $0 \leq j \leq 15$ ,

the DPPM of the permutation is thus:

$$M_{DPP} = \begin{bmatrix} \{e_1, e_2\} & \{-\} & \{e_0\} & \{e_3\} \\ \{e_6\} & \{e_4\} & \{e_7\} & \{e_5\} \\ \{e_9\} & \{e_{10}, e_{11}\} & \{e_8\} & \{-\} \\ \{-\} & \{e_{12}\} & \{e_{13}\} & \{e_{14}, e_{15}\} \end{bmatrix}$$

Here  $\{-\}$  means a null set.

We are now in the position to introduce the definition of permutation class.

**Definition 3:** We define the set of all the permutations corresponding to the same PPM (permutation pattern) to be a *permutation class*.

It should be noted that all permutations in a class share a common PPM, but each permutation has a distinct DPPM. About the number of permutations in a permutation class, we have the following result:

**Theorem 2:** In a permutation class on  $\{0,1,\dots, N-1\}$ , there are at least  $(\sqrt{N!})^{N/2}$  permutations if  $\log_2 N$  is even, and at least  $(\sqrt{2N!})^{N/2}$  permutations if  $\log_2 N$  is odd.

**Proof:** Omitted.

### 5.2 CFPP matrices of a permutation

As indicated in the decomposition Algorithm 1 and Definition 2, the permutation pattern of a permutation of

$N$ -element set  $\{0,1,\dots, N-1\}$  can be decomposed into  $2^{\lfloor (n+1)/2 \rfloor}$  disjoint sub-patterns, and each of these sub-patterns is a perfect match of the specified bipartite graph  $G^* = (V_1, V_2; E)$  of the permutation and corresponds to a SCFPP of the permutation. Since a permutation pattern is completely specified by its permutation pattern matrix (PPM) defined in Definition 2, a sub-pattern discussed above will also be completely specified by a matrix defined in the same way as that of PPM. We introduce the following definition to characterize the matrix defined for a sub-pattern.

**Definition 4:** We define the matrices, which are defined by the  $2^{\lfloor (n+1)/2 \rfloor}$  disjoint sub-patterns decomposed from the permutation pattern of a permutation on  $\{0,1,\dots, N-1\}$  by using Algorithm 1, to be the *CFPP matrices* of the permutation.

We have the following results regarding the CFPP matrices of a permutation:

**Lemma 2:** For any permutation, each of its CFPP matrices is a permutation matrix, and the sums of its CFPP matrices are just the PPM of the permutation.

**Proof:** Omitted.

Since all permutations in one class have a same permutation pattern and the CFPP matrices of a permutation are only determined by its permutation pattern, we have:

**Corollary 1:** All permutations in one permutation class have the same set of CFPP matrices.

### 5.3 Compact CFPP matrix of a permutation class

Since all the permutations in one permutation class have a same permutation pattern and same set of CFPP matrices, we can randomly select one permutation from the class to get the CFPP matrices for the class. After we run the decomposition algorithm 1 over the selected permutation, we can get  $2^{\lfloor (n+1)/2 \rfloor}$  SCFPPs of the permutation and  $2^{\lfloor (n+1)/2 \rfloor}$  disjoint sub-patterns of the specified bipartite graph  $G = (V_1, V_2; E)$  defined for the permutation, and each of these sub-patterns corresponds to a SCFPP. Then  $2^{\lfloor (n+1)/2 \rfloor}$  CFPP matrices of the permutation can be constructed based on these  $2^{\lfloor (n+1)/2 \rfloor}$  SCFPPs, as that of constructing PPM based on the permutation.

Note that every CFPP matrix of a class is just a permutation matrix, and a permutation matrix can be completely specified by the positions of unit entries in the matrix. So we actually only need to calculate and keep the positions of unit entries in the CFPP matrices of a class, such that the  $2^{\lfloor (n+1)/2 \rfloor}$  CFPP matrices of a permutation can be expressed in a more compact way.

**Definition 5:** For a permutation class and its CFPP matrices, we construct a matrix such that each row of the matrix contains the positions of unit entries in a CFPP matrix. We define the matrix as the *compact CFPP matrix* of the class.

