USC-SIPI REPORT #210

Optoelectronic Implementation of Multistage Interconnection Networks

by

Lily Cheng

August 1992

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Building
University Park/MC-2564
Los Angeles, CA 90089 U.S.A.

Dedication

To my parents,

for their constant love, encouragement, and support.

To my family, teachers, and friends,

who make the journey pleasant and fruitful.

Acknowledgments

I would like to thank my thesis advisor, Dr. Alexander A. Sawchuk, for his invaluable support, guidance, and encouragement throughout the course of this work. I am especially grateful for his openness and his broad knowledge, which provide me the complete freedom to look for the seldom or never explored fields and the wide world.

I would also like to thank my dissertation committee members: Dr. Stephen R. Forrest, whose enthusiasm, invaluable advice and effort makes this work better; Dr. B. Keith Jenkins, for his interest in this work and helpful discussions; Dr. Cauligi S. Raghavendra, for his stimulating comments and assistance.

Thanks go as well to all members of Signal and Image Processing Institute (SIPI), who makes this work possible and enjoyable. I gratefully acknowledge Dr. Allan Weber and Mr. Toy Mayeda for their help in computer software and hardware. I am especially grateful to Dr. Jimmy Wang and Mr. M. Govindarajan for their invaluable technical support and inspiring discussions. I would like to thank Dr. Chein-Hsun Wang and Dr. Kung-Shiuh Huang for their leading me to this interesting research area. The friendship and interesting discussions from Mr. Charles Kuznia, Mr. Andrew S. Miller, Mr. Sabino Piazzolla, Mr. Gregory Petrisor, Ms. Clare Waterson, Dr. Qinfen Zheng, Dr. Jason Young, and Dr. Ching-Yih Tseng are acknowledged. The support from the staff of SIPI, especially Ms. Gloria Bullock, Ms. Linda Varilla, Ms. Delsa Tan and Ms. Mayitta Penoliar, are greatly appreciated.

Thanks extended to other friends and teachers from my former institutes — Philips International Institute of Technologies in Eindhoven, the Netherlands, and National Chiao-Tung University in Hsin-Chu, Taiwan. I am especially grateful to Mr. Saeed Rajput, Mr. Chiu-Yeung Ngo, Mr. Meng-Che Li, Mr. Carrie-Lin, and Mr. Chiao-Tsung Young.

This research was supported by DOD/AFOSR University Research Initiative Program (URI) under grant AFOSR-90-0133 and by the National Science Foundation (NSF) under grant ECS-9015797. I would like to acknowledge this financial support which made this work possible.

Finally, the greatest thanks to my parents, my sisters and brother, and my fiance' Shih-Hao, for their love, patience and support during the course of this work.

Contents

List of Figures

List of Tables

Abstract

Communication between signal channels in parallel computing and telecommunication switching systems is accomplished by interconnection networks. With the increasing demand for solving computation-intensive tasks such as image processing, image understanding and telecommunication management, an interconnection network having high throughput is desirable. A 3-D network which occupies a physical volume in space and interconnects a 2-D array of channels or processing elements to another is a promising candidate for such applications. The implementation of such 3-D networks must have high temporal and spatial bandwidth to achieve the high throughput requirement.

The problems of electrical interconnections such as crosstalk, clock skew, pin-in/pin-out and delays, limit the application of electronic technology to a 3-D network. On the other hand, optics has inherent 3-D properties, high bandwidth, parallelism in processing and input/output, and a low degree of interaction between the channels. All these properties make optics a good candidate for implementing 3-D networks. However, because it is very difficult for photons to interact with photons, it is difficult to have logic gates implemented optically. A possible approach is to utilize the capabilities of both optics and electronics in a 3-D network, but many questions remain. From the designer's point of view, there are questions of how to design a *useful* network structure, how to use them for different applications, and how to simplify the hardware complexity. From the builder's point of view, there are questions of how to use these two technologies efficiently, how to design and implement each component, and how to optimize the performance. This thesis attempts to answer these questions and to demonstrate ways of doing so.

We concentrate on a special type of 3-D multistage network, called Omega (shuffle/exchange) network. This network is composed of stages of shuffling on a square array of channels as fixed interconnections combined with dynamic switch elements

at each stage. We discuss an optically efficient implementation of the shuffling operation on 2-D arrays. This method is called the *one-copy algorithm* and achieves 100% light power utilization theoretically as opposed to conventional methods which achieve only 25% efficiency. We have systematically developed the algorithm and have extended it to general types of shuffles. Optical 2-D shuffles using this method can be implemented by a hologram containing four facets. We experimentally built a holographic system to demonstrate this algorithm.

The 2-D version of this shuffle/exchange network, where shuffling is performed on an 1-D vector of channels, has been demonstrated to be equivalent to other multistage networks and has been studied in depth. We discuss the construction of various 3-D shuffle/exchange networks, and investigate their properties and relations to the 2-D networks. By doing so, we automatically understand what components are necessary and how to use various network structures in different applications. A complete mathematical discussion is provided. We design the optoelectronic dynamic switch element systematically by considering its functional structure, signal modulation techniques, and practical implementation issues. A prototype of one of the designs has been built from discrete components.

Because the 3-D shuffle/exchange network contains multiple stages, the hardware for both shuffling and switching in one stage must be duplicated $\log_2 N$ times to make the network *full access*, where N is the number of channels in the square array. We propose a method to spatially arrange the channels such that a single set of optics are used for 2-D shuffles for all the stages and the design of switch elements is simplified. The resulting system requires a feedback system and is an example of a *space multiplexed* optical multistage network. We discuss the channel arrangement, the feedback system, and the complexity issue for this space multiplexed architecture. Experimental results are presented.

Chapter 1

Introduction

Communication between signal channels in parallel computing and telecommunications is accomplished by interconnection networks. Due to the increasing demand for efficient, flexible, and reliable parallel signal processing, interconnection networks have become a critical issue. There are three well known types of interconnection networks: crossbar networks, bus networks and multistage interconnection networks (MINs). Crossbar networks constructed from a single crossbar switch containing N^2 crosspoints, where N is the number of channels, are strictly nonblocking (i.e. any two channels can be connected without altering any existing connection). However, the complexity and cost rise as $O(N^2)$. Bus networks have the simplest structures and are easy to build but they suffer from serious propagation delay (proportional to N) and contention problems. Multistage networks are a compromise between the above. For N channels, where $N = 2^m$, the network delay is proportional to m, whereas the cost and complexity is proportional to $N \times m$. Thus, multistage networks have attracted great interest recently.

A full-access rearrangeable, nonblocking MIN is composed of several stages of fixed link patterns combined with dynamic switch elements at each stage. Different architectures of multistage networks, such as the Omega/shuffle-exchange network [28], Baseline network [60, 61], Banyan network [13], crossover network [25, 35] etc., based on various link patterns have been proposed. It has been shown that all these networks are topologically equivalent [60]. The Omega/perfect shuffle network attracts the most attention because of its relatively regular and simple structure. The link pattern of the Omega network is based on 1-D perfect shuffles. In the 1-D

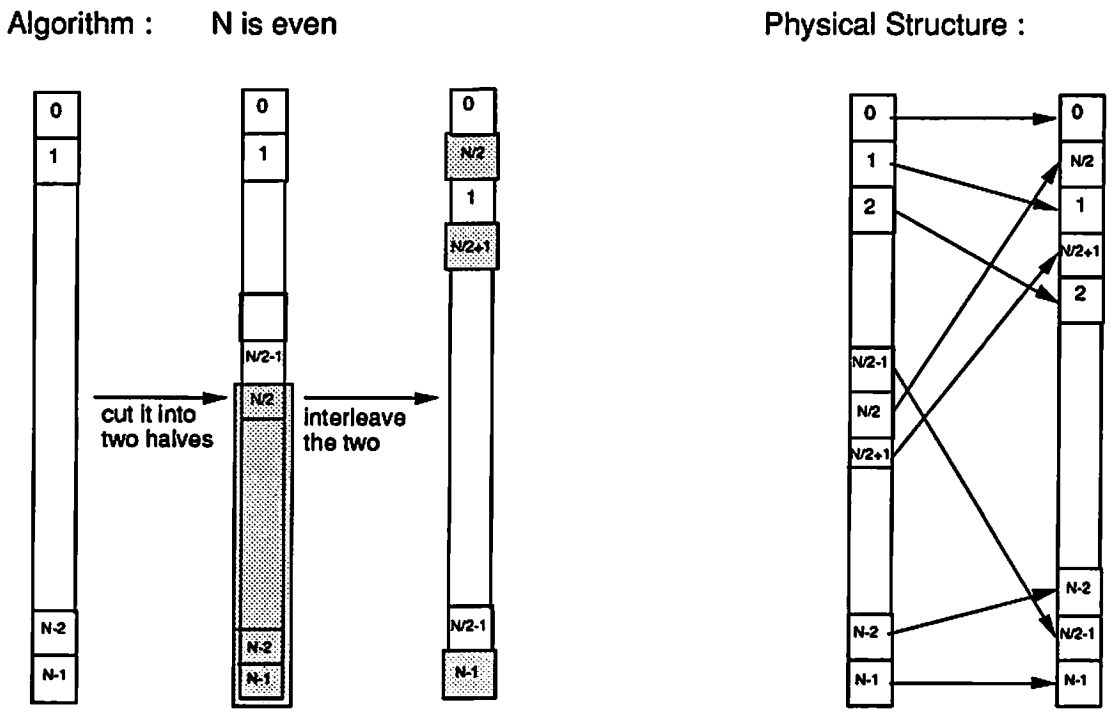


Figure 1.1: 1-D perfect shuffle.

perfect shuffle, the N channels (assuming N is even) are divided into two halves, then these two halves are interleaved perfectly as shown in Fig. 1.1. At each stage, neighboring pairs of 1-D channels are connected through switch elements. As a consequence, a path between any two channels can be established when enough stages are present, and when the switch elements are properly controlled. The Omega network has been implemented with VLSI technology in present systems.

1.1 Optoelectronic Volume Multistage Networks

Here, we describe networks by their construction and data format. A planar network, which can be implemented by electronic VLSI technology, has the channels arranged in a 1-D array and the fixed link patterns (such as 1-D perfect shuffles) arranged in a plane. A 3-D network occupies a physical volume in space, and has its input and output channels arranged in 2-D arrays which are interconnected by fixed link patterns (such as 2-D perfect shuffles) arranged in space.

With the increasing demand for solving computation-intensive tasks such as image processing and image understanding, a volume, dynamic interconnection network that interconnects a 2-D array of channels or processing elements to another is desirable [44]. In this dissertation, we concentrate on the optoelectronic implementation of volume multistage networks. Here, the fixed interconnections are implemented by optical components, and the dynamic switch elements are implemented optoelectronically.

1.1.1 Fixed Optical Interconnections

The limitations of interconnections implemented by electronics are well known [14], and include crosstalk, clock skew, pin-in/pin-out limitations, and delay problems. Other problems including high frequency parasitic coupling between circuit elements, excessive power dissipation, and impedance mismatch, limit the performance of electronically implemented networks. These limitations restrict the utlilzation of electronics in volume networks. On the other hand, optics is ideally suited for providing the fixed interconnections in a volume network. Because optics has the inherent 3-D properties, high temporal and spatial bandwidth, parallelism in processing and input/output, and a low degree of interaction between the channels–properties which are desirable in implementing a volume network.

1.1.2 Dynamic Optoelectronic Switch Elements

Dynamic switch elements at each stage of the multistage networks provide any input the accessibility to outputs besides the dedicated one provided by the fixed interconnections. The propagation direction of light beams can be affected by changing the absorption coefficient and the refraction index of the media. Various optical switches have been proposed. However, it is difficult to obtain a high speed, low power consumption, high signal-to-noise ratio, and cascadable optical switch element with the available technologies. A suggested approach is to utilize the capabilities of both optics and electronics for the dynamic switch elements. Optoelectronic integrated circuit (OEIC) technology is adapted to build the dynamic switch elements.

1.2 Objectives

The objective of this research is to investigate the practicality of an optoelectronic volume multistage network. It covers the discussion of various network structures, the development of high efficient algorithms for implementing fixed interconnections, a unified design on the optoelectronic switch elements for dynamic interconnections and the architecture of the network with simplified hardware. It includes both the theoretical and experimental work on both the fixed optical interconnections and the dynamic optoelectronic switch elements.

1.3 Thesis Organization

This research is concentrated on implementing optoelectronic volume Omega (shuffle/exchange) networks. Figure 1.2 shows the organization of this thesis.

Chapter 2 reviews past work related to this research. The implementation of optical shuffles, both 1-D and 2-D are reviewed in section 2.1; section 2.2 presents several design and implementation of optical and optoelectronic switch elements.

Chapter 3 discusses the properties and relationship of the class of volumn (3-D) Omega networks. The mathematical framework is established along the discussion. Section 3.1 presents the two types of 2-D shuffles and their relation; section 3.2 discusses the types of switch elements and their influences on the input channels; various 3-D Omega networks due to different combinations of these 2-D shuffles and switch elements are presented in section 3.3; the relationship of these networks are also investigated in section 3.3 and summarized in Table 3.4.

Chapter 4 introduces the algorithms perform optical 2-D shuffles. The mathematical notations and some definitions are provided in section 4.1; section 4.2 presents the mathematical description of both the conventional four-copy algorithm and the improved *one-copy algorithm* for 2-D perfect shuffles; section 4.3 generalizes the one-copy algorithm to perform 2-D $r-$shuffles (where r is a prime); section 4.4 discusses the possibility of utilizing the same one-copy algorithm for 2-D r^k-shuffles; the application of one-copy algorithm for the inverse of the above discussed shuffles is investigated in section 4.5 and summarized in Table 4.1.

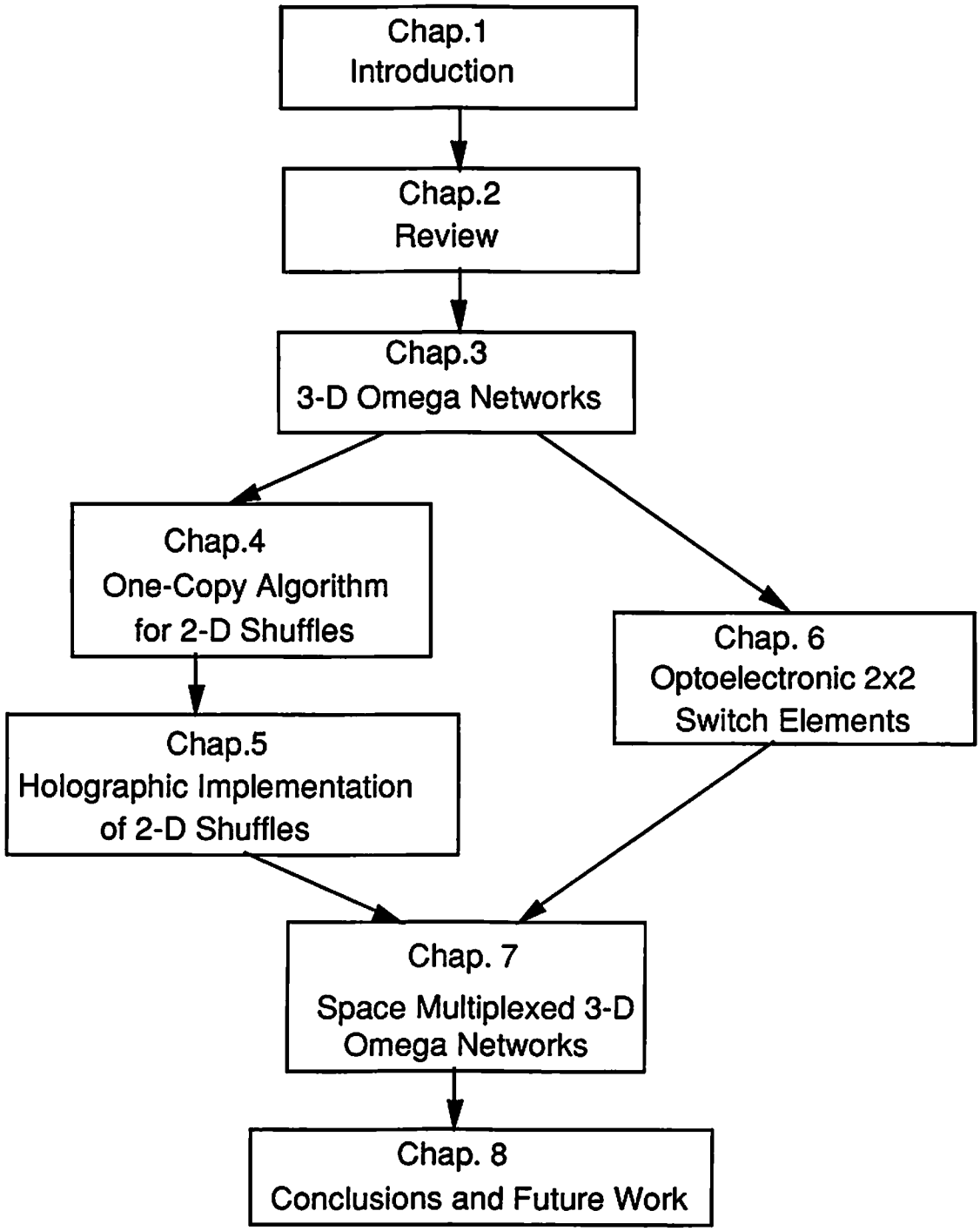


Figure 1.2: Thesis organization.

Chapter 5 presents the holographic implementations of 2-D shuffles based on the one-copy algorithm. Two experiments have been performed. Section 5.2 presents the first experiment using silver halid photographic material; section 5.3 provides some practical considerations and performance analysis; the a second experiment using Du Pont photopolymers and is built to remove some limiting factors in the first experiment is presented in section 5.4

Chapter 6 addresses the design of optoelectronic 2×2 switch elements. The experimental demonstration of an optoelectronic 2×2 switch element based on one of the designs is presented. Section 6.1 introduces the generic design and the basic structures of the switch elements; section 6.2 discusses the methods of modulating the signals, the control, and the power supply of the switch elements; examples of some designs using electrical amplifiers are presented in section 6.3; section 6.4 presents the implementation of an optically powered optoelectronic switch element with polarization control; the control and routing schemes for multistage applications are discussed in section 6.5.

Chapter 7 discusses the different architectures of the 3-D Omega networks and ways to simplify the hardware complexity. The principle of pipelining and different architectural designs of the multistage networks is introduced in section 7.1; section 7.3 presents an implementation of a *space multiplexed* network using the one-copy algorithm; section 7.4 provides the mathematical reasoning for the application of one-copy algorithm in this space multiplexed Omega network; section 7.5 considers the practical issues of such networks including the space-bandwidth requirement and the design of the required feedback system.

Chapter 8 concludes this research. A brief discussion of future research is given.

1.4 Contributions

In this research, we systematically study, design, and simplify optoelectronic volume Omega networks. The contributions of this work are summarized as follows:

- The exploration of the properties and relationship of 3-D Omega networks.
- The investigation of the applications of various structured 3-D Omega networks.

- The development of light efficient one-copy algorithm for optical 2-D shuffles.
- The experimental demonstration of holographic implementation of 2-D shuffles based on the one-copy algorithm.
- The analysis of the holographic implementation of 2-D shuffles.
- The generic designs of optoelectronic 2×2 switch elements.
- The experimental demonstration of an optically powered, optoelectronic 2×2 switch element using polarization control.
- The investigation of simplifying the optical hardware for a multistage networks using spatial multiplexing technique.
- The demonstration of application of one-copy algorithm in a space multiplexed network.
- The design of a simple feedback system for a space multiplexed network.

Chapter 2

Previous Work — a Review

The two fundamental building blocks of any optoelectronic volume multistage networks are:

- fixed optical interconnections, and
- dynamic optical or optoelectronic switch elements.

This research is concerned with an optoelectronic volume Omega (shuffle/exchange) network, where the shuffling operation (as shown in Fig. 1.1) is used as the basic interconnection pattern, and 2 input/2 output switch elements are used for dynamic switching function.

We first review previous work on optical shuffles and optical/optoelectronic switch elements.

2.1 Optical Shuffles

Extensive research has been done for optical shuffles in the past few years.

From the system point of view, they can be classified as space invariant, strictly space variant, and hybrid (space semi-variant) systems [26, 55]. In a space invariant system, there is only one point spread function (PSF) for the complete system. An imaging system is a typical example of a space invariant system. Optical shuffles can be easily implemented with space invariant systems [47, 4, 46, 43, 45, 2]. However, the low utilization of light power is a major drawback. In a strictly space variant

system, the number of PSFs equals the number of channels in the system. A strictly space variant optical shuffle system gives 100% light efficiency in principle because each channel has a dedicated PSF to perform the permutation. However, the complexity involved in such a system is normally far more than desired [14, 38, 39, 19] . In a hybrid system, the number of PSFs is larger than one but smaller than the number of channels in the system. It becomes a compromise between the above two [32, 34, 8, 56]. In chapter 4 and 5, we show a hybrid system which can be used to perform 2-D shuffles with 100% light efficiency (in principle).

From the implementation point of view, the proposed implementation of optical shuffles can be classified as follows:

1. guided–Light beams are confined to traveling in waveguides. This is similar to conventional electronic implementation on shuffles. Optical fibers and integrated optics are the technologies used in this approach [14]. The major drawback here is the need of large bundles of fibers and potential crosstalk problems for a densely connected network.

2. free space–Light beams travel in free space and are deflected or reflected by optical components in the path. This approach has the advantage of very low interactions between the channels, thus, provides a low crosstalk parallel system. It can further be classified according to various components used. Each type implementation is presented with the the problems associated with it.

   (a) classical optics–Lenses, mirrors, and prisms are used in various architectures to perform optical shuffles. They can be used in a range of wavelength illumination. However, most suffer from aberrations. The main streams include:

      i. prisms and lenses–Combinations of prisms and lenses are used to provide the interlacing and shifting operations for shuffles [32, 34, 8, 2, 19]. These implementations can be easily understood by ray tracing. The common drawback is that specially designed prisms or lenses are required.

ii. off-axis lenses–Shuffling in plane can be implemented by duplicating the input with four off-axis lenses and interleaving them as desire [47, 43]. The major drawback of this method is the low utilization of light power (less than 25%), which is discussed in Chapter 4.

iii. Fresnel mirrors and lenses–Sheng [45] utilized Fresnel mirrors with proper tilt in combination with an imaging system to achieve 2-D shuffles. The low light efficiency is also the major drawback.

(b) interferometric setup–Brenner *et. al.* [4] proposed several designs of performing 1-D shuffles with different interferometric setups . Only 50% of the light is used in this approach.

(c) spatial grating–Diffraction gratings used as spatial filters in a 4-f system are proposed to implement both 1-D and 2-D optical shuffles [46, 31]. The complicated grating design process, the limited efficiency, and crosstalk constraints are the issues that need to be considered when used in a dense network.

(d) holographic component–Holographic implementation of optical shuffles is another interesting approach. It can be used in a 4-f system as a spatial filter [4] or simply used as a light deflector [38, 39, 56]. More constraints are imposed when the hologram is used as a spatial filter. Monochromatic illumination should be used in this approach.

The classification and a brief review of the work on optical shuffles is shown in Table. 2.1. The letters "I", "H", and "V" in the "class" column indicate that it is a space invariant, hybrid, and space variant system respectively.

2.2 Optical/Optoelectronic Switches

Photonic technology has been recognized to be of great importance in telecommunication switching networks. The strength of photonic technology lies in:

- high temporal bandwidth–The transmission bandwidth of optical devices normally exceeds 100 Gbits/sec [21], which is far above its electronic counterpart. The switching bandwidth, however, depends on the material and physical

principle used in the device. Multiplexing in spatial or other domains can be used to maximize the utilization of the high temporal bandwidth of photonic devices.

- high spatial bandwidth–A typical optical imaging system exhibits a resolution on the order of 10 μm over a 1 mm field [21]. The extremely high spatial bandwidth of an optical system, combined with its high temporal bandwidth, provides a promising future for high throughput parallel systems.

The optical/optoelectronic switches have been studied by researchers from all different areas ranging from system designers, to device developers, to material scientists. Here, we present a classification scheme for these switches from different aspects. Some of these switches under active development will be studied under the guideline of our classification scheme.

The classification can be viewed from three aspects:

1. system –The switches can be classified according to the way the signals are modulated, the multiplexing method, the switching mechanism, and the control strategy from the system point of view.

   (a) signal modulation–The zeros and ones of signal beams passing the switches can be modulated with different intensities, wavelengths, phases, polarizations, or spatial patterns. The modulation methods can be applied according to the system requirement.

   (b) multiplexing method–To fully utilize the capacity of a network, signals from many channels can be multiplexed into the system. They can be multiplexed in various domains such as space, time, wavelength, frequency, polarization, or the combination of the above domains. Several levels of multiplexing can be used for large systems.

   (c) switching mechanism–The signals have to be switched towards their destinations in the intermediate stations in the network. The mechanisms used in the intermediate stations (nodes) can be circuit switching, message switching, or packet switching.

(d) control strategy–The control of the switching can be centralized in a central controller or be distributed to local controllers. Hybrid control strategy is also possible.

2. device–The devices for optical/optoelectronic switches to date can be characterized in six categories:

   (a) fiber optics–Directional couplers are the most common devices for guided switching.

   (b) optical components–Classical optical elements, holograms, and photorefractive materials can be used for optical switching.

   (c) spatial light modulators (SLM)– By individually addressing the pixels of SLMs, independent control of the switches on a 2-D plane can be achieved.

   (d) liquid crystal (LC)–Liquid crystal has large optical anisotropy and consumes very little power during operation, therefore, offers potential compact 2-D switch element arrays.

   (e) optoelectronic integrated circuit (OEIC)–Using the optical nonlinearity of quantum well structures in semiconductor materials and proper configurations, semiconductor optical amplifiers and bistable laser diodes can be built. These devices, which provide gain and bistability, can further be integrated and used as active switch elements.

3. physics–From a physical point of view, the switches can be classified according to their physical structure, the switching principle, and the control mechanism.

   (a) physical structure–The paths that signals pass through a switch can be either guided or unguided.

   (b) switching principle–The principles that a switch used to perform a "switching" can be:

      i. mechanical

      ii. thermal

      iii. acoustooptic

iv. magnetooptic

v. electrooptic

vi. optical

(c) control–Optical/optoelectronic switches are controlled optically, electrically, or with a hybrid combination.

Most of the recent developments are concentrated in exploring various devices for optical/optoelectronic switch elements. Indeed, devices are the basic building blocks for a system. Once the basic building blocks are fully understood and properly designed, system designers can maximize their strengths in different applications. Here several devices (summarized in Table. 2.2) are presented according to our previously introduced classification scheme. Details of the operation of these switch elements can be understood from the reference papers. However, most of them lack the sufficient information need for characterization from the system point of view. It is left to the system designers to decide how to use them in various applications.

- deformable mirror devices (DMD) [37, 17]– DMDs can be used as unguided switch elements based on mechanical movements of the deformable mirror elements with electrical control. The signals are phase modulated. They are SLM type device.

- smectic liquid crystals [27]– Smectic liquid crystals are used as unguided switch elements based on thermal effects with optical control. The signals are modulated in spatial domain. They are liquid crystal type device.

- acousto-optic cells [20, 54]– Acousto-optic cells can be used as unguided or guided switch elements based on acousto-optic Bragg diffractions with electrical control. The signals are modulated in intensity. They are SLM type device.

- magneto-optic devices — Both mageto-optic spatial light modulators, such as LIGHT-MOD [41] and ferroelectric liquid crystals (FLC) [40] are used as unguided switch elements based on magnetooptic effects with electrical control.

- directional couplers– Directional couplers are demonstrated as active packet-switched guided switch elements based on electrooptic effects with electrical control [11]. The signals are polarization modulated. They are fiber optics type device.

- PLZT ceramics [9, 42]– The electrooptic effects in PLZT ceramics can be used to build unguided switch elements with electrical or optical control. They are SLM type device.

- photorefractive materials [58]– Photorefractive materials can be used as unguided switch elements based on four wave mixing with optical control. The signals are intensity modulated. They are devices built on optical components.

- all-optical devices– Free-space all optical switch elements with optical control were demonstrated [62]. The signals are polarization modulated in their design.

- OEIC– Researches have been made to build unguided OEIC switch elements based on electrooptic effects with either optical or electrical control [23, 29, 52, 50]. The signals are intensity modulated in the demonstrations.

More techniques and devices, which are not described in this short review, have been developed in the past few years. One can get complete view from [12, 53, 22, 36].

year	investigators	class	applications
1984/7	Goodman *et. al.* [14]	v, guided	1-D
1986/5	Lohmann [32, 34]	H, prisms and lenses	1-D, 2-D separable
1987/4	Eichmann *et. al.* [8]	H, prisms and lenses	1-D general
1988/1	Stirk *et. al.* [47]	I, off-axis lenses	2-D folded
1988/1	Brenner *et. al.* [4]	I, interferometric setup, holographic component	1-D
1988/4	Song *et. al.* [46]	I, spatial grating	1-D general
1988/8	Sawchuk *et. al.* [43]	I, off-axis lenses	2-D folded and separable
1988/10	Lohman *et. al.* [31]	V, spatial grating	1-D, 2-D folded and separable
1989/8	Sheng [45]	I, Fresnel mirrors and lenses	2-D folded and separable
1991/1	Bian *et. al.* [2]	I, prisms and lenses	2-D folded and separable
1991/3	Restall *et. al.* [38, 39]	V, holographic component	2-D folded and separable
1991/3	Wang *et. al.* [56]	H, holographic component	2-D folded and separable
1991/7	Haney *et. al* [19]	V, lenses	1-D, 2-D folded and separable

Table 2.1: Review of optical shuffles; I: spaceinvariant; H: hybrid; V: space variant.

switching principle	device name	physical structure	control	device type	signal modulation
mechanical	DMD [37, 17]	unguided	e	SLM	phase
thermal	smectic liquid crystals [27]	unguided	o	LC	spatial
acoustooptic	acousto-optic cells [20, 54]	unguided, guided	e	SLM	intensity
magnetooptic	LIGHT-MOD [41]	unguided	e	SLM	intensity
	FLC [40]	unguided	e	LC	polarization
electrooptic	directional couplers [11]	guided	e	fiber optics	polarization
	PLZT [9, 42]	unguided	e, o	SLM	intensity
	S-SEED [23, 29]	unguided	e	OEIC	intensity
	VSTEP [52, 50]	unguided	o	OEIC	intensity
optical	photorefractive materials [58]	unguided	o	optical components	intensity
	all optical switch [62]	unguided	o	optical components	polarization

Table 2.2: Review of optical/optoelectronic switch elements.

Chapter 3

3-D Omega Networks

Various 3-D Omega networks can be obtained by combining either of the two types of 2-D shuffles with different types of switch elements arranged in various ways. This chapter presents these 3-D Omega networks, and discusses their properties and relationships.

3.1 Shuffling on 2-D Arrays

There are two kinds of 2-D perfect shuffle, namely, the 2-D separable shuffle and the 2-D folded shuffle. Both operations are performed on $N = n \times n = 2^m$ (m is even) channels which are arranged in an $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ array in the 2-D plane. We treat this array as a matrix A, where each channel is represented by a matrix element $a_{p,q}$, p is the row index and q is the column index. Throughout this chapter, we represent these two indices as binary numbers in the form $p = (b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}})_2$, $q = (b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0)_2$, where $b_i \in \{0,1\}$ (The 2's complement of b_i is denoted as $\bar{b}_i$.). With this notation, the decimal values of the indices p and q are

$$p = \sum_{i=0}^{\frac{m}{2}-1} b_{i+\frac{m}{2}} 2^i \tag{3.1}$$

$$q = \sum_{i=0}^{\frac{m}{2}-1} b_i 2^i. \tag{3.2}$$

We denote the matrix A in the compact form $A = [\ a_{p,q}\ ]$. Subscript 2's are used in the notations of some operations to show that the binary number system is used for the indices.

In this section, we discuss the algorithms, the applications of these two 2-D shuffles, and their relationship in detail.

3.1.1 2-D Folded Shuffle

When an optical system is used to perform a 1-D perfect shuffle on N channels, system aberrations and path length differences increase with N. The optical 2-D folded shuffle performs the 1-D perfect shuffle more efficiently by arranging the long 1-D array of channels in a 2-D matrix. Here, the rows and the columns of the input channels are obtained by folding a 1-D input array. A 1-D perfect shuffled version of the input can be obtained after unfolding the output plane of a properly shuffled 2-D output as shown in Fig. 3.1. Although the implementation of the 2-D folded shuffle is entirely different from the 1-D perfect shuffle, the function is the same.

Mathematically, the shuffling operations can be expressed as permutations on the index digits. Let $\sigma(\bullet)$ denote the shuffling permutation performed on the object $\bullet$. The 2-D folded shuffle, denoted as $\sigma_{f,2}$, is actually a 1-D perfect shuffle on the unfolded 2-D plane. Thus, the row and column indices are considered as one entity, and the shuffling operation is a circular left shift on the digits.

Definition 3.1 *2-D Folded shuffle ($\sigma_{f,2}$)*

$$\begin{aligned} \sigma_{f,2}(A) &= \sigma_{f,2}(\ [\ a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}\ ]\ ) \\ &= [\ a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1}, b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}}\ ] \end{aligned} \tag{3.3}$$

The 2-D folded shuffle operation forms a *cyclic group* of order m because A arrives at its original state after m applications of $\sigma_{f,2}$. Because the 1-D perfect shuffle is useful for 1-D array sorting, interconnections, and signal processing (such as the 1-D fast Fourier transform) [48], the 2-D folded shuffle is also useful in these applications.

