國 立 交 通 大 學

應用數學系

碩 士 論 文

合併的光學多級式網路之無開關干擾之可重排性

On the Crosstalk-free Rearrangeability of Combined Optical Multistage Interconnection Networks

研 究 生：黃志文

指導教授：陳秋媛　教授

中 華 民 國 九 十 八 年 一 月

合併的光學多級式網路之
無開關干擾之可重排性

On the Crosstalk-free Rearrangeability of Combined Optical Multistage Interconnection Networks

研 究 生：黃志文　　Student：Chih-Wen Huang

指導教授：陳秋媛　　Advisor：Chiuyuan Chen

國 立 交 通 大 學

應 用 數 學 系

碩 士 論 文

A Thesis
Submitted to Department of Applied Mathematics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Applied Mathematics
January 2009
Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 八 年 一 月

合併的光學多級式網路之無開關干擾之可重排性

研究生：黃志文　　　　　　　　　　　指導老師：陳秋媛　教授

國 立 交 通 大 學

應 用 數 學 系

摘　要

一個多級式連接網路的可重排性，是指這個網路的 N 個輸入到 N 個輸出，在必要時允許重新連線的情況下，是否可以連接所有 $N!$種可能的輸入輸出排列。在文獻[8]中，Das 對於合併的 $2n-1$ 階級的多級式連接網路的可重排性，提出了一個漂亮的充分條件，其中 $n = \log_2 N$，Das 並且對符合這個充分條件的多級式連接網路提出一個時間複雜度為 $O(Nn)$的排列繞送演算法。然而，上述的可重排性的定義以及 Das 的結果，皆只適用於電子的多級式連接網路。如今，光學的多級式連接網路，因其高效能，已是許多人的網路選擇。如同文獻[26]中所提，電子的多級式連接網路、與光學的多級式連接網路，其最大的區別是：在電子的多級式連接網路中，兩個訊息傳送之需求，當它們的傳送路徑的邊均不重覆時，可以同時傳送；而在光學的多級式連接網路中，兩個訊息傳送之需求，只有當它們的傳送路徑的點均不重覆時，才能同時傳送（這意味著這兩條傳送路徑不能同時通過同一個開關，也因此不會有開關干擾的問題產生）。這篇論文的目的便是針對光學的多級式連接網路來重做 Das 的工作。我們對於合併的 $2n-2$ 階級、以及 $2n-1$ 階級的光學多級式連接網路的無開關干擾之可重排性各提出一個充分條件，對於符合充分條件的光學多級式連接網路提出時間複雜度為 $O(Nn)$的排列繞送演算法。另外，我們也針對 baseline 網路提出在四回合之內、點均不重覆的排列繞送演算法。

關鍵詞：多級式連接網路，光學的多級式連接網路，可重排性，排列繞送，開關干擾，Benes 網路，baseline 網路，反向的 baseline 網路。

中 華 民 國 九 十 八 年 一 月

On the Crosstalk-free Rearrangeability of Combined Optical Multistage Interconnection Networks

Student: Chih-Wen Huang Advisor: Chiuyuan Chen

Department of Applied Mathematics

National Chiao Tung University

Abstract

Rearrangeability of a multistage interconnection network (MIN) is that if the MIN can connect its N inputs to its N outputs in all $N!$ possible ways, by rearranging the existing connections if required. In [8], Das formulated an elegant sufficient condition for the rearrangeability of a combined $(2n-1)$-stage MIN, where $n = \log_2 N$, and presented an $O(N \log_2 N)$-time routing algorithm for MINs that satisfy the sufficient condition. However, the above definition of rearrangeability and the results of Das are for electronic MINs. Recently, optical MINs have become a promising network choice for their high performance. As was mentioned in [28], the fundamental difference between an electronic MIN and an optical MIN is that: two routing requests in an electronic MIN can be sent simultaneously if they are link-disjoint, while two routing requests in an optical MIN can be sent simultaneously only when their routing paths are node-disjoint, meaning that these two paths do not pass through the same switching element and therefore there is no crosstalk problem. The purpose of this thesis is to redo the works of Das for optical MINs. In particular, we formulate a sufficient condition for the crosstalk-free rearrangeability of a combined $(2n-2)$-stage and a combined $(2n-1)$-stage optical MIN, we propose an $O(N \log_2 N)$-time routing algorithm for optical MINs that satisfy the sufficient condition. In this thesis we also propose an algorithm to realize any permutation in a baseline network with node-disjoint paths in four passes.

Keywords: Multistage interconnection network; Optical multistage interconnection network; Rearrangeability; Permutation routing; Crosstalk; Benes network; Baseline network; Reverse baseline network.

誌　謝

能夠完成此篇論文首先最要感謝的就是陳秋媛老師，陳老師自從我考上了交大研究所開始就經常撥冗與我討論研究方向及學業上的種種，亦時常教導我們學習該有的心態並且分享許多經驗，除了學業之外，陳老師更是時常關心我們生活方面是否有遭遇什麼問題，因此陳老師對我而言不只是老師同時也是很好的朋友，我真的很慶幸也很感謝這些日子來有陳老師的指導。未來，我定會把陳老師特別交代的兩句話牢記在心並努力實踐，這兩句話便是交大校訓，『知新致遠、崇實篤行』。

同時特別感謝交大應數所組合組的傅恆霖老師、黃大原老師、翁志文老師與符麥克老師的教導與鞭策。於研究期間，感謝學長國元、柏樹、鈺傑與威雄的照顧並且不時給予許多有用的意見，亦感謝信菖、子鴻、松育、世慶、宜君、慧萘以及許許多多的研究同學及學弟妹們提供的一些研究方面的建議及協助，除此之外，學業以外的許許多多照顧、歡笑及不定時的美食分享也都是我最寶貴的回憶，很高興能認識這些學長、同學與學弟妹們，希望你們未來都能夠一帆風順。

除了研究方面的協助，很感謝我的家人們的栽培，父母對我的全力支持與鼓勵，永遠是驅使我不斷向前的動力，我的學業能一路走到這篇論文完成，你們無疑是最大功臣，但願我的未來能不負你們期望。

最後，要感謝的人太多，以上未提及的朋友及同學們再此也一併感謝，謹以此論文獻給我最親愛的家人、師長及同學們！

Contents

List of Figures

1 Introduction

Permutation routing is an important transmission pattern in parallel and distributed computing systems [27]. The purpose of this thesis is to consider the problem of routing all $N!$ possible permutations in an optical multistage interconnection network (MIN).

Given N processors $P_0, P_1, \cdots, P_{N-1}$, an $N \times N$ MIN can be used for communication among these processors as shown in Figures 1 and 2, where $N \times N$ means N inputs and N outputs. In this thesis an MIN denotes both an electronic MIN and an optical MIN, and unless otherwise specified, an MIN means an $N \times N$ MIN. A column in an MIN is called a *stage*. The number of stages in an MIN is denoted by s. The nodes in a stage are called *switches* (or *switching elements* or *crossbars*). Define

$$n = \log_2 N.$$

Each switch is assumed to be of size 2×2 (thus N is even); see [4, 6, 7, 10, 15] for switches of other sizes. It is well known that a 2×2 switch has only two possible states: *straight* and *cross*, as shown in Figure 3.

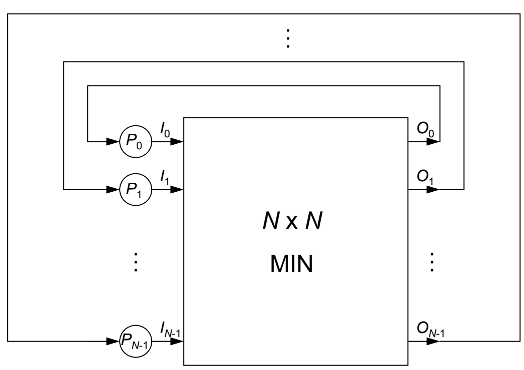

Figure 1: Communications among processors using an MIN.

A *permutation* of an MIN is one-to-one mapping between the inputs and outputs. A permutation is *admissible* of an MIN if it can be realized on that MIN with link-disjoint paths in one pass. An MIN is *rearrangeable* if all $N!$ possible permutations are admissible.

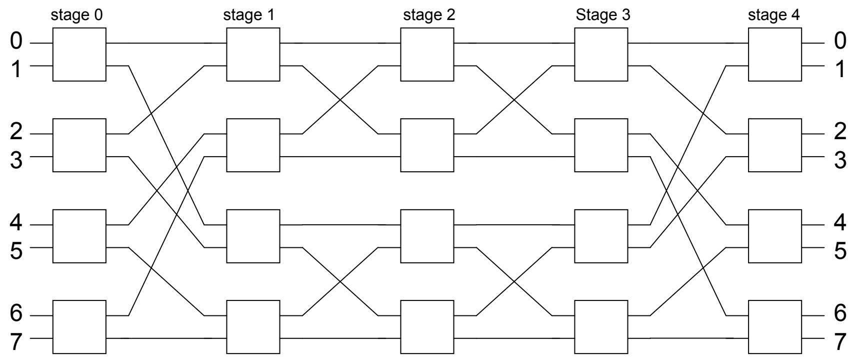

Figure 2: A 5-stage, 8×8 MIN; this MIN is a 8×8 Benes network.

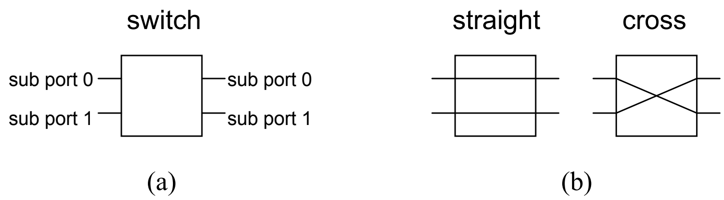

Figure 3: (a) A 2×2 switch and its sub ports. (b) The two possible states.