We present here an algorithm for getting the compact CFPP matrix of a class.

#### Algorithm 2 Getting compact CFPP matrix for a class

**Step 1:** For a permutation class of  $N$ -element set  $\{0,1,\dots, N-1\}$ , we set  $i = \lfloor (n+1)/2 \rfloor - 1$  and take a permutation from

the class. Express the permutation in form of  $(e_0, e_1, \dots, e_{N-1})$  with  $e_j = \begin{pmatrix} x_j \\ y_j \end{pmatrix}$  being the mapping pair.

**Step 2:** Run the decomposition algorithm 1 over the permutation, and summarize the SCFPPs obtained as a matrix:

$$M_1 = \begin{bmatrix} e_{d_0} & e_{d_1} & \dots & e_{d_{\frac{N}{2^{i+1}}-1}} \\ e_{d_{\frac{N}{2^{i+1}}}} & e_{d_{\frac{N}{2^{i+1}}+1}} & \dots & e_{d_{2\frac{N}{2^{i+1}}-1}} \\ \dots & \dots & \dots & \dots \\ e_{d_{(\frac{2^{i+1}-1)\frac{N}{2^{i+1}})}} & e_{d_{(\frac{2^{i+1}-1)\frac{N}{2^{i+1}}+1)}} & \dots & e_{d_{N-1}} \end{bmatrix}$$

Here  $\{d_0, d_1, \dots, d_{N-1}\} = \{0, 1, \dots, N-1\}$ , and each row of the matrix corresponds to a SCFPP of the original permutation.

**Step 3:** Get the following matrix based on matrix  $M_1$ .

$$M_2 = \begin{bmatrix} \left( \left\lfloor \frac{x_{d_0}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_0}}{N/2^{i+1}} \right\rfloor \right) & \left( \left\lfloor \frac{x_{d_1}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_1}}{N/2^{i+1}} \right\rfloor \right) & \dots & \left( \left\lfloor \frac{x_{d_{\frac{N}{2^{i+1}}-1}}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_{\frac{N}{2^{i+1}}-1}}}{N/2^{i+1}} \right\rfloor \right) \\ \left( \left\lfloor \frac{x_{d_{\frac{N}{2^{i+1}}}}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_{\frac{N}{2^{i+1}}}}}{N/2^{i+1}} \right\rfloor \right) & \left( \left\lfloor \frac{x_{d_{\frac{N}{2^{i+1}}+1}}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_{\frac{N}{2^{i+1}}+1}}}{N/2^{i+1}} \right\rfloor \right) & \dots & \left( \left\lfloor \frac{x_{d_{2\frac{N}{2^{i+1}}-1}}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_{2\frac{N}{2^{i+1}}-1}}}{N/2^{i+1}} \right\rfloor \right) \\ \dots & \dots & \dots & \dots \\ \left( \left\lfloor \frac{x_{d_{(\frac{2^{i+1}-1)\frac{N}{2^{i+1}})}}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_{(\frac{2^{i+1}-1)\frac{N}{2^{i+1}})}}}{N/2^{i+1}} \right\rfloor \right) & \left( \left\lfloor \frac{x_{d_{(\frac{2^{i+1}-1)\frac{N}{2^{i+1}}+1)}}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_{(\frac{2^{i+1}-1)\frac{N}{2^{i+1}}+1)}}}{N/2^{i+1}} \right\rfloor \right) & \dots & \left( \left\lfloor \frac{x_{d_{N-1}}}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_{d_{N-1}}}{N/2^{i+1}} \right\rfloor \right) \end{bmatrix}$$

Note that  $M_2$  is converted from  $M_1$  by replacing each mapping pair  $e_j = \begin{pmatrix} x_j \\ y_j \end{pmatrix}$  in  $M_1$  with the number pair

$$\left( \left\lfloor \frac{x_j}{N/2^{i+1}} \right\rfloor, \left\lfloor \frac{y_j}{N/2^{i+1}} \right\rfloor \right).$$

The number pairs in each row of  $M_2$  are just the positions of the units in the CFPP matrix corresponding to the SCFPP in the same row of  $M_1$ . Thus,  $M_2$  is just the compact CFPP matrix of the class.

