2006

Mapping a weak hypercube on an optical slab waveguide

Divya Rengan
Louisiana State University and Agricultural and Mechanical College

MAPPING A WEAK HYPERCUBE ON AN OPTICAL SLAB WAVEGUIDE

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The Department of Electrical and Computer Engineering

by
Divya Rengan
Bachelor of Technology in Information Technology
University of Madras, Chennai, 2004
December 2006

Acknowledgements

I would like to express my gratitude to Dr. R. Vaidyanathan, my major professor, for his guidance and support over the past two years. Without his help this thesis would not have been possible. I would like to thank Dr. Ahmed El-Amawy and Dr. Jerry Trahan for agreeing to be part of my thesis defense committee. I would also like to thank my family, friends and well-wishers.

Table of Contents

Acknowledgements ii

List of Figures iv

Abstract vi

Chapter

1 Introduction 1

2 Preliminaries 5
2.1 General Conventions 5
2.2 Hypercube 6
2.2.1 A Review of Hypercube Properties 6
2.3 Interprocessor Communication on H_d 13
2.4 Optical Implementation of Interprocessor Communication 14
2.4.1 Slab Waveguides 14
2.4.2 Mapping 17
2.4.3 Aggregates 18

3 Dense Mapping 22
3.1 Construction of the Destination Array, Dst 22
3.2 Construction of the Dimension Array, Dim 26
3.3 Construction of Source Array, Src 30
3.4 Analyzing the Mapping 31
3.4.1 Correctness 31
3.4.2 Number of Aggregates 36

4 Sparse Mapping 39
4.1 Construction of the Dimension Array, Dim 39
4.2 Construction of the Destination Array, Dst 44
4.3 Construction of the Source Array, Src 45
4.4 Number of Aggregates 49
4.5 Relation to the Extended Hypercube 52

5 Lower Bounds 57
5.1 A General Lower Bound 57
5.2 Dense Mapping Lower Bound 58

6 Conclusion 66

Bibliography 68

Vita 71

List of Figures

2.1 Examples of hypercubes 7

2.2 A 3-dimensional directed hypercube 8

2.3 Recursive structure of a hypercube 9

2.4 Recursive decomposition of H_4 9

2.5 An illustration of the proof of Lemma 2.4 11

2.6 Binary representations of b and b' 12

2.7 Binary representations of j and j' 12

2.8 Weak communication in a 3-dimensional hypercube 14

2.9 An optical slab 15

2.10 Slab structure considered in this thesis 16

2.11 Mapping 1 20

2.12 Mapping 2 20

2.13 Mapping 3 20

2.14 A way to specify the standard mapping 20

3.1 Structure of the destination array 23

3.2 Array Dst for $d = 3$ 24

3.3 Array Dst for $d = 4$ 24

3.4 Relation between $h_d, \hat{h}_d$ and $\hat{h}_d^x$ 25

3.5 Examples of the array Dim 27

3.6 Positions of α, β and γ in the dimension array 29

3.7 Array Dim for a 4-dimensional hypercube 30

3.8 *Src* array for a 4-dimensional hypercube 30

3.9 Bit values for the proof of Lemma 3.7 when $k > 0$ 34

3.10 Bit values for the proof of Lemma 3.7 when $k = 0$ 35

3.11 Differing bits in the proof of Lemma 3.7 36

4.1 Examples of reverse diagonal arrays 40

4.2 Unit reverse diagonal array, U_2 40

4.3 Recursive construction of $Dim_{3,1}$ 41

4.4 4-dimensional sparse *Dim* array 43

4.5 Sparse destination array for $d = 4$ 44

4.6 A 4-dimensional Sparse *Src* 45

4.7 Mapping for an extended hypercube Q_3 53

4.8 A 5-dimensional extended hypercube 54

5.1 An illustration of the proof of Lemma 5.3 58

5.2 An illustration of the proof of Lemma 5.6 60

5.3 Illustrating the structure for numbers u, a, v, b 60

5.4 Complete and incomplete aggregates 61

5.5 An illustration of the proof of Lemma 5.9 62

5.6 An illustration for the proof of Lemma 5.10 64

Abstract

The communication fabric of a parallel processing system is represented as a directed graph. In a weak topology a processor uses at most one incoming edge or one outgoing edge for communication at any given point in time. Currently, the data rate that can be supported on electronic interconnects is reaching its limits. Optical interconnects have been identified as one of the most promising approaches to the growing demands for today's systems.

The "medium bandwidth" of an optical waveguide is huge (order of petabits per second for a 1mm^2 cross-section optical slab). The challenge lies in utilizing as much of this medium bandwidth as possible. We address this problem by exploiting knowledge of the communication patterns. Key to our approach is a method to map communications to optical channels.

This thesis deals with the mapping of a d-dimensional weak hypercube on to an optical slab waveguide. The weak topology helps reduce the cost of optical components used by allowing component reuse across different channels. We present two mappings, the dense and sparse mappings. The dense mapping for a d-dimensional weak hypercube packs all communication channels into a $d \times 2^d$ array of optical channels and uses $(d-2)2^d+4$ lasers and 2^d detectors (or vice-versa). The sparse mapping uses a $2^{d-1} \times 2^d$ channel array, but does not use all channels to map hypercube edges. We show that this mapping requires 2^d lasers and 2^d detectors. We also define a supergraph of the hypercube, called the extended hypercube, that maximally utilizes the empty channels in a sparse mapping. We establish that the extended hypercube is the largest supergraph of the hypercube that utilizes all available channels, without increasing the number of lasers and detectors used. The mappings defined

for both these sparse cases are optimal. They use N lasers and N detectors, where N is the number of nodes in the topology.