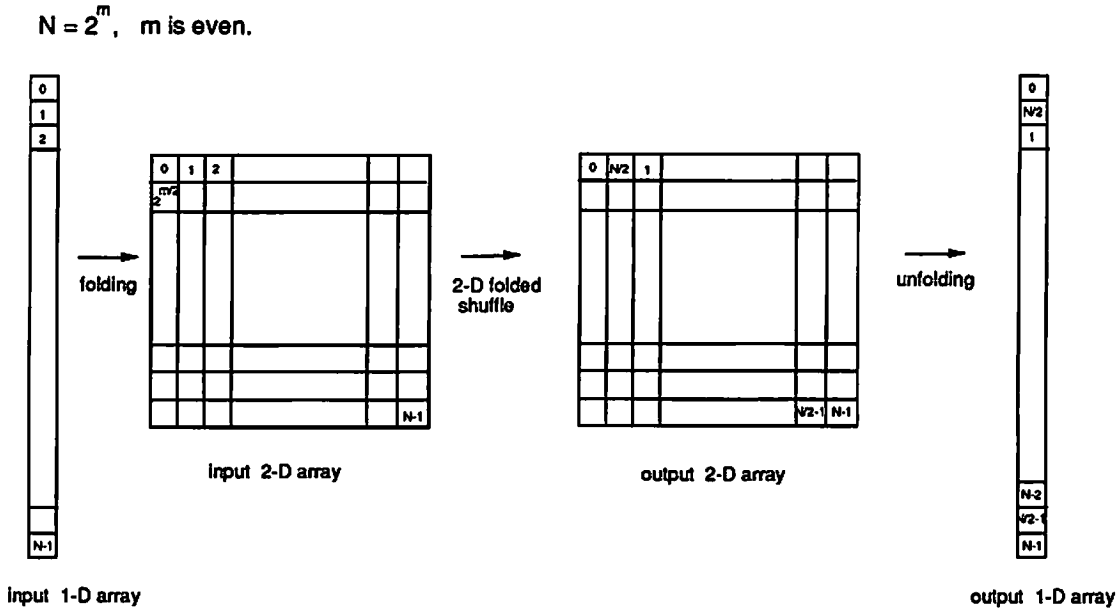


Figure 3.1: 2-D folded shuffle.

3.1.2 2-D Separable Shuffle

Some signal processing applications such as image processing require independent processing on the two dimensions. In this case, a 2-D separable shuffle shown in Fig. 3.2 is useful. Here, the rows and the columns are shuffled independently. The 1-D perfect shuffle is first applied to the rows, then to the columns. The 2-D separable shuffle operation, denoted as $\sigma_{s,2}$, is a circular left shift on the row and column binary indices.

Definition 3.2 *2-D Separable shuffle ($\sigma_{s,2}$)*

$$\begin{aligned} \sigma_{s,2}(A) &= \sigma_{s,2}([\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, \, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,]) \\ &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1}, \, b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{\frac{m}{2}-1}} \,] \end{aligned} \tag{3.4}$$

The 2-D separable shuffle operation also forms a *cyclic group*, but of order $\frac{m}{2}$. It can be considered as the dimensionally extended version of the 1-D perfect shuffle. Therefore, it can be used for 2-D array sorting, interconnections, and signal processing (such as the 2-D FFT).

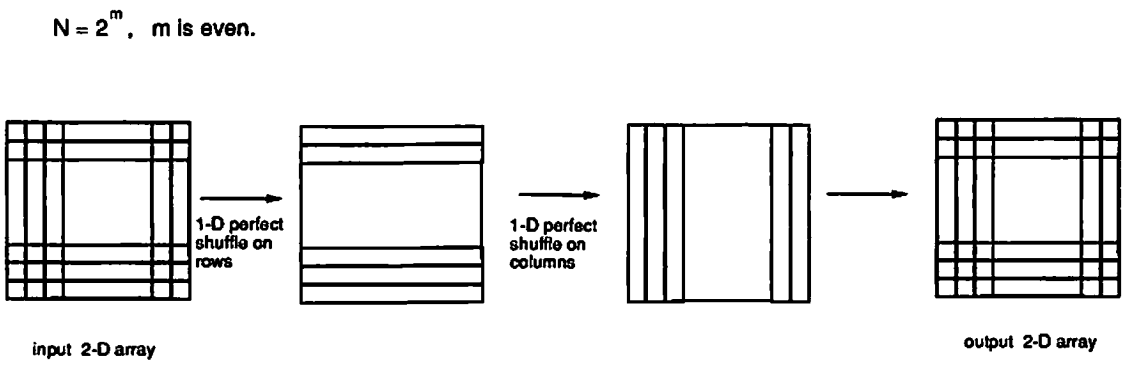


Figure 3.2: 2-D separable shuffle.

3.1.3 Relationship Between Folded and Separable Shuffles

Although the 2-D folded and 2-D separable shuffles are different operations, one can be realized by the other through pre-permutation or post-permutation of the channels. To demonstrate this fact, we define one more permutation on the index bits. Let $\rho_{i,j}(\bullet)$ be the permutation on object $\bullet$ based on the bit exchange of the i^{th} and the j^{th} bits of its index.

Definition 3.3 *bit exchange ($\rho_{i,j}$)*

This operation exchanges the i^{th} and the j^{th} bits, where $0 \leq i, j \leq (m-1)$,

$$\begin{aligned} \rho_{i,j}(A) &= \rho_{i,j}([\, a_{b_{m-1}b_{m-2}\ldots b_{i+1}b_i b_{i-1}\ldots b_{\frac{m}{2}}, \, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_{j+1}b_j b_{j-1}\ldots b_1 b_0} \,]) \\ &= [\, a_{b_{m-1}b_{m-2}\ldots b_{i+1}b_j b_{i-1}\ldots b_{\frac{m}{2}}, \, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_{j+1}b_i b_{j-1}\ldots b_1 b_0} \,]. \end{aligned} \tag{3.5}$$

Due to the symmetry of the bit exchange operation, we have

$$\rho_{i,j}(A) = \rho_{j,i}(A).$$

Obviously,

$$= \rho_{i,j}^{-1}(A).$$

Furthermore, the permutations can be combined to get a new permutation. In our notation, the sequence of a composed permutation is executed from left to right

(i.e. $f \circ g(A)$ denotes that f operates on A before g). We can now find the relationship between $\sigma_{f,2}$ and $\sigma_{s,2}$ (as defined in Eqs. (3.3) and (3.4)).

Observation 3.1 $\sigma_{f,2}(A) = \rho_{m-1,\frac{m}{2}-1} \circ \sigma_{s,2}(A)$

proof :

$$\begin{aligned} A &= [\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,] \\ \rho_{m-1,\frac{m}{2}-1}(A) &= [\, a_{b_{\frac{m}{2}-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{m-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,] \\ \rho_{m-1,\frac{m}{2}-1} \circ \sigma_{s,2}(A) &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1},b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}} \,] \\ &= \sigma_{f,2}(A) \qquad \square \end{aligned}$$

Observation 3.2 $\sigma_{f,2}(A) = \sigma_{s,2} \circ \rho_{\frac{m}{2},0}(A)$

proof :

$$\begin{aligned} A &= [\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,] \\ \sigma_{s,2}(A) &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1},\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{\frac{m}{2}-1}} \,] \\ \sigma_{s,2} \circ \rho_{\frac{m}{2},0}(A) &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1},b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}} \,] \\ &= \sigma_{f,2}(A) \qquad \square \end{aligned}$$

From the above, we get two mechanisms for performing the 2-D folded shuffle by preprocessing on the input channels or postprocessing on the output channels of the 2-D separable shuffle.

Method 1. Block Exchange The input plane is divided into four quadrants. Quadrants I and III are first exchanged, this is an operation corresponds to the bit exchange $\rho_{m-1,\frac{m}{2}-1}(A)$ as in Observation 3.1. Then the 2-D separable shuffle ($\sigma_{s,2}$) is applied. We obtain the 2-D folded shuffle at the output plane as shown in Fig. 3.3(a).

Method 2. Element Exchange First, the 2-D separable shuffle ($\sigma_{s,2}(A)$) is applied to the input plane. The resulting plane is then divided into 2×2 submatrices. The upper right and the lower left channels of each submatrix are exchanged, which corresponds to the $\rho_{\frac{m}{2},0}$ permutation as in Observation 3.2. The 2-D folded shuffled version appears as the output (see Fig. 3.3(b)).

Stucke shows similar results in [49]. Based on the same principle, if we apply inverse operations combined with the 2-D folded shuffle, we can perform the 2-D separable shuffle, as shown in the two following Observations.

Observation 3.3 $\sigma_{s,2}(A) = \rho_{m-1,\frac{m}{2}-1} \circ \sigma_{f,2}(A)$
proof :

$$\begin{aligned} A &= [\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,] \\ \rho_{m-1,\frac{m}{2}-1}(A) &= [\, a_{b_{\frac{m}{2}-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{m-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,] \\ \rho_{m-1,\frac{m}{2}-1} \circ \sigma_{f,2}(A) &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1},\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{\frac{m}{2}-1}} \,] \\ &= \sigma_{s,2}(A) \qquad \square \end{aligned}$$

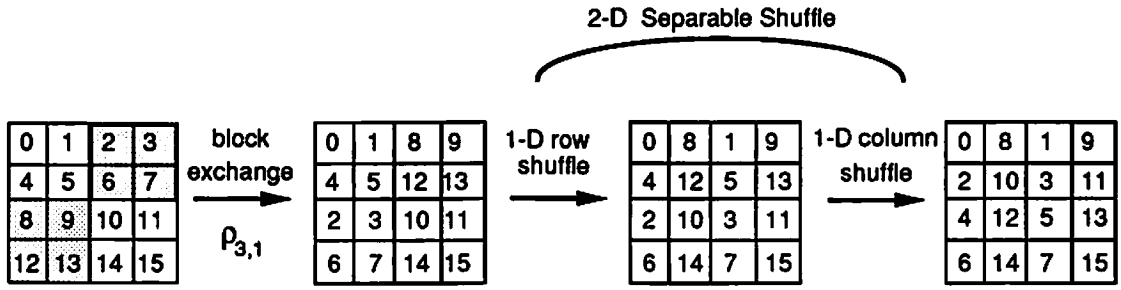
Observation 3.4 $\sigma_{s,2}(A) = \sigma_{f,2} \circ \rho_{\frac{m}{2},0}(A)$
proof :

$$\begin{aligned} A &= [\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,] \\ \sigma_{f,2}(A) &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1},b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}} \,] \\ \sigma_{f,2} \circ \rho_{\frac{m}{2},0}(A) &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1},\ b_{\frac{m}{2}-2}\ldots b_1 b_0 \frac{m}{2}-1} \,] \\ &= \sigma_{s,2}(A) \qquad \square \end{aligned}$$

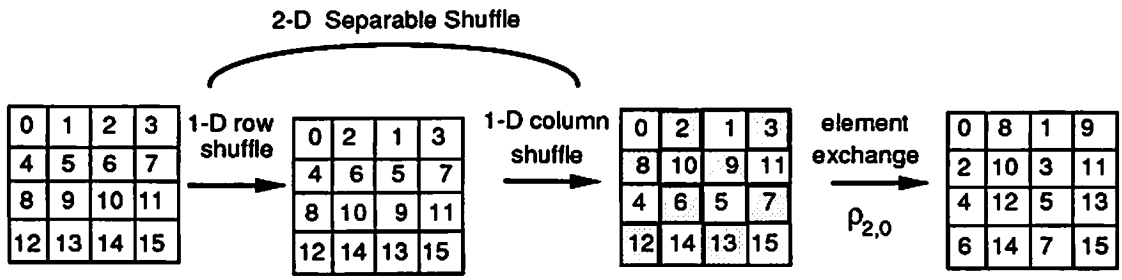
These observations were used by Sawchuk and Glaser [43] in their optical implementation of the folded shuffles. By skewing the lenses properly, the same optical setup for implementing 2-D separable shuffle can also be used to perform 2-D folded shuffle. The *skewing* of the lenses is actually performing the *block exchange* operation as we observed.

3.2 Dynamic Switch Elements in a 2-D Array

A network is called a *full access* network if there is a path between any input channel and any output channel. It is called a *rearrangeable* network if it can perform all possible connections (permutations) between all input channels and output channels by rearranging some of its existing connections [10]. By placing $N/2$ switch elements which perform dynamic bypass or exchange functions on the neighboring



(a)



(b)

Figure 3.3: Transformations between the shuffles. (a) shows the block exchange applied before $\sigma_{s,2}$, (b) shows the element exchange applied after $\sigma_{s,2}$.

pairs of channels at each stage of the N ($N = 2^m$) channel 2-D (planar) Omega network ($\Omega_{2\times2}$), the network can become full access, rearrangeable or nonblocking after a sufficient number of stages [3]. These 2 inputs/2 outputs switch elements are called *2×2 bypass/exchange switch elements.* When the switch elements are capable of broadcasting, the network possesses a broadcasting capability. Here, we consider networks utilizing 2 × 2 bypass/exchange switch elements combined with fixed interconnections (perhaps implemented optically). Figure 3.4 is an example of a 3 stage 2-D (planar) Omega network with 8 channels.

The information reaching the bypass/exchange switch on a channel whose index is

$$(b_{m-1}b_{m-2}\ldots b_1b_0)$$

can reach

$$(b_{m-1}b_{m-2}\ldots b_1\bar{b}_0)$$

or itself (depending on the setting of the switch element) before the next 1-D perfect shuffle takes place. Each of the 2 × 2 bypass/exchange switch elements accounts for two possible states of the network. Because there are a total of $m\frac{N}{2}$ switch elements in an m-stage 2-D Omega network having N channels, the network has $2^{m\frac{N}{2}}$ states in total.

Many design variations of 3-D Omega networks are possible. Figure 3.5 shows a schematic representation of two examples having 2-D shuffle interconnections between stages and switch elements. Figure 3.5(a) shows a network using 2-D separable shuffles and switch elements having 4 inputs and 4 outputs. The switch element itself is shown in Fig. 3.6(a). If 2 × 2 bypass/exchange switch elements are used (Fig. 3.5(b) shown with 2-D folded shuffles), we must choose the orientation of the switch elements, i.e. they can be used to switch the horizontal neighbors as shown or can be arranged vertically. These networks having different structures have different properties and applications. Later we show that the network as shown in Fig. 3.5(a) is *full access*, and the one in Fig. 3.5(b) is not *full access* unless more stages are built in the network.

In the next sections, we study the characteristics of different switch element designs and the resulting properties of 3-D Omega networks.

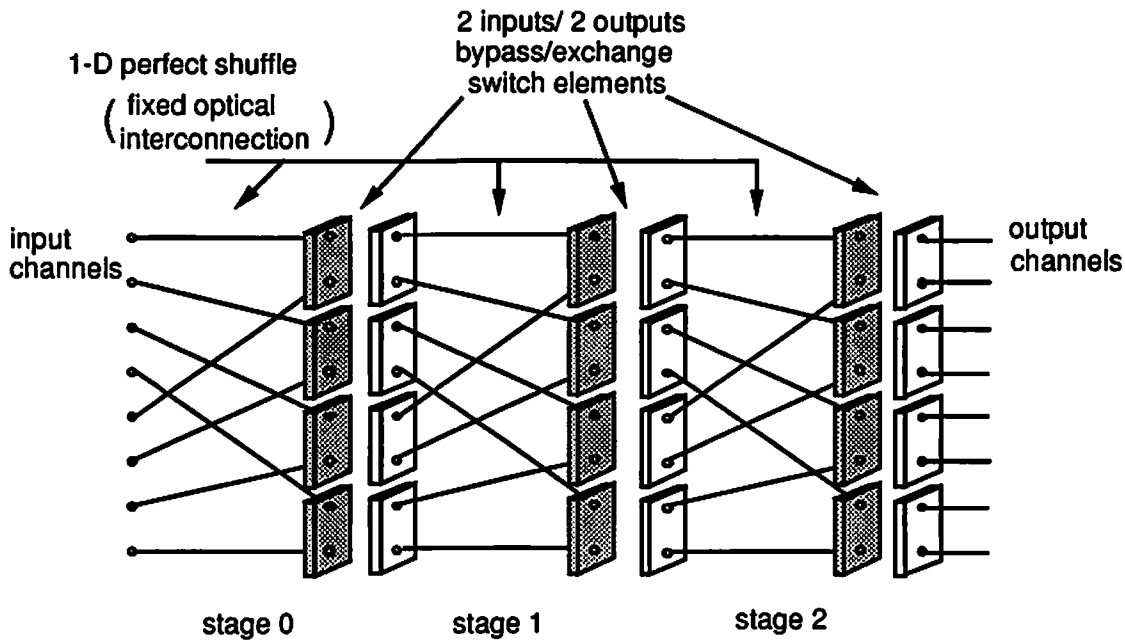


Figure 3.4: 2-D Omega network.

3.2.1 2×2 Switch Elements

Adopting the same notation used in section 3.1, each of the $N = 2^m$ (m is even) channels in the 2-D plane has an index

$$(p, q) = (b_{m-1}b_{m-2} \ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2} \ldots b_1 b_0).$$

When 2×2 switch elements are arranged horizontally in the plane, we bypass or exchange the information from the two channels whose indices differ in b_0. Thus, the information from

$$(b_{m-1}b_{m-2} \ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2} \ldots b_1 b_0)$$

can go to

$$(b_{m-1}b_{m-2} \ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2} \ldots b_1 \bar{b}_0)$$

or itself while passing through the switch element.

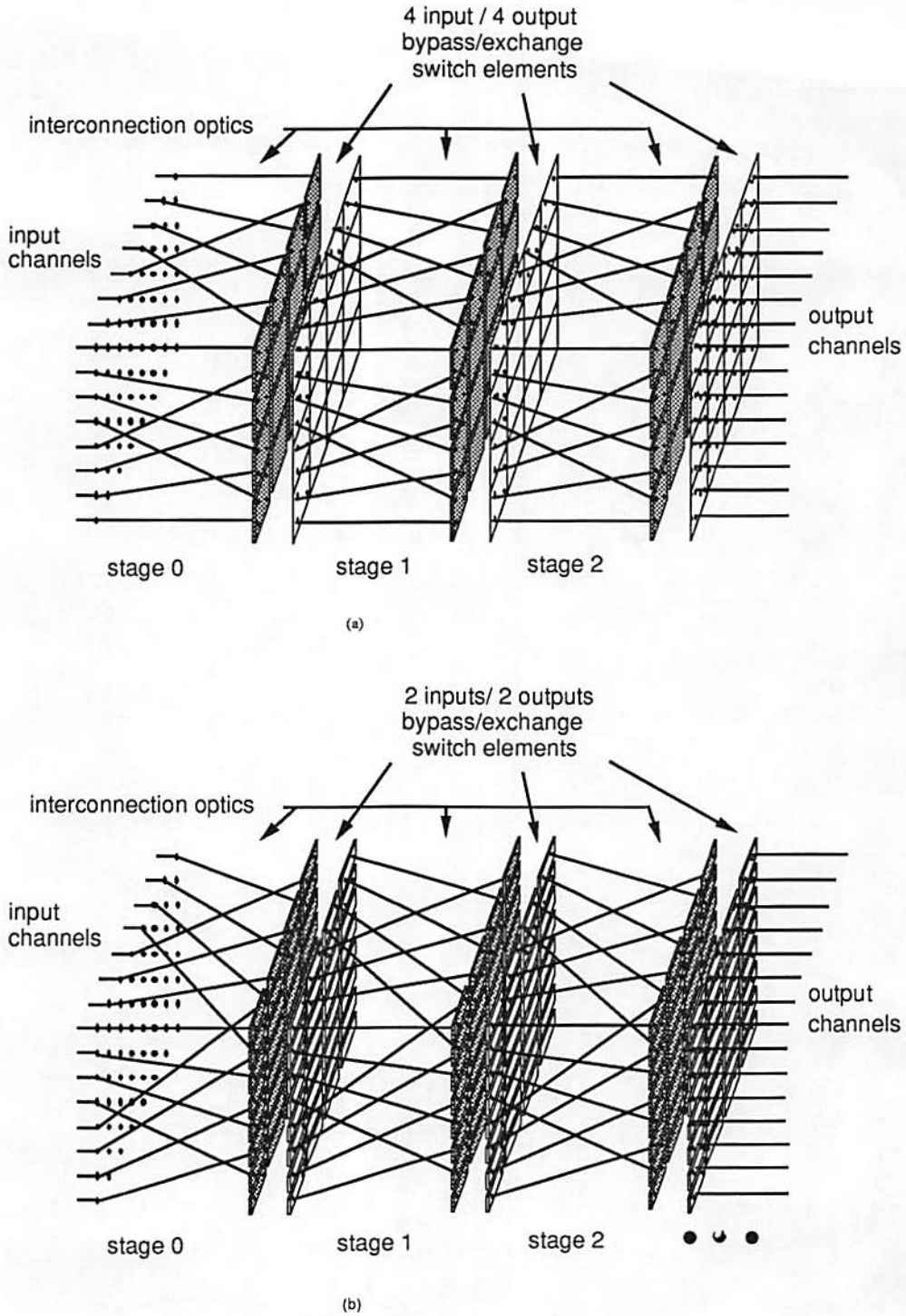


Figure 3.5: Two possible 3-D Omega networks. (a) using 2-D separable shuffles combined with 4×4 switch elements; (b) using 2-D folded shuffles combined with 2×2 switch elements.

The resulting state is denoted as

$$(b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1\ *),$$

where $*$ indicates that the bit is inverted when the switch is in exchange state, and the bit remains unchanged when the switch is in bypass state. Furthermore, the operation of the horizontal 2×2 switch elements on a 2-D plane is defined as a function $S_{2\times2,h}$, then,

$$\begin{aligned} S_{2\times2,h}(A) &= S_{2\times2,h}([\ a_{b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1b_0}\ ]) \\ &= ([\ a_{b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1\ *}\ ]). \end{aligned} \tag{3.6}$$

When 2×2 switch elements are arranged vertically in the plane, we bypass or exchange the information from the two channels whose indices differ in $b_{\frac{m}{2}}$. Thus, the information from

$$(b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1b_0)$$

can go to

$$(b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}+1}\bar{b}_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1b_0)$$

or itself while passing through the switch element.

Similarly, the operation of the vertical 2×2 switch elements on a 2-D plane is defined as a function $S_{2\times2,v}$,

$$\begin{aligned} S_{2\times2,v}(A) &= S_{2\times2,v}([\ a_{b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1b_0}\ ]) \\ &= ([\ a_{b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}+1}\ *,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1b_0}\ ]). \end{aligned} \tag{3.7}$$

3.2.2 4×4 Switch Elements

The most general 4 inputs/4 outputs switch element (Fig. 3.6(a)) have an internal nonblocking crossbar, so that 4! one-to-one permutation states are possible. A simpler 4×4 switch element capable of exchanging information with its three neighbors can be formed by four 2×2 switch elements with one pair lying horizontally,

the other pair standing vertically and a direct connection between the two pairs as shown in Fig. 3.6(b). We call this simpler system a *reduced-state* 4×4 *switch element.* In the reduced-state switch element, the input channel having index

$$(b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1b_0)$$

can communicate with the output channels having indices

$$(b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1\bar{b}_0)$$

$$(b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}\bar{b}_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1b_0)$$

$$(b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}\bar{b}_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1\bar{b}_0)$$

and itself by passing through one horizontal and one vertical 2×2 switch element.

Clearly,

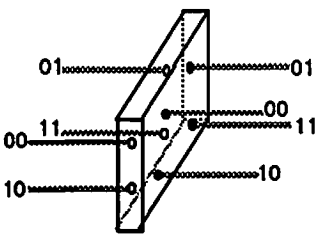
$$\begin{aligned} S_{4\times4}(A) &= S_{4\times4}([\ a_{b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1b_0}\ ]) \\ &= ([\ a_{b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}\ *,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1\ *}\ ]). \end{aligned} \tag{3.8}$$

Each reduced-state 4×4 switch element has 2^4 possible states instead of the 4! states achievable with a one-to-one permutation crossbar switch element.

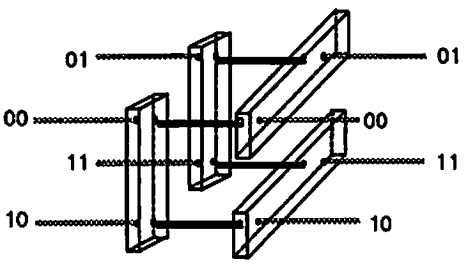
Optical or optoelectronically implementing a reduced-state 4×4 switch element is far easier than the crossbar 4×4 switch element [16]. So we concentrate here on the reduced-state element.

3.3 3-D Omega Networks

The best way to understand the permutation capabilities of various MINs is to start from *graph theory* because the MINs can be easily modeled as graphs by considering the switch elements as nodes and the fixed interconnection links as edges. Two MINs are called *topologically identical* if and only if their graphic models are *identical.* They have exactly the same permutation capability without any rearrangement on the input or output channels. If changing the node or link labelling of one graph can make it *identical* to another graph, then these two graphs are *isomorphic.* Two MINs



(a)



(b)

Figure 3.6: Two types of 4 × 4 switch element; (a) general, (b) reduced-state.

are *topologically equivalent* if and only if their graphical models are *isomorphic.* In other words, the two networks become identical only after certain rearrangement on the input or the output channels. The graph isomorphism is, in general, hard to prove. Agrawal [1] analyzed a class of MINs and provided a theorem to prove their topological equivalence. The MINs he analyzed are constructed in the way that the link patterns at each stage enable the 2×2 switch elements to satisfy the "buddy condition", which means the output of any two switch elements at the i^{th} stage are connected as inputs to only two switch elements at the $(i+1)^{th}$ stage. Feng [60] proved the topological equivalence of a set of full access MINs based on their stage-to-stage relation.

Definition 3.4 *Let a network having $N = 2^m$ channels be described by the mappings β_i which map the nodes in the i^{th} stage to the nodes in the $(i+1)^{th}$ stage, where $0 \leq i \leq m$. Let a set of β_i's describe network N_1 , and a set of $\tilde{\beta}_i$'s describe network N_2. Also define a sequence of composed operations as executed from left to right as earlier. If there is a set of permutation rules γ_i defined on the nodes in the i^{th} stage, such that*

$$\beta_i \circ \gamma_{i+1}(\textit{nodes at } i^{th} \textit{ stage}) = \gamma_i \circ \tilde{\beta}_i(\textit{nodes at } i^{th} \textit{ stage}) \quad \textit{for all } i,$$

then networks N_1 and N_2 are topologically equivalent. If all the γ_i's are the identity mapping for all i, then networks N_1 and N_2 are topologically identical.

Because the shuffling operations do not generate link patterns meeting the "buddy condition", we intend to follow Feng's approach here. Because the 3-D Omega networks are constructed by utilizing either identical cascaded stages of 2-D folded or 2-D separable shuffles (denoted as Ω_f and Ω_s, respectively), we just need to find the mapping rule for one stage if the network is full access. Therefore, determining whether 3-D Omega networks are topologically equivalent or identical consists of two steps:

1. Showing that the networks are full access.
2. Showing that γ_i exists which satisfies Definition 3.4 for a single stage of the two networks.

The full accessibility of a network can be stated in terms of the indices used earlier. That is, if the index of the input channel

$$(b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1b_0)$$

can be turned into

$$\underbrace{(**\ldots\ **,\ **\ldots**)}_{m\text{ bits}}$$

at the output of the network, then the network is full access.

To find the mapping from one network to another, we model the networks according to the following rules:

1. A node is assigned to each switch element.
2. Only the initial inputs and the final outputs are assigned to a node.
3. Nodes of two stages are connected by directed edges.

Here we examine the network properties of Ω_f and Ω_s based on previously described switch element arrangements. In all the networks, we assume there are $N = 2^m$ (m is even) channels.

3.3.1 3-D Omega Networks with 2×2 Switch Elements

Since there are $N = 2^m$ (m is even) channels arranged in a $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ array, 2^{m-1} switch elements are used at each stage.

3.3.1.1 2×2 Switch Elements and the 2-D Folded Shuffle

There are two ways of arranging the 2×2 switch elements in the plane, horizontally or vertically. When the 2×2 switch elements are arranged horizontally in the plane, the network with m stages, denoted as $\Omega_{f,2\times2,h}$, is identical to the traditional m stage 2-D Omega network based on 1-D perfect shuffles and 2×2 switch elements ($\Omega_{2\times2}$).

When the 2×2 switch elements are arranged vertically in the plane, the m stage network, denoted as $\Omega_{f,2\times2,v}$, is topologically equivalent to $\Omega_{f,2\times2,h}$, and hence, $\Omega_{2\times2}$.

This can be proved directly from the definition of *topological equivalence.* We define the transpose of a 2-D array A as $T(A) = A^T$.

Definition 3.5 *Transpose (T)*

The transpose operation exchange the row and the column indices.

$$\begin{aligned} T(A) &= T([\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,]) \\ &= [\, a_{b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0,\ b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}} \,] \\ &= A^T \end{aligned} \tag{3.9}$$

Obviously,

$$T \circ T(A) = T(A^T) = A.$$

Theorem 3.1 *The m stage $\Omega_{f,2\times2,v}$ network and the m stage $\Omega_{f,2\times2,h}$ network are topologically equivalent.*

proof :

$$\begin{aligned} T(A) &= T([\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,]) \\ &= [\, a_{b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0,\ b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}} \,] && (3.10) \\ T \circ \sigma_{f,2}(A) &= \sigma_{f,2}([\, a_{b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0,\ b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}} \,]) \\ &= [\, a_{b_{\frac{m}{2}-2}\ldots b_0 b_{m-1},\ b_{m-2}\ldots b_{\frac{m}{2}-1}} \,] && (3.11) \\ T \circ \sigma_{f,2} \circ S_{2\times2,v}(A) &= S_{2\times2,v}([\, a_{b_{\frac{m}{2}-2}\ldots b_0 b_{m-1},\ b_{m-2}\ldots b_{\frac{m}{2}-1}} \,]) \\ &= [\, a_{b_{\frac{m}{2}-2}\ldots b_0\ *,\ b_{m-2}\ldots b_{\frac{m}{2}-1}} \,] && (3.12) \\ T \circ \sigma_{f,2} \circ S_{2\times2,v} \circ T(A) &= T([\, a_{b_{\frac{m}{2}-2}\ldots b_0\ *,\ b_{m-2}\ldots b_{\frac{m}{2}-1}} \,]) \\ &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}\ldots b_0\ *} \,]. && (3.13) \end{aligned}$$

On the other hand,

$$\begin{aligned} \sigma_{f,2} \circ S_{2\times2,h}(A) &= \sigma_{f,2} \circ S_{2\times2,h}([\, a_{b_{m-1}b_{m-2}\ldots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\ b_1 b_0} \,]) \\ &= S_{2\times2,h}([\, a_{b_{m-2}\ldots\ b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}\ldots\ b_1 b_0\ b_{m-1}} \,]) \\ &= [\, a_{b_{m-2}\ldots\ b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}\ldots\ b_1 b_0\ *} \,]. \end{aligned} \tag{3.14}$$

From Eqs. (3.13) and (3.14), we get

$$T \circ \sigma_{f,2} \circ S_{2\times 2,v} \circ T(A) \;=\; \sigma_{f,2} \circ S_{2\times 2,h}(A). \tag{3.15}$$

Therefore, cascading m stages of both sides of Eq.(3.15), we get

$$(T \circ \sigma_{f,2} \circ S_{2\times 2,v} \circ T)^m(A) \;=\; (\sigma_{f,2} \circ S_{2\times 2,h})^m(A), \tag{3.16}$$

$$T \circ (\sigma_{f,2} \circ S_{2\times 2,v})^m \circ T(A) \;=\; (\sigma_{f,2} \circ S_{2\times 2,h})^m(A). \tag{3.17}$$

Eq. (3.17) gives the pre-permutation of the input channels and post-permutation of the output channels of $\Omega_{f,2\times 2,v}$ necessary to produce the same permutation as $\Omega_{f,2\times 2,h}$. Hence, they are topologically equivalent. □

An example for 16 channels is illustrated in Fig. 3.7. The channels shown in the figure are in pairs, which indicates the arrangement of the 2×2 switch elements. In Fig. 3.7(a) we show that $\Omega_{2\times 2}$ and $\Omega_{f,2\times 2,h}$ are identical. Figure 3.7(b) depicts the topological equivalence of $\Omega_{f,2\times 2,v}$ and $\Omega_{f,2\times 2,h}$. The lightly shaded boxes indicate the single stage permutations of these two networks.

3.3.1.2 2×2 Switch Elements and the 2-D Separable Shuffle

If only horizontal switch elements are used with a 2-D separable shuffle, the resulting $\Omega_{s,2\times 2,h}$ network is not full access.