**Example 3** Compute the compact CFPP matrix for the permutation class containing permutation (3).

From example 1 we know that after we run the decomposition algorithm 1 over permutation (3), the four SCFPPs obtained can be summarized as the following matrix:

$$M_1 = \begin{bmatrix} e_1 & e_7 & e_{10} & e_{14} \\ e_2 & e_5 & e_8 & e_{12} \\ e_0 & e_6 & e_{11} & e_{15} \\ e_3 & e_4 & e_9 & e_{13} \end{bmatrix}$$

Here we number mapping pair as  $e_j = \begin{pmatrix} j \\ - \end{pmatrix}$  for  $0 \leq j \leq 15$ .

The corresponding compact CFPP matrix will be:

$$M_2 = \begin{bmatrix} \left( \left\lfloor \frac{1}{4} \right\rfloor, \left\lfloor \frac{2}{4} \right\rfloor \right) & \left( \left\lfloor \frac{7}{4} \right\rfloor, \left\lfloor \frac{11}{4} \right\rfloor \right) & \left( \left\lfloor \frac{10}{4} \right\rfloor, \left\lfloor \frac{6}{4} \right\rfloor \right) & \left( \left\lfloor \frac{14}{4} \right\rfloor, \left\lfloor \frac{15}{4} \right\rfloor \right) \\ \left( \left\lfloor \frac{2}{4} \right\rfloor, \left\lfloor \frac{0}{4} \right\rfloor \right) & \left( \left\lfloor \frac{5}{4} \right\rfloor, \left\lfloor \frac{13}{4} \right\rfloor \right) & \left( \left\lfloor \frac{8}{4} \right\rfloor, \left\lfloor \frac{9}{4} \right\rfloor \right) & \left( \left\lfloor \frac{12}{4} \right\rfloor, \left\lfloor \frac{5}{4} \right\rfloor \right) \\ \left( \left\lfloor \frac{0}{4} \right\rfloor, \left\lfloor \frac{8}{4} \right\rfloor \right) & \left( \left\lfloor \frac{6}{4} \right\rfloor, \left\lfloor \frac{3}{4} \right\rfloor \right) & \left( \left\lfloor \frac{11}{4} \right\rfloor, \left\lfloor \frac{7}{4} \right\rfloor \right) & \left( \left\lfloor \frac{15}{4} \right\rfloor, \left\lfloor \frac{14}{4} \right\rfloor \right) \\ \left( \left\lfloor \frac{3}{4} \right\rfloor, \left\lfloor \frac{12}{4} \right\rfloor \right) & \left( \left\lfloor \frac{4}{4} \right\rfloor, \left\lfloor \frac{4}{4} \right\rfloor \right) & \left( \left\lfloor \frac{9}{4} \right\rfloor, \left\lfloor \frac{1}{4} \right\rfloor \right) & \left( \left\lfloor \frac{13}{4} \right\rfloor, \left\lfloor \frac{10}{4} \right\rfloor \right) \end{bmatrix} = \begin{bmatrix} (0,0) & (1,2) & (2,1) & (3,3) \\ (0,0) & (1,3) & (2,2) & (3,1) \\ (0,2) & (1,0) & (2,1) & (3,3) \\ (0,3) & (1,1) & (2,0) & (3,2) \end{bmatrix}$$

Since all the permutations in one permutation class have same set of CFPP matrices and thus same compact CFPP matrix, the compact CFPP matrix obtained for

permutation (3) is just the compact CFPP matrix of the permutation class containing permutation (3).

The results in Theorem 2 indicate that there are a huge number of permutations in one permutation class, and all these permutations have the same permutation pattern and thus same crosstalk-free decompositions. In the following, we will present an approach for crosstalk-free decomposition of permutations in one class. The approach consists of two integrated parts: Getting the compact CFPP matrix for a class by running the decomposition algorithm 1 over one of the permutation in the class and applying the obtained compact CFPP matrix to any new permutation in the class to get the crosstalk-free decomposition of the new permutation.