We also derive lower bounds on the number of lasers and detectors needed for a "standard" mapping of a hypercube to an optical slab waveguide. We show that the costs of all the dense and sparse mappings proposed in this thesis match these lower bounds.

Chapter 1
Introduction

The interconnection network between processors in a parallel processing system is one of its most important components. In fact, today's computing systems are limited more by the communication fabric connecting the processors than the processors themselves [5, 10]. In addition, CMOS-transistors are expected to support data rates of around 20 Gbps in the near future [36]. The data rate possible on traditional metal interconnects is fast reaching its limits. There is, therefore, a gap between what these traditional interconnects can deliver and the computing needs of the future [11, 22, 23, 28]. Optical interconnects [8, 13, 16, 20, 35] show great promise in filling this gap. They support faster data rates, higher bandwidths and are less prone to noise. Optical interconnects have been studied well in both long haul networks [2, 6, 14] and at the board/backplane levels [7, 8, 17, 18]. In this thesis we consider optical interconnects between modules that are spatially proximate (such as at the board or backplane levels).

Optical communication can be free-space [4, 27] or waveguided [12, 11, 15, 30]. Free-space systems involve transmitting light in air and directing it using mirrors and prisms. In waveguided systems, light is inserted into a waveguide at one end and detected at the other. Here we consider waveguided systems, specifically, systems with slab waveguides that support transmission of light in several different modes.[1] However, many of the ideas presented are applicable to systems with fibers and free space optics as well.

[1]For this thesis a slab is a waveguide with a sufficiently large cross-section that supports transmission of light in many modes (angles, for our purpose). In contrast, fibers have a much smaller cross-section and allow just a single or a few modes.

A simple slab waveguide can carry a huge bandwidth (easily in the range of petabits per second). Practically, systems can harness just a fraction of this bandwidth (in the terabits per second range). There is a wide gap between the "medium bandwidth" and the "system bandwidth." The main reasons for this gap are the challenges in inserting and detecting light into and from the slab waveguide and packaging all the required fast lasers and detectors used within a small space. Section 2.4.1 elaborates on this idea.

One way to bridge this gap between the system and the medium bandwidths is through technological advances; i.e., by using a large number of small and fast lasers. However, we take a different approach (that is independent of and complements any technological advances). We use to our advantage the fact that most interconnection topologies do not use all available channels for communication at the same time. In this context, we consider a "weak" topology [26], which is one in which each node sends and receives information from at most one of its neighbors at any given instance of time. By mapping a weak topology on a slab waveguide, a single laser or detector can be used over multiple channels to reduce the cost of optical hardware and better utilize the available medium bandwidth. It may also be viewed as an approach that allows a larger instance of the communication topology to be mapped on a slab with fixed hardware and packaging constraints. With a naive mapping, each channel in the communication network corresponds to a laser and detector. On a weak topology, this will force many of these lasers and detectors to be idle most of time. This idle time of lasers and detectors can be used to roll many of them together, so that a single laser or detector serves many channels. For example, if 64 lasers and detectors can be used, then a naive mapping will admit a 16-node hypercube (4-dimensional hypercube) with 4 lasers/detectors per node. With our method a 64 node hypercube can be mapped.

In this thesis we study the mapping of d-dimensional weak hypercube on an optical slab waveguide. A slab waveguide supports multiple modes and wavelengths [12] and each channel in a slab is distinguished by its mode and wavelength. (A doublet, (ν, λ), where ν is the mode and λ is the wavelength at which the channel is operating describes the channel.) Two channels are said to be adjacent if they use the same mode and have adjacent wavelengths or use the same wavelength and have adjacent modes. This notion of adjacency is important because it allows us to aggregate adjacent channels into a single laser (such as a tunable laser), provided these aggregated channels cannot be used simultaneously. The weak topology provides us the basis for ascertaining simultaneity. Similarly, detectors can also be aggregated. These ideas are formally explained in Section 2.4.3.

Previously, Sethuraman [31], studied the mapping of a weak k-ary n-cube on a slab. Our work applies similar techniques to a weak hypercube. Note that while a d-dimensional hypercube is a special case of a k-ary n-cube (where $k = 2$ and $n = d$) the results of Sethuraman are not generalizations of ours. Our mapping uses half the number of edges compared to the work of Sethuraman; i.e., each processor has a degree of only d, rather than the $2d$ required for a d-dimensional torus. Moreover, we also explore the idea of a "sparse" mapping that allows improvements in the number of lasers and detectors used.