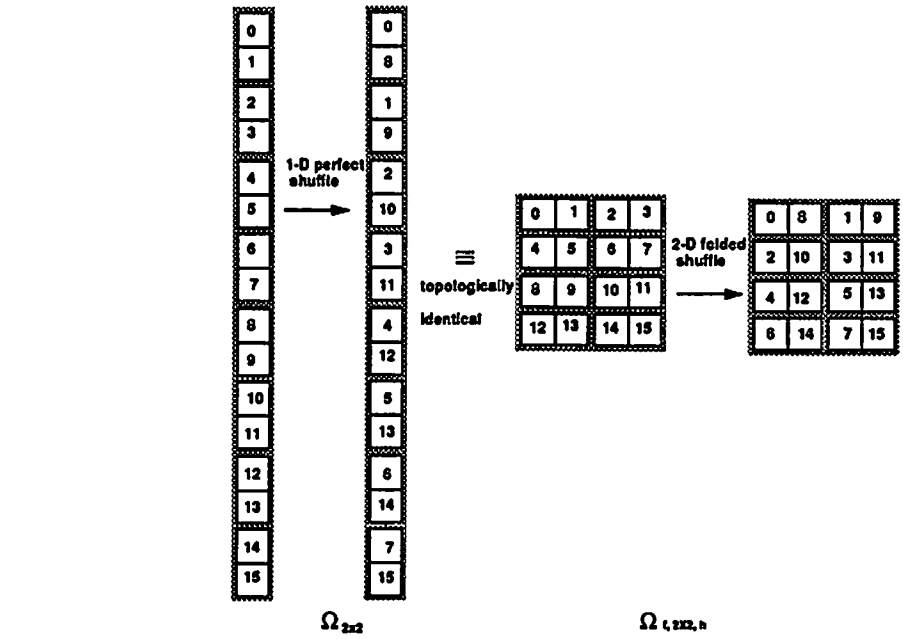
Theorem 3.2 *The m stage $\Omega_{s,2\times 2,h}$ is not a full access network.*

proof :

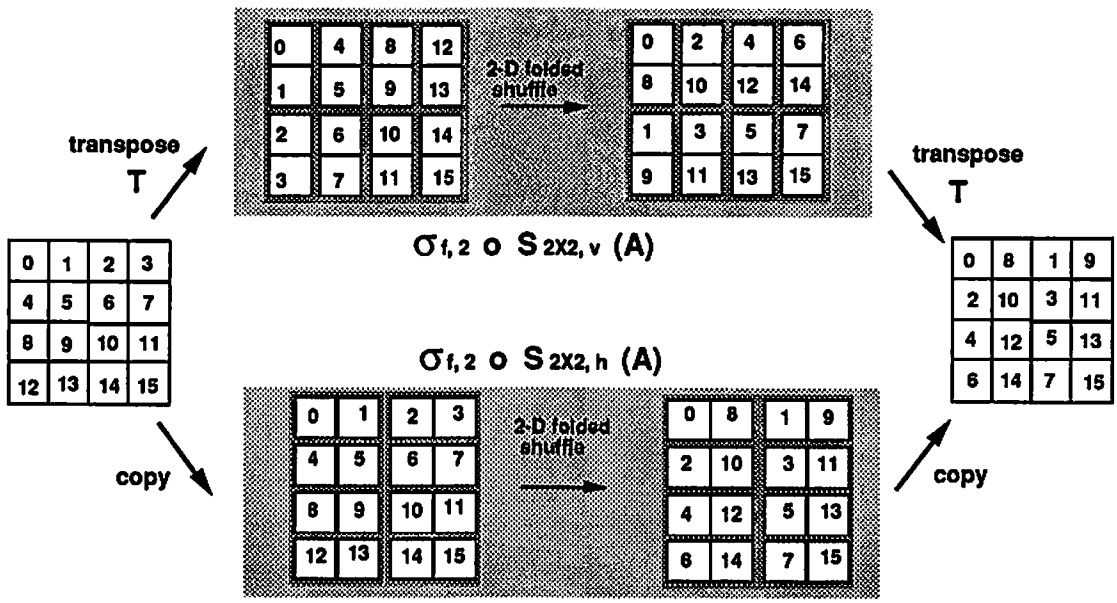
$$\begin{aligned}
\sigma_{s,2} \circ S_{2\times 2,h}(A) &= \sigma_{s,2} \circ S_{2\times 2,h}([\, a_{b_{m-1}b_{m-2}\ldots\, b_{\frac{m}{2}+1}b_{\frac{m}{2}},\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\, b_1b_0} \,]) \\
&= ([\, a_{b_{m-2}\ldots\, b_{\frac{m}{2}+1}b_{\frac{m}{2}}b_{m-1},\; b_{\frac{m}{2}-2}\ldots\, b_1b_0\, *} \,]), \qquad (3.18) \\
(\sigma_{s,2} \circ S_{2\times 2,h})^{\frac{m}{2}}(A) &= (\sigma_{s,2} \circ S_{2\times 2,h})^{\frac{m}{2}}([\, a_{b_{m-1}b_{m-2}\ldots\, b_{\frac{m}{2}+1}b_{\frac{m}{2}},\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\, b_1b_0} \,]) \\
&= ([\, a_{b_{m-1}b_{m-2}\ldots\, b_{\frac{m}{2}},\; \underbrace{**\ldots\, **}_{\frac{m}{2}\text{ bits}}} \,]). \qquad (3.19)
\end{aligned}$$

Also,

$$(\sigma_{s,2} \circ S_{2\times 2,h})^m(A) \;=\; (\sigma_{s,2} \circ S_{2\times 2,h})^m([\, a_{b_{m-1}b_{m-2}\ldots\, b_{\frac{m}{2}+1}b_{\frac{m}{2}},\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots\, b_1b_0} \,])$$



(a)



(b)

Figure 3.7: (a) $\Omega_{f,2\times2,h}$ and $\Omega_{2\times2}$ are identical, (b) $\Omega_{f,2\times2,h}$ and $\Omega_{f,2\times2,v}$ are equivalent.

$$= ([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}}}, \underbrace{**\ldots\ **}_{\frac{m}{2}\text{ bits}}\,]) \qquad \square \qquad (3.20)$$

The m stage network $\Omega_{s,2\times 2,v}$ behaves similarly. Since the 2-D separable shuffle operation forms a *cyclic group* of order $\frac{m}{2}$, neither the m stage $\Omega_{s,2\times 2,h}$ nor $\Omega_{s,2\times 2,v}$ networks are full access. They are, in general, of no interest to any applications and not related to the existing MINs in any sense.

However, we can form a full access 3-D Omega network based on 2-D separable shuffles, using a combination of horizontal and vertical 2×2 switch elements.

Theorem 3.3 *The m stage network $\Omega_{s,2\times 2,h/v}$, which is constructed from $\frac{m}{2}$ stages of 2-D separable shuffles interlaced with $\frac{m}{2}$ stages of horizontal switches, followed by another $\frac{m}{2}$ stages of 2-D separable shuffles interlaced with $\frac{m}{2}$ stages of vertical switches, is full access.*

proof :

$$\begin{aligned}(\sigma_{s,2} \circ S_{2\times 2,h})^{\frac{m}{2}}(A) &= (\sigma_{s,2} \circ S_{2\times 2,h})^{\frac{m}{2}}([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}+1}b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots b_1 b_0}\,]) \\ &= ([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}+1}b_{\frac{m}{2}}}, \underbrace{**\ldots\ **}_{\frac{m}{2}\text{ bits}}\,]) \qquad (3.21)\end{aligned}$$

$$(\sigma_{s,2} \circ S_{2\times 2,h})^{\frac{m}{2}} \circ (\sigma_{s,2} \circ S_{2\times 2,v})^{\frac{m}{2}}(A) = ([\, \underbrace{a_{**\ldots\ **}}_{m\text{ bits}}, **\ldots\ **\,]) \qquad (3.22)$$

$\square$

Figure 3.8 is an example for $N = 16$. The indices in the bottom show the indices of the channels reachable by input channel 0000. Note that if the $\frac{m}{2}$ stages of horizontal and the $\frac{m}{2}$ stages of vertical switches are mixed, the full access network full access property is lost.

3.3.2 3-D Omega Networks with Reduced-State 4×4 Switch Elements

Here we consider building Ω_f and Ω_s based on reduced-state 4×4 switch elements (as shown in Fig. 3.6(b)). In this case, there are 2^{m-2} switch elements at each stage.

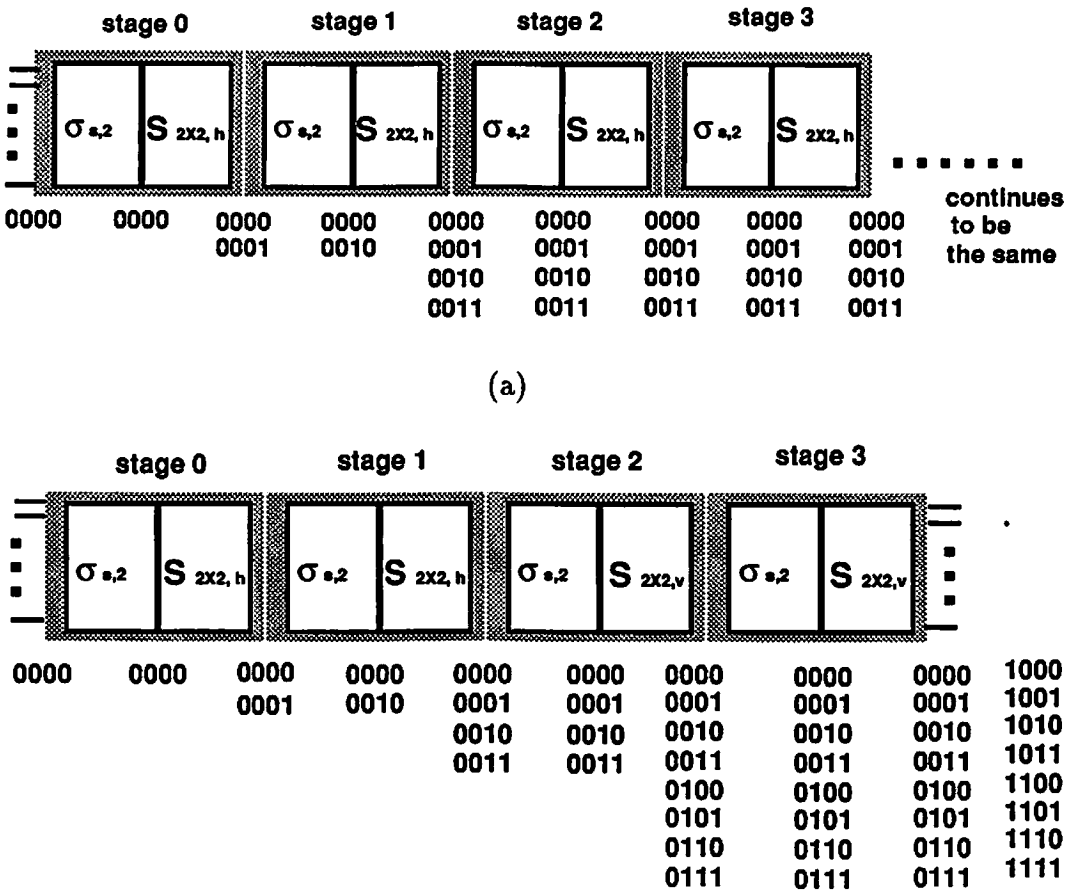


Figure 3.8: Full accessibility of $\Omega_{s,2\times2}$ depends on the orientation of the switch elements at each stage. The indices on bottom show the indices of the channels reachable by input channel 0000. (a) shows for $\Omega_{s,2\times2,h}$; (b) shows for $\Omega_{s,2\times2,h/v}$.

3.3.2.1 Reduced-State 4×4 Switch Elements and the 2-D Folded Shuffle

Theorem 3.4 *An* $\frac{m}{2}$ *stage* $\Omega_{f,4\times4}$ *is full access.*

proof :

$$\begin{aligned} \sigma_{f,2} \circ S_{4\times4}(A) &= \sigma_{f,2} \circ S_{4\times4}([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}+1}b_{\frac{m}{2}},\, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots b_1b_0} \,]) \\ &= ([\, a_{b_{m-2}\cdots b_{\frac{m}{2}+1}b_{\frac{m}{2}}\,*,\, b_{\frac{m}{2}-2}\cdots b_1b_0\,*} \,]). \end{aligned} \tag{3.23}$$

Hence,

$$\begin{aligned} (\sigma_{f,2} \circ S_{4\times4})^{\frac{m}{2}}(A) &= (\sigma_{f,2} \circ S_{4\times4})^{\frac{m}{2}}([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}+1}b_{\frac{m}{2}},\, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots b_1b_0} \,]) \\ &= ([\, a_{\underbrace{**\ldots\ **}_{m \text{ bits}},\, **\ldots**} \,]) \qquad \square \end{aligned} \tag{3.24}$$

3.3.2.2 Reduced-State 4×4 Switch Elements and the 2-D Separable Shuffle

Theorem 3.5 *An* $\frac{m}{2}$ *stage* $\Omega_{s,4\times4}$ *is full access.*

proof :

$$\begin{aligned} \sigma_{s,2} \circ S_{4\times4}(A) &= \sigma_{s,2} \circ S_{4\times4}([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}+1}b_{\frac{m}{2}},\, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots b_1b_0} \,]) \\ &= ([\, a_{b_{m-2}\cdots b_{\frac{m}{2}+1}b_{\frac{m}{2}}\,*,\, b_{\frac{m}{2}-2}\cdots b_1b_0\,*} \,]). \end{aligned} \tag{3.25}$$

Hence,

$$\begin{aligned} (\sigma_{s,2} \circ S_{4\times4})^{\frac{m}{2}}(A) &= (\sigma_{s,2} \circ S_{4\times4})^{\frac{m}{2}}([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}+1}b_{\frac{m}{2}},\, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots b_1b_0} \,]) \\ &= ([\, a_{\underbrace{**\ldots\ **}_{m \text{ bits}},\, **\ldots**} \,]) \qquad \square \end{aligned} \tag{3.26}$$

Therefore, a reduced-state 4×4 switch element can replace the more general 4×4 crossbar switch element and still retain the full accessibility of $\frac{m}{2}$ stage 3-D Omega networks.

3.3.2.3 Relationship Between $\Omega_{f,4\times4}$ and $\Omega_{s,4\times4}$

Adopting the same graph models established earlier, we prove that $\Omega_{f,4\times4}$ and $\Omega_{s,4\times4}$ are topologically identical. Consider the operations at a single stage for both networks. The one-stage operation which maps the nodes in the i^{th} stage to the nodes in the $(i+1)^{th}$ stage in $\Omega_{f,4\times4}$ is

$$\begin{aligned}\beta_{f,4\times4}(\textit{nodes at } 0^{th} \textit{ stage}) &= \sigma_{f,2} \circ S_{4\times4}([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}+1}\,*,\, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots b_1\,*}\,]) \\ &= S_{4\times4}([\, a_{b_{m-2}\cdots b_{\frac{m}{2}+1}\,*\,b_{\frac{m}{2}-1},\, b_{\frac{m}{2}-2}\cdots b_1\,*\,b_{m-1}}\,]) \\ &= ([\, a_{b_{m-2}\cdots b_{\frac{m}{2}+1}\,*\,*,\, b_{\frac{m}{2}-2}\cdots b_1\,*\,*}\,]). \end{aligned} \tag{3.27}$$

Similarly, the one-stage operation in $\Omega_{s,4\times4}$ is

$$\begin{aligned}\beta_{s,4\times4}(\textit{nodes at } 0^{th} \textit{ stage}) &= \sigma_{s,2} \circ S_{4\times4}([\, a_{b_{m-1}b_{m-2}\cdots b_{\frac{m}{2}+1}\,*,\, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots b_1\,*}\,]) \\ &= S_{4\times4}([\, a_{b_{m-2}\cdots b_{\frac{m}{2}+1}\,*\,b_{m-1},\, b_{\frac{m}{2}-2}\cdots b_1\,*\,b_{\frac{m}{2}-1}}\,]) \\ &= ([\, a_{b_{m-2}\cdots b_{\frac{m}{2}+1}\,*\,*\,,\, b_{\frac{m}{2}-2}\cdots b_1\,*\,*}\,]), \end{aligned} \tag{3.28}$$

which is identical to that in $\Omega_{f,4\times4}$. Hence, the mapping γ_i is the identity

$$\gamma_i(\textit{nodes at } i^{th} \textit{ stage}) = (\textit{nodes at } i^{th} \textit{ stage}) \qquad 1 \le i \le m,$$

and we have

$$\beta_{f,4\times4} \circ \gamma_{i+1}(\textit{nodes at } i^{th} \textit{ stage}) = \gamma_i \circ \beta_{s,4\times4}(\textit{nodes at } i^{th} \textit{ stage}) \tag{3.29}$$

proving that $\Omega_{f,4\times4}$ and $\Omega_{s,4\times4}$ are topologically identical. □

Since the 2-D folded shuffle operation on a 2-D array of channels is identical to the 1-D perfect shuffle operation on a 1-D array of channels, the 2-D Omega network utilizing reduced-state 4×4 switch elements ($\Omega_{4\times4}$) is identical to the 3-D $\Omega_{f,4\times4}$ network, hence, it is also topologically identical to the 3-D $\Omega_{s,4\times4}$ network. However, we point out that when a reduced-state 4×4 switch element is physically arranged on the 1-D array, it no longer exchanges the information among four consecutive channels (Fig. 3.9). If we want the reduced-state 4×4 switch elements to operate on adjacent channels in the 1-D array, we actually switch among the channels whose

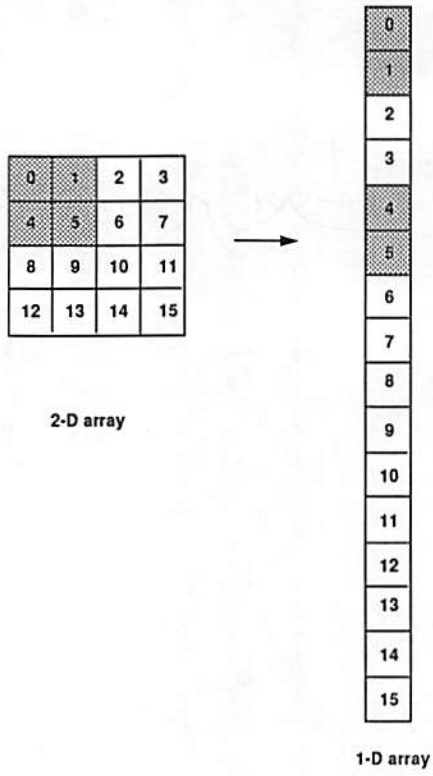


Figure 3.9: The reduced-state 4×4 switch element takes consecutive inputs from a 2-D array and non-consecutive inputs from a 1-D array.

indices differ in the least significant two bits. This operation can be defined as

$$\tilde{S}_{4\times4}([\ a_{b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1b_0}\ ])$$

$$= ([\ a_{b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_2\ *\ *}\ ]).$$

Thus,

$$\sigma_{f,2}\ \circ\ \tilde{S}_{4\times4}([\ a_{b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1b_0}\ ])$$

$$= ([\ a_{b_{m-2}\cdots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}}b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}\cdots\ b_1\ *\ *}\ ]).$$

And,

$$(\sigma_{f,2}\ \circ\ \tilde{S}_{4\times4})^2([\ a_{b_{m-1}b_{m-2}\cdots\ b_{\frac{m}{2}+1}b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\cdots\ b_1b_0}\ ])$$

$$= ([\ a_{b_{m-3}\cdots\ b_{\frac{m}{2}}b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}\cdots\ b_1\ *\ *\ *}\ ]).$$

Obviously, it becomes full access after $(m-1)$ stages.

3.4 Summary

The properties of the various 3-D Omega networks and their relationships with 2-D Omega networks having 2×2 switch elements or reduced-state 4×4 switch elements are summarized in Table . Note that topologically identical is denoted by "$\equiv$" and topologically equivalent is denoted by "$\cong$" in the table. The results show that the 3-D Omega network using 2-D folded shuffles combined with 2×2 bypass/exchange switch element arrays arranged horizontally in the 2-D plane ($\Omega_{f,2\times2,h}$) is identical to 2-D Omega network using 1-D perfect shuffles combined with 2×2 bypass/exchange switch element arrays ($\Omega_{2\times2}$). When the 2×2 bypass/exchange switch element arrays are arranged vertically in the plane ($\Omega_{f,2\times2,v}$), the network is topologically equivalent to $\Omega_{2\times2}$. Networks using 2-D separable shuffles combined with 2×2 bypass/exchange switch element arrays are full access only for a particular arrangement of the switch element arrays ($\Omega_{s,2\times2,h/v}$). When reduced-state 4×4 switch elements arrays are used, 3-D Omega networks with 2-D folded shuffles ($\Omega_{f,4\times4}$) and 2-D separable shuffles ($\Omega_{s,4\times4}$) are identical.

switch dimension	network	shuffle type	switch orientation	total no. of stages needed for full access	relationship to 2-D networks
2×2	$\Omega_{f,2\times2,h}$	folded	horizontal	m	$\equiv \Omega_{2\times2}$
	$\Omega_{f,2\times2,v}$	folded	vertical	m	$\cong \Omega_{2\times2}$
	$\Omega_{s,2\times2,h}$	separable	horizontal	not applicable	not applicable
	$\Omega_{s,2\times2,v}$	separable	vertical	not applicable	not applicable
	$\Omega_{s,2\times2,h/v}$	separable	horizontal vertical	m	not applicable
4×4	$\Omega_{f,4\times4}$	folded		$\frac{m}{2}$	$\equiv \Omega_{4\times4}$
	$\Omega_{s,4\times4}$	separable		$\frac{m}{2}$	$\equiv \Omega_{4\times4}$

Table 3.1: Summary of various 3-D Omega networks.

Chapter 4

One-Copy Algorithm for 2-D Shuffles

In the 1-D perfect shuffle (or two-shuffle), as used in the 2-D Omega network, the N channels (N is even) are divided into two halves, then are interleaved perfectly as shown in Fig. 1.1. If N is divisible by r, similar link patterns called the 1-D r-shuffle can be obtained by interleaving the equally divided r parts of the N channels. We call the network which uses 1-D r-shuffle link patterns *2-D Omega-r network* (the 2-D Omega network is actually a 2-D Omega-2 network for the special case when $r = 2$). Cascading enough stages of the 1-D r-shuffle combined with $r \times r$ switch elements (with r inputs/outputs), can make the network *full access.*

As introduced earlier, there are two types of shuffling operations on the 2-D array of channels for a given r. The *2-D separable r-shuffle* contains independent 1-D r-shuffles on the rows and the columns. In the *2-D folded r-shuffle*, the rows and the columns of the input channels are obtained by folding a 1-D vector of input channels. A 1-D r-shuffled output can be obtained after unfolding the output plane from a 2-D folded r-shuffle. For simplicity, we will drop the r in our discussion when $r = 2$.

Extensive research has been done for optical shuffles, including the use of simple lenses, prisms [32], diffraction gratings [31], and holograms [4, 5, 33], etc.

For the optical 2-D shuffles, Lohmann [32] proposed the implementation of the 2-D separable shuffle using lenses and prisms. Stirk *et al.* [47] used four simple lenses to experimentally achieve an optical 2-D folded shuffle. Sawchuk and Glaser [43]

accomplished both 2-D folded and 2-D separable optical shuffles by a similar approach. In both of the latter two experiments, four copies of the magnified and masked input plane are generated (Hence, we call it *four-copy algorithm* from now on.). By shifting and superimposing these four copies properly, the shuffled version of the input plane will appear in the center as a result of the overlap. In this method, only 25% light efficiency can be obtained at most because only one quadrant of the shifted copies is used in the output.

In this chapter, a novel approach for performing 2-D shuffles is presented. This so called *one-copy algorithm* can achieve 100% light efficiency in principle. The related properties of this algorithm for 2-D r-shuffles and their inverse operations are discussed. Although similar approaches have been used in [32, 49] implicitly, this chapter provides a thorough discussion of the one-copy method which were not presented in previous work. In section 4.1, the mathematical tools for this study based on $r = 2$ are developed. The four-copy algorithm is examined first in detail as an explanation of the procedure used in [47], followed by one-copy algorithms for 2-D perfect shuffles. The extension of the one-copy algorithm for 2-D r-shuffles, where r is a prime number, is presented in section 4.2. The application of the same algorithm for 2-D r^k-shuffle is illustrated in section 4.3. In section 4.4, we discuss the possibilities of applying the one-copy algorithm for inverse shuffle operations.

4.1 Notations and Definitions

Sawchuk and Glaser [43] describe the *"four-copy algorithm"* mathematically in terms of convolution from the geometrical point of view. Another way to describe the algorithm is to enumerate each channel and apply *digit permutation* on its binary representation. Here we describe each step in the *"four-copy algorithm"* mathematically and develop a simplified algorithm called the *"one-copy algorithm"*. In order to do so, we first present some notations and define certain fundamental operations.

Similar as defined earlier, let there be $N = 2^m$ channels arranged in a 2-D plane (m is even) considered as a matrix A, with each channel an entry $a_{p,q}$, where p is the row index and q is the column index. The subscript 2's are used in some operations to show that the binary number system is used for the indices. Later, we generalize this notation to higher indices.

Definition 4.1 *Magnification (M_2)*

To magnify a $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ matrix A into an $2^{\frac{m}{2}+1} \times 2^{\frac{m}{2}+1}$ matrix $\tilde{A}$ is to place the p^{th} row, q^{th} column entry in A into the $\tilde{p}^{th}$ row, $\tilde{q}^{th}$ column entry ($\tilde{p} = 2p$ and $\tilde{q} = 2q$) in $\tilde{A}$ ($\tilde{A} = [\tilde{a}_{\tilde{p},\tilde{q}}]$) and set the other entries to zero. Thus,

$$\begin{aligned} \tilde{A} &= M_2(A) \\ &= M_2([\ a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}\ ]) \\ &= [\ \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\ ] \quad x, y \in \{0,1\} \end{aligned} \tag{4.1}$$

where

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } x = y = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.2 *Shift ($S_{i,j,2}$)*

The shift function $S_{i,j,2}(A)$ of matrix A with dimension $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ corresponds to the circular shift of all the entries in matrix A i rows downward and j columns towards the right.

$$\begin{aligned} S_{i,j,2}(A) &= S_{i,j,2}([\ a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}\ ]) \\ &= [\ a_{[(b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}})+i]\ mod\ 2^{\frac{m}{2}},\ [(b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0)+j]\ mod\ 2^{\frac{m}{2}}}\ ] \end{aligned} \tag{4.2}$$

Note: a negative shift implies a shift in the opposite direction.

Example 4.1 *Suppose $\tilde{A} = M_2(A)$, i and j are 0 or 1, then,*

$$S_{i,j,2}(\tilde{A}) = [\ \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\ ] \quad x, y \in \{0,1\}$$

where

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \textit{if } x = i, \\ & \quad y = j \\ 0 & \textit{otherwise.} \end{cases}$$

Example 4.2 *Suppose* $\tilde{A} = M_2(A)$,

$$S_{b_{\frac{m}{2}-1},b_{m-1},2}(\tilde{A}) \;=\; [\; \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\;] \qquad x,y \in \{0,1\}$$

where

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } x = b_{\frac{m}{2}-1}, \\ & \quad y = b_{m-1} \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.3 *Matrix Breaking*

Let us denote $S_{i,j,2}(\tilde{A})$ by $\tilde{A}_{i,j,2}$, where $\tilde{A}_{i,j,2}$ has size $2^{\frac{m}{2}+1} \times 2^{\frac{m}{2}+1}$, the same as $\tilde{A}$. Each shifted matrix $\tilde{A}_{i,j,2}$ can be partitioned into four equal sized submatrices (of size $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$), denoted as $\alpha_{i,j,2}^{k,l}$, where i, j, k, and l are 0 or 1.

$$\tilde{A}_{i,j,2} \;=\; \begin{bmatrix} \alpha_{i,j,2}^{0,0} & \alpha_{i,j,2}^{0,1} \\ \alpha_{i,j,2}^{1,0} & \alpha_{i,j,2}^{1,1} \end{bmatrix} \tag{4.3}$$

Thus,

$$\alpha_{i,j,2}^{k,l} \;=\; [\; \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\;] \tag{4.4}$$

where,

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } b_{m-1} = k, \\ & \quad b_{\frac{m}{2}-1} = l, \\ & \text{and } x = i, \\ & \quad y = j \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.4 *Nucleus Extraction* ($N_{u,v}$)

The operation $N_{u,v}(C)$ is used to extract the middle most submatrix of size $u \times v$ from matrix C. Suppose C is a matrix of size $2^{\frac{m}{2}+2} \times 2^{\frac{m}{2}+2}$, which comes from magnifying the A matrix four times in both horizontal and vertical dimensions. C

can be represented as

$$C = [\, c_{\, s\, b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\; x\;,\; t\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\; y}\,] \quad s,\ t,\ x \text{ and } y \in \{0,1\}. \quad (4.5)$$

When $u = v = 2^{\frac{m}{2}}$, the middle most submatrix of size $u \times v$ of C becomes

$$N_{u,v}(C) = [\, c_{\, s\, b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\; x\;,\; t\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\; y}\,] \quad s = \bar{b}_{m-1},\ t = \bar{b}_{\frac{m}{2}-1}. \quad (4.6)$$

4.1.1 Algorithms for 2-D Perfect Shuffles

4.1.1.1 2-D Folded Shuffle

1. Four-Copy Algorithm

   With the help of previously developed notations and operations, the four-copy algorithm for the 2-D folded shuffle can be expressed neatly as follows:

   Step 1. magnify-mask: Consider the input data plane as a $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ matrix A. Apply the magnification operation (M_2) on A to get a $2^{\frac{m}{2}+1} \times 2^{\frac{m}{2}+1}$ matrix $\tilde{A}$, such that $\tilde{A} = M_2(A)$.

   Step 2. shift: Four copies of matrix $\tilde{A}$ are made and shifted individually by $S_{0,0,2}$, $S_{0,1,2}$, $S_{1,0,2}$, and $S_{1,1,2}$. Thus, $\tilde{A}_{0,0,2}$, $\tilde{A}_{0,1,2}$, $\tilde{A}_{1,0,2}$, and $\tilde{A}_{1,1,2}$ are obtained.

   Step 3. superimpose: The final step is to superimpose $\tilde{A}_{0,0,2}$, $\tilde{A}_{0,1,2}$, $\tilde{A}_{1,0,2}$, and $\tilde{A}_{1,1,2}$, such that $\alpha^{j,i}_{i,j,2}$, where i and $j \in \{0,1\}$, are added to get the shuffled result. This is achieved by considering the $\tilde{A}_{i,j,2}$ matrices as quadrant submatrices of some $2^{\frac{m}{2}+2} \times 2^{\frac{m}{2}+2}$ matrices C_t, where $1 \leq t \leq 4$. There are two ways of doing so; either we can let

$$C_1 = \begin{bmatrix} \tilde{A}_{0,0,2} & [0] \\ [0] & [0] \end{bmatrix}$$

$$C_2 = \begin{bmatrix} \tilde{A}_{0,1,2} & [0] \\ [0] & [0] \end{bmatrix}$$

$$C_3 = \begin{bmatrix} \tilde{A}_{1,0,2} & [0] \\ [0] & [0] \end{bmatrix}$$

$$C_4 = \begin{bmatrix} \tilde{A}_{1,1,2} & [0] \\ [0] & [0] \end{bmatrix},$$

where the [0] denotes a quadrant of zeros of size $2^{\frac{m}{2}+1} \times 2^{\frac{m}{2}+1}$. Let C_{folded} be the sum of the shifted versions of these four matrices, as expressed by

$$\begin{aligned} & C_{folded} \\ = \; & S_{\frac{3m}{2},\frac{3m}{2},2}(C_1) + S_{\frac{m}{2},\frac{3m}{2},2}(C_2) + S_{\frac{3m}{2},\frac{m}{2},2}(C_3) + S_{\frac{m}{2},\frac{m}{2},2}(C_4) \qquad (4.7) \\ = \; & [\, c_{\,s\, b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\; x\;,\; t\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\; y} \,] \quad s,\ t,\ x \text{ and } y \in \{0,1\}. \end{aligned}$$

Then, the desired result appears as the center $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ matrix of C_{folded}, i.e.,

$$\alpha^{0,0}_{0,0,2} + \alpha^{1,0}_{0,1,2} + \alpha^{0,1}_{1,0,2} + \alpha^{1,1}_{1,1,2} = N_{2^{\frac{m}{2}},2^{\frac{m}{2}}}(C_{folded}). \qquad (4.8)$$

Or we can let

$$C_1 = \begin{bmatrix} [0] & [0] \\ [0] & \tilde{A}_{0,0,2} \end{bmatrix}$$

$$C_2 = \begin{bmatrix} [0] & \tilde{A}_{0,1,2} \\ [0] & [0] \end{bmatrix}$$

$$C_3 = \begin{bmatrix} [0] & [0] \\ \tilde{A}_{1,0,2} & [0] \end{bmatrix}$$

$$C_4 = \begin{bmatrix} \tilde{A}_{1,1,2} & [0] \\ [0] & [0] \end{bmatrix}$$

Now,

$$\begin{aligned} & C_{folded} \\ = \; & S_{\frac{-m}{2},\frac{-m}{2},2}(C_1) + S_{\frac{m}{2},\frac{-m}{2},2}(C_2) + S_{\frac{-m}{2},\frac{m}{2},2}(C_3) + S_{\frac{m}{2},\frac{m}{2},2}(C_4) \qquad (4.9) \\ = \; & [\, c_{\,s\, b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\; x\;,\; t\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\; y} \,] \quad s,\ t,\ x \text{ and } y \in \{0,1\}. \end{aligned}$$

The desired result also appears as a $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ matrix in the center of C_{folded}, i.e.,

$$\alpha^{0,0}_{0,0,2} + \alpha^{1,0}_{0,1,2} + \alpha^{0,1}_{1,0,2} + \alpha^{1,1}_{1,1,2} \;=\; N_{2^{\frac{m}{2}},2^{\frac{m}{2}}}(C_{folded}).$$

This completes our discussion of four-copy algorithm for 2-D folded shuffle.