#### 5.4 Crosstalk-free decomposition based on the compact CFPP matrix of a class

By the definition of detailed permutation pattern matrix (DPPM) and permutation pattern matrix (PPM), we know that there is a one-to-one mapping between a permutation pattern and a PPM, and there is a one-to-one mapping between a permutation and a DPPM. For a permutation, its PPM has the same structure as that of its DPPM except that every entry in PPM is only the number of mapping pair(s) of the corresponding entry in DPPM. All the permutations in one class have a common PPM and permutation pattern, but every permutation in a class has its distinct DPPM.

For a permutation of  $N$ -element set  $\{0,1,\dots,N-1\}$ , we can get its  $2^{\lfloor (n+1)/2 \rfloor}$  SCFPPs based on its  $2^{\lfloor (n+1)/2 \rfloor}$  CFPP matrices as follows: For the first CFPP matrix, we take one mapping pair from each position in the DPPM of the permutation, where 1 occurs in the same position in this CFPP matrix, and then get a set of mapping pairs corresponding to the first CFPP matrix. For the second CFPP matrix, we take one mapping pair from each position in the DPPM, where 1 occurs in the same position in this second CFPP matrix, and then get a set of mapping pairs corresponding to the second CFPP matrix. We repeat this until the last CFPP matrix. Its easy to see that each set of mapping pairs corresponding to each CFPP matrix is just a SCFPP because of the definitions of CFPP and SCFPP. Since the DPPM of a permutation is specified by the PPM of the permutation, and the sums of all the CFPP matrices of the permutation is just the PPM of the permutation, the above process works for any permutation if we obtain its CFPP matrices.

The algorithm 2 indicates that all the CFPP matrices of a permutation class can be summarized as a single compact CFPP matrix, so the crosstalk-free decomposition of the permutation can also be implemented by using its compact CFPP matrix. We present here an efficient algorithm to decompose any permutation of a class into SCFPPs based on the compact CFPP matrix of the class.

#### Algorithm 3 Crosstalk-free Decomposition Based on Compact CFPP Matrix of a Class

**Step 1:** For a permutation in the permutation class of  $N$ -element set  $\{0,1,\dots,N-1\}$ , construct the DPPM of the permutation.

**Step 2:** For every row in the compact CFPP matrix of the class, we take the mapping pairs from the DPPM of the

permutation according to the positions indicated in the row of the compact CFPP matrix, and then get a set of mapping pairs corresponding to a row of the compact CFPP matrix. Each set of mapping pairs obtained is just a SCFPP of the permutation.

The above algorithm is correct by the meanings of SCFPP and compact CFPP matrices. Since Step 1 takes  $O(N)$  time to construct the DPPM of a permutation, and Step 2 takes  $O(N)$  time to get the SCFPPs of the permutation based on its DPPM and compact CFPP matrices, the time complexity of the algorithm is  $O(N)$ .

**Example 4** Crosstalk-free decomposition of the permutation (4) based on its compact CFPP matrix.

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 2 & 12 & 8 & 0 & 10 & 13 & 3 & 6 & 1 & 9 & 5 & 4 & 14 & 11 & 15 & 7 \end{pmatrix} \quad (4)$$

Since permutation (4) has the same PPM as that of permutation (3), they belong to the same class and thus have the same compact CFPP matrix. From example 3, we know that the compact CFPP matrix of the permutation is:

$$M_2 = \begin{bmatrix} (0,0) & (1,2) & (2,1) & (3,3) \\ (0,0) & (1,3) & (2,2) & (3,1) \\ (0,2) & (1,0) & (2,1) & (3,3) \\ (0,3) & (1,1) & (2,0) & (3,2) \end{bmatrix}$$

If we number mapping pairs as  $e_j = \begin{pmatrix} j \\ - \end{pmatrix}$  for  $0 \leq j \leq 15$ ,

the DPPM of the permutation will be:

$$M_{DPP} = \begin{bmatrix} \{e_0, e_3\} & \{-\} & \{e_2\} & \{e_1\} \\ \{e_6\} & \{e_7\} & \{e_4\} & \{e_5\} \\ \{e_8\} & \{e_{10}, e_{11}\} & \{e_9\} & \{-\} \\ \{-\} & \{e_{15}\} & \{e_{13}\} & \{e_{12}, e_{14}\} \end{bmatrix}$$