We use two approaches to mapping a d-dimensional weak hypercube (that has $N = 2^d$ processors) on the slab. The mapping uses an $M \times W$ channel array (where M is the number of modes and W is the number of wavelengths supported by the slab). In our case, we use $W = N =$ number of processors (nodes) in the topology, as this reduces optical hardware. The "dense mapping" employs an array of size $d \times 2^d$ and uses $(d-1)2^d + 4$ lasers and 2^d detectors (or 2^d lasers and $(d-1)2^d + 4$ detectors). This is the minimum array size

for any "standard" mapping of the hypercube topology on a slab. Our second approach is the "sparse mapping" that employs a $2^{d-1} \times 2^d$ channel array and uses 2^d lasers and 2^d detectors. Obviously, not all channels of this array are used as the d-dimensional hypercube only has $d2^d$ edges. We identify the largest supergraph of the hypercube (called the extended hypercube) that can utilize these empty channels without increasing the required numbers of lasers and detectors. The above sparse mapping uses the smallest possible numbers of lasers and detectors (1 per processor) for any strongly connected topology. We formally prove this optimal. In Chapter 5 we derive a non-trivial lower bound that proves the number of lasers in a dense mapping to also be optimal.

The remainder of this thesis is organized as follows. Chapter 2 discusses some preliminary ideas that will be important to follow ideas in the remaining chapters. Chapter 3 elaborates on the dense mapping approach. Chapter 4 deals with the sparse mapping and defines the extended hypercube. We also present a possible direction of unifying the dense and sparse mapping into one more-general mapping. Chapter 5 derives lower bounds for mapping a hypercube to an optical slab waveguide. Here, we also show that our results for the dense and sparse mappings match the lower bound. Finally, in Chapter 6 we summarize our results and identify some directions for future work.

Chapter 2
Preliminaries

This chapter discusses some important concepts that are used in the rest of this thesis. Specifically, we describe the hypercube topology, the idea of interprocessor communication in point-to-point networks, and issues in an optical implementation of interprocessor communication.

2.1 General Conventions

Unless obvious from the context, a binary number will be suffixed with a subscripted 2. For example, 11_2 is 3 (in decimal). A binary b-bit number has bits indexed 0 to $b-1$, with bit 0 as the least significant bit (lsb) and bit $b-1$ as the most significant bit (msb). Let $i = i_{b-1}i_{b-2}\cdots i_1 i_0$ be a b-bit integer, where $i_j \in \{0,1\}$ and $0 \leq j < b$.

Let $S \subseteq \{0,1,\cdots,b-1\}$. Then $i|_S = \hat{i}_{b-1}\hat{i}_{b-2}\cdots\hat{i}_1\hat{i}_0$, where for $0 \leq j < b$,

$$\hat{i}_j = \begin{cases} i_j & \text{if } j \notin S \\ i'_j & \text{if } j \in S \end{cases}$$

That is, $i|_S$ is the integer obtained from i by complementing the bits whose indicies are in S. When $S = \{a\}$ is a singleton set we will simply write $i|_a$ instead of $i|_{\{a\}}$. For example, $7|_2 = 111|_2 = \overline{1}11_2 = 011_2 = 3$, whereas $7|_3 = 0111|_3 = \overline{0}111_2 = 1111_2 = 15$. Also $7|_{\{2,3\}} = 0111|_{\{2,3\}} = 1011_2 = 11$.

Definition 2.1 Let $A = [a_{ij}]$ be an $x \times y$ array, all of whose cells are either integers or empty. We define the following operations on A.

(a) For any real r, array $rA = [r \cdot a_{ij}]$ is the array with the non-empty cells of A multiplied by r; empty cells remain empty.

(b) For any set S of non-negative integers, $A|_S = [a_{ij}|_S]$ is the array with non-empty cells obtained by complementing the binary representation of the non-empty cells of A, according to the bit positions in S; empty cells remain empty. □

2.2 Hypercube

The topology considered in this thesis is a *d-dimensional hypercube* that is defined as follows

Definition 2.2 For any $d \geq 1$, a *d-dimensional hypercube* is an undirected graph $H_d = (V, E)$ where $V = \{0, 1, \cdots, 2^d - 1\}$ is the set of nodes. Two nodes p and q are connected by an edge if and only if the binary representations of p and q differ in exactly one bit position. □

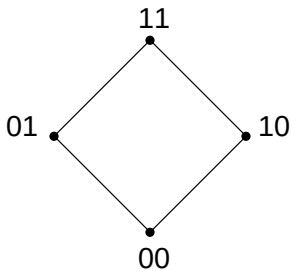
That is, (p, q) is an edge of the hypercube iff there exists $0 \leq j < d$ such that $p|_j = q$. If nodes p and q differ in bit b (where $0 \leq b < d$), then the edge (p, q) is called a *dimension-b* edge.

Examples of hypercubes are shown in Figure 2.1.

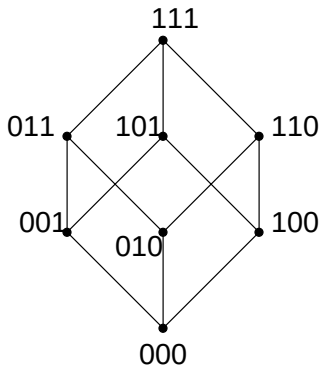
The "undirected hypercube" described so far readily extends to a directed graph where each undirected edge is replaced by two oppositely directed edges. We will use this *directed hypercube* in this thesis. For simplicity we will still call it a hypercube and denote it by H_d. An example of a 3-dimensional (directed) hypercube is shown in Figure 2.2.