2. One-Copy Algorithm

Next, we propose a new method of performing the 2-D folded shuffle which utilizes only one copy of the magnified matrix. The first step of magnification is the same as in the four-copy algorithm. However, steps 2 and 3 become:

Step 2. shift: Four different shifting operations should be applied on the four quadrant submatrices of $\tilde{A}$. The shifting operation applied on $\tilde{A}$ is $S_{b_{\frac{m}{2}-1},b_{m-1},2}$, thus,

$$\begin{aligned} S_{b_{\frac{m}{2}-1},b_{m-1},2}(\tilde{A}) &= S_{b_{\frac{m}{2}-1},b_{m-1},2}(\tilde{A}_{0,0,2}) \\ &= S_{b_{\frac{m}{2}-1},b_{m-1},2}\left(\begin{bmatrix} \alpha^{0,0}_{0,0,2} & \alpha^{0,1}_{0,0,2} \\ \alpha^{1,0}_{0,0,2} & \alpha^{1,1}_{0,0,2} \end{bmatrix} \right) \\ &= \begin{bmatrix} \alpha^{0,0}_{b_{\frac{m}{2}-1},b_{m-1},2} & \alpha^{0,1}_{b_{\frac{m}{2}-1},b_{m-1},2} \\ \alpha^{1,0}_{b_{\frac{m}{2}-1},b_{m-1},2} & \alpha^{1,1}_{b_{\frac{m}{2}-1},b_{m-1},2} \end{bmatrix} \end{aligned} \qquad (4.10)$$

Step 3. superimpose: Finally, the $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ shuffled matrix is obtained by adding these four shifted quadrants (submatrices) of the $\tilde{A}_{0,0,2}$. This step is performed by neglecting the first digits of the row and the column indices.

$$\alpha^{k,l}_{b_{\frac{m}{2}-1},b_{m-1},2} \;=\; [\; \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\; x,\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\; y} \;] \qquad (4.11)$$

where,

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}$$

$$= \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } b_{m-1} = k, \\ & b_{\frac{m}{2}-1} = l, \\ & x = b_{\frac{m}{2}-1}, \\ & y = b_{m-1} \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the desired output is

$$\begin{aligned} & \alpha^{0,0}_{0,0,2} + \alpha^{1,0}_{0,1,2} + \alpha^{0,1}_{1,0,2} + \alpha^{1,1}_{1,1,2} \\ &= [\ \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}}\ ] \\ &= [\ a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}}\ ] \\ &= \sigma_{f,2}(A). \end{aligned}$$

We illustrate an example for $N = 16 = 2^4$ in Figs. 4.1 and 4.2. The shifted $\tilde{A}$'s are shown in Fig. 4.1. The numbers shown in the boxes are the decimal index of the channels and the zero elements are shown as black in these figures. The superposed C_i's and extraction of the nucleus portion for the four-copy algorithm is shown in Fig. 4.2 (a). The operation of the one-copy algorithm is depicted in Fig. 4.2 (b).

4.1.1.2 2-D Separable Shuffle

1. Four-Copy Algorithm

   Although the four-copy algorithm described by Stirk *et al* [47] was used for the 2-D folded shuffle only, we can use a similar approach to get a 2-D separable shuffle. The only change is made at the superposition step. Here, we want to add up $\alpha^{i,j}_{i,j,2}$, where i, and $j \in \{0,1\}$, to get the 2-D separable shuffled output. The formation of C_t, where $1 \leq t \leq 4$, remains the same as in the 2-D folded shuffle.

   In the first method where $\tilde{A}_{i,j,2}$'s are used as the upper left quadrant of C_t,

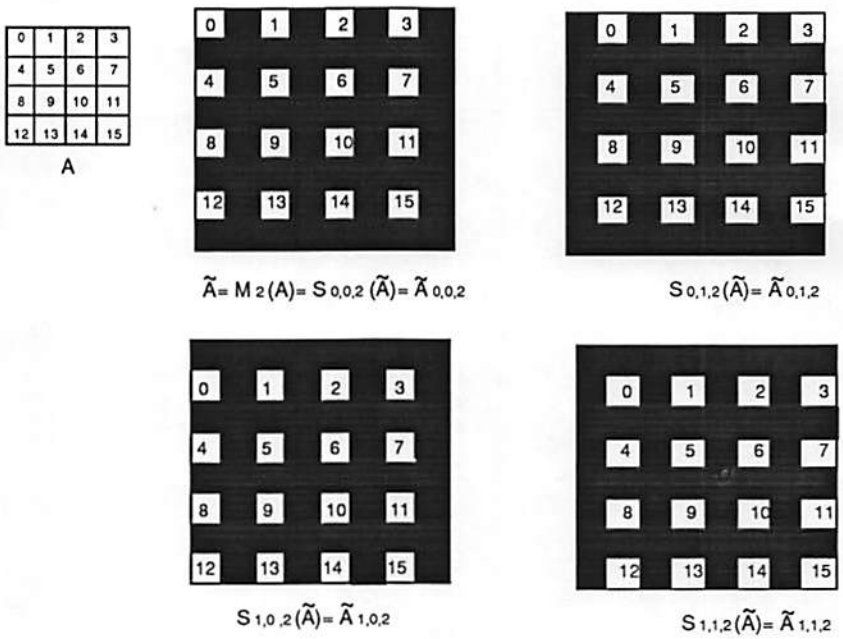


Figure 4.1: The four different shifts of the magnified input plane.

$C_{separable}$

$$\begin{aligned} &= S_{\frac{3m}{2},\frac{3m}{2},2}(C_1) + S_{\frac{3m}{2},\frac{m}{2},2}(C_2) + S_{\frac{m}{2},\frac{3m}{2},2}(C_3) + S_{\frac{m}{2},\frac{m}{2},2}(C_4) \qquad (4.12)\\ &= [\ c_{\ s\ b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ t\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\ ] \quad s,\ t,\ x \text{ and } y\ \in \{0,1\}. \end{aligned}$$

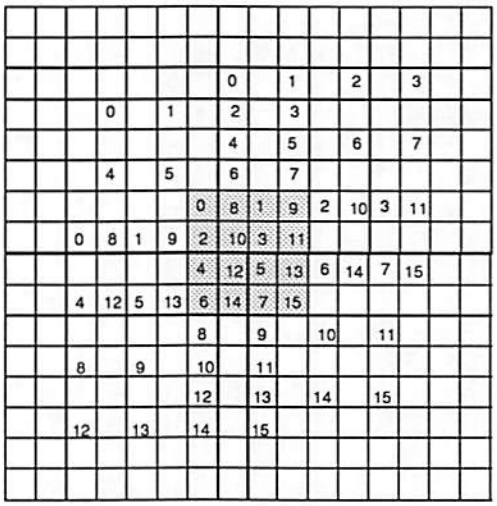
The desired result appears as the center $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ matrix of $C_{separable}$, i.e.,

$$\alpha^{0,0}_{0,0,2} + \alpha^{0,1}_{0,1,2} + \alpha^{1,0}_{1,0,2} + \alpha^{1,1}_{1,1,2} \ = \ N_{2^{\frac{m}{2}},2^{\frac{m}{2}}}(C_{separable}). \qquad (4.13)$$

In the second method where $\tilde{A}_{i,j,2}$'s are used as different quadrants of the C_t,

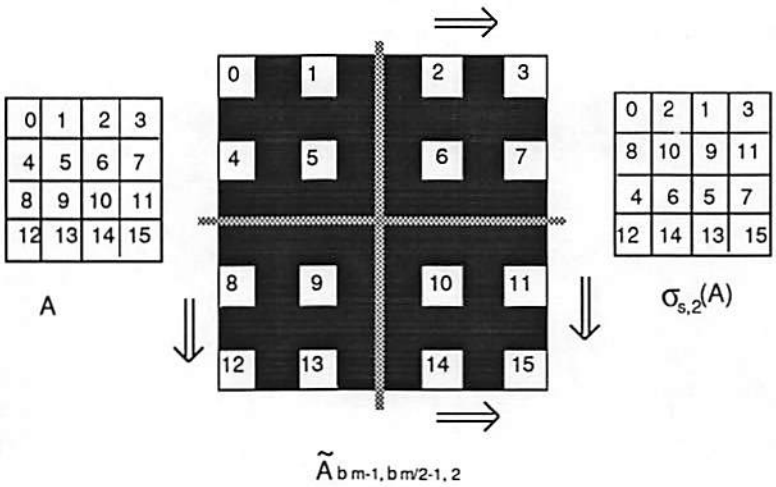
$C_{separable}$

$$\begin{aligned} &= S_{\frac{-m}{2},\frac{-m}{2},2}(C_1) + S_{\frac{3m}{2},\frac{-3m}{2},2}(C_2) + S_{\frac{-3m}{2},\frac{3m}{2},2}(C_3) + S_{\frac{m}{2},\frac{m}{2},2}(C_4) \quad (4.14)\\ &= [\ c_{\ s\ b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ ,\ t\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\ ] \quad s,\ t,\ x \text{ and } y\ \in \{0,1\}. \end{aligned}$$



$\sigma_{f,2}(A) = N_{4,4}(C_{folded})$

(a)



(b)

Figure 4.2: Illustrations of four-copy and one-copy algorithms for 2-D folded shuffle, $N = 16$; (a) shows the output as the center portion of the superposition of the four shifts; (b) shows the one-copy algorithm which performs 2-D folded shuffle operation by applying different shifts on the four quadrants of the magnified input plane.

Similarly, the desired result appears as the center $2^{\frac{m}{2}} \times 2^{\frac{m}{2}}$ matrix of $C_{separable}$, i.e.,

$$\alpha^{0,0}_{0,0,2} + \alpha^{0,1}_{0,1,2} + \alpha^{1,0}_{1,0,2} + \alpha^{1,1}_{1,1,2} \;=\; N_{2^{\frac{m}{2}},2^{\frac{m}{2}}}(C_{separable}).$$

2. One-Copy Algorithm

Similarly, the one-copy method can also be used for 2-D separable shuffle. Here, the shifting operation applied on $\tilde{A}$ in step 2 is $S_{b_{m-1},b_{\frac{m}{2}-1},2}$, thus,

$$\begin{aligned} S_{b_{m-1},b_{\frac{m}{2}-1},2}(\tilde{A}) &= S_{b_{m-1},b_{\frac{m}{2}-1}}(\tilde{A}_{0,0,2}) \\ &= S_{b_{m-1},b_{\frac{m}{2}-1},2}\left(\begin{bmatrix} \alpha^{0,0}_{0,0,2} & \alpha^{0,1}_{0,0,2} \\ \alpha^{1,0}_{0,0,2} & \alpha^{1,1}_{0,0,2} \end{bmatrix} \right) \\ &= \begin{bmatrix} \alpha^{0,0}_{b_{m-1},b_{\frac{m}{2}-1},2} & \alpha^{0,1}_{b_{m-1},b_{\frac{m}{2}-1},2} \\ \alpha^{1,0}_{b_{m-1},b_{\frac{m}{2}-1},2} & \alpha^{1,1}_{b_{m-1},b_{\frac{m}{2}-1},2} \end{bmatrix} \end{aligned} \qquad (4.15)$$

The final step is to add up these shifted four quadrant submatrices of $\tilde{A}_{0,0,2}$.

$$\alpha^{k,l}_{b_{m-1},b_{\frac{m}{2}-1},2} \;=\; [\; \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} \;] \qquad (4.16)$$

where

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } b_{m-1} = k, \\ & b_{\frac{m}{2}-1} = l, \\ & p = b_{m-1}, \\ & q = b_{\frac{m}{2}-1} \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the desired output is

$$\begin{aligned} \alpha^{0,0}_{0,0,2} + \alpha^{0,1}_{0,1,2} + \alpha^{1,0}_{1,0,2} + \alpha^{1,1}_{1,1,2} &= [\; \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{\frac{m}{2}-1}} \;] \\ &= [\; a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1},\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{\frac{m}{2}-1}} \;] \\ &= \sigma_{s,2}(A). \end{aligned}$$

The case of $N = 16$ for both algorithms is shown in Fig. 4.3. Figure 4.3 (a) shows the superposition and extraction steps of the four-copy algorithm. The different shifts applied to the four quadrants of the magnified input plane of the one-copy algorithm are depicted in Fig. 4.3 (b).

4.2 One-Copy Algorithm for 2-D r-Shuffles

To further generalize our one-copy algorithm, we show that similar mechanisms can be applied to produce 2-D folded and separable r-shuffles, where r is a prime. In this case, all the terms defined before should be expressed as powers of r. The notation should be modified as follows:

$$\begin{aligned} N &= r^m \\ p &= (b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}})_r \\ &= \sum_{i=0}^{\frac{m}{2}-1} b_{i+\frac{m}{2}} r^i \qquad (4.17) \\ q &= (b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0)_r \\ &= \sum_{i=0}^{\frac{m}{2}-1} b_i r^i, \qquad (4.18) \end{aligned}$$

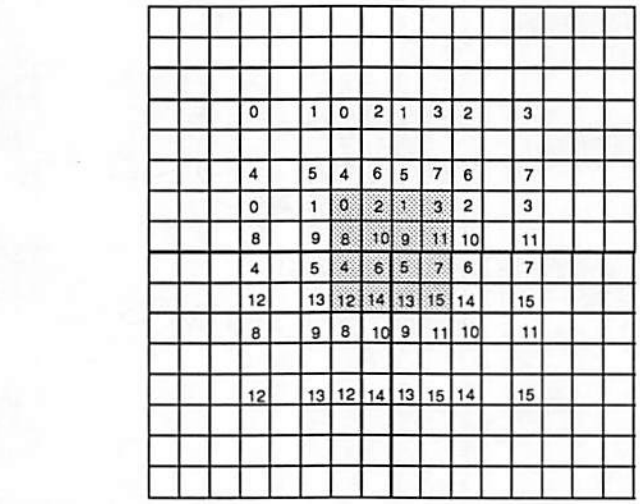
where $b_i \in \{0, 1, 2, \ldots, r-1\}$ and the r's complement (the additive inverse) of b_i is $\bar{b}_i$ (i.e., $b_i + \bar{b}_i = r - 1$). In the following definitions, r's are used in the subscripts showing that base r number system is used.

Similar to the 2-D folded shuffle, the 2-D folded r-shuffle, denoted as $\sigma_{f,r}$, is actually a 1-D r-shuffle on the unfolded 2-D plane.

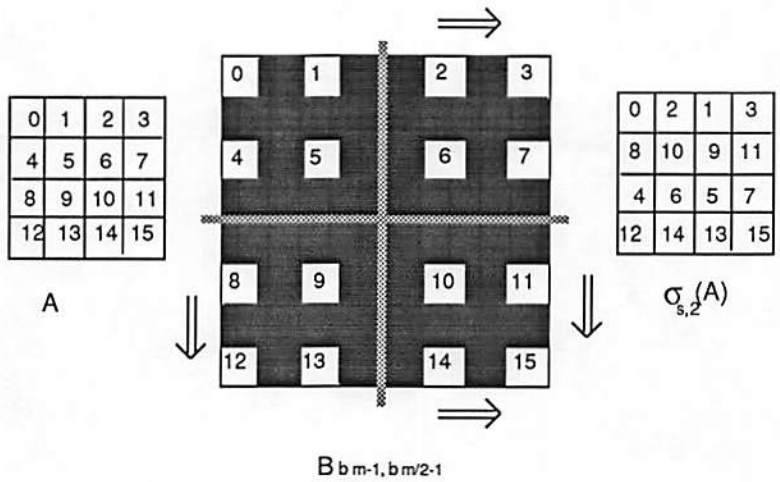
Definition 4.5 *2-D folded r-shuffle ($\sigma_{f,r}$)*

$$\begin{aligned} \sigma_{f,r}(A) &= \sigma_{f,r}(\ [\ a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}\ ]\ ) \\ &= [\ a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}}\ ] \qquad (4.19) \end{aligned}$$

The 2-D folded r-shuffle operation on $N = r^m$ channels forms a *cyclic group* of order m because A arrives at its original state after m applications of $\sigma_{f,r}$.



(a)



(b)

Figure 4.3: Illustrations of four-copy and one-copy algorithms for 2-D separable shuffle, $N = 16$. (a) shows the output as the center portion of the superposition of the four shifts; (b) shows the one-copy algorithm which performs 2-D separable shuffle operation by applying different shifts on the four quadrants of the magnified input plane.

The 2-D separable r-shuffle contains independent circular left shifts on the row and the column r-based indices.

Definition 4.6 *2-D separable r-shuffle ($\sigma_{s,r}$)*

$$\begin{aligned} \sigma_{s,r}(A) &= \sigma_{s,r}(\ [\ a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}\ ]\ ) \\ &= [\ a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1}, b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{\frac{m}{2}-1}}\ ] \end{aligned} \tag{4.20}$$

The 2-D separable r-shuffle operation on $N = r^m$ channels also forms a *cyclic group*, but of order $\frac{m}{2}$.

Definition 4.7 *Magnification (M_r)*

To magnify an $r^{\frac{m}{2}} \times r^{\frac{m}{2}}$ matrix A into an $r^{\frac{m}{2}+1} \times r^{\frac{m}{2}+1}$ matrix $\tilde{A}$ is to place the p^{th} row, q^{th} column entry in A into the $\tilde{p}^{th}$ row, $\tilde{q}^{th}$ column entry ($\tilde{p} = rp$ and $\tilde{q} = rq$) in $\tilde{A}$ ($\tilde{A} = [\tilde{a}_{\tilde{p},\tilde{q}}]$) and set the other entries to zero. Thus,

$$\begin{aligned} \tilde{A} &= M_r(A) \\ &= M_r([\ a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}\ ]) \\ &= [\ \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ , b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\ ] \quad x, y \in \{0, 1, \ldots, r-1\} \end{aligned} \tag{4.21}$$

where

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x\ , b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } x = y = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.8 *Shift ($S_{i,j,r}$)*

The $S_{i,j,r}(A)$ shift corresponds to the circular shift of all the entries in matrix A i rows downward and j columns towards the right.

$$\begin{aligned} S_{i,j,r}(A) &= S_{i,j,r}([\ a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}\ ]) \\ &= [\ a_{[(b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}})+i]\ mod\ r^{\frac{m}{2}}, [(b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0)+j]\ mod\ r^{\frac{m}{2}}}\ ] \end{aligned} \tag{4.22}$$

Note: a negative shift implies a shift in the opposite direction.

Example 4.3 *Suppose* $\tilde{A} = M_r(A)$,

$$S_{b_{\frac{m}{2}-1},b_{m-1},r}(\tilde{A}) \;=\; [\; \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\; x\;,\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\; y}\;] \qquad x,y \in \{0,1,\ldots,r-1\}$$

where

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\; x\;,\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\; y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\; b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } x = b_{\frac{m}{2}-1}, \\ & \quad y = b_{m-1} \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.9 *Matrix Breaking*

Let us denote $S_{i,j,r}(\tilde{A})$ by $\tilde{A}_{i,j,r}$. Each shifted matrix $\tilde{A}_{i,j,r}$ which has the same size as $\tilde{A}$ ($r^{\frac{m}{2}+1} \times r^{\frac{m}{2}+1}$), can be partitioned into r^2 equal sized submatrices (of size $r^{\frac{m}{2}} \times r^{\frac{m}{2}}$), denoted as $\alpha^{k,l}_{i,j,r}$, where i, j, k, and l are all in $\{0,1,\ldots,r-1\}$.

$$\tilde{A}_{i,j,r} \;=\; \begin{bmatrix} \alpha^{0,0}_{i,j,r} & \alpha^{0,1}_{i,j,r} & \cdots & \alpha^{0,r-1}_{i,j,r} \\ \alpha^{1,0}_{i,j,r} & \alpha^{1,1}_{i,j,r} & \cdots & \alpha^{1,r-1}_{i,j,r} \\ \vdots & & & \\ \alpha^{r-1,0}_{i,j,r} & \alpha^{r-1,1}_{i,j,r} & \cdots & \alpha^{r-1,r-1}_{i,j,r} \end{bmatrix} \tag{4.23}$$

4.2.1 2-D Folded r-shuffle

Similar to the 2-D folded shuffle, the three steps of one-copy algorithm for 2-D folded r-shuffle are:

Step 1. magnify-mask: Consider the input data plane as an $r^{\frac{m}{2}} \times r^{\frac{m}{2}}$ matrix A. Apply the magnification operation (M_r) on A to get an $r^{\frac{m}{2}+1} \times r^{\frac{m}{2}+1}$ matrix $\tilde{A}$, such that $\tilde{A} = M_r(A)$.

Step 2. shift: Various shifting operations should be applied on the submatrices of $\tilde{A}$. The shifting operation applied on $\tilde{A}$ is $S_{b_{\frac{m}{2}-1},b_{m-1},r}$, thus,

$$S_{b_{\frac{m}{2}-1},b_{m-1},r}(\tilde{A}) \;=\; S_{b_{\frac{m}{2}-1},b_{m-1},r}(\tilde{A}_{0,0,r})$$

$$= S_{b_{\frac{m}{2}-1},b_{m-1},r}\left(\begin{bmatrix} \alpha^{0,0}_{0,0,r} & \alpha^{0,1}_{0,0,r} & \cdots & \alpha^{0,r-1}_{0,0,r} \\ \alpha^{1,0}_{0,0,r} & \alpha^{1,1}_{0,0,r} & \cdots & \alpha^{1,r-1}_{0,0,r} \\ \vdots & & & \\ \alpha^{r-1,0}_{0,0,r} & \alpha^{r-1,1}_{0,0,r} & \cdots & \alpha^{r-1,r-1}_{0,0,r} \end{bmatrix}\right)$$

$$= \begin{bmatrix} \alpha^{0,0}_{b_{\frac{m}{2}-1},b_{m-1},r} & \alpha^{0,1}_{b_{\frac{m}{2}-1},b_{m-1},r} & \cdots & \alpha^{0,r-1}_{b_{\frac{m}{2}-1},b_{m-1},r} \\ \alpha^{1,0}_{b_{\frac{m}{2}-1},b_{m-1},r} & \alpha^{1,1}_{b_{\frac{m}{2}-1},b_{m-1},r} & \cdots & \alpha^{1,r-1}_{b_{\frac{m}{2}-1},b_{m-1},r} \\ \vdots & & & \\ \alpha^{r-1,0}_{b_{\frac{m}{2}-1},b_{m-1},r} & \alpha^{r-1,1}_{b_{\frac{m}{2}-1},b_{m-1},r} & \cdots & \alpha^{r-1,r-1}_{b_{\frac{m}{2}-1},b_{m-1},r} \end{bmatrix} \quad (4.24)$$

Step 3. superimpose: Finally, the folded r-shuffled matrix is obtained by adding these r^2 shifted submatrices of the $\tilde{A}_{0,0,r}$. This step is performed by neglecting the first digits of the row and the column indices. We have

$$\alpha^{k,l}_{b_{\frac{m}{2}-1},b_{m-1},r} = [\ \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\ ] \quad (4.25)$$

where

$$\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } b_{m-1}=k, \\ & b_{\frac{m}{2}-1}=l, \\ & x=b_{\frac{m}{2}-1}, \\ & y=b_{m-1} \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the desired output is,

$$\sum_{i=0}^{r-1}\sum_{j=0}^{r-1}\alpha^{j,i}_{i,j,r} = [\ \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}}\ ]$$
$$= [\ a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}}\ ]$$
$$= \sigma_{f,r}(A).$$

4.2.2 2-D Separable r-shuffle

The three steps of one-copy algorithm for 2-D separable r-shuffle are:

Step 1. magnify-mask: This is the same as for the folded case.

Step 2. shift: Various shifting operations are applied on the submatrices of $\tilde{A}$. The shifting operation applied on $\tilde{A}$ is $S_{b_{m-1},b_{\frac{m}{2}-1},r}$, thus,

$$
\begin{aligned}
S_{b_{m-1},b_{\frac{m}{2}-1},r}(\tilde{A}) &= S_{b_{m-1},b_{\frac{m}{2}-1},r}(\tilde{A}_{0,0,r}) \\
&= S_{b_{m-1},b_{\frac{m}{2}-1},r}\left(\begin{bmatrix} \alpha_{0,0,r}^{0,0} & \alpha_{0,0,r}^{0,1} & \cdots & \alpha_{0,0,r}^{0,r-1} \\ \alpha_{0,0,r}^{1,0} & \alpha_{0,0,r}^{1,1} & \cdots & \alpha_{0,0,r}^{1,r-1} \\ \vdots & & & \\ \alpha_{0,0,r}^{r-1,0} & \alpha_{0,0,r}^{r-1,1} & \cdots & \alpha_{0,0,r}^{r-1,r-1} \end{bmatrix}\right) \\
&= \begin{bmatrix} \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{0,0} & \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{0,1} & \cdots & \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{0,r-1} \\ \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{1,0} & \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{1,1} & \cdots & \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{1,r-1} \\ \vdots & & & \\ \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{r-1,0} & \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{r-1,1} & \cdots & \alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{r-1,r-1} \end{bmatrix}
\end{aligned}
$$

Step 3. superimpose: Finally, the separable r-shuffled matrix is obtained by adding these r^2 shifted submatrices of the $\tilde{A}_{0,0,r}$ using

$$
\alpha_{b_{m-1},b_{\frac{m}{2}-1},r}^{k,l} = [\ \tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y}\ ] \tag{4.26}
$$

where

$$
\tilde{a}_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}\ x,\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0\ y} = \begin{cases} a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} & \text{if } b_{m-1}=k,\ b_{\frac{m}{2}-1}=l,\ x=b_{m-1},\ y=b_{\frac{m}{2}-1} \\ 0 & \text{otherwise.} \end{cases}
$$

Thus, the desired output is

$$\begin{aligned}\sum_{i=0}^{r-1}\sum_{j=0}^{r-1}\alpha_{i,j,r}^{i,j} &= [\, b_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}} \,] \\ &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1}},\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{\frac{m}{2}-1}} \,] \\ &= \sigma_{s,r}(A).\end{aligned}$$

4.2.3 Summary

In summary, the one-copy algorithm for both 2-D folded and separable r-shuffles, where r is a prime number, can be described in three steps:

1. Magnify the input plane A to get $\tilde{A}_{0,0,r}$, s.t. $\tilde{A}_{0,0,r} = M_r(A)$.
2. Break $\tilde{A}_{0,0,r}$ into r^2 equal size submatrices, $\alpha_{0,0,r}^{i,j}$, where $0 \le i,j \le (r-1)$.
3. Apply different shifts on these submatrices and superimpose them. For the folded r-shuffle, the shift on $\alpha_{0,0,r}^{i,j}$ is $S_{j,i,r}$, whereas, for the separable r-shuffle, it is $S_{i,j,r}$.

4.3 One-Copy Algorithm for 2-D r^k-Shuffles

We have shown that the one-copy algorithm can be generalized for performing 2-D r-shuffles on $N = r^m$ channels, where r is a prime and m is an even integer. The 2-D r^k-shuffles are basically k-digit circular left shifts on the r-ary indices. They are defined as follows, and are shown here that they can be realized by cascading k stages of 2-D r-shuffle. Notice that throughout sections 4.3 and 4.4, the indices are represented as r-ary digits. The subscripts of σ's denote the type of shuffles, i.e. folded or separable, r^k-shuffle or r-shuffle.

Definition 4.10 *2-D folded r^k-shuffle (σ_{f,r^k})*

$$\begin{aligned}\sigma_{f,r^k}(A) &= \sigma_{f,r^k}([\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,]) \\ &= [\, a_{b_{m-k-1}\ldots b_{\frac{m}{2}-k}},\ b_{\frac{m}{2}-k-1}\ldots b_{m-k}} \,] \qquad (4.27)\end{aligned}$$

Definition 4.11 *2-D separable r^k-shuffle (σ_{s,r^k})*

$$\begin{aligned}\sigma_{s,r^k}(A) &= \sigma_{s,r^k}([\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,]) \\ &= [\, a_{b_{m-k-1}\ldots b_{\frac{m}{2}}b_{m-1}\ldots b_{m-k}, b_{\frac{m}{2}-k-1}\ldots b_0 b_{\frac{m}{2}-1}\ldots b_{\frac{m}{2}-k}} \,] \end{aligned} \tag{4.28}$$

Theorem 4.1 $\sigma_{f,r^k}(A) = \sigma^k_{f,r}(A)$.

proof :

$$\begin{aligned}\sigma_{f,r}(A) &= \sigma_{f,r}([\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,]) \\ &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1}, b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}} \,] \\ \sigma^2_{f,r}(A) &= [\, a_{b_{m-3}\ldots b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}, b_{\frac{m}{2}-3}\ldots b_0 b_{m-1}b_{m-2}} \,]\end{aligned}$$

Therefore,

$$\begin{aligned}\sigma^k_{f,r}(A) &= [\, a_{b_{m-k-1}\ldots b_{\frac{m}{2}-k}, b_{\frac{m}{2}-k-1}\ldots b_{m-k}} \,] \\ &= \sigma_{f,r^k}(A). \qquad \square\end{aligned}$$

Theorem 4.2 $\sigma_{s,r^k}(A) = \sigma^k_{s,r}(A)$.

proof :
Similarly,

$$\begin{aligned}\sigma_{s,r}(A) &= \sigma_{s,r}([\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,]) \\ &= [\, a_{b_{m-2}\ldots b_{\frac{m}{2}}b_{m-1}, b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{\frac{m}{2}-1}} \,] \\ \sigma^2_{s,r}(A) &= [\, a_{b_{m-3}\ldots b_{\frac{m}{2}}b_{m-1}b_{m-2}, b_{\frac{m}{2}-3}\ldots b_0 b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}} \,]\end{aligned}$$

Therefore,

$$\begin{aligned}\sigma^k_{s,r}(A) &= [\, a_{b_{m-k-1}\ldots b_{\frac{m}{2}}b_{m-1}\ldots b_{m-k}, b_{\frac{m}{2}-k-1}\ldots b_0 b_{\frac{m}{2}-1}\ldots b_{\frac{m}{2}-k}} \,] \\ &= \sigma_{s,r^k}(A). \qquad \square\end{aligned}$$

Next we show that when $1 \leq k \leq \frac{m}{2}$, the 2-D r^k-shuffles on $N = r^m$ channels can be obtained simply by single stage of the extended one-copy algorithm.

Theorem 4.3 *The one-copy algorithm can be generalized to perform 2-D r^k-shuffles on $N = r^m$ channels if $1 \leq k \leq \frac{m}{2}$. In the algorithm, the input plane is first magnified r^k times. Then the magnified plane is divided into r^{2k} equal size submatrices. The shuffled version can be obtained after applying different shifts and superimposing the submatrices.*

proof :
The row index p and the column index q of any channel in the input 2-D plane can be denoted as $\frac{m}{2}$-tuple r-ary numbers:

$$(p,q) = (\underbrace{b_{m-1}b_{m-2}\ldots b_{m-k}b_{m-k-1}\ldots b_{\frac{m}{2}}}_{\frac{m}{2}\text{ digits}}, \underbrace{b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_{\frac{m}{2}-k}b_{\frac{m}{2}-k-1}\ldots b_0}_{\frac{m}{2}\text{ digits}})_r.$$

The first step in the one-copy algorithm is to magnify the matrix r^k times, which implies the appending of k digits of zero in both the row and the column indices. Thus, the indices of the channels in the magnified matrix become

$$\begin{aligned}(\tilde{p},\tilde{q}) = (&\underbrace{b_{m-1}b_{m-2}\ldots b_{m-k}}_{k\text{ digits}}\underbrace{b_{m-k-1}\ldots b_{\frac{m}{2}}}_{(\frac{m}{2}-k)\text{ digits}}\underbrace{0\ldots0}_{k\text{ digits}},\\ &\underbrace{b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_{\frac{m}{2}-k}}_{k\text{ digits}}\underbrace{b_{\frac{m}{2}-k-1}\ldots b_0}_{(\frac{m}{2}-k)\text{ digits}}\underbrace{0\ldots0}_{k\text{ digits}})_r.\end{aligned} \tag{4.29}$$

The first k digits of the row and the column indices represent the indices (i,j) of the submatrices $\alpha^{i,j}_{0,0,r}$. In the second step of one-copy algorithm for the folded r^k-shuffle, a shift $S_{j,i,r}$ is applied on the submatrix $\alpha^{i,j}_{0,0,r}$, thus,

$$\begin{aligned}(\hat{p},\hat{q}) = (&b_{m-1}b_{m-2}\ldots b_{m-k}b_{m-k-1}\ldots b_{\frac{m}{2}}0\ldots0 + b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_{\frac{m}{2}-k},\\ &b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_{\frac{m}{2}-k}b_{\frac{m}{2}-k-1}\ldots b_00\ldots0 + b_{m-1}b_{m-2}\ldots b_{m-k})_r \bmod r^{\frac{m}{2}}.\end{aligned}$$

The superposition part of one-copy algorithm is simply to ignore the effect of the first k digits of the row and the column indices. Therefore, the indices of the channels in the output plane are

$$(p,q) = (b_{m-k-1}\ldots b_{\frac{m}{2}}b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_{\frac{m}{2}-k}, b_{\frac{m}{2}-k-1}\ldots b_0b_{m-1}b_{m-2}\ldots b_{m-k})_r,$$

which is the result from the folded r^k-shuffle.

Similarly, the second step of the one-copy algorithm for the separable r^k-shuffle is to apply a shift $S_{i,j,r}$ on the submatrix $\alpha^{i,j}_{0,0,r}$. After the superposition, the indices of the channels in the output plane become

$$\begin{aligned}(p,q) &= (b_{m-k-1}\dots b_{\frac{m}{2}}0\dots 0 + b_{m-1}b_{m-2}\dots b_{m-k},\\ &\quad b_{\frac{m}{2}-k-1}\dots b_0 0\dots 0 + b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\dots b_{\frac{m}{2}-k})_r \bmod r^{\frac{m}{2}}\\ &= (b_{m-k-1}\dots b_{\frac{m}{2}}b_{m-1}b_{m-2}\dots b_{m-k}, b_{\frac{m}{2}-k-1}\dots b_0 b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\dots b_{\frac{m}{2}-k})_r.\end{aligned}$$

From Eq. (4.29), it is clear that if $k > \frac{m}{2}$, there is a conflict in the portion of $(\frac{m}{2} - k)$ digits. Hence, the one-copy algorithm can be extended for obtaining 2-D r^k-shuffles only when $1 \le k \le \frac{m}{2}$. □

4.4 One-Copy Algorithm for 2-D Inverse Shuffles

Here we consider the inverse operation of the 2-D shuffles discussed earlier. Because the shuffling operation is a circular left shift of the indexing digits, the inverse shuffling operation is a circular right shift of the indexing digits.