The *rearrangeability* of an MIN is that if the MIN can connect its N inputs to its N outputs in all $N!$ possible ways, by rearranging the existing connections if required.

The Benes network has been proposed as a popular architecture for rearrangeable MINs and it uses the theoretically minimum number of stages [2]. More precisely, a Benes network is a $(2n-1)$-stage MIN and it is essentially the concatenation of the baseline network and the reverse baseline network with the last stage of the baseline network overlapped with the first stage of the reverse baseline network. The shuffle-exchange network is also a widely studied architecture for rearrangeable MINs. In [23], Waksman proved that if a shuffle-exchange network is rearrangeable, then it has at least $2n-1$ stages. Later on, Stone [20] showed that an n^2-stage shuffle-exchange network is rearrangeable. In [24], Wu and Feng proposed an algorithm for realizing an arbitrary permutation on a $(3n-1)$-stage shuffle-exchange network. In [1] and [9], Babu et al. had proven that $3n-3$ is an upper bound on the number of stages for a shuffle-exchange network to be rearrangeable; this upper bound was later improved to be $3n-4$ by Linial and Tarsi [13]. In [5], Cam proved that a $(2n-1)$-stage shuffle-exchange network is rearrangeable, but the proof is still doubtful for $n > 4$; see [21].

Do notice that the above definition of rearrangeability and the results of [1, 2, 5, 9, 13, 20, 23, 24] are for electronic MINs. Recently, optical MINs have become a promising network choice for their high performance and faster speed; see [11, 16, 19, 22, 25, 26, 27, 28]. As was mentioned in [28], electronic MINs and optical MINs have many similarities, but there are some fundamental differences between them. The major difference between them is that the optical MINs have the *crosstalk* problem (see [28]) and therefore two messages can not pass through the same switch at the same time. As a result, two routing requests can be sent simultaneously in an electronic MIN if they are link-disjoint; they can be sent simultaneously in an optical MIN only when their routing paths are node-disjoint.

Due to the crosstalk problem, the results for electronic MINs may not be applied on optical MINs. Yang et al. [28] observed that the maximum number of input-output pairs that can be routed simultaneously in an optical MIN is $\frac{N}{2}$. Thus they introduced the definition of *semi-permutation*, which is a partial permutation with $\frac{N}{2}$ input-output pairs. A partial permutation $\begin{pmatrix} a_0 & a_1 & \cdots & a_{\frac{N}{2}-1} \\ b_0 & b_1 & \cdots & b_{\frac{N}{2}-1} \end{pmatrix}$ of an N-element set $\{0, 1, 2, \ldots, N-1\}$, where $a_i, b_i \in \{0, 1, 2, \ldots, N-1\}$, is a *semi-permutation* of the N-element set if

$$\left\{\left\lfloor \frac{a_0}{2} \right\rfloor, \left\lfloor \frac{a_1}{2} \right\rfloor, \ldots, \left\lfloor \frac{a_{\frac{N}{2}-1}}{2} \right\rfloor\right\} = \left\{\left\lfloor \frac{b_0}{2} \right\rfloor, \left\lfloor \frac{b_1}{2} \right\rfloor, \ldots, \left\lfloor \frac{b_{\frac{N}{2}-1}}{2} \right\rfloor\right\} = \left\{0, 1, \ldots, \frac{N}{2}-1\right\}.$$

For example, when $N = 8$, $\begin{pmatrix} 1 & 2 & 5 & 6 \\ 7 & 2 & 0 & 4 \end{pmatrix}$ is a semi-permutation since

$$\{\lfloor \tfrac{1}{2} \rfloor, \lfloor \tfrac{2}{2} \rfloor, \lfloor \tfrac{5}{2} \rfloor, \lfloor \tfrac{6}{2} \rfloor\} = \{\lfloor \tfrac{7}{2} \rfloor, \lfloor \tfrac{2}{2} \rfloor, \lfloor \tfrac{0}{2} \rfloor, \lfloor \tfrac{4}{2} \rfloor\} = \{0, 1, 2, 3\}.$$

A semi-permutation ensures that there is no crosstalk at the first and the last stages.

In [3], Bao and Li defined a routing to be *crosstalk-free* (*conflict-free*) if any two paths used in the routing are node-disjoint (link-disjoint). They showed that the crosstalk-free routing on any bit permutation network (BPN) is equivalent to the conflict-free routing on a BPN of smaller size and with fewer stages. They defined "CF-rearrangeable" and proved that the minimum number of stages for a BPN to be CF-rearrangeable is $2n-2$. In particular, an MIN is *crosstalk-free rearrangeable* (*CF-rearrangeable*) if every semi-permutations can be realized with node-disjoint paths in one pass. Since the maximum number of input-output pairs that can be routed simultaneously in an optical MIN is $\frac{N}{2}$, at least two passes are required for realizing a permutation. Yang et al. [28] proved an important result: *Any permutation can be decomposed into two semi-permutations.* Thus a CF-rearrangeable MIN can realize any permutation in two passes and this is optimal. Yang et al. [28] proved that a Benes network is CF-rearrangeable and proposed a permutation routing algorithm for a Benes network. Lu and Zheng [14] also proposed a permutation routing algorithm for the same network.

In [26], Yang and Wang proposed a permutation routing algorithm for the baseline (or reverse baseline) network with node-disjoint paths in four passes; they said that the proposed algorithm can work efficiently only for long message. Later, in [27], Yang and Wang presented a permutation routing algorithm for the baseline (or reverse baseline) network with node-disjoint paths in four passes and they claimed that this algorithm is suitable for messages of any length.

Recently, Das [8] formulated a sufficient condition for the rearrangeability of a *combined* $(2n-1)$-stage electronic MIN and presented an $O(N \log_2 N)$-time permutation routing algorithm for MINs that satisfy the sufficient condition. However, the results of Das are for electronic MINs. Therefore, due to the crosstalk problem, these results can not be applied on optical MINs. The purpose of this thesis is to transform the results of Das into results applicable to optical MINs. In particular, we formulate a sufficient condition for the CF-rearrangeability of a combined $(2n-2)$-stage optical MIN and a sufficient condition for the CF-rearrangeability of a combined $(2n-1)$-stage optical MIN. We propose an $O(N \log_2 N)$-time semi-permutation routing algorithm for optical MINs that satisfy the sufficient condition. We also improve the decomposition algorithm proposed in [28] and the permutation routing algorithm proposed in [27].

This thesis is organized as follows: Section 2 lists some preliminaries. Section 3 improves the decomposition algorithm proposed in [28]. Section 4 contains our results on CF-rearrangeability of optical MINs. Section 5 improves the permutation routing algorithm in [27]. Concluding remarks are given in the last section.

2 Preliminaries

An s-stage $N \times N$ MIN is represented as follows (see Figure 4 for an illustration):

- label the *inputs* of the MIN as $0, 1, 2, \ldots, N-1$ and represent each input by an n-bit binary number $x_{n-1}x_{n-2}\cdots x_0$;
- label the *outputs* of the MIN as $0, 1, 2, \ldots, N-1$ and represent each output by an n-bit binary number $y_{n-1}y_{n-2}\cdots y_0$;
- label the *stages* as $0, 1, 2, \ldots, s-1$;
- label the *switches* of each stage as $0, 1, 2, \ldots, \frac{N}{2}-1$ and represent each switch by an $(n-1)$-bit binary number $z_{n-2}z_{n-3}\cdots z_0$; and
- label the *upper* and the *lower output links* of every switch j, $0 \le j \le \frac{N}{2}-1$, as $2j$ and $2j+1$ respectively, and represent each link by an n-bit binary number $x_{n-1}x_{n-2}\cdots x_0$.

In an s-stage MIN, a path from an input to an output can be described by a sequence $r_0r_1\cdots r_{s-1}$ of labels that label the successive links on this path. Such a sequence of labels is called the *routing bits* [8] (or *control tag* [17], *tag* [6], or *path descriptor* [12]). Routing bits can be used as the header for routing a message: each successive switch uses the first routing bit to route the message, and then discards it. In particular, routing bit r_k controls the switch at stage k, and if $r_k = 0$ (respectively, $r_k = 1$), then a connection is

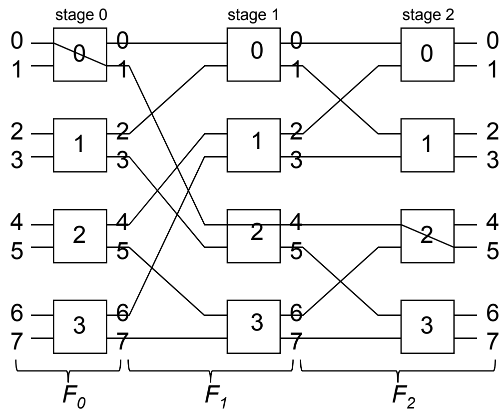

Figure 4: An 8×8 baseline network with labels and i-mappings.

made to sub port 0 (respectively, sub port 1). For example, in Figure 4, input 0 can get to output 5 by using routing bits 101, which means the routing request can be fulfilled by a path via sub port 1 at stage 0, sub port 0 at stage 1, and sub port 1 at stage 2.