2.2.1 A Review of Hypercube Properties

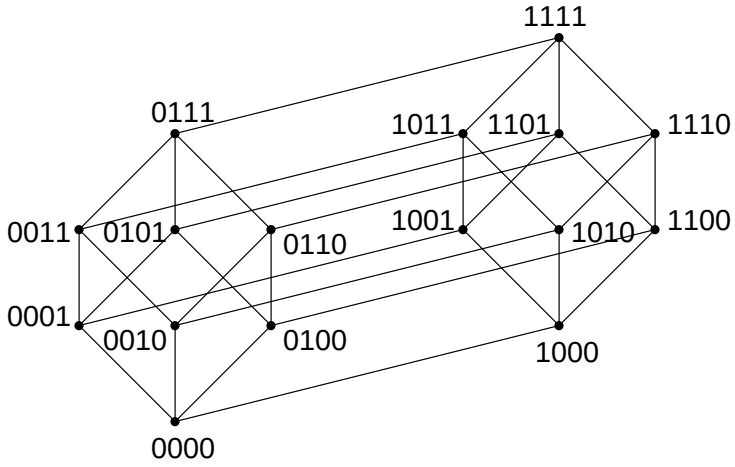
We now state some well-known properties of the hypercube. More details may be found in Leighton [19].



(a) $d = 2$



(b) $d = 3$



(c) $d = 4$

FIGURE 2.1: Examples of hypercubes

Basic Topological Properties: A d-dimensional (undirected) hypercube H_d has 2^d nodes, each of in-degree d and out-degree d. Hypercube H_d has a diameter of d and is a node-symmetric graph. Each node is incident on a dimension-b edge for each $0 \leq b < d$.

Recursive Structure: A d-dimensional hypercube H_d can be expressed in terms of two smaller $(d-1)$-dimensional hypercubes as follows

Let H_{d-1} and H'_{d-1} be two $(d-1)$-dimensional hypercubes. Number the nodes of H_{d-1} in the usual way from 0 to $2^{d-1}-1$ and those of H'_{d-1} from 2^{d-1} to 2^d-1. Now, H_d can

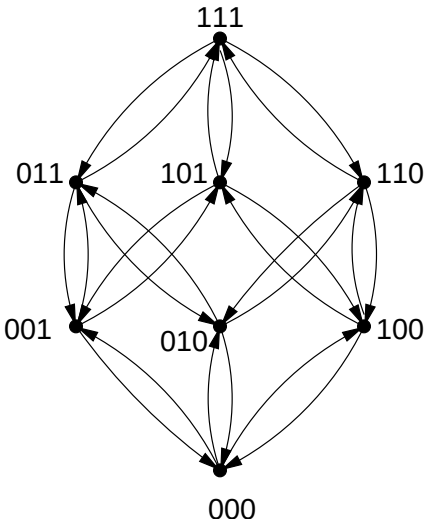


FIGURE 2.2: A 3-dimensional directed hypercube

be constructed by connecting node i of H_{d-1} to node $i + 2^{d-1}$ of H'_{d-1}. Figure 2.3 illustrates the general idea and Figure 2.4 shows an example with $d = 4$.

Hamiltonian path: A *Hamiltonian path* (resp., *cycle*) of a graph is a path (resp., cycle) that traverses every node in the graph exactly once. It is well known that a hypercube H_d with $d \geq 2$ has a Hamiltonian cycle and, hence, a Hamiltonian path. Here we describe one such path and term it the *standard Hamiltonian path* for a hypercube. Observe that every path (including a Hamiltonian path) can be specified as a string of nodes of the graph.

Definition 2.3 For any string s, let $\bar{s}$ denote the reversed string. For strings s_1 and s_2, let $s_1 \circ s_2$ denote the concatenation of s_1 and s_2. Let $s = \langle s_0, s_1, \cdots, s_k \rangle$ be a string of reals, and let r be a real number. Then $r + s$ is the string $\langle r + s_0, r + s_1, \cdots, r + s_k \rangle$. □

For example, if $s_1 = \langle a, b \rangle$ and $s_2 = \langle c, d \rangle$, then $\bar{s_1} = \langle b, a \rangle$ and $s_1 \circ s_2 = \langle a, b, c, d \rangle$. If $s = \langle 1, 2, 3, 4 \rangle$, then $4 + s = \langle 5, 6, 7, 8 \rangle$.