Definition 4.12 *2-D inverse folded r^k-shuffle (σ^{-1}_{f,r^k})*

$$\begin{aligned}\sigma^{-1}_{f,r^k}(A) &= \sigma^{-1}_{f,r^k}(\ [\ a_{b_{m-1}b_{m-2}\dots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\dots b_1 b_0}\ ]\ )\\ &= [\ a_{b_{k-1}\dots b_0 b_{m-1}\dots b_{\frac{m}{2}+k},\ b_{\frac{m}{2}-1+k}\dots b_k}\ ]\\ &= \sigma_{f,r^{m-k}}(A) \qquad (4.30)\end{aligned}$$

Definition 4.13 *2-D inverse separable r^k-shuffle (σ^{-1}_{s,r^k})*

$$\begin{aligned}\sigma^{-1}_{s,r^k}(A) &= \sigma^{-1}_{s,r^k}(\ [\ a_{b_{m-1}b_{m-2}\dots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\dots b_1 b_0}\ ]\ )\\ &= [\ a_{b_{\frac{m}{2}+k-1}\dots b_{\frac{m}{2}}b_{m-1}\dots b_{\frac{m}{2}+k},\ b_{k-1}\dots b_0 b_{\frac{m}{2}-1}\dots b_k}\ ]\\ &= \sigma_{s,r^{\frac{m}{2}-k}}(A) \qquad (4.31)\end{aligned}$$

4.4.1 Single Stage Implementation

In certain special cases, the 2-D inverse shuffles can be performed in a single stage. These cases are:

1. For 2-D folded shuffle — From Theorem 4.3, the 2-D folded r^k-shuffle can be performed by the one-copy algorithm if $1 \leq k \leq \frac{m}{2}$. Because $\sigma^{-1}_{f,r^k}(A) = \sigma_{f,r^{m-k}}(A)$ (Definition 4.12), the inverse folded r^k-shuffle can be obtained by the similar algorithm only when $1 \leq (m-k) \leq \frac{m}{2}$. Therefore, for the 2-D folded shuffle, the folded and the inverse folded r^k-shuffles can both be performed by utilizing the one-copy algorithm if both $1 \leq k \leq \frac{m}{2}$ and $1 \leq (m-k) \leq \frac{m}{2}$. Hence, the only possible case for this to be true is when $k \leq \frac{m}{2}$.

2. For 2-D separable shuffle — Similarly, applying Theorem 4.3 and Definition 4.13, we conclude that the 2-D separable r^k-shuffle and its inverse can both be obtained from the one-copy algorithm if both $1 \leq k \leq \frac{m}{2}$ and $1 \leq (\frac{m}{2}-k) \leq \frac{m}{2}$. Therefore, k can range from 1 to $\frac{m}{2}$ here.

4.4.2 Multistage Implementation

Another way of doing the 2-D inverse shuffles by the one-copy algorithm uses the concept of *cyclic group*. From algebraic definitions, we know that any cyclic group has the following properties:

1. Let $(G, \cdot)$ be a cyclic group with generator a, then every element g of G is a power of a, i.e. $g = a^n$ for some integer n.

2. Let G be a cyclic group of order λ and $g \in G$, then $g^{-t} = g^{(\alpha\lambda - t)}$ for any integer t and α.

It is obvious that α can be selected to make $(\alpha\lambda - t)$ a positive number. This implies that the inverse of any element in a cyclic group is equal to some positive power of the element. Since the shuffling is a circular left shift operation on the digits, it forms a *cyclic group*. In the following, we prove that the 2-D inverse r^k-shuffles can be achieved by cascading several stages of the 2-D r^k-shuffle. The number of stages required is dependent on the type of 2-D inverse r^k-shuffle on $N = r^m$ channels.

Definition 4.14 *The order of the 2-D folded r^k-shuffle on $N = r^m$ channels ($\lambda_{f,m,k}$) is defined as the minimum number of k-digit circular left shifts on the indices of the channels needed to reach the original state, while considering the row and the column indices as one entity.*

Definition 4.15 *The order of the 2-D separable r^k-shuffle on $N = r^m$ channels ($\lambda_{s,m,k}$) is defined as the minimum number of k-digit circular left shifts on the indices of the channels needed to reach the original state, while considering the row and the column indices independently.*

Theorem 4.4 *The order of the 2-D folded r^k-shuffle on $N = r^m$ channels is*

$$\lambda_{f,m,k} = \frac{m}{\gcd(m,k)},$$

where $\gcd(m,k)$ *is the greatest common divisor of* m *and* k.

proof :
The index of the channel at the beginning is

$$\underbrace{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}_{m \text{ digits}}.$$

The 2-D folded r^k-shuffle is an operation of k-digit circular left shift on the index, thus, we expect,

$$\begin{aligned} \lambda_{f,m,k}k &= \text{lcm}\,(m,k) \\ &= \frac{mk}{\gcd(m,k)}, \end{aligned}$$

where lcm (m,k) is the least common multiple of m and k. Hence,

$$\lambda_{f,m,k} = \frac{m}{\gcd(m,k)}. \qquad \square$$

Corollary 4.1 *For any 2-D folded r-shuffle,* $\lambda_{f,m,1} = m$.

proof :

$$\begin{aligned} \text{Since } \lambda_{f,m,k} &= \frac{m}{\gcd(m,k)}, \\ \lambda_{f,m,1} &= \frac{m}{\gcd(m,1)} \\ &= \frac{m}{1} \\ &= m. \end{aligned}$$ □

Theorem 4.5 *The order of the 2-D separable r^k-shuffle on $N = r^m$ channels is*

$$\lambda_{s,m,k} = \frac{m}{2\gcd(\frac{m}{2},k)}.$$

proof :
The index of the channels at the beginning is

$$\underbrace{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}}_{\frac{m}{2}\text{ digits}}, \underbrace{b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}_{\frac{m}{2}\text{ digits}}.$$

The 2-D separable r^k-shuffle is an operation of k-digit circular left shifts on the row and the column indices, thus, we expect,

$$\begin{aligned} \lambda_{s,m,k}k &= \text{lcm}\,(\frac{m}{2},k) \\ &= \frac{mk}{2\gcd(\frac{m}{2},k)} \\ \lambda_{s,m,k} &= \frac{m}{2\gcd(\frac{m}{2},k)}. \end{aligned}$$ □

Corollary 4.2 *For any 2-D separable r-shuffle, $\lambda_{s,m,1} = \frac{m}{2}$.*

proof :

$$\begin{aligned} \text{Since } \lambda_{s,m,k} &= \frac{m}{2\gcd(\frac{m}{2},k)}, \\ \lambda_{s,m,1} &= \frac{m}{2\gcd(\frac{m}{2},1)} \end{aligned}$$

$$= \frac{m}{2}. \qquad \square$$

By applying Theorems 4.4 and 4.5 to the second property of the cyclic group, we can show that the minimum number of stages required for 2-D inverse r^k-shuffles with $N = r^m$ channels is $(\lambda_{f,m,k} - 1)$ for the folded case, and is $(\lambda_{s,m,k} - 1)$ for the separable case.

Theorem 4.6 $\sigma^{-1}_{f,r^k}(A) = \sigma^{\lambda_{f,m,k}-1}_{f,r^k}(A)$.

proof :
The proof follows from the above by letting $g = \sigma_{f,r^k}$, $t = \alpha = 1$, thus,

$$\sigma^{-1}_{f,r^k}(A) \;=\; \sigma^{\lambda_{f,m,k}-1}_{f,r^k}(A) \qquad \square$$

Theorem 4.7 $\sigma^{-1}_{s,r^k}(A) = \sigma^{\lambda_{s,m,k}-1}_{s,r^k}(A)$.

proof :
Similarly, it can be proved by applying second property of the cyclic group and Theorem 4.5. $\square$

Furthermore, the number of stages required for $\sigma^{-1}_{f,r^k}(A)$ can be reduced from $(\lambda_{f,m,k} - 1)$ to $(\frac{\lambda_{f,m,k}}{2} - 1)$ if $\lambda_{f,m,k}$ is an even integer. In this case, $\sigma^{-1}_{f,r^k}(A)$ can be obtained by $(\frac{\lambda_{f,m,k}}{2} - 1)$ stages of σ_{f,r^k} on the transpose of the input matrix (A^T).

Lemma 4.1 *If $\lambda_{f,m,k}$ is an even integer, the transpose of the input matrix (A^T) can be obtained by applying σ_{f,r^k} on the input matrix $\frac{\lambda_{f,m,k}}{2}$ times, i.e.,*

$$A^T = \sigma^{\frac{\lambda_{f,m,k}}{2}}_{f,r^k}(A).$$

proof :
Let us define τ to be the number of stages of σ_{f,r^k} applied on A which results in A^T, then,

$$b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_0$$

$$\downarrow\ \tau \text{ times } \sigma_{f,r^k}$$

$$b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_0,\ b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}.$$

Therefore,

$$\begin{aligned} \tau k &= \frac{1}{2}\text{lcm}\,(m,k) \\ &= \frac{1}{2}\frac{mk}{\gcd(m,k)} \\ \tau &= \frac{1}{2}\frac{m}{\gcd(m,k)} \\ &= \frac{1}{2}\lambda_{f,m,k}. \end{aligned}$$

Because the number of stages must be an integer, if $\lambda_{f,m,k}$ is an odd integer, τ can not be an integer, which results in the contradiction. □

Theorem 4.8 *If $\lambda_{f,m,k}$ is an even integer,*

$$\sigma_{f,r^k}^{-1}(A) = \sigma_{f,r^k}^{\lambda_{f,m,k}-1}(A) = \sigma_{f,r^k}^{\frac{\lambda_{f,m,k}}{2}-1}(A^T).$$

proof :
Applying the Lemma 4.1 to Theorem 4.4,

$$\begin{aligned} \sigma_{f,r^k}^{-1}(A) &= \sigma_{f,r^k}^{\lambda_{f,m,k}-1}(A) \\ &= \sigma_{f,r^k}^{\frac{\lambda_{f,m,k}}{2}-1}(\sigma_{f,r^k}^{\frac{\lambda_{f,m,k}}{2}}(A)) \\ &= \sigma_{f,r^k}^{\frac{\lambda_{f,m,k}}{2}-1}(A^T). \end{aligned}$$ □

Example 4.4 *For the 2-D inverse folded r-shuffle,*

$$\sigma_{f,r}^{-1}(A) = \sigma_{f,r}^{m-1}(A) = \sigma_{f,r}^{\frac{m}{2}-1}(A^T).$$

proof :

$$\begin{aligned} A &= [\, a_{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}},\ b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0} \,] \\ \sigma_{f,r}^{m-1}(A) &= [\, a_{b_0 b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}+1},\ b_{\frac{m}{2}}b_{\frac{m}{2}-1}\ldots b_1} \,] \\ &= \sigma_{f,r}^{-1}(A) \\ A^T &= [\, a_{b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0,\ b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}} \,] \end{aligned}$$

$$\begin{aligned}\sigma_{f,r}^{\frac{m}{2}-1}(A^T) &= [\, a_{b_0 b_{m-1} b_{m-2} \ldots b_{\frac{m}{2}+1},\, b_{\frac{m}{2}} b_{\frac{m}{2}-1} \ldots b_1} \,] \\ &= \sigma_{f,r}^{-1}(A)\end{aligned}$$

Thus,

$$\sigma_{f,r}^{-1}(A) = \sigma_{f,r}^{m-1}(A) = \sigma_{f,r}^{\frac{m}{2}-1}(A^T). \qquad \square$$

Example 4.5 *For the 2-D inverse separable r-shuffle,*

$$\sigma_{s,r}^{-1}(A) = \sigma_{s,r}^{\frac{m}{2}-1}(A).$$

proof :

$$\begin{aligned}\sigma_{s,r}^{\frac{m}{2}-1}(A) &= [\, a_{b_{\frac{m}{2}} b_{m-1} b_{m-2} \ldots b_{\frac{m}{2}+1},\, b_0 b_{\frac{m}{2}-1} \ldots b_1} \,] \\ &= \sigma_{s,r}^{-1}(A) \quad \square\end{aligned}$$

4.4.3 Summary

The previously discussed properties of the one-copy algorithm applied for 2-D r^k folded and separable shuffles on a 2-D array having r^m channels can be summarized in Table 4.1. For a 2-D array having r^m channels (where r is prime, m is even), both the 2-D r^k-folded and separable shuffles can be realized using one-copy algorithm in a single stage whenever $1 \leq k \leq \frac{m}{2}$. However, the 2-D inverse folded r^k-shuffle using the one-copy algorithm can be realized in a single stage only when $k = \frac{m}{2}$. On the other hand, the 2-D inverse separable r^k-shuffle using one-copy algorithm can be realized in a single stage whenever $1 \leq k \leq \frac{m}{2}$. When a multistage one-copy algorithm implementation of the inverse shuffles is considered, the order is important. It has been found out that the order of the 2-D inverse folded r^k-shuffle on r^m channels is $\lambda_{f,m,k} = \frac{m}{\gcd(m,k)}$ and that for 2-D inverse separable r^k-shuffle is $\lambda_{s,m,k} = \frac{m}{2\gcd(\frac{m}{2},k)}$. When $\lambda_{f,m,k}$ is even, the inverse folded shuffle can be realized in $(\frac{\lambda_{f,m,k}}{2} - 1)$ stages, otherwise, $(\lambda_{f,m,k} - 1)$ stages are required. The multistage implementation of inverse separable shuffle always requires $(\lambda_{s,m,k} - 1)$ stages.

	single stage	multistage	
		order	no. of stages
folded	possible if $1 \leq k \leq \frac{m}{2}$		
inverse folded	possible if $k = \frac{m}{2}$	$\lambda_{f,m,k}$	$(\frac{\lambda_{f,m,k}}{2} - 1)$ if $\lambda_{f,m,k}$ is even, $(\lambda_{f,m,k} - 1)$ otherwise.
separable	possible if $1 \leq k \leq \frac{m}{2}$		
inverse separable	possible if $1 \leq k \leq \frac{m}{2}$	$\lambda_{s,m,k}$	$(\lambda_{s,m,k} - 1)$

Table 4.1: Possible applications of one-copy algorithms for 2-D r^k-shuffles on r^m channels.

Chapter 5

Holographic Implementation of 2-D Perfect Shuffles

We present two experimental implementations of optical perfect shuffles on a 2-D plane of 64 channels based on one-copy algorithm. A hologram containing four facets is used as a light deflector to realize the shuffling operation. In the first experiment, a silver halide amplitude hologram made on Kodak 649F photographic plate is used. This leads to low diffraction efficiency although the basic principle is demonstrated. In the second experiment, a volume phase hologram made on Du Pont photopolymer is used. The results show the potential of future applications.

5.1 First Experiment

Realizing the one-copy algorithm requires a system that applies different shifts to the four input submatrices and superimposes them. Considering the four input submatrices individually, the deflecting operation on each can be achieved by a space-invariant system, e.g., each facet having a distinct point spread function. Figure 5.1 shows our first experimental system. The two beams shown in shaded are used to make the facet hologram on the hologram plane H. The hologram deflects the light beams of the four quadrants and superimpose them on the observation plane P. During hologram recording the input objects are absent. A specially designed mask at the hologram plane allows only one quadrant to be exposed at a time. We adjust lens L_1 in both horizontal and vertical directions for each exposure

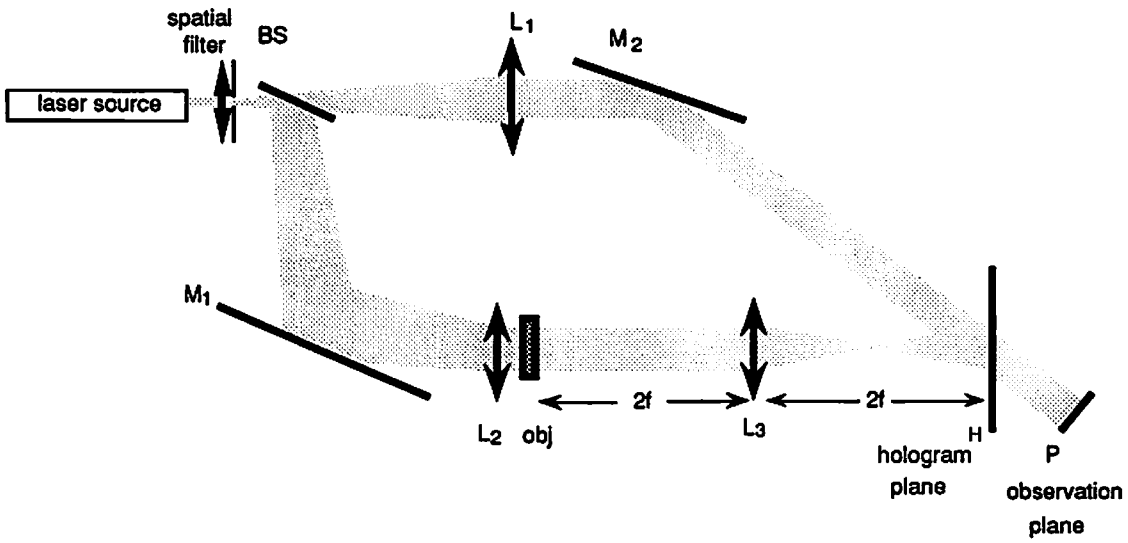


Figure 5.1: Holographic experimental setup for one-copy algorithm. BS is a beam splitter, M_1 and M_2 are mirrors, and L_1, L_2, L_3 are lenses.

to get different point spread functions for each facet. While reconstructing, the reference beam is blocked, and a pre-magnified input object is inserted. The lens L_3 images the four quadrants of the pre-magnified input object on the corresponding facets. The 2-D shuffled pattern, as the superimposition appears at the observation plane.

The magnification factor of two needed for the one-copy algorithm was not included in our imaging system. For simplicity, a pre-magnified object was used as the input in our experiment. Figure 5.2 shows the computer simulated and the experimental results for both 2-D folded and 2-D separable shuffles. The objects are 8×8 binary inverse shuffled patterns of the Chinese character "light". These 64 channels are considered as 64 points in the experiment. In this experimental system, each point is 0.625 mm in diameter, the center-to-center spacing between two points is 1.25 mm in both vertical and horizontal directions, the hologram size is 10 mm × 10 mm, the observation plane size is 5 mm × 5 mm, the angle between object and reference beams is 40°, and the center-to-center distance between the hologram plane and the observation plane is 250 mm. With this set of parameters, the maximum differential path length from object plane to observation plane is approximately 3 mm or 10 ps.

The experiment was not optimized for light efficiency. A silver halide amplitude hologram (made on Kodak 649F photographic plate) was used in this experiment, therefore the diffraction efficiency was low (approximately 2.9%). Later, we demonstrate that higher light efficiency can be obtained using different materials.

5.2 Analysis

The design has the advantage of high light efficiency and low system complexity. As pointed out earlier, this is a hybrid system which contains only four point spread functions. The complexity is much less than a strictly space variant system. Theoretically, the one-copy algorithm gives 100% light efficiency for the perfect shuffle operation, which is not possible with a space invariant system. Practically, the light efficiency of the system is limited by the diffraction efficiency of the hologram. Also, the system can be simplified and made more compact if the input is in direct contact with the hologram plane, so that no lenses or prisms are used. The system has several generalizations. By moving L_1 properly in both the horizontal and vertical direction, and proper masking on the hologram plane while making the four facets, the system can be utilized for both 2-D separable and 2-D folded shuffles. Moreover, it can similarly be used for general 2-D shuffles, such as the 2-D folded r-shuffle and the 2-D separable r-shuffle, etc. A final property is that the system minimizes the effects of holographic aberrations. As discussed in [57], distortion due to the inherent holographic imaging property is the major obstacle in holographic spatial invariant interconnection architectures. Although some work has been done to compensate for the distortion using a symmetric architecture [51], it is still undesirable due to its complexity. In our proposed system, the hologram acts as a light deflector. The light which is diffracted by the hologram propagates in free space between the hologram plane and the observation plane. The final output does not suffer from any holographic aberrations because it does not form an image.

However, the capacity of this system is limited by the diffraction effects appear in the free space propagation from the hologram plane to the observation plane. It is important to calculate the space-bandwidth product (SBWP) requirement of this system and find out its limit. Here we analyze the space-bandwidth product and the limiting diffraction effects of the system.

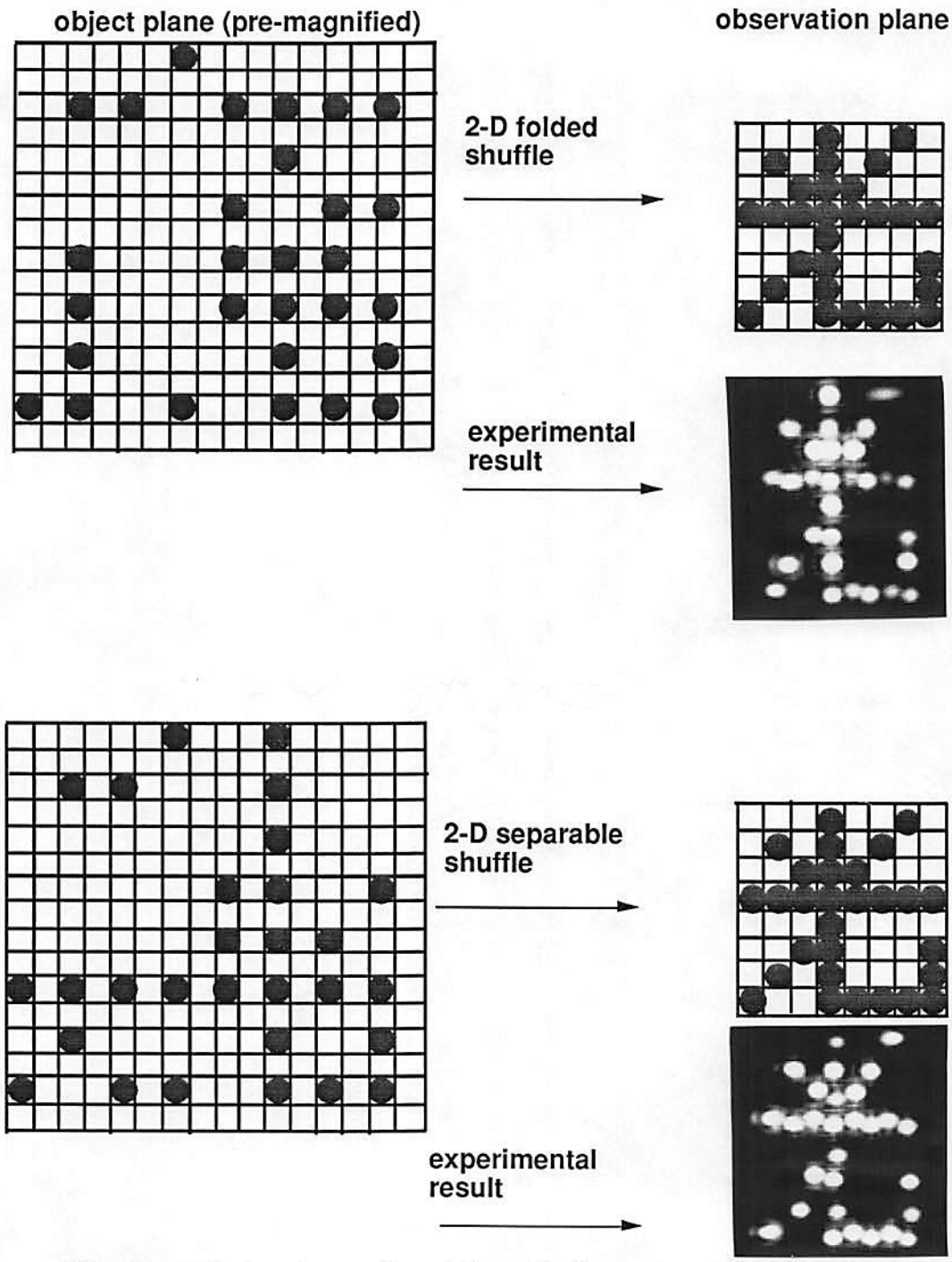


Figure 5.2: Theoretical and experimental results for one-copy folded and separable shuffles.

5.2.1 Space-Bandwidth Product

For reference, we assume that an ordinary shift-invariant 1-D imaging system is capable of resolving points separated by distance d over a field of size D. The 1-D SBWP is thus

$$n = \frac{D}{d} \tag{5.1}$$

in both x and y directions, for a total 2-D SBWP of n^2 [26]. The four-copy systems [47, 43] require four imaging systems, each capable of resolving points separated by distance $\frac{d}{2}$ (for interlacing) over a field of total size $1.25D$. Thus the four-copy 1-D SBWP is

$$n_{four-copy} = \frac{1.25D}{d/2} = 2.5n \tag{5.2}$$

or a total 2-D SBWP of 6.25 n^2.

The one-copy algorithm procedure also requires an imaging system capable of resolving points separated by distance $\frac{d}{2}$ (for interlacing) over a field of total size D. Thus the one-copy imaging system 1-D SBWP is

$$n_{one-copy\ (imaging)} = \frac{D}{d/2} = 2n. \tag{5.3}$$

Each subhologram in the one-copy system requires a 1-D SBWP of

$$n_{one-copy\ (subhologram)} = \frac{D/2}{d/2} = n \tag{5.4}$$

and there are two subholograms, so the SBWP of the complete 1-D hologram is

$$n_{one-copy\ (hologram)} = \frac{2D}{d} = 2n. \tag{5.5}$$

So, the minimum required SBWP of the 1-D one-copy system is $2n$, and the total 2-D SBWP of the one-copy system is $4n^2$.

5.2.2 Diffraction Effects

Because the hologram in our system acts as a light deflector, the free-space diffraction of light propagation from H to P limits the number of independent channels (resolvable points) in the design. The physical limitations are estimated by estimating the size of the Fresnel diffraction pattern. Because an off-axis hologram is used, the final image is compressed in the horizontal direction, and the diffraction effects are more severe in this dimension than in the vertical dimension. (This also explains why the points are closer to each other in the horizontal direction than in the vertical direction in the experimental results.) Here we analyze these effects.

We assume that a pre-magnified input object is used, and let the diameter of each input point be a mm, and the center-to-center spacing between two points be b mm. Let the total number of input points be $N = n^2$; then the size of the input plane (and also its image on H) is $D = nb$ mm. Moreover, let x denote the horizontal direction, and y be the vertical direction. Then the point size in the hologram plane, H, is a mm in both x and y directions. However, its effective size projected on the observation plane P is

$$a'_x = a\cos\theta \tag{5.6}$$

$$a'_y = a, \tag{5.7}$$

where θ is the angle between the two beams. In the observation plane, P, the size of the diffracted point can be calculated by

$$a''_x = \frac{2\lambda}{a'_x}L + a'_x \tag{5.8}$$

$$a''_y = \frac{2\lambda}{a'_y}L + a'_y, \tag{5.9}$$

where λ is the wavelength, and L is the distance from H to P. It is evident that the distance L should be as short as possible to minimize the diffraction effects. On the other hand, L should be large enough to assure the diffracted light to be separable from the zeroth order. Figure 5.3 depicts the analysis. The minimum value of L,

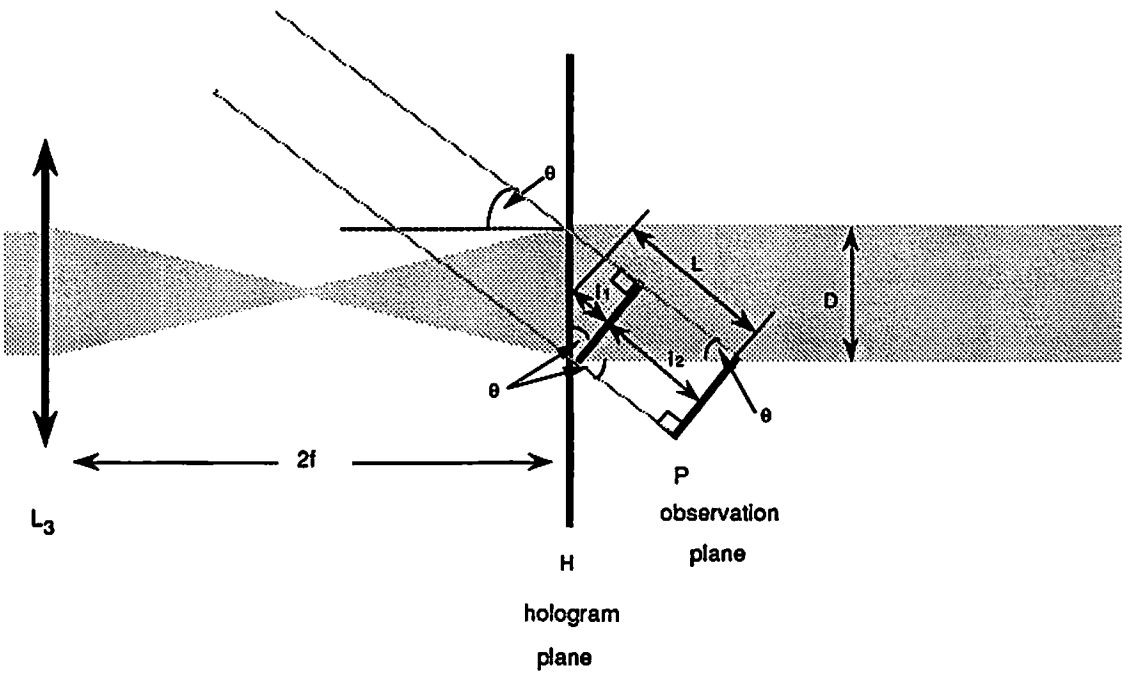


Figure 5.3: Diffraction effects analysis of the experimental setup.

L_{min} can be expressed as

$$L_{min} = l_1 + l_2, \tag{5.10}$$

where

$$l_1 = \frac{D}{2}\cos\theta \tag{5.11}$$

$$l_2 = D\cos\theta\ \cot\theta, \tag{5.12}$$

therefore,

$$L_{min} = D\cos\theta\ (\frac{1}{2} + \cot\theta). \tag{5.13}$$

As an example, assume D= 20 mm, $b = 2a$, $\lambda = 0.6$ μm and $\theta = 40°$. If half of the point diameter is taken to be the tolerance for diffraction, it is possible to use a 32×32 array as the input.

5.3 Second Experiment

There are two major drawbacks of the experiment described above. The first one is the distortion of the points. Due to the projection of the points from the hologram plane H to the observation plane P, the points have different size from quadrant to quadrant. They are also compressed in the horizontal direction. The second one is the low diffraction efficiency due to the holographic material. These two drawbacks make the experimental results unsatisfying. We built another system to eliminate these two annoying factors by correcting the experimental setup, using elliptical points as input patterns to compensate the distortion, and by using Du Pont photopolymers to make high efficient volume holograms.

In this second experiment, the setup is changed in two parts (Fig. 5.4). First, the hologram plane H is now perpendicular to the beam collimated by lens L_1. Secondly, the 4-f system for the object beam is modified by a pair of lenses L_3 and L_4 to get collimated image of the input object. This modified setup introduces an expansion (instead of compression) factor of $\frac{1}{\cos\theta}$ in the horizontal direction, where θ is the angle between the reference beam and the object beam. Thus, the input points are compressed in the horizontal direction for compensation. The objects are 8×8 binary inversed shuffled patterns of a letter "H". Following the same notation as used in our previous analysis, the input points have size $a_x = 0.2621$ mm, $a_y = 0.3125$ mm; the center-to-center spacing between two points is $b_x = 1.0483$ mm, $b_y = 1.25$ mm when $\theta = 33°$.

Du Pont HRF-150 transmission type photopolymer is used in this experiment. From our measurement, the optimum exposure energy for this material is approximately 100 mJ/cm^2. The photopolymer hologram needs a curing process, by which the monomers without being exposed to the light will polymerize. The fixing energy is around 577 mJ/cm^2. The changes in the exposure energy and the fixing energy influence the final diffraction efficiency of a hologram. In our experiment, optimum exposure energy for each facet is applied. After the recording, the photopolymer hologram is cured by a UV light source but the fixing energy is not optimized. Figure 5.5 shows the computer simulated and the experimental results. The real diffraction efficiency, which is defined as the ratio of the intensity of the diffracted light to the incident light, is found to be 60% on average of the four facets. The

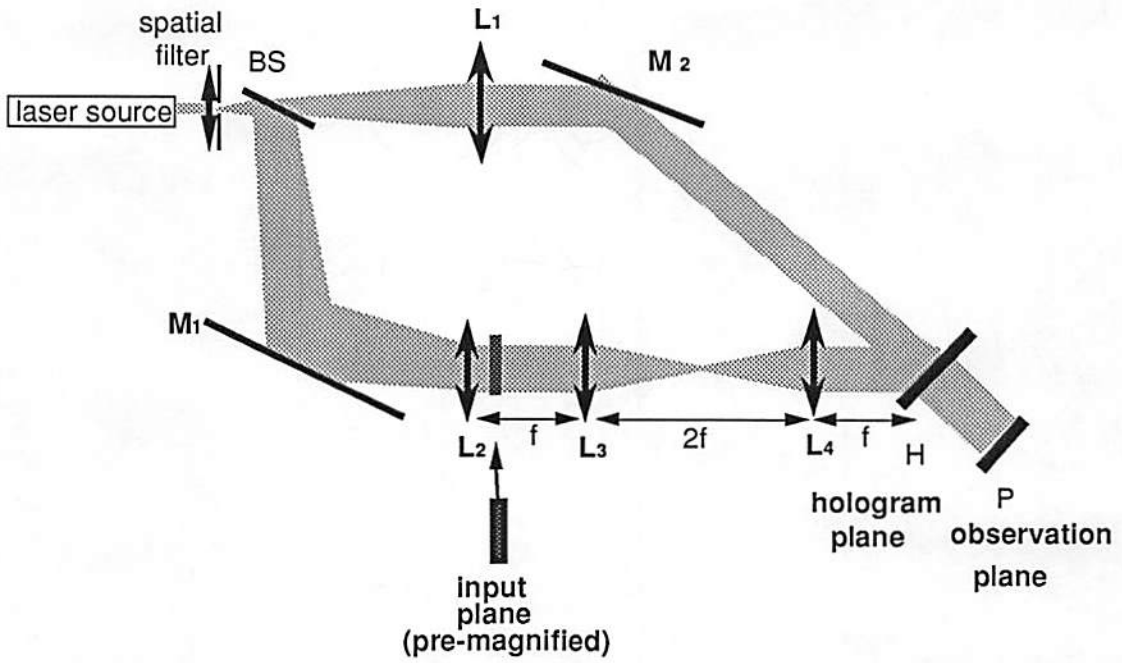


Figure 5.4: Refined experimental setup for one-copy algorithm. BS is a beam splitter, M_1 and M_2 are mirrors, and L_1, L_2, L_3 , L_4 are lenses.

scattering of the photopolymer is around 21% and the reflection loss is about 9.2%. However, the effective diffraction efficiency, which is defined to be the ratio of the diffracted light to the sum of the diffracted and the zeroth order light, is measured to be 93.8% on average.