In this thesis, the links connecting inputs and switches of stage 0 are regarded as *output links of switches of stage* (-1) although there are no switches of stage (-1). The following definition, *i-mapping*, is crucial for this thesis and was first proposed in [8]. Given an MIN, if for each stage i, $0 \leq i \leq s-1$, there exists a mapping

$$F_i: \; x_{n-1}x_{n-2}\cdots x_1x_0 \;\longrightarrow\; y_{n-1}y_{n-2}\cdots y_1r_i$$

between the output links $x_{n-1}x_{n-2}\cdots x_1x_0$ of switches of stage $(i-1)$ and the output links $y_{n-1}y_{n-2}\cdots y_1r_i$ of switches of stage i, where $y_{n-1}y_{n-2}\cdots y_1$ is a permutation of any $n-1$ bits of $x_{n-1}x_{n-2}\cdots x_1x_0$ and r_i is the routing bit, such that the link $x_{n-1}x_{n-2}\cdots x_1x_0$ at stage $(i-1)$ is connected to the link $y_{n-1}y_{n-2}\cdots y_1r_i$ at stage i, then F_i's are defined as *i-mappings* for the MIN. For example, the i-mappings for the 8×8 baseline network in Figure 4 are: $F_0 : x_2x_1x_0 \longrightarrow x_2x_1r_0,\; F_1 : x_2x_1x_0 \longrightarrow x_0x_2r_1,\; F_2 : x_2x_1x_0 \longrightarrow x_2x_0r_2$.

An s-stag MIN is said to *follow destination tag routing* if the routing bits $r_0r_1\cdots r_{s-1}$ of every message equal to the n-bit binary representation $y_{n-1}y_{n-2}\cdots y_0$ of the destination. In other words, if an MIN follows destination tag routing, then $s = n$ and the routing bits of a message sent to output $y_{n-1}y_{n-2}\cdots y_0$ are $y_{n-1}y_{n-2}\cdots y_0$; also, the destination of a message with routing bits $r_0r_1\cdots r_{s-1}$ is $r_0r_1\cdots r_{s-1}$. Consider the 8×8 baseline network shown in Figure 4. From its i-mappings, a message from input $x_2x_1x_0$ with routing bits $r_0r_1r_2$ will reach output $r_0r_1r_2$ since $x_2x_1x_0 \xrightarrow{F_0} x_2x_1r_0 \xrightarrow{F_1} r_0x_2r_1 \xrightarrow{F_2} r_0r_1r_2$. Thus the 8×8 baseline network follows destination tag routing. In general, an $N\times N$ baseline network satisfies $s=n$ and has i-mappings:

$$\begin{array}{llll}
F_0: & x_{n-1}x_{n-2}\cdots x_1x_0 & \longrightarrow & x_{n-1}x_{n-2}\cdots x_1r_0,\\
F_1: & x_{n-1}x_{n-2}\cdots x_1x_0 & \longrightarrow & x_0x_{n-1}x_{n-2}\cdots x_2r_1,\\
F_2: & x_{n-1}x_{n-2}\cdots x_1x_0 & \longrightarrow & x_{n-1}x_0x_{n-2}\cdots x_2r_2,\\
 & & \vdots & \\
F_{n-1}: & x_{n-1}x_{n-2}\cdots x_1x_0 & \longrightarrow & x_{n-1}x_{n-2}\cdots x_2x_0r_{n-1}.
\end{array}$$

A baseline network follows destination tag routing since

$$x_{n-1}x_{n-2}\cdots x_1x_0 \xrightarrow{F_0} x_{n-1}x_{n-2}\cdots x_1r_0 \xrightarrow{F_1} r_0x_{n-1}x_{n-2}\cdots x_2r_1 \xrightarrow{F_2} \cdots \xrightarrow{F_{n-1}} r_0r_1\cdots r_{n-1}.$$

We now define a combined MIN $M_1(n_1)\oplus M_2(n_2)$. Let $M_1(n_1)$ and $M_2(n_2)$ denote an n_1-stage MIN and an n_2-stage MIN, respectively. The combined MIN $M_1(n_1)\oplus M_2(n_2)$ is the concatenation of $M_1(n_1)$ and $M_2(n_2)$ with the last stage of $M_1(n_1)$ overlapped with the first stage of $M_2(n_2)$. Clearly, $M_1(n_1)\oplus M_2(n_2)$ is an (n_1+n_2-1)-stage MIN.

Recall that $n=\log_2 N$. A n-stage shuffle-exchange network is the well-known *omega network* and is usually denoted as Ω. In [8], Das formulated the following sufficient condition for the rearrangeability of a combined $(2n-1)$-stage MIN $\Delta\oplus\Delta'$, where Δ and Δ' are two n-stage Ω-equivalent networks (notice that an Ω-equivalent network follows destination tag routing).

Theorem 1. [8] *In a combined $(2n-1)$-stage MIN $\Delta\oplus\Delta'$, if i-mappings exist for all i, $0\le i\le 2n-2$, and each AR-bit r_j, $0\le j\le n-2$, occurs only in each S_k, $j+1\le k\le 2n-2-j$, then $\Delta\oplus\Delta'$ is rearrangeable.*

For the definitions of AR-bit and S_k, see Section 4. The Benes network is an example of networks that satisfy this sufficient condition.

3 Decompose a permutation into semi-permutations

The purpose of this section is to improve the decomposition algorithm proposed in [28]. Throughout this section, $P = \begin{pmatrix} a_0 & a_1 & \cdots & a_{N-1} \\ b_0 & b_1 & \cdots & b_{N-1} \end{pmatrix}$ denotes a given permutation. In [28], Yang et al. proposed an efficient algorithm to decompose a given permutation P into two semi-permutations L and R. This algorithm first constructs an undirected bipartite graph $G = (V_1, V_2; E)$ for P. The vertex sets of G are given by $V_1 = \{A_0^{[1]}, A_1^{[1]}, \ldots, A_{\frac{N}{2}-1}^{[1]}\}$ and $V_2 = \{A_0^{[2]}, A_1^{[2]}, \ldots, A_{\frac{N}{2}-1}^{[2]}\}$, where $A_j^{[1]}$ and $A_j^{[2]}$ correspond to inputs and outputs, respectively, and both $A_j^{[1]}$ and $A_j^{[2]}$ are the 2-element set $\{2j, 2j+1\}$ for all j, $0 \leq j \leq \frac{N}{2}-1$. The edge set E is defined by: $(A_{j_1}^{[1]}, A_{j_2}^{[2]}) \in E$ if and only if there exists a pair $\begin{pmatrix} a_i \\ b_i \end{pmatrix}$ in P such that $a_i \in A_{j_1}^{[1]}$ and $b_i \in A_{j_2}^{[2]}$. Clearly, G is 2-regular, $|V_1| = |V_2| = \frac{N}{2}$, and $|E| = N$. The algorithm in [28] takes $O(N)$ time and is listed in Algorithm 1.

Algorithm 1 DECOMPOSITION ALGORITHM in [28]

Require: A permutation P.

Ensure: Two semi-permutations L and R of P.

1: Construct a bipartite graph $G = (V_1, V_2; E)$ for P.

2: For each connected component of G, start from a vertex of this component in V_1, traverse through an unvisited edge to the neighbor vertex in V_2, back and forth until returning to the starting vertex. (During the traverse, a visited edge is marked "forward" if the traverse direction on this edge is from V_1 to V_2 and marked "backward" if the direction is opposite.)

3: Take all one-pair mappings corresponding to edges marked with "forward" to form semi-permutation L; take all one-pair mappings corresponding to edges marked with "backward" to form semi-permutation R.

In Algorithm 1, a bipartite graph has to be constructed explicitly. We now propose a decomposition algorithm, which abandon the requirement for constructing a bipartite graph and still takes $O(N)$ time. Without loss of generality, in our algorithm, assume the given permutation P is of the form $P = \begin{pmatrix} 0 & 1 & \cdots & N-1 \\ b_0 & b_1 & \cdots & b_{N-1} \end{pmatrix}$ and is represented as an array also called P with $P[i] = b_i$. An array Q is used to store the inverse permutation of P; that is, $Q[b_i] = i$ if and only if $P[i] = b_i$. Semi-permutation L is represented by two arrays L_a and L_b such that $L_a[\lfloor \frac{i}{2} \rfloor] = i$ and $L_b[\lfloor \frac{i}{2} \rfloor] = b_i$ if and only if $\begin{pmatrix} i \\ b_i \end{pmatrix} \in L$; R is represented by two arrays R_a and R_b such that $R_a[\lfloor \frac{i}{2} \rfloor] = i$ and $R_b[\lfloor \frac{i}{2} \rfloor] = b_i$ if and only if $\begin{pmatrix} i \\ b_i \end{pmatrix} \in R$. Define $mate(v) = v + 1$ if v is an even number and $v - 1$ if v is an odd number. Thus 0 and 1 are the mates of each other, 2 and 3 are the mates of each other, and so on. The following is our decomposition algorithm.