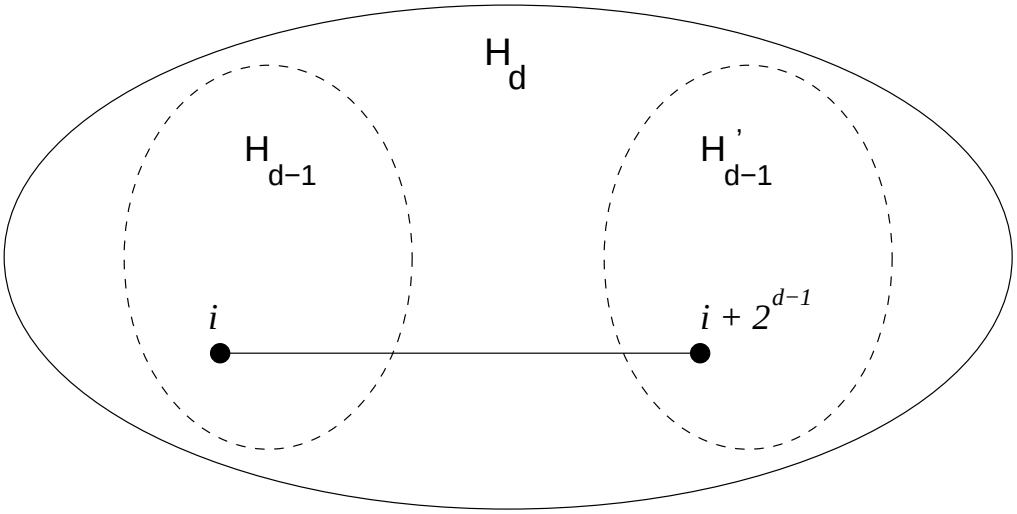


FIGURE 2.3: Recursive structure of a hypercube

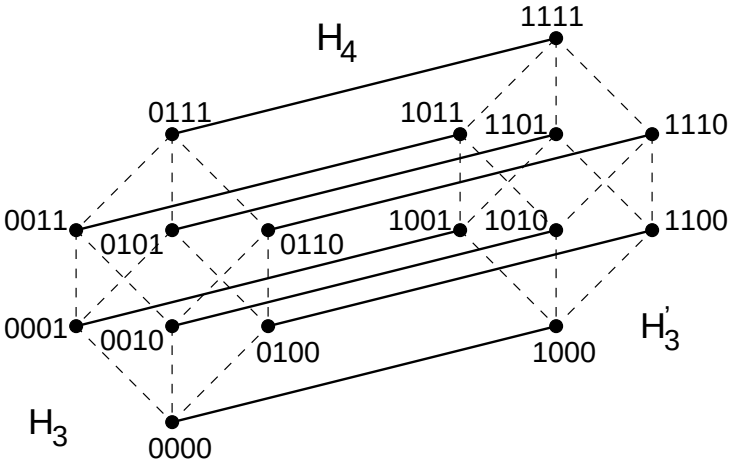


FIGURE 2.4: Recursive decomposition of H_4

Let h_d denote the string representing the standard Hamiltonian path of hypercube H_d. We now give a recursive definition of h_d. Define h_0 to be $\langle 0 \rangle$. For $d > 0$, let hypercube H_d be recursively decomposed into subcubes H_{d-1} and H'_{d-1}, and let their standard Hamiltonian paths be h_{d-1} and h'_{d-1}, respectively. Recall that H_{d-1} and H'_{d-1} are identical except that node i of H_{d-1} corresponds to node $i + 2^{d-1}$ of H'_{d-1}. Similarly $h'_{d-1} = 2^{d-1} + h_{d-1}$ is simply

h_{d-1} with 2^{d-1} added to each element. Then for $d > 0$, h_d can be defined as follows:

$$h_d = h_{d-1} \circ \overline{2^{d-1} + h_{d-1}}. \tag{2.1}$$

For example,

$$\begin{aligned}
h_0 &= \langle 0 \rangle \\
h_1 &= h_0 \circ \overline{h_0'} = \langle 0 \rangle \circ \langle 0 + 2^0 \rangle = \langle 0 \rangle \circ \langle 1 \rangle = \langle 0, 1 \rangle \\
h_2 &= \langle 0, 1 \rangle \circ \langle 1 + 2^1, 0 + 2^1 \rangle = \langle 0, 1, 3, 2 \rangle \\
h_3 &= \langle 0, 1, 3, 2 \rangle \circ \langle 6, 7, 5, 4 \rangle = \langle 0, 1, 3, 2, 6, 7, 5, 4 \rangle
\end{aligned}$$

Subsequently, we will denote the j^{th} element of h_d by $h_d(j)$. Where the dimension d is implicit, we will denote h_d simply by h. The following lemma specifies the elements of h_d in a closed form.

Lemma 2.4 *For integers b and j, let $b_{d-1}b_{d-2}\cdots b_1b_0$ be the binary representation of b and let $j_{d-1}j_{d-2}\cdots j_1j_0$ be the binary representation of j. For any $0 \leq j < 2^d$, let $h_d(j)$, the j^{th} element of the standard Hamiltonian path h_d, be b. Then for each $0 \leq i < d$,*

$$b_i = j_i \oplus j_{i+1},$$

where $j_d = 0$.

Proof: We proceed by induction on $d \geq 1$. For $d = 1$, $h_1 = \langle 0, 1 \rangle$ and $b = j$. Here $j_1 = 0$, so $b_0 = j_0 \oplus j_1 = j_0$. This agrees with $b = j$.

Assume the lemma to hold for all $d \geq 0$ and consider $h_{d+1} = h_d \circ \overline{h_d'}$. Let b be the j^{th} element of the standard Hamiltonian path h_{d+1}. We now consider two cases.

Case $0 \leq j < 2^d$ (b is in the first half of the Hamiltonian path): By the induction hypothesis for $0 \leq i < d$, we have $b_i = j_i \oplus j_{i+1}$ as required. Also, since $j_{d+1} = j_d = 0$, we have $b_d = 0 = j_d \oplus j_{d+1}$ as required.