The physical limitation of this system is also limited by diffraction. The diffraction effects appear in two places, the first one comes from the light propagates from the image plane to the hologram plane. The second one comes from the light propagates from the hologram plane to the observation plane. It is a continuous process but can be considered separately. Because the maximum number of channels can be arrange in the system depends on the number of resolvable points in this optical system, the diffraction effects limits the system capacity.

Figure 5.6 shows part of the system for the analysis purpose. The lenses L_2 and L_3 form a symmetric imaging system, and produce the image of the pre-magnified input on the image plane. The light propagates from the image plane to the hologram plane, results in a projected, and diffracted pattern. Let the diameter of each

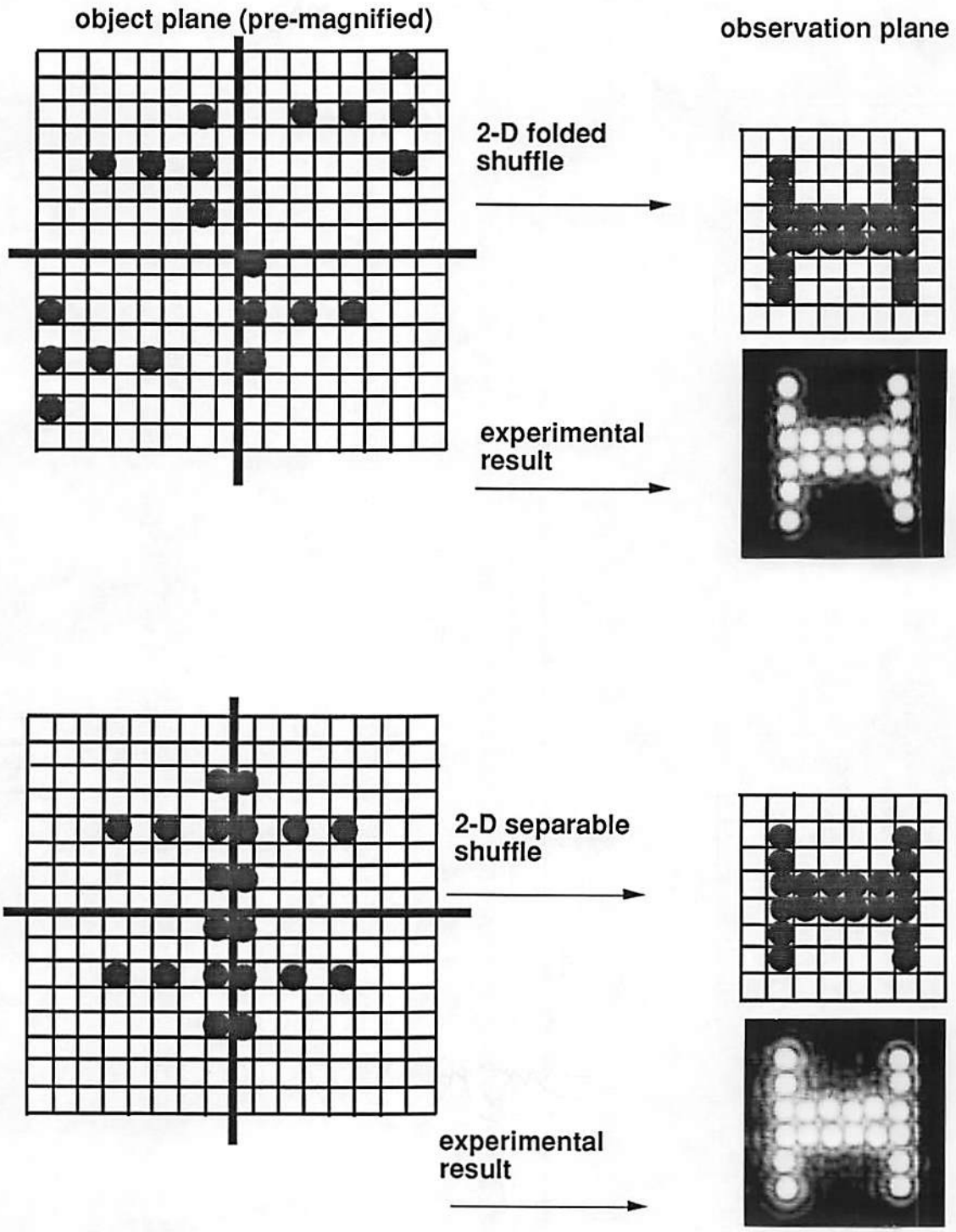


Figure 5.5: Theoretical and experimental results for one-copy folded and separable shuffles.

input point (channel) in the pre-magnified input plane be a mm, and the center-to-center spacing between two points be b mm. Let the total number of input points be $N = n \times n$; then the size of the input object plane (and also its image on the image plane I) in 1-D is $D = nb$ mm. Moreover, let x denote the horizontal direction, and y be the vertical direction. Let the angle of the two hologram construction beams be θ, and the maximum angle between the image of the pre-magnified input and its diffracted beam be α. The maximum distance from the image plane to the hologram plane is l_1, and $l_1 = \frac{D}{2}\tan\theta$. The resulting point size in the hologram plane becomes

$$a'_x = \frac{a}{\cos\theta} + 2r' \tag{5.14}$$

$$a'_y = a + 2r', \tag{5.15}$$

where $r' = l_1 \frac{\sin\alpha}{\sin(90^\circ-\alpha-\theta)} = l_1 \frac{\sin\alpha}{\cos(\alpha+\theta)}$. Thus,

$$a'_x = \frac{a}{\cos\theta} + D\frac{\sin\alpha\tan\theta}{\cos(\alpha+\theta)} \tag{5.16}$$

$$a'_y = a + D\frac{\sin\alpha\tan\theta}{\cos(\alpha+\theta)}. \tag{5.17}$$

Let the distance from the hologram plane to the observation plane be l_2 and the maximum angle of the diffracted beam due to free space propagation in this distance be β. The point in the observation plane has size

$$a''_x = a'_x + 2r'' \tag{5.18}$$

$$a''_y = a'_y + 2r'', \tag{5.19}$$

where $r'' = l_2\tan\beta$. The angle β can be obtained from the grating equations. Let d be the grating spacing in the hologram. We have

$$d(\sin(\alpha+\theta) - \sin\beta) = \lambda, \text{and} \tag{5.20}$$

$$d\sin\theta = \lambda. \tag{5.21}$$

Thus,

$$\beta = sin^{-1}[\sin(\alpha+\theta) - \sin\theta]. \tag{5.22}$$

Since both α and β are small, $\alpha \approx \sin\alpha \approx \frac{\lambda}{a}$ and $\tan\beta \approx \beta$, where λ is the wavelength. Hence

$$a''_x = \frac{a}{\cos\theta} + D\frac{\frac{\lambda}{a}\tan\theta}{\cos(\frac{\lambda}{a}+\theta)} + 2l_2\beta \tag{5.23}$$

$$a''_y = a + D\frac{\frac{\lambda}{a}\tan\theta}{\cos(\frac{\lambda}{a}+\theta)} + 2l_2\beta. \tag{5.24}$$

Because $\frac{\lambda}{a} << \theta$, the above can further simplified as

$$a''_x = \frac{a}{\cos\theta} + D\frac{\frac{\lambda}{a}\tan\theta}{\cos\theta} + 2l_2\beta \tag{5.25}$$

$$a''_y = a + D\frac{\frac{\lambda}{a}\tan\theta}{\cos\theta} + 2l_2\beta. \tag{5.26}$$

The minimum distance for l_2 is

$$\begin{aligned} l_2 &= \frac{D}{2\sin\theta} + \frac{D}{4\sin\theta} \\ &= \frac{3D}{4\sin\theta}. \end{aligned} \tag{5.27}$$

Thus, the final results contain three terms: the projection of the image to the hologram, the diffraction due to this projection, and the diffraction due to free space propagation from the hologram plane to the observation plane. The point size in the observation plane placed a limit on the maximum packaging density. As an example, assume $D = 20\text{mm}$, $b = 2a$, $\lambda = 0.6\mu\text{m}$ and $\theta = 40°$. If half of the point diameter is taken to be the tolerance for diffraction, it is possible to use a 36×36 array as the input. To get higher packing density, we should optimize the related parameters.

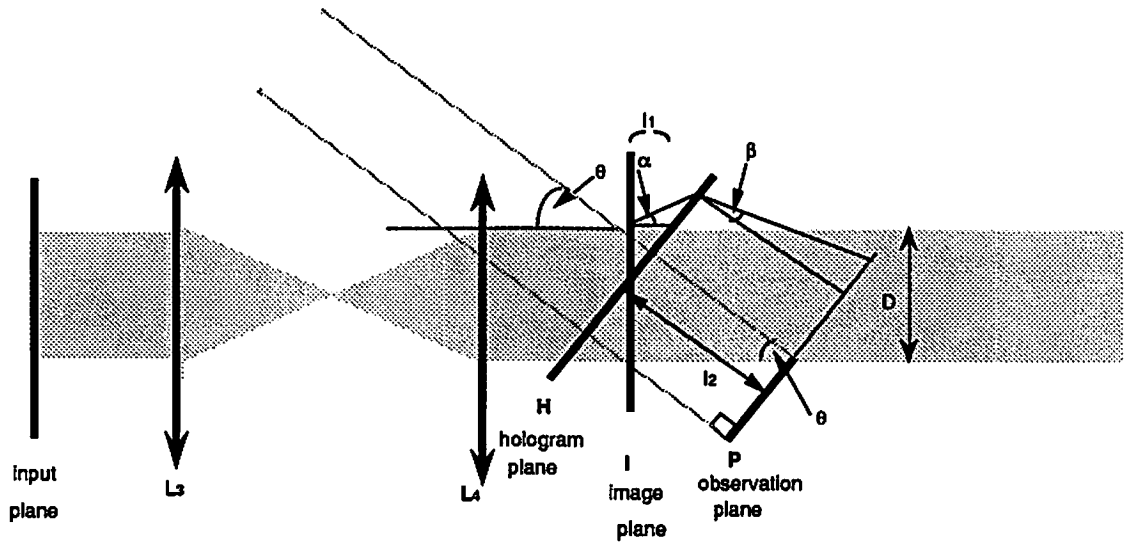


Figure 5.6: Diffraction analysis for the experimental setup.

Chapter 6

Optoelectronic Switches

A unified approach to designing the switch elements from the building block point of view is presented. This approach allows us to design various switch elements having the same functions. In order to further analyze the performance of different structured switch elements from practical aspects, signal modulations are taken into consideration. Several designs utilizing electronic amplifiers are considered. An experimental demonstration of an optically powered, optoelectronic 2×2 bypass/exchange switch element using polarization routing control is presented. Finally, we discuss the controlling schemes of such switch elements when used in the Omega network.

6.1 Structural Design on Optoelectronic 2×2 Switch Elements — a Unified Approach

The basic functions of a 2×2 switch element for multistage network applications are: to receive the signal, to amplify the signal, to set up the information path, and to transmit the signal. Therefore, an optoelectronic switch element consists of the following four building blocks:

- R — receiver (optical detector)
- C — connector (routing device)
- A — data amplifier

- T — transmitter (optical source).

The receiver detects the optical signal and transforms it to electrical form, whereas the transmitter, transforms the electrical signal to its optical form. The signal processing such as routing and amplification, take place in either electrical or optical domain depends on different designs. Various switch designs can be obtained by cascading these building blocks in different order. Practically, while cascading these blocks, the following rules have to be satisfied for consistency:

1. T must be after R.
2. Any block in between R and T should operate in the electrical domain.
3. Any block before R or after T should operate in the optical domain.
4. When R and T are directly connected, the structure is reduced to an optical device which can route and amplify the signals. This type of device is not in our scope of optoelectronic switches, thus, is not discussed here.

After applying these rules, we are left with six possible structures as shown in Table 6.1. We can also consider the application of these switch elements in a cascaded fashion, so that the equivalence between certain categories can further reduce the number of possible structures. For example, RCTA is equivalent to ARCT, and RATC is equivalent to CRAT in repeated sequence. The first and last element or group of elements may be different depending on the interface to outside world.

To decide which structure to use is actually a decision very dependent on the technology maturity. The inside structure of each building block and its performance are, also, related to the ways how the signals are modulated. We discuss the modulation methods in detail next.

6.2 Modulation Techniques

In general, there are several possible modulation methods that can be applied to binary optical signals. They are:

- intensity modulation (IM) — Intensity is used to modulate the 1's and 0's. For instance, bright beam represents 1's and dark beam represents 0's. It is similar to the amplitude modulation (AM) for electrical signals.

- wavelength modulation (WM) — Zeros and ones are carried by light beams of different wavelengths. For instance, wavelength λ_1 is used for 0's and wavelength λ_2 is used for 1's. It is similar to frequency modulation (FM) for electrical signals.

- phase modulation (PhM) — Phase is used to modulate the 1's and 0's of the signal.

- polarization modulation (PM) — The signal can be modulated as TE or TM mode according to its level (0 or 1). This kind of modulation is sometimes used in satellite communication.

- spatial modulation (SM) — Different levels of signals can also be modulated as different spatial patterns (this can be referred to "symbolic substitution"). If this modulation method is adopted, all the operations on the signal must be image-based type of operations.

Among these modulation techniques, phase modulation and spatial modulation are less attractive. Because phase modulation requires coherent demodulation, thus, interferometric techniques should be applied to retrieve the phase information. As in most applications, the phase of a light beam is very easily distracted while propagating. Therefore, the results from phase modulation and demodulation may not be cost-effective. Spatial modulation requires "pattern recognition" type demodulation techniques. This implies a complicated detection process, which may not be advantageous. Therefore, we focus on using the intensity, wavelength, and polarization modulations in this work.

6.2.1 Modulation Methods for Optically Powered Switch Elements

Optical powering has been shown to be an effective way to reduce interchannel electrical crosstalk in a dense optoelectronic integrated circuitry [30] [15]. We use

the same concept in our switch designs, and obtain optically powered optoelectronic switch element. Hence, the signals that pass through these switch elements include

1. data signal — Data information is represented as binary digits, thus, two levels are used to modulate the data signal.

2. control signal — The routing information for setting the switch to be in bypass state or in exchange state, is also a binary signal. In the self-routed packet switching environment, the control signal travels with the data packet and route the data packet accordingly.

3. power supply — The constant power supply for the operation of the switch element is transmitted along with the above two signals.

Various ways to modulate these signals can be applied although some may be impractical. For the moment, we show an exhaustive listing of all the possible combinations of modulation methods (Table 6.2).

For best performance, the modulation used for switch control signal should have the highest contrast ratio. PM seems to provide the highest contrast ratio, making it more suitable for this purpose. Furthermore, the modulation used for the data signal requires the fastest transition between the two levels to achieve higher data rate. Intensity modulation is considered appropriate for the data signal because it is easier to adjust the intensity of data signals in high speed. The number of optical sources required also must be taken into consideration. In most cases, the less optical sources required, the more cost-effective the design is. After all the above considerations, we conclude that scheme 10 and 27, both use polarization for the routing information and intensity modulation for the data signal, and utilize two wavelengths, are the best choices in practice. In the next section, we present examples of switch designs using various structures combined with these modulation schemes.

6.3 Single Stage 2×2 Switch Element Design Using Electrical Amplifier

Since optical amplifiers are still an active research area, we only consider the switch elements composed of electrical amplifiers. The structures we are interested in, thus, are RCAT, RACT, RATC, and CRAT. We can consider RCAT and RACT as in one class because the sequence of C and A in one structure can be inverted to get the other. Also, RATC and CRAT are similar in the sense that *RAT* are connected in both structures.

6.3.1 RCAT and RACT Structured Switch Element

They are both optically powered, electrically connected switches with electrical amplifiers. Figure 6.1 shows the schematic design using RCAT structure combined with modulation scheme 10, where the input signals have two wavelengths with one polarized to carry the routing information. The λ_1 beam is intensity modulated to carry the power signal and is polarization modulated carrying routing information. The λ_2 beam is pure data signal. The first pair filters which placed in front of the p-i-n photodiodes are wavelength selective filters. They are used to separate the data signals in wavelength λ_2 from the power/control signals in wavelength λ_1. Each λ_1 beam is split into two parts, one is designated to provide the power for the switch circuitry, the other one is processed further by a polarization beam splitter to activate the selected path. If λ_1 beam is polarized in $\odot$ mode, the switch element performs exchange function; if it is polarized in the direction perpendicular to $\odot$ mode, the switch element is bypassed. Broadcasting occurs when both modes are present.

As mentioned earlier, the RACT design is just a C-A exchanged version of RCAT switch element. One can obtain the RACT design based on similar principles and even combined with different modulation techniques.

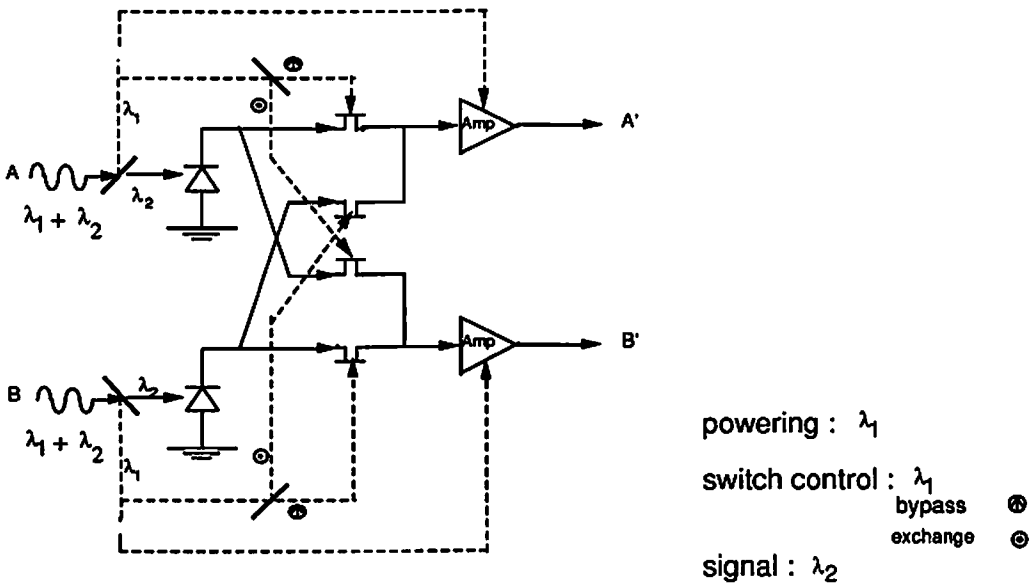


Figure 6.1: Optoelectronic 2 × 2 RCAT switch element design.

6.3.2 CRAT and RATC Structured Switch Element

They are optically powered, optically connected switches with electrical amplifiers. Figure 6.2 shows the schematic design using CRAT structure combined with modulation scheme 27, where the input signals have two wavelengths with one polarized carrying routing information. The λ_1 beam is intensity modulated to carry the power signal. The λ_2 beam is intensity modulated to carry the data signal and is polarization modulated to carry the routing information. The first pair of filters are wavelength selective filters used to filter out the power beams. The power beams are used to supply the power of the switch circuitry. The rest of the signals are routed by polarization beam splitters (shown as the second pair thick bars in the diagram) according to their polarizations. If λ_1 beam is polarized in $\odot$ mode, the switch element performs exchange function; if it is polarized in the direction perpendicular to $\odot$ mode, the switch element is bypassed. Broadcasting occurs when both modes are present.

To get the RATC structured switch element, we simply place the connector, which is located at the input end of the CRAT structure, to the output end. Various modulation schemes can be applied with minor changes.

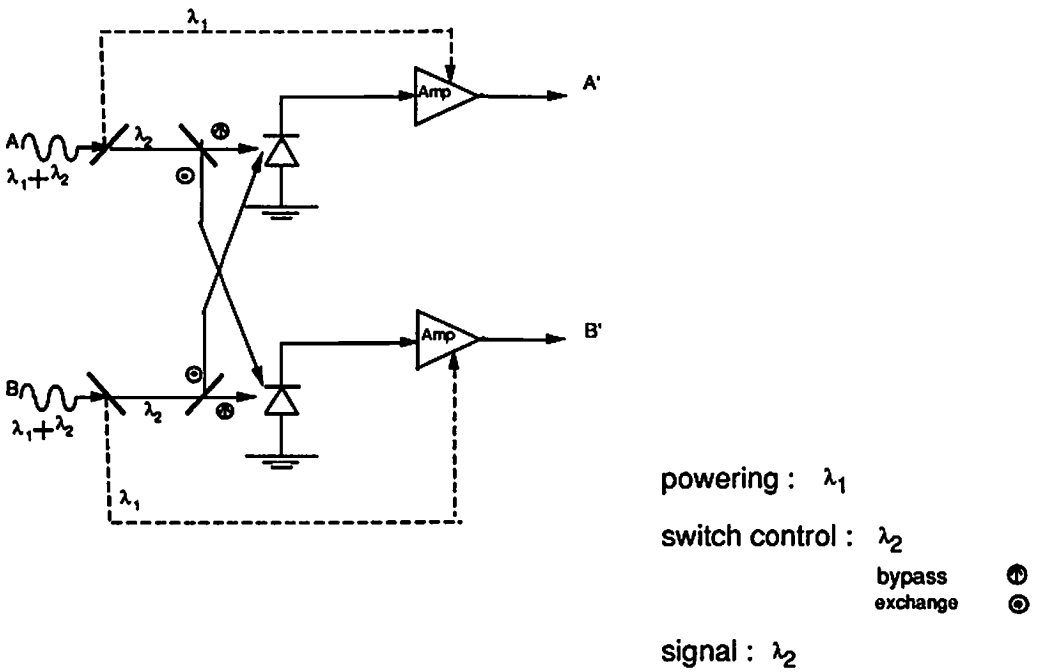


Figure 6.2: Optoelectronic 2×2 CRAT switch element design.

6.4 Implementation of Optically Powered Opto-electronic Switch with Polarization Control

Here we present an experimental implementation of an optically powered 2 × 2 bypass/exchange switch element using CRAT structure and polarization control.

6.4.1 Experimental Setup

The schematic design is shown in Fig. 6.2 and is described in section 6.3.2. Figure 6.3 shows the experimental setup. In this experiment, the power signals are provided in wavelength 0.8 μm by GaAs laser diodes, having typical output power 15 mW, for both channels. The data signals are provided by two linearly polarized 1.3 μm wavelength InGaAsP laser diodes modulated by the signal sources. These data laser diodes are pig-tailed with single mode fibers having core diameter of 60 μm. We mechanically modulate the polarization of the data signals by rotating the fiber tip in front of the polarizaion plates before the data signals are transmitted. In the switch element, two four-segment large area GaAs photovoltaic (PV) cell arrays receive the power signals and provide the open circuit voltage of 3.6 V to the receiver circuitry each. The polarization beam splitters (PBSs) route the incoming data signals to the designated receiver circuitry according to their polarizations. Each receiver circuit contains a planar $In_{0.53}Ga_{0.47}As$ p-i-n photodiode sensitive to wavelengths < 1.65 μm, and a high bandwidth Si bipolar transistor emitter follower circuit. Thus, it can receive and amplify the data signals.

6.4.2 Experimental Results

Figure 6.4 shows the curves obtained from the scope connected to the two output channels. The data signal from one input channel is routed to one output channel or the other by changing its polarization. This demonstrates the switching function of the setup.

The performance of a switch element, however, should be viewed from its power consumption, the crosstalk between the channels, the data bandwidth and the

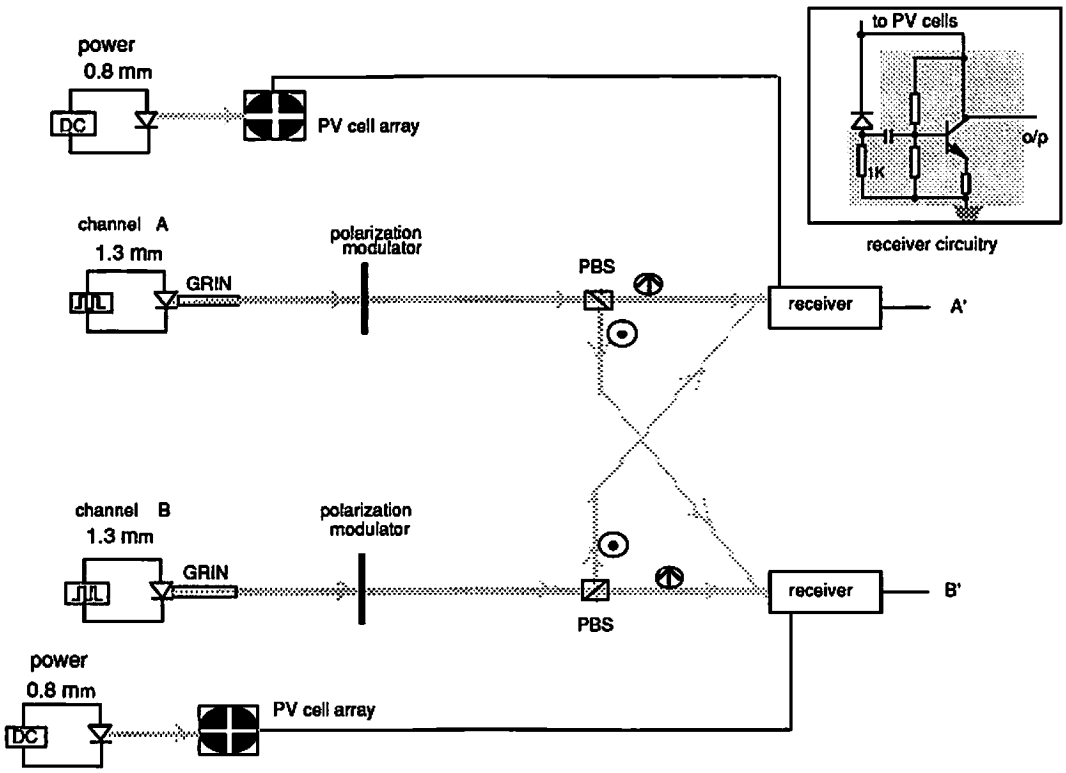


Figure 6.3: Experimental setup for optoelectronic 2×2 CRAT switch element.

switching speed. The first two are closely related to the packing density of an integrated switch circuitry. The last two set a limit to the real-time applications.

In this experiment, the optical signal power incident on each photodiode was 20 μW. The optical interconnection efficiency of the data channel was found to be 50% (measured from sending dc power on the receiver from a data laser). The quiescent power dissipation in each receiver was 2 mW. The optical-electrical power conversion efficiency of the PV cell was measured as 50%. Thus, the net power dissipation in each receiver (with PV cells) is then 4 mW.

The crosstalk χ is defined as the ratio of the root-mean-square (rms) signal voltage induced in one receiver due to an optical signal incident on the other. The electrical crosstalk between the two channels for various data rates using either electrical or optical powering was measured and is shown in Fig. 6.5. The polarization beam splitters were removed to eliminate the optical crosstalk. In the electrical powered case, the same power supply was used for both receiver circuits using 0.1 μF decoupling capacitors at the power supply terminals to improve the electrical isolation. As Fig. 6.5 shows, the crosstalk in the electrical powered case saturates at about -25 dB for data rates greater than 150 Mb/s, assuming non-return-to-zero (NRZ) coding. The crosstalk in the optically powered case was very small and hence could not be accurately measured. The -50 dB line is the limit of our measurement.

Detailed analysis of packing densities based on this experiment is developed [16]. It is concluded that even assuming modest device performance, switch packing densities greater than 100/cm^2 can be easily achieved using this switch design.

The data rate of this switch element is limited by the frequency response of the receiver circuits. In this experiment, the frequency response was not optimized. The 3 dB bandwidth of the two receivers was measured to be 140 Mb/s and 190 Mb/s (assuming NRZ coding). The switching speed is limited by the speed of modulating the polarization of the data signals. Since we only mechanically modulated the data signals, the switching speed should not be measured unless the modulation device is implemented. For this purpose, we suggest to use electro-optic modulator [18] or a polarization switchable laser [59] which can change the polarization of light beams at high frequency ($>$ 100 MHz).

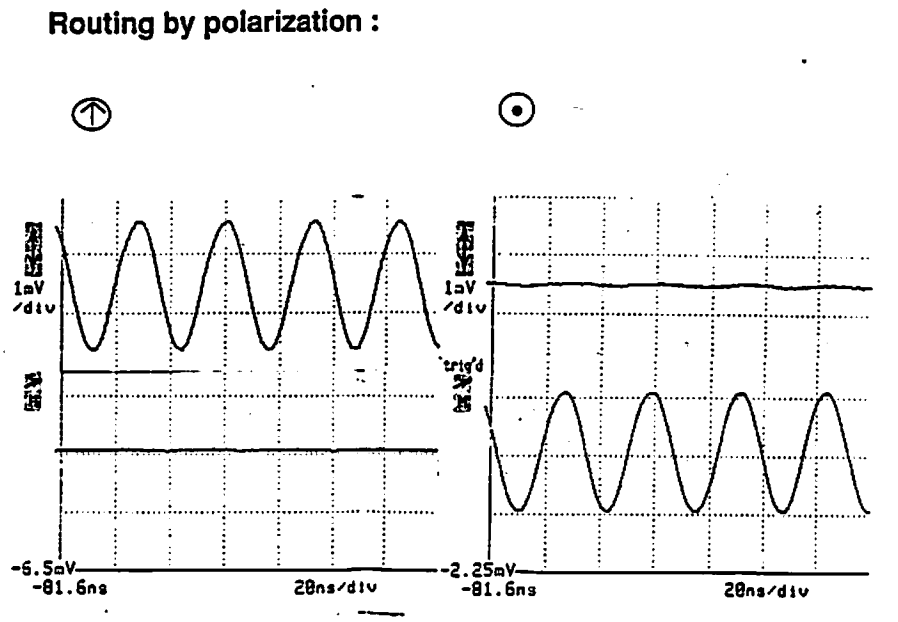


Figure 6.4: Demonstration of switching function of the experimented switch element.

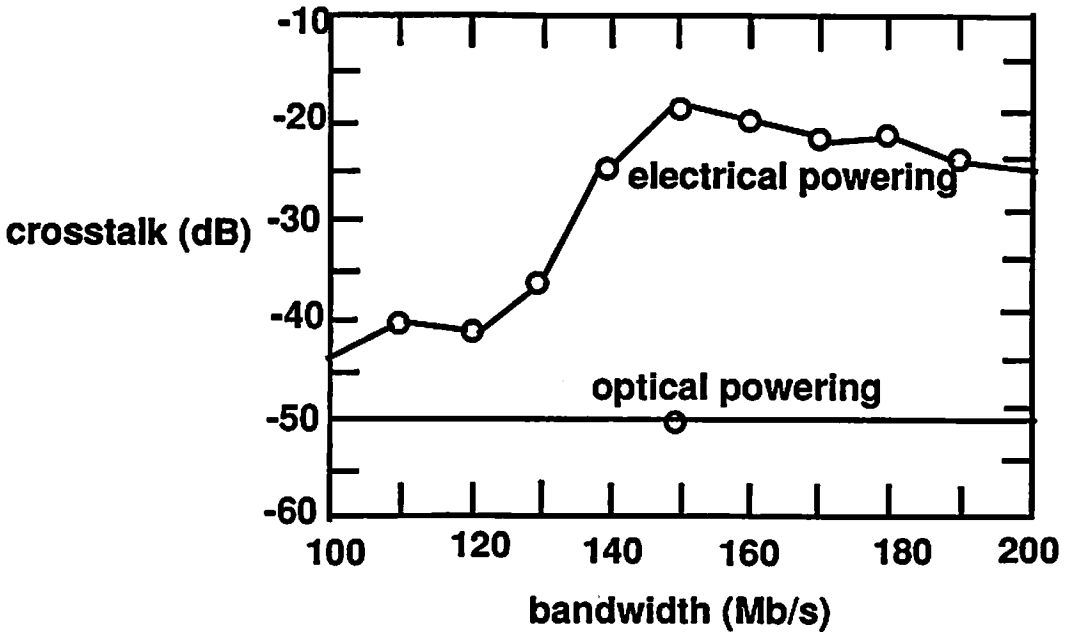


Figure 6.5: Crosstalk vs. frequency plot for optical and electrical powering the experimented switch element.

6.5 Switch Elements in Multistage Network

In the previous sections, we build the frame of an optoelectronic 2×2 switch element for a single stage network. To use these single stage switch elements in a multistage network, extra circuitry have to be added to apply various control strategy and switching mechanism. As introduced in section 2.2, either centralized control or distributed control can be used as control strategy, and switching mechanism including circuit switching, message switching, and packet switching can be applied. We here discuss from these two aspect here.