Algorithm 2 OUR DECOMPOSITION ALGORITHM

Require: A permutation P.
Ensure: Two semi-permutations L and R of P.

for $i \leftarrow 0$ to $N-1$ **do**
  $A[i] \leftarrow 0;$
end for
for $i \leftarrow 0$ to $N-1$ **do**
  $Q[P[i]] \leftarrow i;$
end for
while there exists i such that $A[i] = 0$ **do**
  $head \leftarrow i;$
  $next \leftarrow i;$
  repeat
    $L_a[\lfloor \frac{next}{2} \rfloor] \leftarrow next;$
    $L_b[\lfloor \frac{next}{2} \rfloor] \leftarrow P[next];$
    $A[next] \leftarrow 1;$
    $next \leftarrow Q[mate(P[next])];$
    $R_a[\lfloor \frac{next}{2} \rfloor] \leftarrow next;$
    $R_b[\lfloor \frac{next}{2} \rfloor] \leftarrow P[next];$
    $A[next] \leftarrow 1;$
    $next \leftarrow mate(next);$
  until $(next = head);$
end while

We now give an example for Algorithm 2. Suppose $P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 7 & 2 & 6 & 3 & 0 & 4 & 1 \end{pmatrix}$.
In the 1st iteration, $L = \begin{pmatrix} 0 & ? & ? & ? \\ 5 & ? & ? & ? \end{pmatrix}$ and $R = \begin{pmatrix} ? & ? & ? & 6 \\ ? & ? & ? & 4 \end{pmatrix}$.
In the 2nd iteration, $L = \begin{pmatrix} 0 & ? & ? & 7 \\ 5 & ? & ? & 1 \end{pmatrix}$ and $R = \begin{pmatrix} ? & ? & 5 & 6 \\ ? & ? & 0 & 4 \end{pmatrix}$.
In the 3rd iteration, $L = \begin{pmatrix} 0 & ? & 4 & 7 \\ 5 & ? & 3 & 1 \end{pmatrix}$ and $R = \begin{pmatrix} ? & 2 & 5 & 6 \\ ? & 2 & 0 & 4 \end{pmatrix}$.
In the final iteration, we obtain the two semi-permutations

$$L = \begin{pmatrix} 0 & 3 & 4 & 7 \\ 5 & 6 & 3 & 1 \end{pmatrix} \text{ and } R = \begin{pmatrix} 1 & 2 & 5 & 6 \\ 7 & 2 & 0 & 4 \end{pmatrix}.$$

We now analyze Algorithm 2.

Theorem 2. *Algorithm 2 is correct and takes $O(N)$ time.*

Proof. Let G be the bipartite graph in the algorithm in [28]. Our algorithm is based on the observation that: for all v, $0 \leq v < N$, $\{v, mate(v)\}$ is the vertex $A_i^{[1]}$ and also the vertex $A_i^{[2]}$ in G, $A_i^{[1]}$ is adjacent to $A_j^{[2]}$, where $i = \lfloor \frac{v}{2} \rfloor$ and $j = \left\lfloor \frac{P[v]}{2} \right\rfloor$. Since G is a 2-regular bipartite graph, each connected component of G is an even cycle. The repeat-loop in lines 10 to 19 corresponds to traversing a cycle in G and inserting edges of the cycle into L and R alternatingly. The while-loop ensures that all the connected components of G are traversed. Hence the resultant L and R are two semi-permutations. Lines 1 to 6 of this algorithm take $O(N)$ time. Lines 7 to 20 take $O(N)$ time since each input-output pair in P is considered exactly once. Thus Algorithm 2 takes $O(N)$ time. ∎

4 CF-rearrangeability of optical MINs

The purpose of this section is to formulate a sufficient condition for the crosstalk-free rearrangeability of a combined $(2n-2)$-stage and a combined $(2n-1)$-stage optical MIN and to propose a routing algorithm for MINs that satisfy the sufficient condition. Before going further, we give three definitions: AR-bits, OW_k, and OS_k.

Recall that in this thesis, an MIN is an $N \times N$ MIN and $n = \log_2 N$. Let $M_1(n') \oplus M_2(n)$ be a combined optical MIN in which $M_2(n)$ follows destination tag routing and set $s = n' + n - 1$ for easy writing. A path from an input to an output through the MIN is referred to as an input-output path. Since $M_2(n)$ follows destination tag routing, for a particular input-output path, the routing bits for stages k, $s - n \leq k \leq s - 1$, are predetermined by the n-bit binary representation of the destination (i.e., the output), but the routing bits r_k, $0 \leq k \leq s - n - 1$, can be arbitrary and are referred to as *arbitrary routing bits* (*AR-bits*).

Suppose the i-mappings of $M_1(n') \oplus M_2(n)$ are $F_0, F_1, \ldots, F_{s-1}$. Then an input-output path from input x to output y can be represented as $L_0 \to L_1 \to L_2 \to \cdots \to L_{s-1} \to L_s$, where $L_0 = x$, $L_s = y$, and L_k, $1 \leq k \leq s-1$, is the output of stage $k-1$ followed by the path. Note that $L_k = F_{k-1}(L_{k-1})$ for all k, $1 \leq k \leq s$. The path can also be represented as $E_0 \rightarrow E_1 \rightarrow \cdots \rightarrow E_{s-1}$, where E_k, $0 \leq k \leq s-1$, is the switch passed by the path at stage k. It is not difficult to see that the binary representation of E_k, $0 \leq k \leq s-1$, can be obtained by deleting the rightmost bit of the binary representation of L_{k+1}.

Given a semi-permutation on $M_1(n') \oplus M_2(n)$, at any stage k, $0 \leq k \leq s-1$, the set of switches passed by individual input-output paths can be represented by an $\frac{N}{2} \times (n-1)$ matrix, called *optical window* OW_k, where each row j, $0 \leq j \leq \frac{N}{2}-1$, of OW_k is the $(n-1)$-bit binary representation of the switch at stage k that is passed by the path started from input $2j$ (if $2j$ belongs to the semi-permutation) or $2j+1$ (if $2j+1$ belongs to the semi-permutation). Note that each optical window OW_k can be represented uniquely by a string OS_k obtained by deleting the rightmost bit of S_{k+1}, where $S_0 = x_{n-1}x_{n-2}\cdots x_0$ and $S_k = F_{k-1}(S_{k-1})$, $1 \leq k \leq s$. OS_k is called the *characteristic string* of OW_k.

A dilated Benes network is a $(2n-2)$-stage MIN and it is the concatenation of the baseline network and the reverse baseline network with the last two stages of the baseline network overlapped with the first two stages of the reverse baseline network.

Take the 6-stage 16×16 dilated Benes network shown in Figure 5 as an example. Suppose the binary representations of the input and the output are $x_3x_2x_1x_0$ and $y_3y_2y_1y_0$, respectively. Then $r_2r_3r_4r_5 = y_3y_2y_1y_0$; the i-mappings, S_k's, and OS_k's are:

$$
\begin{array}{lll}
F_0 : x_3x_2x_1x_0 \longrightarrow x_3x_2x_1r_0 & S_0 = x_3x_2x_1x_0 & \\
F_1 : x_3x_2x_1x_0 \longrightarrow x_0x_3x_2r_1 & S_1 = F_0(S_0) = x_3x_2x_1r_0 & OS_0 = x_3x_2x_1 \\
F_2 : x_3x_2x_1x_0 \longrightarrow x_3x_0x_2r_2 & S_2 = F_1(S_1) = r_0x_3x_2r_1 & OS_1 = r_0x_3x_2 \\
F_3 : x_3x_2x_1x_0 \longrightarrow x_3x_2x_0r_3 & S_3 = F_2(S_2) = r_0r_1x_3r_2 & OS_2 = r_0r_1x_3 \\
F_4 : x_3x_2x_1x_0 \longrightarrow x_3x_1x_0r_4 & S_4 = F_3(S_3) = r_0r_1r_2r_3 & OS_3 = r_0r_1r_2 \\
F_5 : x_3x_2x_1x_0 \longrightarrow x_2x_1x_0r_5 & S_5 = F_4(S_4) = r_0r_2r_3r_4 & OS_4 = r_0r_2r_3 \\
 & S_6 = F_5(S_5) = r_2r_3r_4r_5 & OS_5 = r_2r_3r_4
\end{array}
$$

The following observation is crucial for the remaining discussions: *A semi-permutation can be realized on $M_1(n') \oplus M_2(n)$ if and only if all rows of each optical window OW_k, $0 \leq k \leq s-1$, are distinct.* We now are ready to propose our sufficient condition.

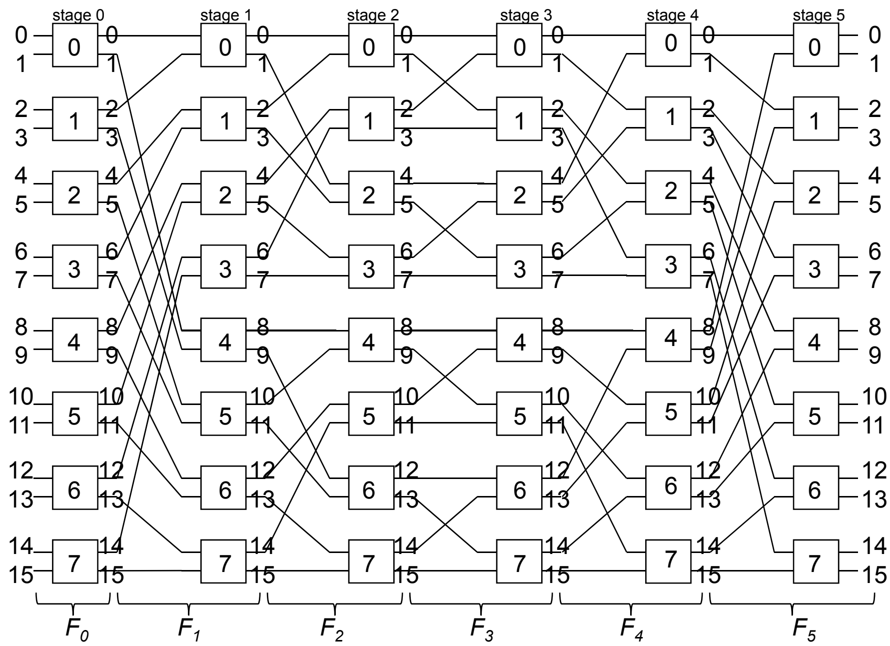

Figure 5: A 16×16 dilated Benes network.