Case $2^d \leq j < 2^{d+1}$ (b is in the second half of the Hamiltonian path): We are given that the j^{th} element of h_{d+1} is b. Since j points to an element in the latter half of h_{d+1}, which is $\overline{h'_d} = \overline{h_d + 2^d}$, we have $2^d \leq b < 2^{d+1}$. Therefore $b_d = j_d = 1$ and $j_{d+1} = 0$, which implies that $b_d = j_d \oplus j_{d+1}$. So we now consider indices $0 \leq i < d$. The strategy is to express the j^{th} element, b, of h_{d+1} as the $(j')^{th}$ element, b', of h_d, developing a relationship between j, b and j', b'. Once this is done, the induction hypothesis will establish the result.

The j^{th} element of h_{d+1} is the $(j-2^d)^{th}$ element of $\overline{h'_d}$ that makes the second half of h_{d+1} (see Figure 2.5). If the x^{th} element of h'_d is y, then the $(2^d - 1 - x)^{th}$ element of h_d is

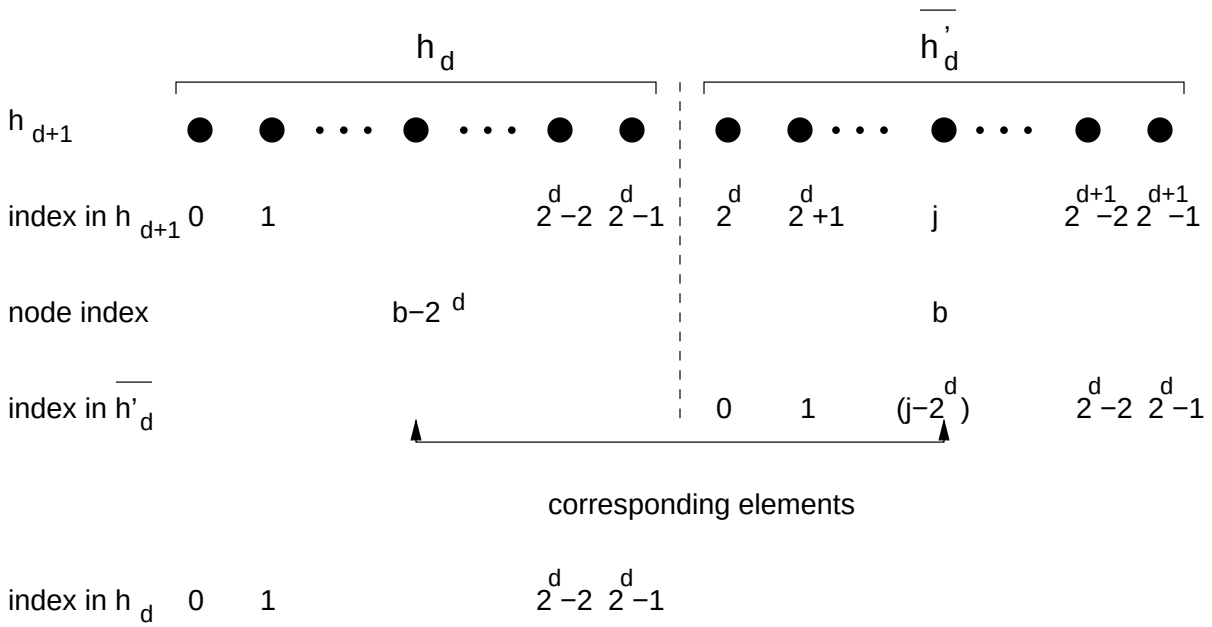


FIGURE 2.5: An illustration of the proof of Lemma 2.4

$(y-2^d)$. Therefore, if the $(j-2^d)^{th}$ element of $\overline{h'_d}$ is b, then the $j' = (2^d-1-(j-2^d))^{th} = (2^{d+1}-1-j)^{th}$ element of h_d is $b' = b - 2^d$. This element satisfies $b'_i = j'_i \oplus j'_{i+1}$ by the induction hypothesis.

Note that b' and b differ only in bit d where $b'_i = 0$ while $b_i = 1$. For $0 \leq i < d$, $b'_i = b_i$. In fact, if $b = X + 2^d$, then $b' = X$ (see Figure 2.6). Let $j = 2^d + Y$. Then

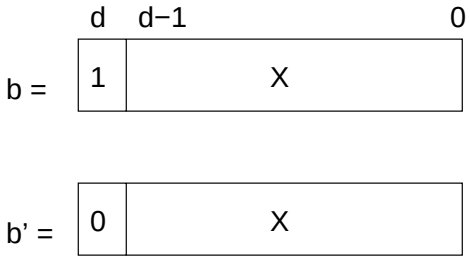


FIGURE 2.6: Binary representations of b and b'

$j' = (2^{d+1}-1) - j = 2^d - 1 - Y$ (see Figure 2.7). For each $0 \leq i < d$, the i^{th} bit of

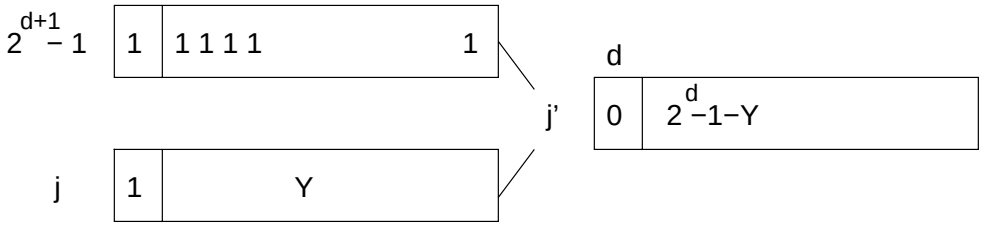


FIGURE 2.7: Binary representations of j and j'