6.5.1 Centralized Control vs. Distributed Control

In centralized control, a powerful central controller takes the responsibility of the routing, congestion resolving, and sends out the command to each switch element. Here, the central controller should be very "intelligent" and the switch elements can be simple. The communication between the central controller and the switch elements should be reliable and efficient.

In distributed control, the duty of routing, congestion resolving, are distributed into the switch elements. There is no central controller to manage the network. Each switch element is built with "intelligence". This avoids the dependency on the central controller but increases the complexity of the switch elements.

6.5.2 Circuit Switching vs. Packet Switching

The three switch mechanisms considered here are circuit switching, message switching, and packet switching.

Circuit switching implies that there is a dedicated physical path between the input and the output channels. The communication is achieved in three phases: physical path establishment, data transfer, and physical path disconnect. This mechanism is easy to implement but it suffers from inefficient utilization of the channel capacity.

In message switching, the address of the destination is appended to the message itself and pass around the network. At each switch element, the entire message is received, stored and then transmitted to the next stage. Some signal processing,

such as arranging the message priorities, error control and recovery, can occur in each switch element after the message is stored. This scheme becomes more flexible and reliable by adding extra function in the switch elements. However, the delay time in the communication is large.

Packet switching is a mechanism which combines the advantages of the above two while minimizing the disadvantages of both. It is more like message switch, whereas, the message is chopped into packets for transmission.

structure	connector		amplifier	
	electrical	optical	electrical	optical
RCAT	*		*	
RACT	*		*	
RCTA	*			*
RATC		*	*	
ARCT	*			*
CRAT		*	*	

Table 6.1: Possible structures of optoelectronic 2×2 switch elements.

	power supply	control signal	data signal	no. of wavelengths used
1	λ_1	λ_1 (IM)	λ_1 (PM)	1
2	λ_1	λ_1 (IM)	λ_1, λ_2 (WM)	2
3	λ_1	λ_1 (PM)	λ_1 (IM)	1
4	λ_1	λ_1 (PM)	λ_1, λ_2 (WM)	2
5	λ_1	λ_1, λ_2 (WM)	λ_1 (IM)	2
6	λ_1	λ_1, λ_2 (WM)	λ_1 (PM)	2
7	λ_1	λ_1 (IM)	λ_2 (IM)	2
8	λ_1	λ_1 (IM)	λ_2 (PM)	2
9	λ_1	λ_1 (IM)	λ_2, λ_3 (WM)	3
10	λ_1	λ_1 (PM)	λ_2 (IM)	2
11	λ_1	λ_1 (PM)	λ_2 (PM)	2
12	λ_1	λ_1 (PM)	λ_2, λ_3 (WM)	3
13	λ_1	λ_1, λ_2 (WM)	λ_3 (IM)	3
14	λ_1	λ_1, λ_2 (WM)	λ_3 (PM)	3
15	λ_1	λ_1, λ_2 (WM)	λ_3, λ_4 (WM)	4
16	λ_1	λ_2 (IM)	λ_1 (IM)	2
17	λ_1	λ_2 (IM)	λ_1 (PM)	2
18	λ_1	λ_2 (IM)	λ_1, λ_3 (WM)	3
19	λ_1	λ_2 (PM)	λ_1 (IM)	2
20	λ_1	λ_2 (PM)	λ_1 (PM)	2
21	λ_1	λ_2 (PM)	λ_1, λ_3 (WM)	3
22	λ_1	λ_2, λ_3 (WM)	λ_1 (IM)	3
23	λ_1	λ_2, λ_3 (WM)	λ_1 (PM)	3
24	λ_1	λ_2, λ_3 (WM)	λ_1, λ_4 (WM)	4
25	λ_1	λ_2 (IM)	λ_2 (PM)	2
26	λ_1	λ_2 (IM)	λ_2, λ_3 (WM)	3
27	λ_1	λ_2 (PM)	λ_2 (IM)	2
28	λ_1	λ_2 (PM)	λ_2, λ_3 (WM)	3
29	λ_1	λ_2, λ_3 (WM)	λ_2 (IM)	3
30	λ_1	λ_2, λ_3 (WM)	λ_2 (PM)	3
31	λ_1	λ_2 (IM)	λ_3 (IM)	3
32	λ_1	λ_2 (IM)	λ_3 (PM)	3
33	λ_1	λ_2 (IM)	$\lambda_{3a}, \lambda_{3b}$ (WM)	4
34	λ_1	λ_2 (PM)	λ_3 (IM)	3
35	λ_1	λ_2 (PM)	λ_3 (PM)	3
36	λ_1	λ_2 (PM)	λ_3, λ_4 (WM)	4
37	λ_1	λ_2, λ_3 (WM)	λ_4 (IM)	4
38	λ_1	λ_2, λ_3 (WM)	λ_4 (PM)	4
39	λ_1	λ_2, λ_3 (WM)	λ_4, λ_5 (WM)	5

Table 6.2: Possible modulations used in an optically powered optoelectronic 2×2 switch element.

Chapter 7

Space Multiplexed 3-D Omega Networks

In this chapter, we investigate various approaches to reduce the overall complexity of optical 3-D Omega networks. We present techniques for spatially arranging the channels so that all stages of shuffles can be performed by a single set of optics. Throughout the discussion, we assume that network routing is performed by an intelligent central controller. We assume there are no network contention problems or the contentions are resolved by the controller. The network is considered as a synchronous packet switching network such that all packets of data at every channel and every stage have the same length. Also, we assume that the propagation time for the optical shuffles and the processing time at the switch elements are the same for every channel at every stage.

7.1 Background

7.1.1 Principle of Pipelining

Pipelining is used to exploit temporal parallelism and to improve the throughput of existing resources in an overlapped manner. The temporal parallelism is achieved by considering the processing as a sequence of stations in an assembly line, and dividing the operations into smaller parts that are processed by the sequence of stations. The concept of pipelining in computer architecture was introduced in 1970s [6]. Various

types of pipelining have been developed from hardware levels to software levels, from static to dynamic fashion, from one dimension to multi-dimensions [24].

We define two important parameters to measure the performance of a pipelined system, namely, *speedup* and *efficiency*. Speedup (S) is defined as the gain obtained in time complexity through pipelining. Efficiency (η) is defined as the percentage time the pipeline is totally occupied. Assume that a job is divided into p subtasks, that each subtask must be processed by m stations, and that each processing step requires a single unit of time. Without pipelining, the total processing time for the job is pm; for a pipelined system, the total processing time becomes the number of steps needed to move p pieces of data past m stations, or $p+m-1$. Therefore, the speedup is

$$S = \frac{pm}{p+m-1}. \tag{7.1}$$

The efficiency of the pipe is

$$\eta = \frac{p}{p+m-1}. \tag{7.2}$$

7.1.2 Pipelining in Volume MINs

A synchronous packet switched multistage interconnection network can be considered as pipeline. The packets of data to be transmitted are the subtasks of a divided job. The fixed connections (such as 2-D shuffles) and switch elements at each stage are the processing stations. The packets pass through the multistage network and reach the destination in a pipelined, synchronized way. If we assume that all the switch elements have the same processing speed, and that the propagation time is the same for all channel connections at each stage, the bottleneck problem created by the slowest processing station in a general pipelined computer system is eliminated.

7.1.2.1 Linear Multistage Networks

Figure 7.1 shows the data flow in a 3-D Omega network having m stages and $N = 2^m = n \times n$ channels. Here, I_i is the i^{th} stage input plane, and O_i is the i^{th}

stage output plane, where $i \in \{0, 1, \ldots, m-1\}$. The optics between these input and output planes performs the 2-D perfect shuffles at every stage. The optoelectronic 2×2 bypass/exchange switch elements, each one contains two sources, detectors, and performs the dynamic interconnections, are placed at the interface of O_{i-1} and I_i. Buffers can be implemented at each stage to resolve the network contention problems. However, we concentrate on the architectural design here, and assume there is no contention problems or the contentions are solved by central controller.

Assume there are p packets of data to be transmitted for each of the N channels (they are analogous to the p subtasks in a pipeline). For one packet, let the propagation time for a single stage optical shuffle be t_O, and the switch element processing time in a single stage be t_{OE}. The total time for all the Np packets to pass through the m stage network containing N channels in parallel is $(p + m - 1)(t_O + t_{OE})$ (same for each channel). This time is defined as the *time complexity.*

We define the *mechanical complexity* as the amount of work to build the system. It is different from measures of the *capacity* of a system, such as *space-bandwidth product* (SBWP) of an optical system and the *packing density* of an optoelectronic integrated circuit. The capacity of a system is closely related to the implementation devices and the algorithms used. We discuss the capacity issue later. Let the optics that performs a single stage 2-D shuffle have mechanical complexity h_O, and the optoelectronic switch element array at each stage have mechanical complexity h_{OE}. The total mechanical complexity of an m stage *linear multistage Omega network* is $m(h_O + h_{OE})$.

7.1.2.2 Time Multiplexed Multistage Networks

Because the fixed interconnection patterns performed by optics are the same at each stage of a volume Omega network, replicating the same set of optics for each stage may not be necessary. *Time multiplexing* can be used to avoid the hardware replication.

As shown in Fig. 7.2, packets of data are injected to the input plane, they pass through the optical shuffle, and through the bypass/exchange switch array after the output plane. These data packets are then fed back to the input plane for injection into the next stage. Since the bypass/exchange switch elements must be

reset properly for each stage, there must be buffers used to hold the data while the switch elements are reset. Once the entire switch array is reset for the following stage, the data is released from the buffers into the shuffle. This process repeats until the data has traversed all m stages, at which point it emerges from the system. Then a second set of packets may be injected, and the process repeats. Therefore, the packets of data can enter the network only after the previous ones have been processed recursively by the single stage network m times. Although we avoid the replication of optics, the embedded pipelined processing capability of multistage networks is lost.

To consider the complexity of a *time multiplexed* multistage network, we let the mechanical complexity of the feedback system be $h_{F,T}$, where the subscript F indicates the feedback system and the subscript T indicates the *time multiplexing.* Let the processing time for one packet be $t_{F,T}$. Let the buffers have mechanical complexity h_B and time complexity t_B (where the subscript B indicates buffers). Then, the mechanical complexity of this system is $h_O + h_{OE} + h_{F,T} + h_B$, while the time complexity becomes $pm(t_O+t_{OE}+t_{F,T}+t_B)$. For such a simple feedback system, both the mechanical and time complexity can be considered small. As compared to a *linear multistage network*, the savings in mechanical complexity translates to increased time consumption even if t_B is negligible.

7.1.2.3 Space Multiplexed Multistage Networks

An alternative to time multiplexing is *space multiplexing.* In a time multiplexed multistage network, only one set of optics and optoelectronic switch elements is built, and is reused for every stage. In a space multiplexed multistage network, there are switch arrays dedicated to each individual stage, only the optical hardware (called *macro optics*) are shared by all stages simultaneously.

Figure 7.3 shows the scheme. The data packets flow in a helix-type track. They enter the network at I_0, propagate through the macro optics which performs shuffle operation, then they reach O_0, pass through the dynamic bypass/exchange switch array, and are fed back to I_1, continue in the same fashion until they reach O_{m-1}. Similar to the linear multistage networks, buffers can be implemented at each stage to resolve the network contention problems, but they are not used to hold the data

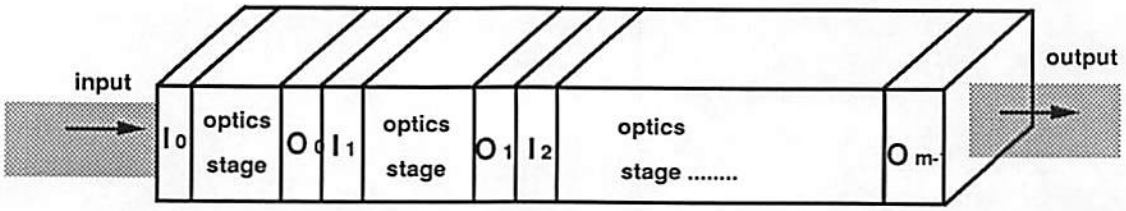


Figure 7.1: Data flow in linear 3-D Omega network.

	time complexity	mechanical complexity
linear	$(p+m-1)(t_O+t_{OE})$	$m(h_O+h_{OE})$
time multiplexing	$pm(t_O+t_{OE}+t_{F,T}+t_B)$	$h_O+h_{OE}+h_{F,T}+h_B$
space multiplexing	$(p+m-1)(t_O+t_{OE}+t_{F,S})$	$xh_O+mh_{OE}+h_{F,S}$

Table 7.1: Comparison of different structures for 3-D Omega networks.

packets as in the time multiplexed multistage networks. Thus, we do not consider the buffers in this architectural design.

Similarly, we let the mechanical complexity of the feedback system be $h_{F,S}$, and the processing time be $t_{F,S}$, where subscript F indicates the feedback system and S indicates the space multiplexing. The mechanical complexity of such system is $xh_O + mh_{OE} + h_{F,S}$, where $x \geq 1$ is a factor describing the relative mechanical complexity of building the macro optics shuffles compared to the complexity h_O. The time complexity is $(p+m-1)(t_O+t_{OE}+t_{F,S})$. To compare this with a linear multistage network, the time complexity is increased by $(p+m-1)t_{F,S}$. The saving in mechanical complexity is achieved only when $(xh_O + h_{F,S}) < mh_O$.

Table 7.1 shows the comparison of these three structures. It is clear that the linear multistage network has minimum time complexity, and the time multiplexed network has maximum. For mechanical complexity, the time multiplexed network has a minimum, but the comparison between the linear network and the space multiplexed network depends on the implementation of the macro optics and the feedback system. Next, we investigate the implementation of a space multiplexed multistage network, and compare it with a linear multistage network.

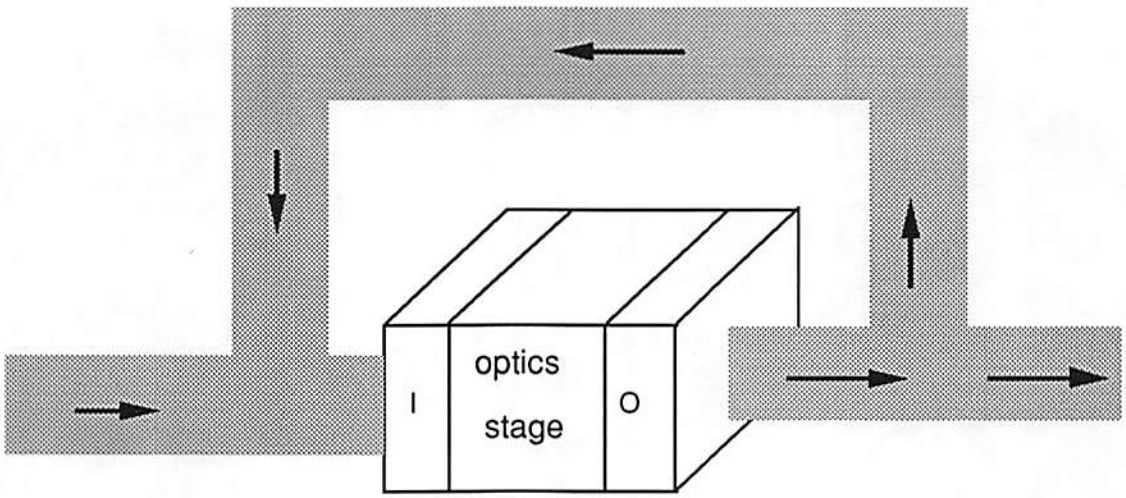


Figure 7.2: Data flow in time multiplexed 3-D Omega network.

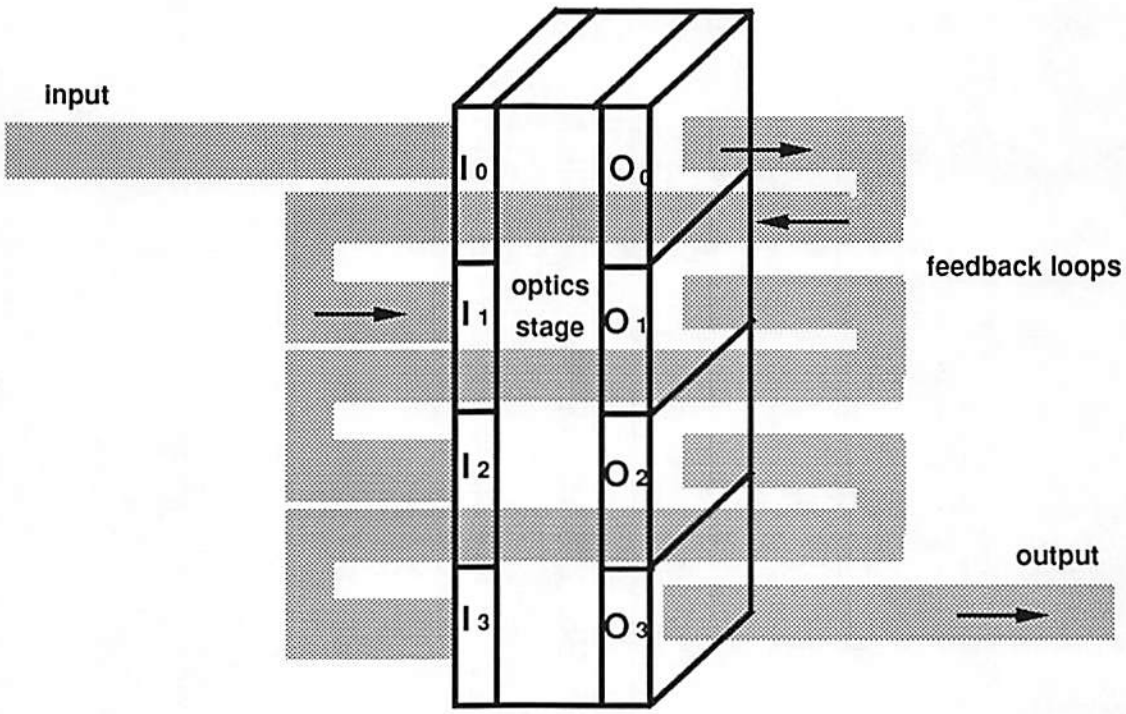


Figure 7.3: Data flow in space multiplexed 3-D Omega network.

7.2 Implementation of Space Multiplexed Omega Networks

Here we demonstrate that the one-copy algorithm can be used in space multiplexed Omega networks.

7.2.1 One-Copy Algorithm and Space Multiplexed Omega Networks

In the *one-copy algorithm* [7], the input plane contains four quadrants, labelled Q_i, where $i \in \{0,1,2,3\}$. First, the input plane is magnified twice. The four magnified quadrants are shifted differently and then are stacked to produce the 2-D shuffled results (Figs. 4.2(b) and 4.3(b)). The system which performs the one-copy algorithm can be described as a system containing four shift-invariant subsystems, capable of performing different shifts on the four quadrants of a magnified input plane, and superimposing these four shifted quadrants.

In a multistage Omega network, the 2-D shuffles are performed at every stage. This indicates that using the one-copy algorithm, the same quadrants from each of the different stages must be processed by the same subsystem. Hence, in a space multiplexed Omega network, the *quadrants* are to be spatially multiplexed when the one-copy algorithm is used. This applies to the networks either use 2-D folded or 2-D separable shuffles.

The method for the networks using 2-D folded shuffles is shown schematically in Fig. 7.4. The magnified inputs of all the stages are divided into four quadrants Q_0, Q_1, Q_2 and Q_3. In this figure, all quadrants from the same stage of the multistage network have the same shading. These quadrants are placed into the four blocks of a macro input plane according to the embedded index of the quadrants. This macro input plane, which contains the input channels from all the stages, is processed by a macro facet hologram still having four facets. The output channels of different stages then appear in non-overlapping regions in the macro output plane. Thus, instead of replicating the four-facet holograms for all stages, we are able to perform 2-D shuffles optically for all stages with a single four-facet hologram. The consequences of such an arrangement are discussed in section 7.4.

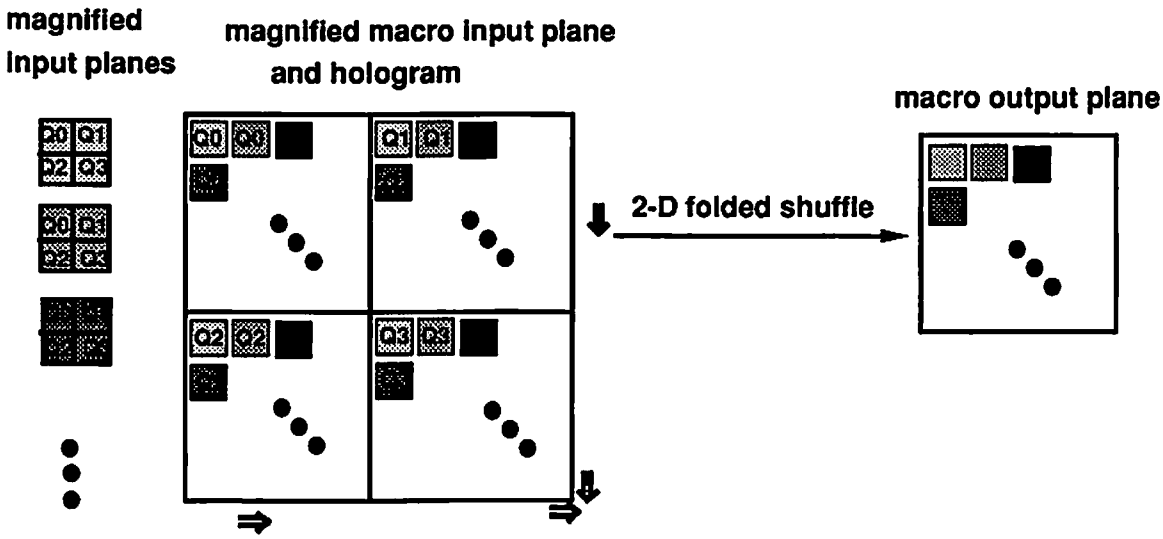


Figure 7.4: Space multiplexing many stages of an Omega network in quadrants.

7.2.2 Demonstration with a Facet Hologram System

We demonstrate a four-stage, space multiplexed Omega network having 16 channels. In this demonstration, we assume that all the switch elements at all the four stages are in bypass state. The initial input is a 4 × 4 binary pattern of a letter "P". If this pattern passes through four stages of 2-D folded shuffle and bypass state switch arrays, the output will be identical to the input. Figure 7.5 shows the array of propagating signals at each stage (clock cycle) of the Omega network.

A diagram of the same data pattern in a space multiplexed Omega network using the one-copy algorithm described previously is shown in Fig. 7.6. Notice that there is only one macro stage, and the steps in the sequence show the signals in corresponding stages in a linear Omega network. Here the input pattern at every clock cycle is split into four quadrants and arranged in corresponding locations in the macro input plane. At the macro output plane, the pattern for each stage output appears in one quadrant. It passes through the bypass state switch array and is fed back (with proper magnification and segmentation into quadrants) to the appropriate locations in the macro input plane.

We can also collapse these four clock cycles into one step to demonstrate the space multiplexing of the channels from all four stages (Fig. 7.7). The macro

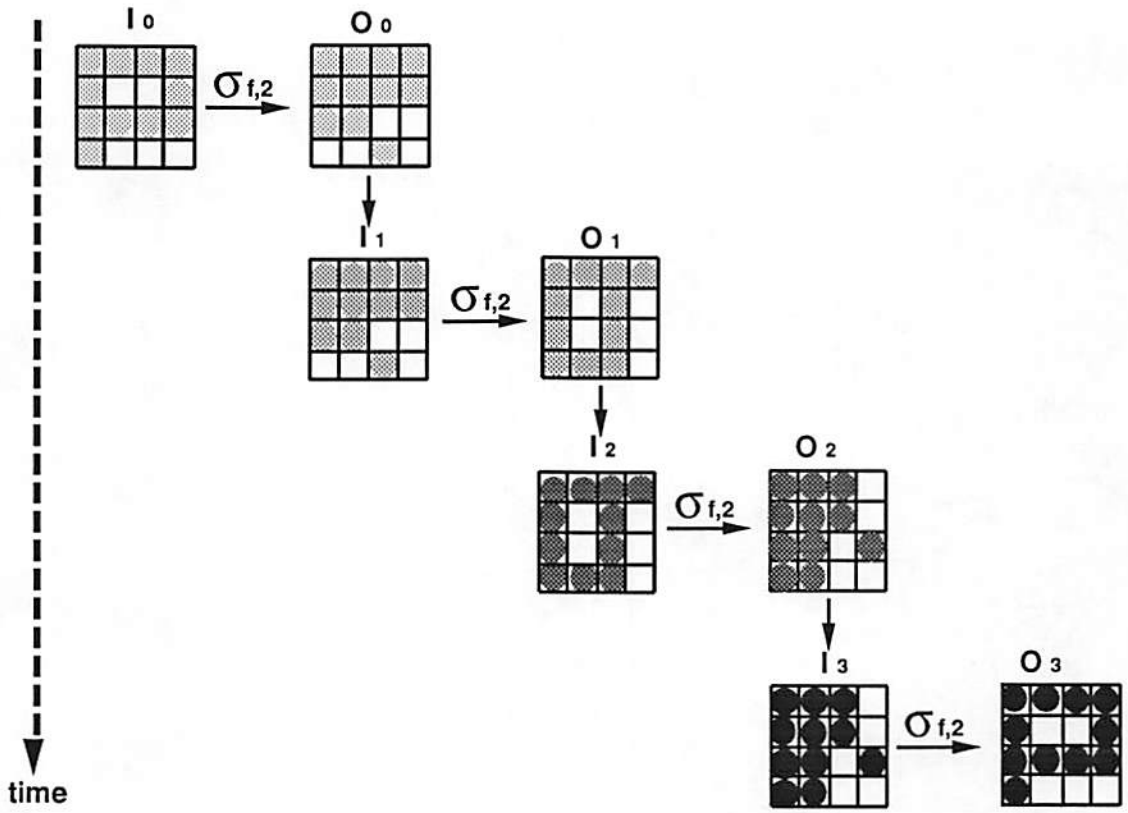


Figure 7.5: Linear pipelined 4 stage Omega network having 16 channels, "P" as the initial input pattern, and all switches in bypass state.

input plane is an overlap of the input planes of four consecutive clock cycles (i.e. four stages) and the output plane contains the outputs of all four consecutive clock cycles (i.e. four stages). The shuffle operations for these four stages take place simultaneously.

We have experimentally demonstrated the space multiplexing using a four clock cycle overlapped macro input plane with the setup in Fig. 5.4. The combined input plane, the expected output plane, and the experimental result are depicted in Fig. 7.7.

7.3 Mathematical Reasoning

The reason why the one-copy algorithm works perfectly for a space multiplexed Omega network can be explained mathematically using the same set of notation

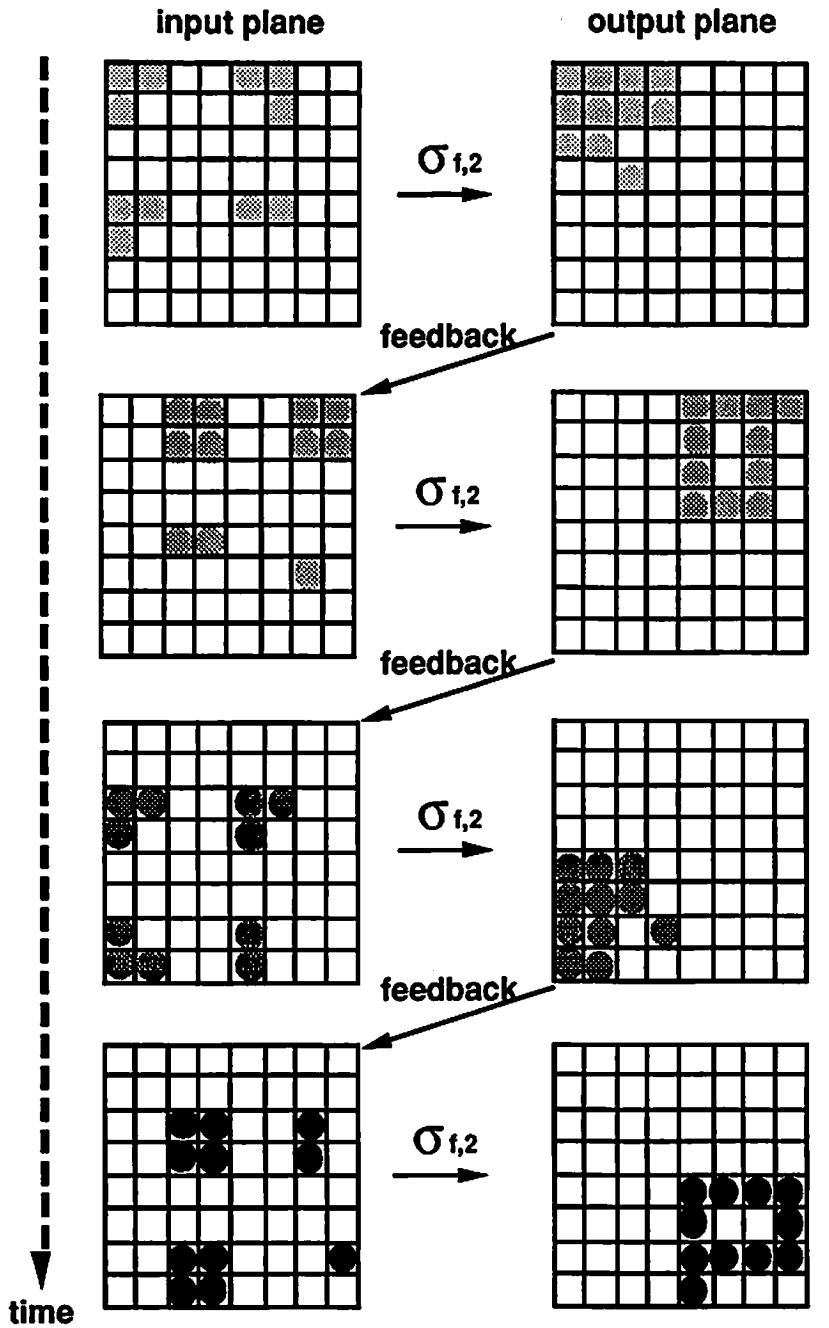


Figure 7.6: Spatial multiplexed 4 stage Omega network having 16 channels, "P" as the initial input pattern, and all switches in bypass state.

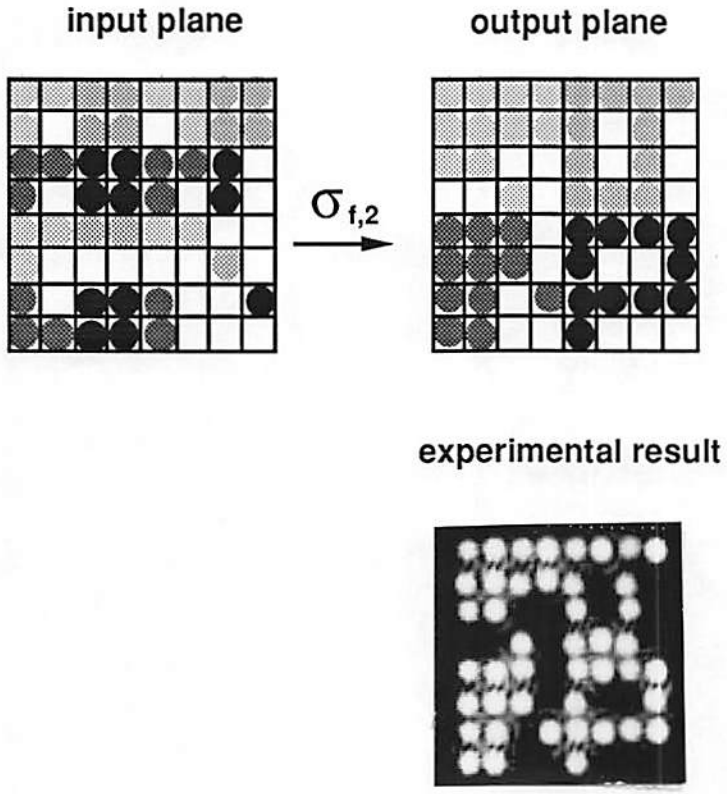


Figure 7.7: Demonstration of space multiplexing in one step.

developed in Chapter 4. Let there be $N = 2^m = n \times n$ channels in the network. The spatial location of any channel in a single stage is denoted as the row index and column index pair,

$$(p, q) = (\underbrace{b_{m-1}b_{m-2}\ldots b_{\frac{m}{2}}}_{\frac{m}{2}\text{ digits}}, \underbrace{b_{\frac{m}{2}-1}b_{\frac{m}{2}-2}\ldots b_1 b_0}_{\frac{m}{2}\text{ digits}})_2, \tag{7.3}$$

where b_is are binary digits.

In a space multiplexed Omega network, in order to arrange all the m stage inputs (or outputs) in a macro input (or output) plane, we must let the macro input (or output) plane be $cn \times cn$ in size, where c is an expansion factor such that $c^2 \geq m$. The macro planes contains extra unused spatial channels when $c^2 > m$. We enumerate all the m stages ranging from 0 to $m-1$, and represent this stage index as (y_r, y_c), where $0 \leq y_r, y_c < c$. The $(y_{r,i}, y_{c,i})^{th}$ pair is used to separate the channels of i^{th} stage from other stages. Thus, each stage has a distinct (y_r, y_c) pair. For example, the i^{th} stage is described as the $(y_{r,i}, y_{c,i})^{th}$ stage, and $i = y_{r,i}c + y_{c,i}$.