Theorem 3. (Sufficient condition for a combined $(2n-2)$-stage optical MIN) *In a combined $(2n-2)$-stage optical MIN $M_1(n-1) \oplus M_2(n)$ in which $M_2(n)$ follows destination tag routing, if i-mappings exist for all i, $0 \le i \le 2n-3$, and each AR-bit r_k, $0 \le k \le n-3$, occurs only in each OS_ℓ, for $k+1 \le \ell \le 2n-4-k$, then the network is CF-rearrangeable.*

Proof. To prove this theorem, it suffices to prove that $M_1(n-1) \oplus M_2(n)$ can realize each semi-permutation $\mathcal{P}$ with node-disjoint paths in one pass. In this proof, j is an integer in $\{0, 1, \ldots, \frac{N}{2}-1\}$. By the definition of a semi-permutation, exactly one of inputs $2j$ and $2j+1$ is in $\mathcal{P}$; denote the one in $\mathcal{P}$ as j^*. Let $r_0(j)r_1(j)\cdots r_{2n-3}(j)$ be the routing bits of j^*; in particular, $r_k(j)$ is the routing bit of input j^* at stage k.

Before going further, we define *conjugate rows*. By the constraints of this theorem, AR-bit r_k, $0 \le k \le n-3$, appears only in $OS_{k+1}, OS_{k+2}, \ldots, OS_{2n-4-k}$. Thus r_0 appears only in $OS_1, OS_2, \ldots, OS_{2n-4}$; r_1 appears only in $OS_2, OS_3, \ldots, OS_{2n-5}$; r_2 appears only in $OS_3, OS_4, \ldots, OS_{2n-6}$; and so on. Therefore, we have two properties.

(i) For all k, $1 \le k \le n-2$, $n-2$ columns of OW_k appear in OW_{k-1} and the remaining one column is composed of $r_{k-1}(0),\ r_{k-1}(1),\ \ldots,\ r_{k-1}(\frac{N}{2}-1)$.

(ii) For all k, $n-1 \leq k \leq 2n-4$, $n-2$ columns of OW_k appear in OW_{k+1} and the remaining one column is composed of $r_{2n-4-k}(0),\ r_{2n-4-k}(1),\ \ldots,\ r_{2n-4-k}(\frac{N}{2}-1)$.

Based on the above two properties, we define *conjugate rows* as follows.

(i) For all k, $1 \leq k \leq n-2$, two rows j and j' of OW_k are the *conjugate row* of each other if these two rows are identical except at $r_{k-1}(j)$ and $r_{k-1}(j')$.

(ii) For all k, $n-1 \leq k \leq 2n-4$, two rows j and j' of OW_k are the *conjugate row* of each other if these two rows are identical except at $r_{2n-4-k}(j)$ and $r_{2n-4-k}(j')$.

Since $M_2(n)$ follows destination tag routing, $r_{n-2}(j)r_{n-1}(j)\cdots r_{2n-3}(j)$ are predetermined by the binary representation of the destination. Hence, to prove this theorem, it suffices to prove that for each j, routing bits $r_0(j)r_1(j)\cdots r_{n-3}(j)$ (i.e., AR-bits) exist such that all rows of each optical window OW_k, $0 \leq k \leq 2n-3$, are distinct.

First consider OW_0 and OW_{2n-3}. Since OS_0 is composed of the leftmost $n-1$ bits of the inputs in $\mathcal{P}$, all rows of OW_0 are distinct. Since OS_{2n-3} is composed of the leftmost $n-1$ bits of the outputs in $\mathcal{P}$, all rows of OW_{2n-3} are distinct. In the following, we will show that it is possible to assign the values of $r_0(0),\ r_0(1),\ \ldots,\ r_0(\frac{N}{2}-1)$ so that all rows of OW_1 are distinct and all rows of OW_{2n-4} are distinct, too. After $r_0(0),\ r_0(1),\ \ldots,\ r_0(\frac{N}{2}-1)$ are assigned, we will show that it is possible to assign the values of $r_1(0),\ r_1(1),\ \ldots,\ r_1(\frac{N}{2}-1)$ so that all rows of OW_2 are distinct and all rows of OW_{2n-5} are distinct, too. In general, after $r_{k-1}(0),\ r_{k-1}(1),\ \ldots,\ r_{k-1}(\frac{N}{2}-1)$ are assigned, we will show that it is possible to assign the values of $r_k(0),\ r_k(1),\ \ldots,\ r_k(\frac{N}{2}-1)$ so that all rows of OW_{k+1} are distinct and all rows of OW_{2n-4-k} are distinct, too.

Consider the pair of optical windows OW_{k+1} and OW_{2n-4-k} in the order $k = 0, 1, \ldots, n-3$. We now show that it is possible to use the idea of conjugate rows to assign the values of $r_k(0),\ r_k(1),\ \ldots,\ r_k(\frac{N}{2}-1)$ so that all rows of OW_{k+1} are distinct and all rows of

OW_{2n-4-k} are distinct, too. Among the $n-1$ columns of OW_{k+1}, $n-2$ of them appear in OW_k and the remaining one column is composed of $r_k(0),\ r_k(1),\ \ldots,\ r_k(\frac{N}{2}-1)$; also, among the $n-1$ columns of OW_{2n-4-k}, $n-2$ of them appear in OW_{2n-3-k} and the remaining one column is composed of $r_k(0),\ r_k(1),\ \ldots,\ r_k(\frac{N}{2}-1)$. Each row j of the $n-2$ columns of OW_{k+1} that appear in OW_k has a conjugate row j'. Hence all rows of OW_{k+1} are distinct if and only if for each pair of conjugate rows j and j', $r_k(j) \neq r_k(j')$. Similarly, each row j of the $n-2$ columns of OW_{2n-4-k} that appear in OW_{2n-3-k} has a conjugate row j'. Hence all rows of OW_{2n-4-k} are distinct if and only if for each pair of conjugate rows j and j', $r_k(j) \neq r_k(j')$. We start with an arbitrary row j of OW_{k+1} and set $r_k(j) = 0$. Then we find the conjugate row j' of row j in OW_{k+1} and set $r_k(j') = 1$. In this way, rows j and j' in OW_{k+1} can be made distinct. Next, find the conjugate row j'' of row j' in OW_{2n-4-k} and set $r_k(j'') = 0$. Again, in this way, rows j' and j'' in OW_{2n-4-k} can be made distinct. Repeat the above process until $r_k(j)$ is assigned for all j. Thus it is possible to assign the values of $r_k(0),\ r_k(1),\ \ldots,\ r_k(\frac{N}{2}-1)$ so that all rows of OW_{k+1} are distinct and all rows of OW_{2n-4-k} are distinct, too. We now have this theorem. ∎

By using the idea of conjugate rows (defined in the proof of Theorem 3), we now propose an algorithm to determine the AR-bits $r_0 r_1 \cdots r_{n-3}$ for optical MINs that satisfy Theorem 3.

Since AR-bits $r_0 r_1 \cdots r_{n-3}$ together with $r_{n-2} r_{n-1} \cdots r_{2n-3}$ (the predetermined routing bits) can be used to route a given semi-permutation with node-disjoint paths in one pass, Algorithm 3 is called the ROUTING ALGORITHM.

Take the 16×16 dilated Benes network shown in Figure 5 and the semi-permutation $\mathcal{P} = \begin{pmatrix} 0 & 2 & 5 & 7 & 8 & 11 & 13 & 15 \\ 13 & 11 & 2 & 0 & 9 & 14 & 5 & 7 \end{pmatrix}$ as an example of Algorithm 3. The first and the

Algorithm 3 ROUTING ALGORITHM

Require: An arbitrary semi-permutation $\mathcal{P}$, the characteristic strings OS_k, $1 \le k \le 2n-4$, of the combined $(2n-2)$-stage optical MIN, and the optical windows OW_0 and OW_{2n-3} derived from $\mathcal{P}$.

Ensure: AR-bits $r_0 r_1 \cdots r_{n-3}$, each represented as an $(\frac{N}{2})$-bit array such that $r_k(j)$, $0 \le j \le \frac{N}{2}-1$, represents the routing bit of input j^* at stage k, where $j^* = 2j$ if input $2j$ is in $\mathcal{P}$ and $j^* = 2j+1$ if input $2j+1$ is in $\mathcal{P}$.

1: **for** $k = 0$ **to** $n-3$ **do**
2: use OS_{k+1} and W_k to form $(n-2)$ of the $(n-1)$ columns of OW_{k+1};
/* the remaining one column is for AR-bit r_k and is determined below */
3: use OS_{2n-4-k} and W_{2n-3-k} to form $(n-2)$ of the $(n-1)$ columns of OW_{2n-4-k};
/* the remaining one column is for AR-bit r_k and is determined below */
4: $j \leftarrow 0$;
5: $r_k(j) \leftarrow 0$;
6: set the corresponding entry in the remaining one column of OW_{k+1} to 0;
7: find the conjugate row j' in OW_{k+1};
8: $r_k(j') \leftarrow 1$;
9: set the corresponding entry in the remaining one column of OW_{k+1} to 1;
10: find the conjugate row j'' in OW_{2n-4-k};
11: $r_k(j'') \leftarrow 0$;
12: set the corresponding entry in the remaining one column of OW_{2n-4-k} to 0;
13: **repeat** lines 7 to 12 **until** $j'' = j$;
14: **if** there exists a row j in OW_{k+1} such that $r_k(j)$ is not assigned **then** go to line 5;
15: **end for**

last optical windows OW_0 and OW_5 are as follows.

$$\left(\begin{array}{cccccc}
\leftarrow & OW_0 & \rightarrow & \leftarrow & OW_5 & \rightarrow \\
x_3 & x_2 & x_1 & y_3\,(=r_2) & y_2\,(=r_3) & y_1\,(=r_4) \\
0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 1
\end{array}\right)$$

Algorithm 3 determines r_0 from OW_1 and OW_4 as follows. Note that for convenience, the columns in OW_1 is given in the order $x_3 x_2 r_0$ instead of the order $r_0 x_3 x_2$.