After we apply the compact CFPP matrix to the DPPM according to algorithm 3, we get immediately the four SCFPPs of the permutation:  $(e_0, e_4, e_{10}, e_{12})$ ,  $(e_3, e_5, e_9, e_{15})$ ,

$(e_2, e_6, e_{11}, e_{14})$ ,  $(e_1, e_7, e_8, e_{13})$ , which are:

$$\begin{pmatrix} 0 & 4 & 10 & 12 \\ 2 & 10 & 5 & 14 \end{pmatrix}, \begin{pmatrix} 3 & 5 & 9 & 15 \\ 0 & 13 & 9 & 7 \end{pmatrix}, \begin{pmatrix} 2 & 6 & 11 & 14 \\ 8 & 3 & 4 & 15 \end{pmatrix}, \begin{pmatrix} 1 & 7 & 8 & 13 \\ 12 & 6 & 1 & 11 \end{pmatrix}.$$

#### 5.5 Numbering the permutation class

Since a permutation class is identified by its permutation pattern matrix (PPM), we can number each permutation class by assigning a distinct number to its distinct PPM. For a permutation class of  $N$ -element set  $\{0,1,\dots,N-1\}$  and if  $\log N$  is even, its PPM will be in the form  $M_{PP} = (a_{jk})_{\sqrt{N} \times \sqrt{N}}$  with  $0 \leq a_{jk} \leq \sqrt{N}$ . Then we can assign the PPM a distinct number as:

$$\sum_{j=0}^{\sqrt{N}-1} \sum_{k=0}^{\sqrt{N}-1} a_{jk} \cdot \sqrt{N}^{j \cdot \sqrt{N} + k} \quad (5)$$

When  $\log N$  is odd, the distinct number can be evaluated in a similar way. Noted that the numbering of a PPM can be finished in  $O(N)$  time.

#### 5.6 The overall decomposition algorithm based on permutation class

The results in Section 5.3 indicate that we only need to run the decomposition algorithm 1 once for a permutation class to get the compact CFPP matrix of the class, then the crosstalk-free decomposition of any permutation in the class can be implemented in a simple way as discussed in Section 5.4.

Since a permutation class is identified by its PPM, we can determine a class for a permutation by checking its PPM. We are now in the position to give a high-level description of the overall decomposition algorithm based on the idea of permutation class.

#### Algorithm 4 Crosstalk-free Decomposition based on Permutation Class

**Step 1:** For a new permutation of  $N$ -element set  $\{0,1,\dots,N-1\}$ , construct its PPM and calculate its class number.

**Step 2:** Check the address corresponding to the class number. Go to Step 3 if the compact CFPP matrix is available in the address, go to Step 4 otherwise.

**Step 3:** Use Algorithm 3 in Section 5.4 to get the SCFPPs of the permutation.

**Step 4:** Use Steps 2-3 in Algorithm 2 to get the SCFPPs and compact CFPP matrix of the permutation, and save the compact CFPP matrix to the address corresponding to the class number.

Since the Steps 1-3 take  $O(N)$  time and Step 4 takes  $O(M\log N)$  time, then we have the following results regarding time complexity of Algorithm 4.

**Theorem 3:** To decompose a permutation of  $N$ -element set  $\{0,1,\dots,N-1\}$  into SCFPPs based on Algorithm 4, the average time complexity is  $O(N)$ .

**Proof:** Omitted.

Since a  $RB(N)$  network consists of multi Banyan networks which are self-routing, so the routing complexity is  $O(\log_2 N)$ . Thus, the overall time complexity of the new scheme is  $O(N) + O(\log_2 N) \sim O(N)$  to realize a permutation crosstalk-free in a  $RB(N)$  network.