If we suppose the expansion factor is $c = 2^t$, the spatial locations of the channels from all the m stages in the macro input plane can then be described by new row and column indices, where each index contains $\frac{m}{2} + t$ bits. To spatially multiplex the quadrants from all the stages, the channels of the i^{th} stage become

$$\begin{aligned}(p', q') &= (b_{m-1}(y_{r,i})b_{m-2}\ldots b_{\frac{m}{2}}, b_{\frac{m}{2}-1}(y_{c,i})b_{\frac{m}{2}-2}\ldots b_0) \\ &= (\underbrace{b_{m-1}\ \underbrace{0\ldots0}_{t\text{ digits}}\ b_{m-2}\ldots b_{\frac{m}{2}} + \frac{n}{2}y_{r,i}}_{(\frac{m}{2}+t)\text{ digits}}, \\ &\quad \underbrace{b_{\frac{m}{2}-1}\ \underbrace{0\ldots0}_{t\text{ digits}}\ b_{\frac{m}{2}-2}\ldots b_1 b_0 + \frac{n}{2}y_{c,i}}_{(\frac{m}{2}+t)\text{ digits}}).\end{aligned} \tag{7.4}$$

Similarly, we apply the one-copy algorithm on this macro input plane. In the first step, we magnify the macro input plane twice. The indices of the channels become

$$(\tilde{p}', \tilde{q}') = (\underbrace{(b_{m-1}\ \underbrace{0\ldots0}_{t\text{ digits}}\ b_{m-2}\ldots b_{\frac{m}{2}}0 + ny_{r,i},}_{(\frac{m}{2}+t+1)\text{ digits}}$$
$$\underbrace{b_{\frac{m}{2}-1}\ \underbrace{0\ldots0}_{t\text{ digits}}\ b_{\frac{m}{2}-2}\ldots b_1 b_0 0 + ny_{c,i}).}_{(\frac{m}{2}+t+1)\text{ digits}} \tag{7.5}$$

In the second step, the shift operations take place. Thus, the indices of the channels become

$$(\hat{p}', \hat{q}') = (\underbrace{(b_{m-1}\ \underbrace{0\ldots0}_{t\text{ digits}}\ b_{m-2}\ldots b_{\frac{m}{2}} b_{\frac{m}{2}-1} + ny_{r,i},}_{(\frac{m}{2}+t+1)\text{ digits}}$$
$$\underbrace{b_{\frac{m}{2}-1}\ \underbrace{0\ldots0}_{t\text{ digits}}\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1} + ny_{c,i}).}_{(\frac{m}{2}+t+1)\text{ digits}} \tag{7.6}$$

After the superposition step, the most significant bits of both row and column indices are removed, and the spatial location of the channels becomes

$$(p', q') = (\underbrace{(\underbrace{0\ldots0}_{t\text{ digits}}\ b_{m-2}\ldots b_{\frac{m}{2}} b_{\frac{m}{2}-1} + ny_{r,i},}_{(\frac{m}{2}+t)\text{ digits}}$$
$$\underbrace{\underbrace{0\ldots0}_{t\text{ digits}}\ b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1} + ny_{c,i}).}_{(\frac{m}{2}+t)\text{ digits}} \tag{7.7}$$

Since $n = 2^{\frac{m}{2}}$, the binary representation for n is a one followed by $\frac{m}{2}$ digits of zeros. Then, $ny_{r,i}$ can be represented as a $(\frac{m}{2} + t)$ digit binary number having the binary representation of $y_{r,i}$ as the most significant t digits followed by $\frac{m}{2}$ digits zeros. Similar reasoning is applied for $ny_{c,i}$. Therefore, Eq. (7.7) becomes

$$(p', q') = ((y_{r,i})b_{m-2}\ldots b_{\frac{m}{2}} b_{\frac{m}{2}-1},\ (y_{c,i})b_{\frac{m}{2}-2}\ldots b_1 b_0 b_{m-1}), \tag{7.8}$$

where the pair $(y_{r,i}, y_{c,i})$ indicates the spatial location of the output channels of stage i in the macro output plane. Because the (y_r, y_c) pair is different for each stage, the output of all the stages appears as non-overlapped regions. The rest of the digits show that a 2-D folded shuffle is performed.

Similar derivation can be obtained for 2-D separable shuffles. These derivations are also valid for the case when $c \neq 2^t$. Here, the only difference is in that $y_{r,i}$ and $y_{c,i}$ are $(t+1)$-tuple binary numbers (assuming $2^t < c < 2^{t+1}$).

7.4 Practical Considerations

We have demonstrated a hybrid system to implement a space multiplexed Omega network with a facet hologram. For practical applications, it is important to consider limiting factors such as the space-bandwidth product (SBWP) of the optics, the physical size of the system when building space multiplexed Omega networks.

We have discussed the diffraction effects on this system and the SBWP requirement in Chapter 5. In short, the packing density depends on the resolution in the observation plane, and is diffraction limited. For a single stage network having N channels, the total SBWP of the system using the one-copy algorithm is $4N$. In the space multiplexed m stage network, the total SBWP requirement becomes $4mN$. In practical implementation, when the macro plane has size $cn \times cn$, the SBWP requirement is $4c^2N$, where $c^2 \geq m$. This generally means that a larger hologram and larger lenses are necessary.

The physical size and the complexity of the complete system depend on how to break the channels from the macro output plane into quadrants and place these quadrants appropriately within the macro input plane. This break-and-put function is also required for the initial system input. Thus, we need a *pre-processing* step for the initial input, and a *backward processing* step embedded in the feedback loop. We call the step of performing 2-D shuffles *forward processing*. Therefore, the complete system is composed of *pre-processing*, *forward processing*, switch array, and *backward processing* as shown in Fig. 7.8.

From our analysis in section 5.3, the distance (l_2) from forward processing to the switch array depends on the input plane size and the angle between the two beams which make the forward processing hologram (Eq. 5.27). If a space multiplexed

plane has size D mm $\times D$ mm, and the angle is θ, then $l_2 = \frac{3D}{4\sin\theta}$ mm. A 4-f imaging system is required to image the output from the switch array to the front plane of the backward processing. Suppose the aperture of the imaging lenses is $1.5D$ mm, and the F number is 5, the focal length of these lenses is then $7.5D$ mm. Let the same imaging system be used to image the output of the backward processing to the front plane of the forward processing. The total path for one stage is approximately $60D + l_2$ mm. The latency per stage is

$$\text{latency/stage} = \frac{(60D + \frac{3D}{4\sin\theta}) \text{ mm}}{3 \times 10^{11} \text{ mm/sec}}. \tag{7.9}$$

The maximum path length difference in one stage is between the upper left quadrant and the lower right quadrant. It is estimated as $\sqrt{(D/\sqrt{2})^2 + l_2^2}$ mm. The maximum time skew in a stage is then

$$\text{max time skew/stage} = \frac{\sqrt{(D/\sqrt{2})^2 + l_2^2} \text{ mm}}{3 \times 10^{11} \text{ mm/sec}}. \tag{7.10}$$

Assuming typical values of $D = 20$ mm and $\theta = 40^o$, so that more than 1000 channels are in the system, the latency per stage is about 4 nsec, and the time skew is approximately 91 psec.

Consider the same example used in our demonstration (Figs. 7.6 and 7.7). The output from each stage must be broken into four quadrants and then put into the input plane separated into the proper locations. Figure 7.9 is a schematic diagram of the forward and backward operations as quadrants. The notation Q_i^j indicates the i^{th} quadrant of the j^{th} stage. Thus for an m stage network, $0 \leq i \leq 3$ and $0 \leq j < m$. The mathematical description of this backward processing is to move the channels

$$(p', q') = ((y_{r,i})b_{m-2}b_{m-3}\ldots b_{\frac{m}{2}-1},\ (y_{c,i})b_{\frac{m}{2}-2}b_{\frac{m}{2}-3}\ldots b_{m-1}) \tag{7.11}$$

from the output plane to the channels

$$(p', q') = (b_{m-2}(y_{r,i+1})b_{m-3}\ldots b_{\frac{m}{2}-1},\ b_{\frac{m}{2}-2}(y_{c,i+1})b_{\frac{m}{2}-3}\ldots b_{m-1}) \tag{7.12}$$

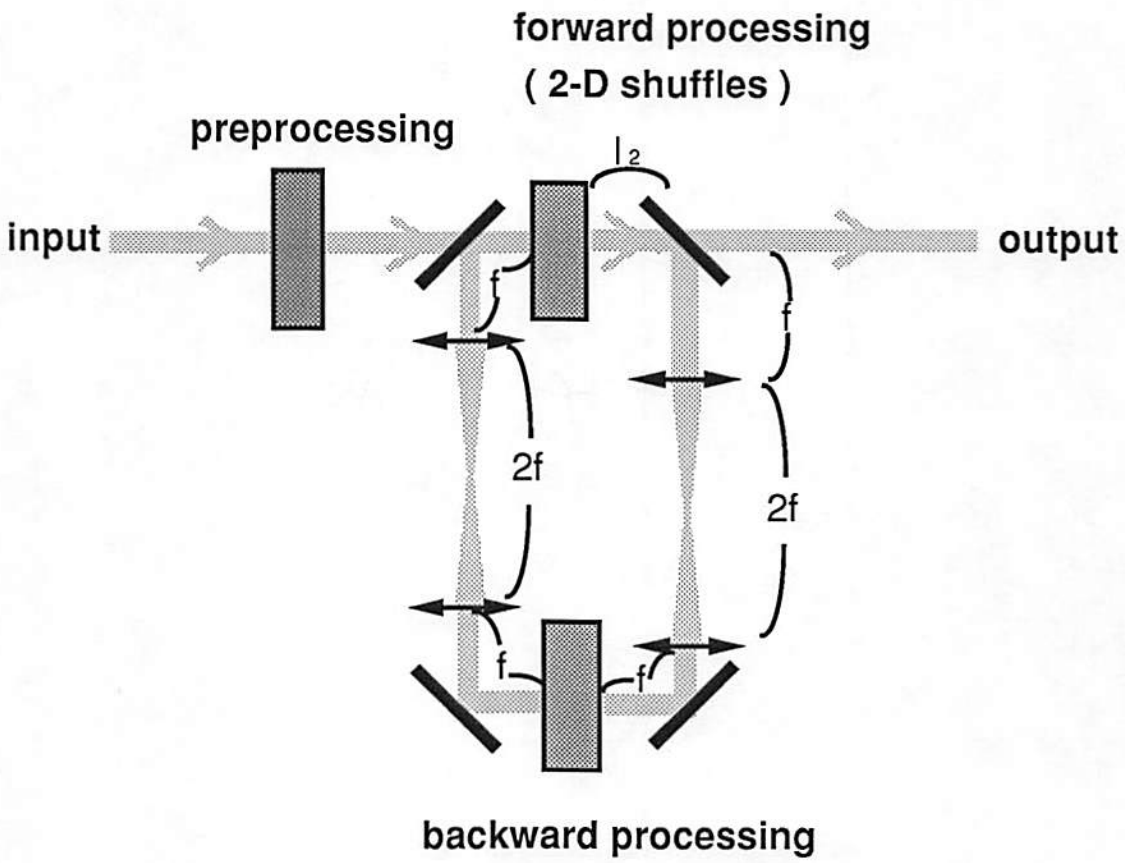


Figure 7.8: The complete system includes pre-processing, forward, and backward processing.

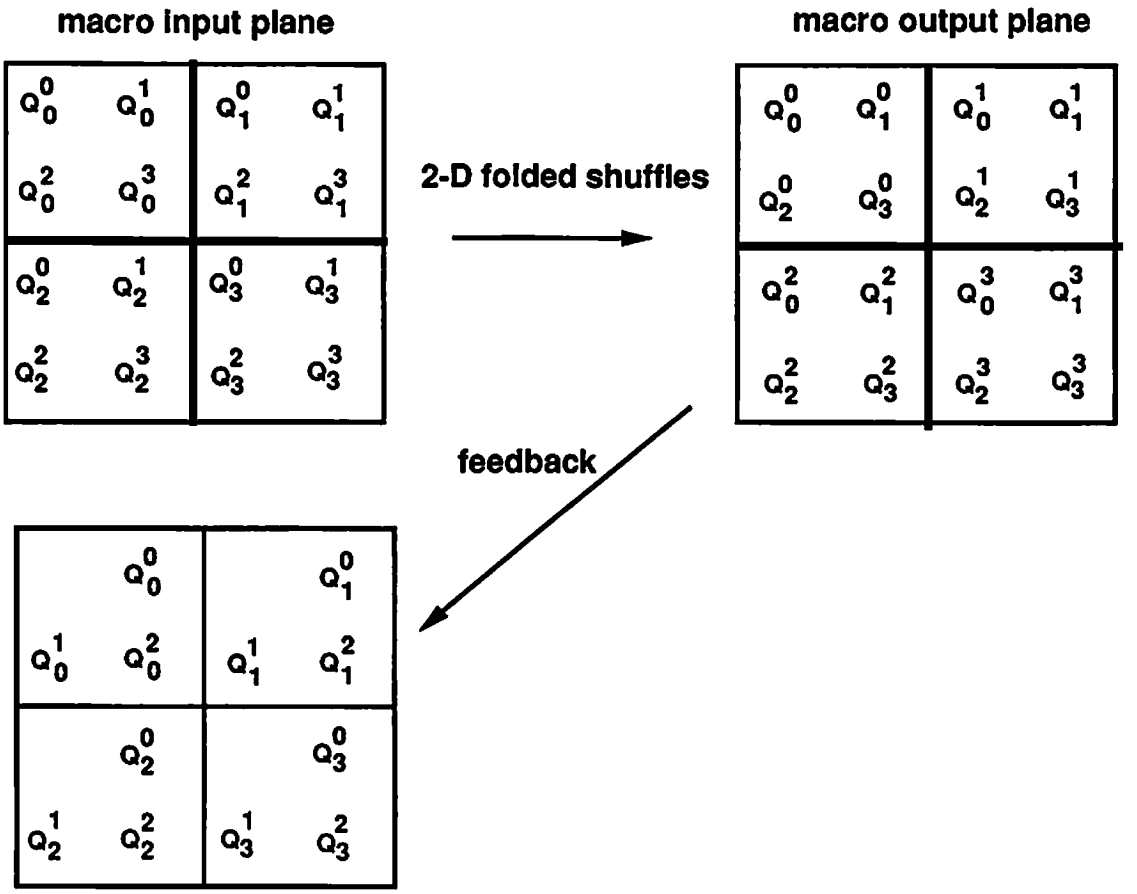


Figure 7.9: The forward and backward processings based on quadrants .

in the input plane, where $0 \leq i < (m-1)$.

A possible backward processing system is to use distinct facet holograms for each quadrant of the output stages to project these quadrants on the appropriate locations in the input plane. Figure 7.10 shows how such system redirects the quadrants from the macro output plane to the proper locations in the macro input plane. The arrows in each of the facets of the backward processing system indicate the direction and the unit counted as *quadrants* of the shift. Because there is no need to feed back the final stage output, backward processing is necessary for only $(m-1)$ stages. Each stage requires four facets, so the hologram contains a total of 12 facets in our example. In general, for an m stage space multiplexed network using the block method, this backward processing system will contain at most $4(m-1)$ facets. Similarly, we can use a four-facet hologram for the pre-processing step to

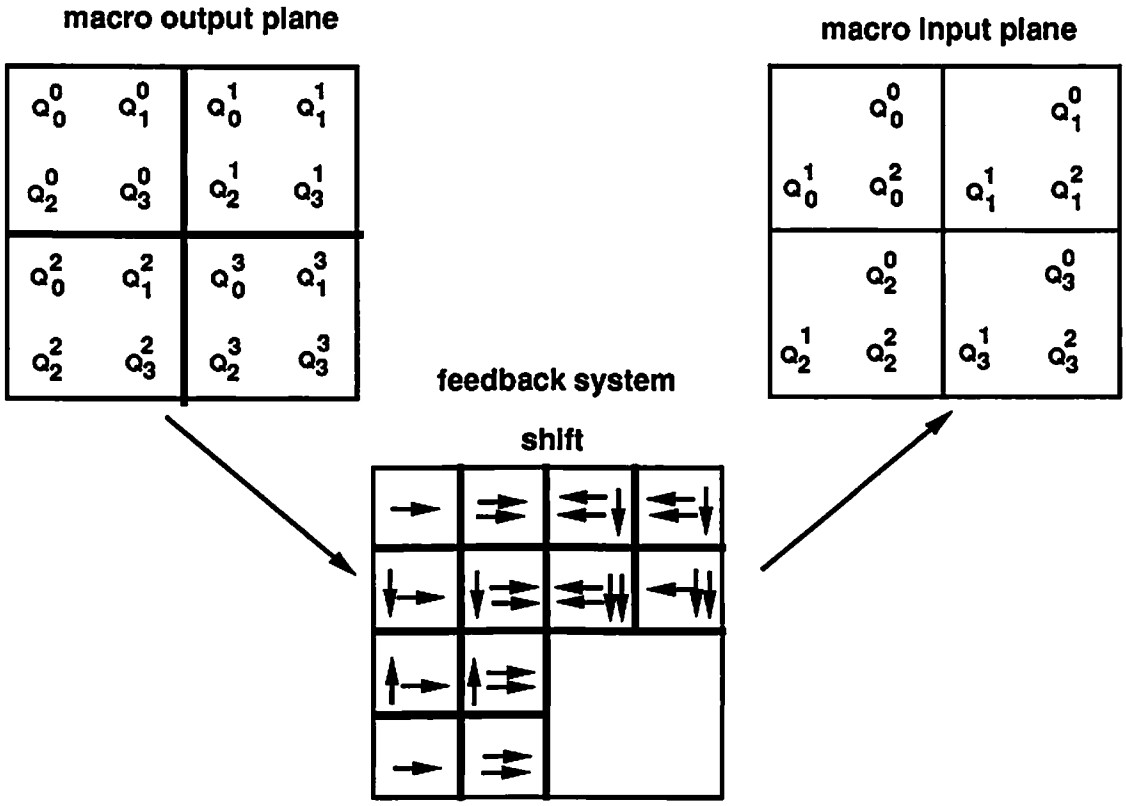


Figure 7.10: A possible backward processing holographic system for the demonstration example using 12 facets.

arrange the initial input on the macro input plane. Remembering that the forward processing step, the shuffle, also uses a four-facet hologram, then the complete system requires $4(m+1)$ facets for an m-stage space multiplexed volume Omega network. It is slightly more than a linear network which requires $4m$ facets to operate. However, the alignment effort is reduced and the overall system is more compact.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This dissertation presents the optoelectronic implementation of multistage interconnection networks. Volume (3-D) Omega networks which aim at utilizing the high spatial and temporal bandwidth of optics thoroughly are discussed.

A detailed study of the network properties of various volume Omega networks and their relationships, provides a direction in designing the architecture for any specific application.

Optical 2-D shuffles are used in volume Omega networks for the fixed interconnections; and the optoelectronic bypass/exchange switches are used for the dynamic switching. The one-copy algorithm provides a light efficient approach in implementing the optical 2-D general shuffles. The holographic system that was built to demonstrate the one-copy algorithm is a simple and powerful method of implementation. A systematic design method considering both the structure and the signal modulation techniques was developed for the optoelectronic bypass/exchange switches. An optically powered, optoelectronic switch based on one of the designs with polarization control was experimentally implemented. The experimental results show the potential of integrating a large array of this design, and use the integrated array of switches in optoelectronic volume networks.

A space multiplexed volume Omega network was introduced. This structure provides a way to reuse the optics to perform 2-D shuffles for all the stages, and

preserves the pipelining property of a multistage network. The effort needed to build a multistage network and to align these stages is also reduced. However, these savings require a higher space bandwidth product of the optical system and a relatively simple feedback system (backward processing).

In conclusion, this dissertation demonstrates the promise of implementing volume multistage interconnection networks optoelectronically. Further efforts should be made on both architecture study, and device and material development.

8.2 Future Work

Following are some important areas of research in the future:

- In this work, we concentrate on unidirectional networks which can be implemented using transmissive optical components. In practice, bidirectional networks which provide duplex communication are desired. Reflective optics can be used for both types of networks and to make them more compact.

- In practical applications, it is important to have fault tolerance in a multistage interconnection network. The implementation of fault tolerance in optoelectronic volume networks efficiently is a critical issue.

- The detailed design of bypass/exchange switches depends on the applications. To be specific, the scheme to control these switches plays an important role in the design of a switch. An important area of research lies in optimizing the schemes for various applications.

- In the space multiplexed structure, preprocessing and backward processing steps must be performed by the system. One improvement that can be made is to simplify these two steps. In order to do so, the arrangement of the channels from all the stages may be different from the block method. In addition, the optimized solution may exist for networks which are equivalent to Omega networks, but not for Omega networks.

- Many applications of volume networks can be obtained as an extension of their planar (2-D) versions. Other applications in the areas of computing, telecommunication, or processing should be studied in detail.

- Further investigation on the capabilities and limitations of optoelectronic implementation of volume multistage networks is important.

References

[1] D. P. Agrawal, "Graph Theoretical Analysis and Design of Multistage Interconnection Networks," *IEEE Trans. on Computers*, No. 7, pp. 637–648, 1983.

[2] S. Bian, K. Xu, and J. Hong, "Optical Perfect Shuffle Using Wollaston Prisms," *Appl. Opt.*, Vol. 30, No. 2, pp. 173–174, 1991.

[3] R. Boppana and C. S. Raghavendra, "On Self-Routing in Benes and (2n-1)-stage Shuffle/Exchange Networks," *Proc. Int. Conf. Parallel Processing*, pp. 196–200, 1988.

[4] K. Brenner and A. Huang, "Optical Implementations of the Perfect Shuffle Interconnection," *Appl. Opt.*, Vol. 27, No. 1, pp. 135 – 137, 1988.

[5] K. Brenner and F. Sauer, "Diffractive-Reflective Optical Interconnects," *Appl. Opt.*, Vol. 27, No. 10, pp. 4251 –4254, 1988.

[6] T. C. Chen, "Parallelism, Pipelining, and Computer Efficiency," *Computer Design*, pp. 69 — 74, 1971.

[7] L. Cheng and A. A. Sawchuk, "A One-Copy Algorithm for 2-D Shuffles for Optical Omega Networks," *to appear in Journal of Parallel and Distributed Computing.*

[8] G. Eichmann and Y. Li, "Compact Optical Generalized Perfect Shuffle," *Appl. Opt.*, Vol. 26, No. 7, pp. 1167–1169, 1987.

[9] S. C. Esener, J. H. Wang, T. J. Drabik, M. A. Title, and S. H. Lee, "One-Dimensional Silicon/PLZT Spatial Light Modulators," *Opt. Eng.*, Vol. 26, No. 5, pp. 406 — 413, 1987.

[10] T.-Y. Feng, "A Survey of Interconnection Networks," *IEEE Computers*, Vol. 14, No. 12, pp. 12–27, 1981.

[11] R. M. Fortenberry, W. L. Ha, and R. S. Tucker, "Photonic Fast Packet Switch With Gain," *Phtonic Switching Technical Digest Series*, pp. 128 — 131, 1991.

[12] J. Glinski, "Photonic Switching Technology: An update," *Technical Report TR87-0031, Bell-Northern Research Ltd.*, Ottwa, Ontario, Canada, July 1987.

[13] L. R. Goke and G. J. Lipovski, "Banyan Networks for Partitioning Multiprocessing Systems," *Proc. First Annual Computer Architecture Conf.*, pp. 21–28, 1973.

[14] J. W. Goodman, F. I. Leonberger, S. Y. Kung, and R. A. Athale, "Optical Interconnections for VLSI Systems," *Proc. IEEE*, Vol. 72, No. 7, pp. 850–866, 1984.

[15] M. Govindarajan and S. R. Forrest, "Optically Powered Arrays for Optoelectronic Interconnection Networks," *Appl. Opt.*, Vol. 10, 1991.

[16] M. Govindarajan, S. R. Forrest, L. Cheng, and A. A. Sawchuk, "Optically Powered 2×2 Optoelectronic Bypass/Exchange Switch with Polarization Routing," *IEEE Photonic Tech. Let.*, Vol. 3, No. 7, pp. 669 — 672, 1991.

[17] D. A. Gregory, R. D. Juday, J. Sampsell, R. Gale, R. W. Cohn, and J. S.E. Monroe, "Optical Characteristics of a Deformable-mirror Spatical Light Modulator," *Opt. Lett.*, Vol. 13, No. 1, pp. 10 — 12, 1988.

[18] J. Grinberg, A. D. Jacobson, W. P. Bleha, L. Miller, L. Fraas, D. Boswell, and G. Mayer, "A New Real-Time Noncoherent to Coherent Light Image Converter: The Hybrid Field Effect Liquid Crystal Light Valve," *Opt. Eng.*, Vol. 14, No. 3, pp. 217 –225, 1975.

[19] M. W. Haney and J. J. Levy, "Optically Efficient Free-Space Folded Perfect Shuffle Network," *Appl. Opt.*, Vol. 30, No. 20, pp. 2833–2840, 1991.

[20] D. O. Harris and A. VanderLugt, "Acousto-Optic Photonic Switch," *Opt. Lett.*, Vol. 14, No. 21, pp. 1177 — 1179, 1989.

[21] H. S. Hinton, "Architectural Considerations for Photonic Switching Networks," *IEEE J. Selective Areas in Communications*, Vol. 6, No. 7, pp. 1209 — 1226, 1988.

[22] H. S. Hinton, "Photonic Switching Fabrics," *IEEE Communication Magazine*, Vol. 28, No. 4, pp. 71 —89, 1990.

[23] H. S. Hinton, T. J. Cloonan, S. J. Hinterlong, A. L. Lentine, F. B. McCormick, and F. A. P. Tooley, "Photonic Switching Fabrics Based On S-SEED Arrays," *Photonic Switching Technical Digest Series*, 1990.

[24] K. Hwang and F. A. Briggs, "Computer Architecture and Parallel Processing," *McGrow Hill, Inc.*, 1984.

[25] J. Jahns and M. J. Murdocca, "Crossover Networks and Their Optical Implementation," *Appl. Opt.*, Vol. 27, No. 15, pp. 3155–3160, 1988.

[26] B. K. Jenkins, P. Chavel, R. Forchheimer, A. A. Sawchuk, and T. C. Strand, "Architectural Implications of a Digital Optical Processor," *Appl. Opt.*, Vol. 23, No. 19, pp. 3465–3474, 1984.

[27] F. K. Kahn, "Ir-Laser-Addressed Thermal-Optic Smetic Liquid-Crystal Story Displays," *Appl. Phys. Lett*, Vol. 22, No. 3, pp. 111 — 113, 1973.

[28] D. H. Lawrie, "Access and Alignment of Data in an Array Processor," *IEEE Trans. on Computers*, No. 12, pp. 1145–1155, 1975.

[29] A. L. Lentine, L. M. F. Chirovsky, M. W. Focht, J. M. Freund, G. D. Guth, R. E. Leibenguth, G. P. Przybylek, L. E. Smith, L. A. D'Asaro, and D. A. B. Miller, "Integrated Self Electro-Optic Effect Device Photonic Switching Nodes," *Photonic Switching Technical Digest Series*, pp. 170 — 173, 1991.

[30] Y. Liu, J. J. Brown, D. C. W. Lo, and S. R. Forrest, "Optically Powered Optical Interconnection System," *IEEE Photonic Technology Letters*, Vol. 1, No. 1, pp. 21 - 23, 1989.

[31] G. E. Lohman and A. W. Lohmann, "Optical Interconnection Network Utilizing Diffraction Gratings," *Opt. Eng.*, Vol. 27, No. 10, pp. 893 - 900, 1988.

[32] A. W. Lohmann, "What Classical Optics Can Do for The Digital Optical Computer," *Appl. Opt.*, Vol. 25, No. 10, pp. 1543–1549, 1986.

[33] A. W. Lohmann and F. Sauer, "Holographic Telescope Arrays," *Appl. Opt.*, Vol. 27, No. 14, pp. 3003 - 3007, 1988.

[34] A. W. Lohmann, W. Stork, and G. Stueke, "Optical Perfect Shuffle," *Appl. Opt.*, Vol. 25, No. 10, pp. 1530–1531, 1986.

[35] M. Murdocca and T. J. Cloonan, "Optical Design of a Digital Switch," *Appl. Opt.*, Vol. 28, No. 13, pp. 2505–2517, 1989.

[36] J. A. Neff, R. A. Athale, and S. H. Lee, "Two-Dimensional Spatial Light Modulators: A Tutorial," *Proc. of IEEE*, Vol. 78, No. 5, pp. 826 — 855, 1990.

[37] D. R. Pape and L. J. Hornbeck, "Characteristics of the Deformable Mirror Device for Optical Information Processing," *Opt. Eng.*, Vol. 22, No. 6, pp. 675 — 681, 1983.

[38] E. J. Restall, B. Robertson, M. R. Taghizadeh, and A. C. Walker, "Two Dimensional Spatially Variant Optical Interconnects," *Optical Computing Technical Digest Series*, pp. 49–52, 1991.

[39] B. Robertson, E. J. Restall, , M. R. Taghizadeh, and A. C. Walker, "Space-Variant Holographic Optical Elements in Dichromated Gelatin," *Appl. Opt.*, Vol. 30, No. 17, pp. 2368–2375, 1991.

[40] M. G. Robinson, K. M. Johnson, and D. Doroski, "Polarization-Independent, Broadband, Bistable 2X2 Optical Exchange Switch," *Opt. Lett.*, Vol. 15, No. 2, pp. 145 — 147, 1990.

[41] W. E. Ross, D. Psaltis, and R. H. Anderson, "Two-Dimensional Magneto-Optic Spatial Light Modulator for Signal Processing," *Opt. Eng.*, Vol. 22, No. 4, pp. 485 — 490, 1983.

[42] Y. Sampei, S. Naito, and Y. Kurita, "PLZT Fiber-Optics Switch," *IEEE Journal of Lightwave Technology*, Vol. 5, No. 9, pp. 1203 — 1206, 1987.

[43] A. A. Sawchuk and I. Glaser, "Geometries for Optical Implementations of the Perfect Shuffle," *SPIE Proc.*, Vol. 963, pp. 270– 282, 1988.

[44] A. A. Sawchuk and B. K. Jenkins, "Dynamic Optical Interconnections for Parallel Processors," *Proc. SPIE*, Vol. 625, pp. 143–153, 1986.

[45] Y. Sheng, "Light Effictive 2-D Optical Perfect Shuffle Using Fresnel Mirrors," *Appl. Opt.*, Vol. 28, No. 15, pp. 3290–3292, 1989.

[46] Q. W. Song and T. S. Yu, "Generalized Perfect Shuffle Using Optical Spatial Filtering," *Appl. Opt.*, Vol. 27, No. 7, pp. 1222–1223, 1988.

[47] C. W. Stirk, R. A. Athale, and M. W. Haney, "Folded perfect shuffle optical processor," *Appl. Opt.*, Vol. 27, No. 2, pp. 202–203, 1988.

[48] H. S. Stone, "Parallel Processing with the Perfect Shuffle," *IEEE Trans. on Computers*, No. 2, pp. 153–161, 1971.

[49] G. Stucke, "A Complete 2D-Shuffle/Exchange-Stage for Large 1D Data Arrays," *Optik*, Vol. 78, No. 2, pp. 84–85, 1988.

[50] S. Suzuki and K. Kasahara, "Photonic Packet Switch Based on VSTEP Two-Dimensional Array," *Photonic Switch Technical Digest Series*, pp. 124 — 127, 1991.

[51] J. Taboury, J. M. Wang, P. Chavel, and F. Devos, "Optical Cellular Processor Architecture 2 : Illustration and System Considerations," *Appl. Opt.*, Vol. 28, pp. 3138–3147, 1989.

[52] Y. Tashiro, N. Hamao, M. Sugimoto, N. Takado, S. Asada, and K. Kasahara, "Vertical to Surface Transmission Electrophotonic Device with Selectable Output Light Channels," *Appl. Phys. Lett.*, Vol. 54, No. 4, pp. 329 — 331, 1989.

[53] H. F. Taylor, "Application of Guided-Wave Optics in Signal Processing and Sensing," *Proc. of IEEE*, Vol. 75, No. 11, pp. 1524 — 1535, 1987.

[54] C. S. Tsai and P. Le, "A 4X4 Nonblocking Acoutooptic Waveguide Space Switch," *Photonic Switching Technical Digest Series*, pp. 67 — 70, 1991.

[55] K. S. Urquhart, P. Marchand, Y. Fainman, and S. H. Lee, "Design of Free-Space Optical Interconnection System Using Diffractive Optics," *OSA Annual meeting Technical Digest*, Vol. 17, p. 70, 1991.

[56] J. M. Wang, L. Cheng, and A. A. Sawchuk, "Holographic Implementation of 2-D Perfect Shuffles Based on a One-Copy Algorithm," *Photonic Switching Technical Digest Series*, pp. 180–183, 1991.

[57] J. M. Wang and A. A. Sawchuk, "Holographic Interconnections and Their Applications," *OSA Annual Meeting Technical Digest*, Vol. 15, p. 241, 1990.

[58] S. Weiss, M. Segev, S. Sternklar, and B. Fisher, "Photorefractive Dynamic Optical Interconnects," *Appl. Opt.*, Vol. 27, No. 16, pp. 3422 - 3428, 1988.

[59] T. Wolf and F. Kappeler, "Fast Polarization and Wavelength Switching in Quasi-Index Guided InGaAsP Twin-Stripe Lasers by Direct Current Modulation," *Proc. IEE Part J.*, Vol. 137, pp. 49 - 54, 1990.

[60] C.-L. Wu and T.-Y. Feng, "On a Class of Multistage Interconnection Networks," *IEEE Trans. on Computers*, No. 8, pp. 694–702, 1980.

[61] C.-L. Wu and T.-Y. Feng, "On a Distributed-Processor Communication Architecture," *Proc. Compcon Fall 1980*, pp. 599–605, 1980.

[62] L. Zhang, R. Jin, C. W. Stirk, G. Khirova, R. A. Athale, H. M. Gibbs, H. M. Chou, R. W. Sprague, and H. A. Macleod, "All-Optical Compare-and-Exchange Switches," *IEEE Journal of Selected Areas in Communications*, Vol. 6, No. 7, pp. 1273 — 1279, 1988.