$$\left(\begin{array}{ccccc}
\leftarrow & OW_1 & \rightarrow & & \\
 & & \leftarrow & OW_4 & \rightarrow \\
x_3 & x_2 & r_0 & y_3\,(=r_2) & y_2\,(=r_3) \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & & 1 & 0 \\
0 & 1 & & 0 & 0 \\
0 & 1 & & 0 & 0 \\
1 & 0 & & 1 & 0 \\
1 & 0 & & 1 & 1 \\
1 & 1 & & 0 & 1 \\
1 & 1 & & 0 & 1
\end{array}\right)$$

$$
\begin{pmatrix}
\leftarrow & OW_1 & \rightarrow & & \\
 & & \leftarrow & OW_4 & \rightarrow \\
x_3 & x_2 & r_0 & y_3\,(=r_2) & y_2\,(=r_3) \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & & 0 & 0 \\
0 & 1 & & 0 & 0 \\
1 & 0 & & 1 & 0 \\
1 & 0 & & 1 & 1 \\
1 & 1 & & 0 & 1 \\
1 & 1 & & 0 & 1
\end{pmatrix}
$$

$$
\begin{pmatrix}
\leftarrow & OW_1 & \rightarrow & & \\
 & & \leftarrow & OW_4 & \rightarrow \\
x_3 & x_2 & r_0 & y_3\,(=r_2) & y_2\,(=r_3) \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & & 0 & 0 \\
0 & 1 & & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & & 1 & 1 \\
1 & 1 & & 0 & 1 \\
1 & 1 & & 0 & 1
\end{pmatrix}
$$

$$
\begin{pmatrix}
\leftarrow & OW_1 & \rightarrow & & \\
 & & \leftarrow & OW_4 & \rightarrow \\
x_3 & x_2 & r_0 & y_3\,(=r_2) & y_2\,(=r_3) \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 1
\end{pmatrix}
$$

Algorithm 3 determines r_1 from OW_2 and OW_3 as follows. Again, for convenience, the columns in OW_2 is given in the order $x_3r_0r_1$ instead of the order $r_0r_1x_3$.

$$
\begin{pmatrix}
\leftarrow & OW_2 & \rightarrow & \\
 & \leftarrow & OW_3 & \rightarrow \\
x_3 & r_0 & r_1 & y_3\,(=r_2) \\
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0
\end{pmatrix}
$$

The routing bits $r_0r_1r_2r_3r_4r_5$ for $\mathcal{P}$ is listed below, in which r_0 and r_1 are the AR-bits derived by Algorithm 3, and each of r_2, r_3, r_4, r_5 is represented as an $(\frac{N}{2})$-bit array such that $r_2(j)r_3(j)r_4(j)r_5(j)$, $0 \leq j \leq \frac{N}{2} - 1$, is the binary representation of the output of input j^* in $\mathcal{P}$, where $j^* = 2j$ if input $2j$ is in $\mathcal{P}$ and $j^* = 2j+1$ if input $2j+1$ is in $\mathcal{P}$.

$$\begin{pmatrix} r_0 & r_1 & r_2 & r_3 & r_4 & r_5 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Figure 6 shows the routing paths of $\mathcal{P}$ when the above $r_0r_1r_2r_3r_4r_5$ is used.

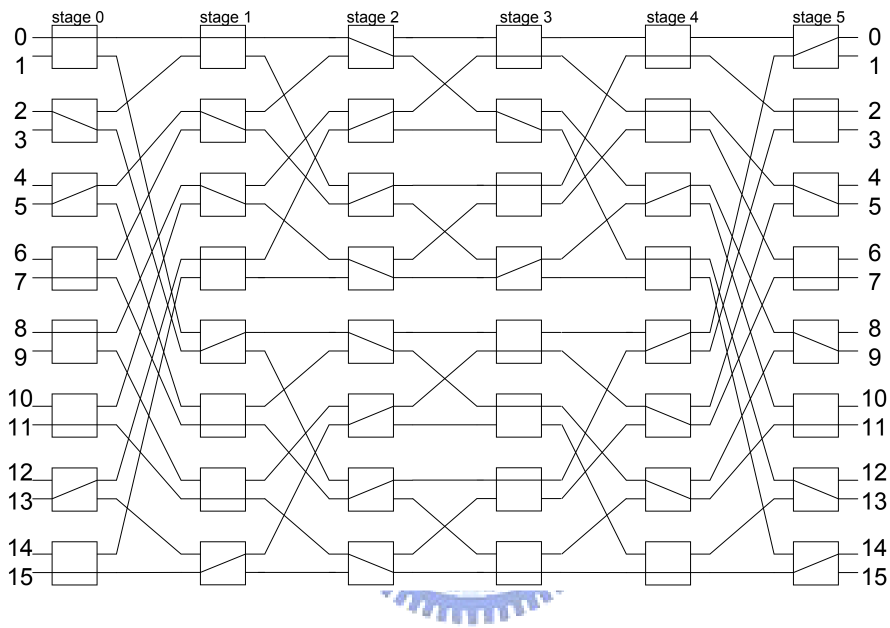

Figure 6: The routing paths obtained by our algorithm.

We now prove a theorem.

Theorem 4. *Algorithm 3 is correct and takes $O(N \log_2 N)$ time. Moreover, it leads to an $O(N \log_2 N)$-time semi-permutation routing algorithm.*

Proof. The correctness of Algorithm 3 follows from the proof of Theorem 3. Since there are $(n-2) \times \frac{N}{2}$ $r_k(j)$'s and each of them can be determined in $O(1)$ time, Algorithm 3 takes $O((\log_2 N - 2) \times \frac{N}{2}) = O(N \log_2 N)$ time. It is not difficult to see that the AR-bits $r_0r_1 \cdots r_{n-3}$ obtained by Algorithm 3 together with the n predetermined routing bits $r_{n-2}r_{n-1} \cdots r_{2n-3}$ can be used to route a given semi-permutation with node-disjoint paths in one pass in $O(N \log_2 N)$ time. ∎

The following is a sufficient condition for the CF-rearrangeability of a combined $(2n-1)$-stage optical MIN. Since the proof is similar to that of Theorem 3, the proof is omitted.

Theorem 5. (Sufficient condition for a combined $(2n-1)$-stage optical MIN) *In a combined $(2n-1)$-stage optical MIN $M_1(n) \oplus M_2(n)$ in which $M_2(n)$ follows destination tag routing, if i-mappings exist for all i, $0 \leq i \leq 2n-2$, and each AR-bit r_k, $0 \leq k \leq n-2$, occurs only in each OS_ℓ, for $k+1 \leq \ell \leq 2n-3-k$, then the network is CF-rearrangeable.*

Before ending this section, we list the characteristic strings of a $(2n-2)$-stage dilated Benes network. These strings will be used in the next section.

$$\begin{array}{ll} OS_0 & = x_{n-1}x_{n-2}\cdots x_3x_2x_1 \\ OS_1 & = r_0x_{n-1}x_{n-2}\cdots x_3x_2 \\ OS_2 & = r_0r_1x_{n-1}x_{n-2}\cdots x_3 \\ & \vdots \\ OS_{n-2} & = r_0r_1r_2\cdots r_{n-3}x_{n-1} \\ OS_{n-1} & = r_0r_1r_2\cdots r_{n-3}r_{n-2} \\ & \vdots \\ OS_{2n-5} & = r_0r_1r_{n-2}r_{n-1}\cdots r_{2n-6} \\ OS_{2n-4} & = r_0r_{n-2}r_{n-1}\cdots r_{2n-6}r_{2n-5} \\ OS_{2n-3} & = r_{n-2}r_{n-1}\cdots r_{2n-6}r_{2n-5}r_{2n-4} \end{array} \quad (1)$$

Here $x_{n-1}x_{n-2}\cdots x_0$ denotes an input and $y_{n-1}y_{n-2}\cdots y_0$ denotes an output. Note that $r_{n-2}r_{n-1}\cdots r_{2n-3} = y_{n-1}y_{n-2}\cdots y_0$. It can be verified that a dilated Benes network satisfies the sufficient condition stated in Theorem 3 and hence is CF-rearrangeable.

5 A permutation routing algorithm in the baseline (or reverse baseline) network

Recall that both the Benes network and the dilated Benes network are the concatenation of the baseline network and the reverse baseline network. The Benes network is rearrangeable, whereas the dilated Benes network is CF-rearrangeable. For convenience, call the output links of switches of stage $(n-1)$ followed by routing paths in a permutation (respectively, semi-permutation) of the Benes network (respectively, dilated Benes network) the *intermediate destinations*. In [27], by using the intermediate destinations of a Benes network, Yang and Wang proposed an algorithm (for convenience, call it Algorithm YW) to route an arbitrary permutation in a baseline (or reverse baseline) network with node-disjoint paths in four passes.

Algorithm YW uses Algorithm 1 to decompose a given permutation P into two semi-permutations L and R. Recall that Algorithm 1 has to construct a bipartite graph explicitly. Also, to use the intermediate destinations of a Benes network, Algorithm YW has to run Algorithm 1 to decompose each of L and R into two semi-permutations, say, LL, LR, RL, and RR. Then, Algorithm YW has to run Algorithm 1 to further decompose each of LL, LR, RL, and RR into two semi-permutations, say, LLL, LLR, LRL, LRR, RLL, RLR, RRL, and RRR. The same process repeats until each semi-permutation contains only one input-output pair.

The purpose of this section is to improve Algorithm YW. To achieve this purpose, Algorithm 2 is used instead of Algorithm 1; also, the intermediate destinations of a dilated Benes network are used instead of the intermediate destinations of a Benes network. See the following for details.