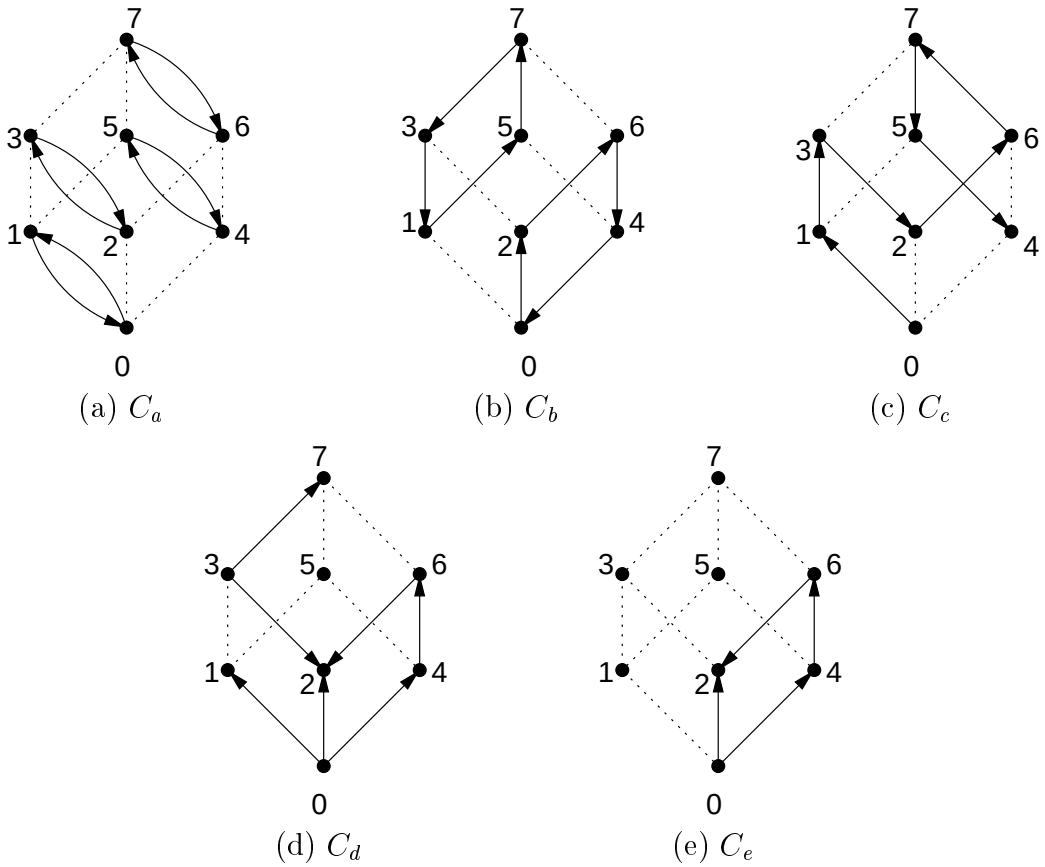
$2^{d+1}-1$ is a 1. So $j'_i = 1$ iff $j_i = 0$. That is, j'_i is the one's complement of j_i. Now for any $0 \leq i < d$, $b_i = b'_i = j'_i \oplus j'_{i+1} = \bar{j}_i \oplus \overline{j_{i+1}} = j_i \oplus j_{i+1}$. Also, $b_d = 1 = 1 \oplus 0 = j_d \oplus j_{d+1}$.

■

2.3 Interprocessor Communication on H_d

A *point-to-point*, N-processor network consists of a set $P = \{p_0, p_1, \cdots, p_{N-1}\}$ of processors connected pairwise by communication links. This system can be represented by a directed graph $G = (P, E)$ where edge set E is the set of communication links between processor pairs. The hypercube outlined in the previous section is an example of such a network.

A *communication* C in a point-to-point network is simply a set of edges of its graph; that is, $C \subseteq E$. Figure 2.8 shows five communications C_a – C_e for hypercube H_3. Notice that the communication in Figure 2.8(e) is a subset of that in Figure 2.8(d). In this thesis we consider weak communications. In a *weak communication* a processor sends at most one message and receives at most one message at a given step, regardless of the number of neighboring processors in the underlying topology. Figure 2.8 (a)–(c) illustrate weak communications on H_3. Here, node 0 has nodes $1, 2, 4$ as neighbors. Therefore in a step, node 0 can send to any one of nodes $1, 2$ or 4 and receive from any one of nodes $1, 2$ or 4. In Figure 2.8 (a) node 0 sends and receives from node 1. In fact, in this figure all communications are along dimension 0 of H_3. If all nodes in the hypercube either send or receive a message through a particular dimension b edge at a given time, then that kind of communication is called a *uniform communication* [32]. However, a weak communication need not be confined to one dimension as shown in Figure 2.8 (b). Here node 0 sends to node 2 but receives from node 4. In Figure 2.8 (c) node 1 sends to node 3 and receives from node 0. In fact, this figure shows a standard Hamiltonian path of H_3 (see Section 2.2.1). Figures 2.8 (d) and (e) do not represent weak communications as, for example, node 0 sends to nodes 2 and 4 simultaneously.