## 6. Conclusions

In this paper, we have proposed a new scheme for crosstalk-free realization of permutations in rearrangeable and self-routing optical MINs built on vertical stacking. We have introduced the idea of permutation class in the scheme to make the crosstalk-free decomposition permutations more efficient. We have shown that there is a large number of permutations in a permutation class that share the same permutation pattern and thus the same crosstalk-free decompositions. By getting the crosstalk-free decomposition pattern of a class based on the Euler-split technique and applying it to all the permutations in the class, the crosstalk free decompositions of these permutations can be made efficient. Our new scheme has an average time complexity of  $O(N)$  instead of  $O(M\log N)$  required previously for realizing any crosstalk-free permutation on  $\{0,1,\dots,N-1\}$  in this type of optical MINs.

## References

[1] V.R.Chinni et al., "Crosstalk in a lossy directional coupler switch," *J.Lightwave Technol.*, vol.13, no.7, pp.1530-1535, July 1995.

[2] H.S.Hinton, *An introduction to Photonic Switching Fabrics*, New York: Plenum, 1993.

[3] G.R.Goke and G.J.Lipovski, "Banyan networks for partitioning multiprocessor systems," *Proc.1<sup>st</sup> Annu. Symp. Comp. Arch.*, pp.21-28,1973.

[4] B.Kolman, R.C.Busby and S.Ross, *Discrete Mathematical Structures, 3rd Edition*, Prentice Hall, 1996.

[5] C.Kruskal and M.Snir, "The performance of multistage interconnection networks for multiprocessors," *IEEE Trans. Commun.*, vol.COM-32, pp.1091-1098, Dec.1983.

[6] C.-T. Lea, "Multi- $\log_2 N$  networks and their applications in high speed electronic and photonic switching systems," *IEEE Trans. Commun.*, vol.38, pp.1740-1749, Oct. 1990.

[7] C.-T. Lea, "Crossover minimization in directional coupler-based photonic switching systems," *IEEE Trans. Commun.*, vol.36, pp.355-363, Mar.1988.

[8] J.Patel, "Performance of processor-memory interconnections for multiprocessors," *IEEE Trans. Comput.*, vol.C-30, pp.771-780, Oct.1981.

[9] Xiaohong Jiang, Md. Mamun-ur-Rashid Khandker and S.Horiguchi, "Nonblocking Optical MINs Under Crosstalk-free Constraint", Proceedings of the 2001 IEEE Workshop on High Performance Switching and Routing, pp.307-311, May.2001, Dallas, USA.

[10] M.M.Vaez and C.-T. Lea, "Strictly nonblocking directional-coupler-based switching networks under crosstalk constraint," *IEEE Trans. Commun.*, vol.48, no.2, pp.316-323, Feb. 2000.

[11] M.M.Vaez and C.-T. Lea, "Space-wavelength tradeoff in the design of nonblocking directional coupler based network under crosstalk constraint," *J. Lightwave Technol.*, vol.16, pp.1373-1379, Aug.1998.

[12] G.Maier and A.Pattavina, "Design of photonic rearrangeable networks with zero first-order switching-element-crosstalk," *IEEE Trans. Commun.*, vol.49, no.7, pp.1268-1279, July.2001.

[13] K.Padmanabhan and A.Netravali, "Dilated networks for photonic switching," *IEEE Trans. Commun.*, vol.COM-35, pp.1357-1365, Dec.1987.

[14] D.Li, "Elimination of crosstalk in directional coupler switches," *Optical Quantum Electron.*, vol.25, no.4, pp.255-260, Apr.1993.

[15] T.-S. Wong and C.-T. Lea, "Crosstalk reduction through wavelength assignment in WDM photonic switching networks," *IEEE Trans. Commun.*, vol.49, no.7, pp.1280-1287, July.2001.

[16] M.M.Vaez and C.-T. Lea, "Wide-sense nonblocking Banyan-type switching systems based on directional couplers," *IEEE J. Select. Areas Commun.*, vol.16, pp.1327-1332, Sept.1998.

[17] Xiaohong Jiang, Md. Mamun-ur-Rashid Khandker, Hong Shen and S.Horiguchi, "Realizing Crosstalk-free Permutation in Vertically Stacked Optical Multistage Interconnection Networks," *Submitted for publication*.