We first use Algorithm 2 to decompose a given permutation P into two semi-permutations L and R; then, route L in a baseline network with node-disjoint paths in two passes, and route R in a baseline network with node-disjoint paths in two passes. In the following, we only present an algorithm to route an arbitrary semi-permutation $\mathcal{P}$ in a baseline network with node-disjoint paths in two passes. Obviously, setting $\mathcal{P} = L$ and $\mathcal{P} = R$ will route an arbitrary permutation in a baseline network with node-disjoint paths in four passes.

Suppose the semi-permutation is $\mathcal{P} = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{\frac{N}{2}-1} \\ b_0 & b_1 & b_2 & \cdots & b_{\frac{N}{2}-1} \end{pmatrix}$. Note that we have assumed that the links connecting inputs and switches of stage 0 are regarded as output links of switches of stage (-1). Therefore, an input-output path from input a_i to output b_i, $0 \le i \le \frac{N}{2} - 1$, in a dilated Benes network can be represented as

$L_{i,0} \to L_{i,1} \to L_{i,2} \to \cdots \to L_{i,2n-3} \to L_{i,2n-2}$, where $L_{i,0} = a_i$, $L_{i,2n-2} = b_i$, and $L_{i,k}$, $1 \le k \le 2n-3$, is the output link of switches of stage $k-1$ followed by the path. Our algorithm is based on the observation that $L_{0,n}, L_{1,n}, \ldots, L_{\frac{N}{2}-1,n}$ are the intermediate destinations of a dilated Benes network for $\mathcal{P}$ and can be obtained by Algorithm 3. Let $\mathcal{P}_1 = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{\frac{N}{2}-1} \\ L_{0,n} & L_{1,n} & L_{2,n} & \cdots & L_{\frac{N}{2}-1,n} \end{pmatrix}$ and $\mathcal{P}_2 = \begin{pmatrix} L_{0,n} & L_{1,n} & L_{2,n} & \cdots & L_{\frac{N}{2}-1,n} \\ b_0 & b_1 & b_2 & \cdots & b_{\frac{N}{2}-1} \end{pmatrix}$.

The following lemma was proven in [27].

Lemma 6. [27] *The set of all semi-permutations realized by a baseline network with node-disjoint paths in one pass is exactly the set of all semi-permutations realized by a reverse baseline network with node-disjoint paths in one pass.*

We now prove a lemma.

Lemma 7. *Both $\mathcal{P}_1$ and $\mathcal{P}_2$ can be realized by a baseline (or reverse baseline) network with node-disjoint paths in one pass. Moreover, routing bits for $\mathcal{P}_1$ and $\mathcal{P}_2$ are the n-bit binary representations of $L_{0,n}, L_{1,n}, \ldots, L_{\frac{N}{2}-1,n}$ and $b_0, b_1, \ldots, b_{\frac{N}{2}-1}$, respectively.*

Proof. Since a dilated Benes network is CF-rearrangeable, $\mathcal{P}$ can be realized in it with node-disjoint paths in one pass. Consider $\mathcal{P}_1$. The first n stages of a dilated Benes network form a baseline network. Thus $\mathcal{P}_1$ can be realized by a baseline network with node-disjoint paths in one pass. By Lemma 6, $\mathcal{P}_1$ can also be realized by a reverse baseline network with node-disjoint paths in one pass. Now consider $\mathcal{P}_2$. Since the last n stages of a dilated Benes network form a reverse baseline network, $\mathcal{P}_2$ can be realized in the last $n-2$ stages (i.e., stages 2, 3, $\ldots$, $n-1$) of a reverse baseline network with node-disjoint paths in one pass. Thus if we can prove that, in a reverse baseline network, input $L_{i,n}$, $0 \le i \le \frac{N}{2}-1$, can get to output link $L_{i,n}$ of stage 1, then $\mathcal{P}_2$ can be realized by a reverse baseline network with node-disjoint paths in one pass. For input $L_{i,n}$, choose its routing bit at stage 0 to be 0 if $\left\lfloor \frac{L_{i,n}}{2} \right\rfloor$ is even and 1 if $\left\lfloor \frac{L_{i,n}}{2} \right\rfloor$ is odd; choose its routing bit at stage 1 to be 0 if $L_{i,n}$ is even and 1 if $L_{i,n}$ is odd. It is not difficult to see that the above choices of routing bits ensure input $L_{i,n}$ to get to output link $L_{i,n}$ of stage 1. Therefore, $\mathcal{P}_2$ can be realized by a reverse baseline network with node-disjoint paths in one pass. By Lemma 6, $\mathcal{P}_2$ can also be realized by a baseline network with node-disjoint paths in one pass. Since a baseline (or reverse baseline) network follows destination tag routing, routing bits for $\mathcal{P}_1$ and $\mathcal{P}_2$ are the n-bit binary representations of $L_{0,n}, L_{1,n}, \ldots, L_{\frac{N}{2}-1,n}$ and $b_0, b_1, \ldots, b_{\frac{N}{2}-1}$, respectively. ∎

The following is our algorithm for routing an arbitrary semi-permutation $\mathcal{P}$ in an $N \times N$ baseline (or reverse baseline) network with node-disjoint paths in two passes.

Algorithm 4 ROUTING A SEMI-PERMUTATION IN A BASELINE OR REVERSE BASELINE NETWORK

Require: An arbitrary semi-permutation $\mathcal{P}$ and the characteristic strings OS_k, $1 \le k \le 2n-4$, of a $(2n-2)$-stage dilated Benes network.

Ensure: Routing bits $r_0 r_1 \ldots r_{n-1}$ and $r'_0 r'_1 \ldots r'_{n-1}$, each represented as an $(\frac{N}{2})$-bit array such that $r_k(j)$, $0 \le k < n$ and $0 \le j \le \frac{N}{2}-1$, represents the routing bit of input j^* at stage k for the first pass, and $r'_k(j)$, $0 \le k < n$ and $0 \le j \le \frac{N}{2}-1$, represents the routing bit of input j^* at stage k for the second pass, where $j^* = 2j$ if input $2j$ is in $\mathcal{P}$ and $j^* = 2j+1$ if input $2j+1$ is in $\mathcal{P}$.

use $\mathcal{P}$ to derive the optical windows OW_0 and OW_{2n-3} for a $(2n-2)$-stage dilated Benes network and use Algorithm 3 to find AR-bits $r_0 r_1 \ldots r_{n-3}$;
for $j = 0$ **to** $\frac{N}{2} - 1$ **do**
  let $y_{n-1} y_{n-2} \cdots y_0$ be the n-bit binary representation of the destination of input j^*;
  $r_{n-2}(j) \leftarrow y_{n-1}$;
  $r_{n-1}(j) \leftarrow y_{n-2}$;
  for $i = 0$ **to** $n-1$ **do**
    $r'_i(j) \leftarrow y_{n-1-i}$;
  end for
end for

In this algorithm, $r_0 r_1 \cdots r_{n-1}$ are the routing bits for the first pass (i.e., for $\mathcal{P}_1$), and $r'_0 r'_1 \cdots r'_{n-1}$ are the routing bits for the second pass (i.e., for $\mathcal{P}_2$). This algorithm uses the characteristic strings OS_k, $1 \le k \le 2n-4$, of a $(2n-2)$-stage dilated Benes network to find $r_0 r_1 \cdots r_{n-3}$. It sets r_{n-2} and r_{n-1} to the leftmost two bits of the n-bit binary representation of the destination and sets $r'_0, r'_1, \ldots, r'_{n-1}$ to the n-bit binary representation of the destination. Note that OS_k, $1 \le k \le 2n-4$, of a $(2n-2)$-stage dilated Benes network can be obtained before this algorithm runs; see (1). Thus we assume OS_k, $1 \leq k \leq 2n-4$, are inputs, too.

We now give an example for Algorithm 4. Let $\mathcal{P} = \begin{pmatrix} 0 & 2 & 5 & 7 & 8 & 11 & 13 & 15 \\ 13 & 11 & 2 & 0 & 9 & 14 & 5 & 7 \end{pmatrix}$ be the given semi-permutation. Then Algorithm 4 obtains the following routing bits.

$$\begin{pmatrix} r_0 & r_1 & r_2 & r_3 & r'_0 & r'_1 & r'_2 & r'_3 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Do notice that the routing bits obtained by Algorithm 4 works for both a baseline and a reverse baseline network. Routing paths in a baseline network are shown in Figure 7; those in a reverse baseline network are shown in Figure 8. It is not difficult to see that $\mathcal{P}_1 = \begin{pmatrix} 0 & 2 & 5 & 7 & 8 & 11 & 13 & 15 \\ 3 & 10 & 4 & 12 & 6 & 15 & 1 & 9 \end{pmatrix}$ and $\mathcal{P}_2 = \begin{pmatrix} 3 & 10 & 4 & 12 & 6 & 15 & 1 & 9 \\ 13 & 11 & 2 & 0 & 9 & 14 & 5 & 7 \end{pmatrix}$.

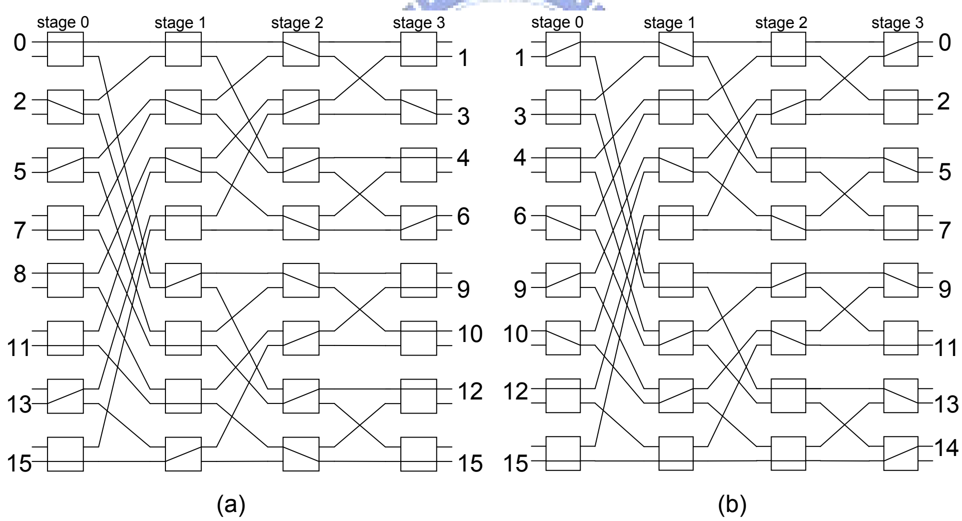

Figure 7: (a) Routing paths in the first pass. (b) Routing paths in the second pass.

We now analyze Algorithm 4.

Theorem 8. *Algorithm 4 takes $O(N \log_2 N)$ time and it can realize any semi-permutation with node-disjoint paths in a baseline (or reverse baseline) network in two passes.*

Proof. It is not difficult to see that Algorithm 4 takes $O(N \log_2 N)$ time. By Lemmas 7, it suffices to prove that: (i) $r_0 r_1 \cdots r_{n-1}$ are the n-bit binary representations

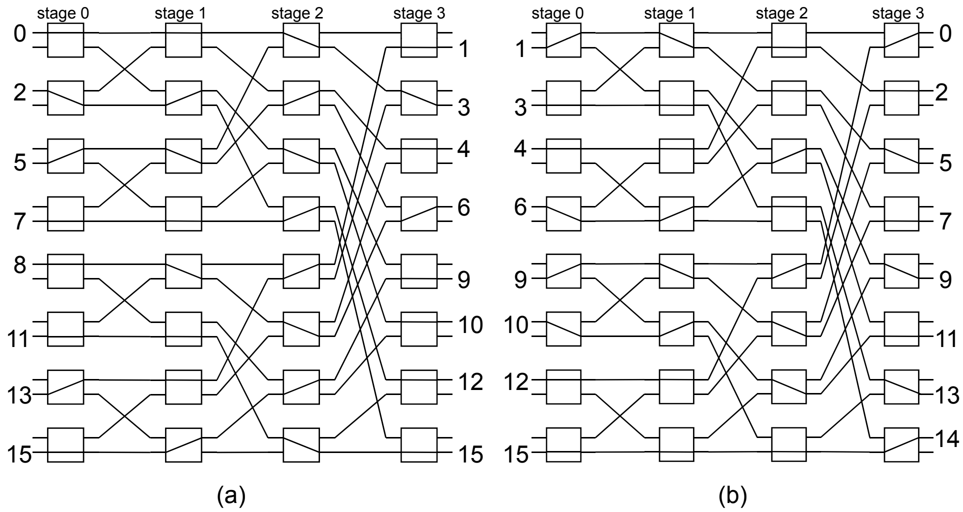

Figure 8: (a) Routing paths in the first pass. (b) Routing paths in the second pass.

of $L_{0,n}, L_{1,n}, \ldots, L_{\frac{N}{2}-1,n}$, and (ii) $r'_0 r'_1 \cdots r'_{n-1}$ are the n-bit binary representations of $b_0, b_1, \ldots, b_{\frac{N}{2}-1}$. Statement (ii) follows from lines 3, 6, 7, 8 of Algorithm 4. Since $L_{0,n}, L_{1,n}, \ldots, L_{\frac{N}{2}-1,n}$ are the intermediate destinations of an $(n-2)$-stage dilated Benes network for $\mathcal{P}$, if we set $r_0 r_1 \cdots r_{n-1}$ to the first n routing bits of the $2n-2$ routing bits obtained by Algorithm 3, then we have (i). By lines 1, 3, 4, 5 of Algorithm 4, $r_0 r_1 \cdots r_{n-1}$ are set accordingly. Thus we have (i). ∎

6 Concluding remarks

This thesis considers the crosstalk-free rearrangeability of combined optical MINs. In [8], Das formulated an elegant sufficient condition for the rearrangeability of a combined $(2n-1)$-stage electronic MIN and presented an $O(N \log_2 N)$-time permutation routing algorithm for MINs that satisfy the sufficient condition. In this thesis, we have formulated a sufficient condition for the crosstalk-free rearrangeability of a combined $(2n-2)$-stage optical MIN and a sufficient condition for the crosstalk-free rearrangeability of a combined $(2n-1)$-stage optical MIN. We have proposed an $O(N \log_2 N)$-time semi-permutation routing algorithm for optical MINs that satisfy the sufficient condition, and improved the decomposition algorithm in [28] and the permutation routing algorithm in [27].

References

[1] C. S. Babu and C. S. Raghavendra, "On the invariants of shuffle/exchange networks," in *Proceedings of the 1st IEEE Symposium on Parallel and Distributed Processing, IEEE Computer Society, Los Alamitos, CA*, pp. 249-256, 1989.

[2] V. E. Benes, "Optimal rearrangeable multistage connecting networks," *Bell System Tech. J.*, vol. 43, pp. 1641-1656, 1964.

[3] X. Bao and Q. Li, "Crosstalk-free rearrangeable multistage intercinnection networks," *Taiwan. J. Math.*, Vol. 10, No. 5, pp. 1225-1243, 2006.

[4] G. J. Chang, F. K. Hwang, and L. D. Tong, "Characterizing bit permutation networks," *Networks*, vol. 33, no. 4, pp. 261-267, 1999.

[5] H. Cam, "Rearrangeability of $(2n-1)$-stage shuffle-exchange network," *SIAM J. Comput.*, vol. 32, pp. 557-585, 2003.

[6] Z. Chen, Z. Liu, and Z. Qiu, "Bidirectional shuffle-exchange network and tag-based routing algorithm," *IEEE Commun. Lett.*, vol. 7, no. 3, pp. 121-123, 2003.

[7] C. Chen and J. K. Lou, "An efficient tag-based routing algorithm for the backward network of a bidirectional general shuffle-exchange network," *IEEE Commun. Lett.*, vol. 10, no. 4, pp. 296-298, 2006.

[8] N. Das, "More on rearrangeability of combined $(2n-1)$-stage networks," *J. Syst. Architect.*, vol. 51, pp. 207-222, 2005.

[9] S. T. Huang and S. K. Tripathi, "Finite state model and compatibility theory: New analysis tools for permutation networks," *IEEE Trans. Comput.*, vol. 35, pp. 591-601, 1986.

[10] F. K. Hwang, “The mathematical theory of nonblocking switching networks,” *Series on Applied Mathematics*, vol. 15, ch. 1, pp. 12-22, 2004.

[11] X. Jiang, P. H. Ho, H. Shen and S. Horiguchi, “A class of Benes-based optical multistage interconnection networks for crosstalk-free realization of permutations,” *IEICE Trans. Commun.*, vol. E89VB, no. 1, pp. 19-27, 2006.

[12] C. P. Kuruskal, “A unified theory of interconnection network structure,” *Theor. Comput. Sci.*, vol. 48, pp. 75-94, 1986.

[13] N. Linial and M. Tarsi, “Interpolation between bases and the shuffle-exchange network,” *Eur. J. Comb.*, vol. 10, pp. 29-39, 1989.

[14] E. Lu and S. Q. Zheng, “High-speed crosstalk-free routing for optical multistage interconnection networks,” in *Proceedings of International Conference on Computer Communications and Networks*, 2003.

[15] V. W. Liu, C. Chen, and R. B. Chen, “Optimal all-to-all personalized exchange in d-nary banyan multistage interconnection networks,” *J. Comb. Optim.*, vol. 14, pp. 131-142, 2007.

[16] 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, 2001.

[17] K. Padmanabhan, “Design and analysis of even-sized binary shuffle-exchange networks for multiprocessors,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 2, no. 4, pp. 385-397, 1991.

[18] K. Padmanabhan and A. N. Netravali, “Dilated networks for photonic switching,” *IEEE Trans. Commun.*, vol. 35, no. 12, pp. 1357-1365, 1987.

[19] C. Qiao and L. Zhou, "Scheduling switching element disjoint connections in stage-controlled photonic banyans," *IEEE Trans. Commun.*, vol. 47, no. 1, pp. 139-148, 1999.

[20] H. S. Stone, "Parallel processing with the perfect shuffle," *IEEE Trans. Comput.*, vol. 20, pp. 153-161, 1971.

[21] X. TAN and S. Y. R. LI, "Rearrangeability of tandem cascade of banyan-type networks," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 1, pp. 67-74, 2007.

[22] 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, 2000.

[23] A. Waksman, "A permutation network," *J. ACM*, vol. 15, pp. 159-163, 1969.

[24] C. Wu and T.-Y. Feng, "The universality of the shuffle-exchange network," *IEEE Trans. Comput.*, vol. 30, pp. 324-331, 1981.

[25] Y. Yang, J. Wang, "Optimal all-to-all personalized exchange in a class of optical multistage networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 9, pp. 567-582, 2001.

[26] Y. Yang and J. Wang, "Routing permutations with link-disjoint and node-disjoint paths in a class of self-routable interconnects," *IEEE Trans. Parallel Distrib. Syst.*, vol 14, no. 4, pp. 383-393, 2003.

[27] Y. Yang and J. Wang, "Routing permutations on baseline networks with node-disjoint paths," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 8,pp. 737-746, 2005.

[28] Y. Yang, J. Wang and Y. Pan, "Permutation capability of optical multistage interconnection networks," *J. Parallel Distrib. Comput.*, vol. 60, no. 1, pp. 72-91, 2000.