(a) C_a (b) C_b (c) C_c

(d) C_d (e) C_e

FIGURE 2.8: Weak communication in a 3-dimensional hypercube

2.4 Optical Implementation of Interprocessor Communication

In this section we discuss a general optical slab waveguide and explain how a communication can be performed on this structure.

2.4.1 Slab Waveguides

An optical slab waveguide is made of any transparent material that can be used to transmit light. It can be viewed as consisting of three sections: (a) the input, (b) the stem and (c) the output (see Figure 2.9). The input section is the portion through which light is inserted into the slab. Depending on the slab, this portion may have different geometries. Figure 2.9

simply shows a flat surface through which light is inserted into the slab. Similarly the output is the portion of the slab through which light exits. Again this can be of different geometries and Figure 2.9 shows one possibility. The portion between the input and the output is the stem and it is simply a conduit for the light that maintains any distinguishing property of the inserted light; for our purpose such distinguishing properties would be the mode and wavelength of the light.

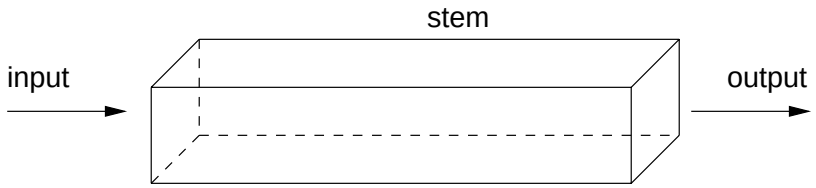


FIGURE 2.9: An optical slab

The input light can be in a range of wavelengths. Theoretically, this wavelength can be from 400nm to 1550nm [1]. In fact, several different wavelengths can be transmitted simultaneously on the slab. This is called Wavelength Division Multiplexing (WDM) [24, 29, 34]. This multiplexed signal is separated (demultiplexed) at the output. Current technology (DWDM-Dense Wavelength Division Multiplexing) for fibers allows more than 150 wavelengths to be multiplexed [9]. These wavelengths are in the 1550 nm range for which attenuation is small in fibers. For slabs (that are used over much shorter distances and have insignificant attenuation), the range of useful wavelengths is much bigger and the number of signals that can be multiplexed is much larger.

Unlike fibers that need to avoid dispersion and transmit a single or a few modes, a slab (used over short distances) can support multiple modes. This allows light to be input in different modes (that roughly translate to "angles" for our purpose). If the slab preserves

the modes, then signals input at different modes will also exit the slab at different modes. Thus two signals of the same wavelength can be distinguished by their modes. This form of multiplexing is called Mode Division Multiplexing (MDM) [3, 11].

In short, a slab can support both WDM and MDM. As noted above, currently it is possible to have much more than 150 wavelengths and a slab of cross section 1 mm^2 can support over 2000 MDM angles [11]. So this slab can carry in excess of 300,000 signals. The data rate of these signals is limited only by the capabilities of the lasers and detectors (not by the slab itself). Currently, lasers and detectors operating at 40 Gbps are available [21]. Thus the slab can carry in excess of 300,000 channels, each operating at 40 Gbps, or a total of 12 Pbps.

Figure 2.10 illustrates the optical setting that we consider in this thesis. The input to the slab waveguide is a set of lasers. As mentioned above these lasers can operate in a range of wavelengths and modes. Each input signal occupies a *channel* in the slab.

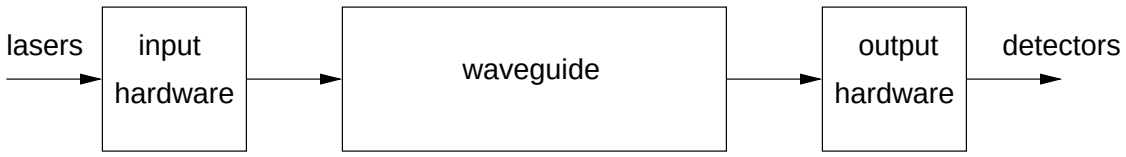


FIGURE 2.10: Slab structure considered in this thesis

The input to the slab can be single/multiple lasers operating at different wavelengths, or it could be single/multiple lasers operating at different modes. For the waveguide to maintain the mode and wavelength of the light that passes through it, the input to the slab must be collimated. To input light at different wavelengths, either several lasers, each capable of generating a single wavelength, or a single tunable laser capable of tuning to a relatively few adjacent frequencies (one at a time), can be used. If the application uses

adjacent wavelengths, it is possible to save on optical hardware by replacing several lasers each operating at a fixed wavelength by a single tunable laser capable of tuning to one of a set of adjacent frequencies.

To input light in multiple modes, either multiple lasers, each transmitting collimated light at a fixed mode (angle), or a single laser transmitting in different modes can be used. To select one out of a range of modes, one could use "spatial light modulators" [25]. As in the case of tunable lasers, it is possible to save on optical hardware by using a spatial light modulator in the place of several single mode lasers.

If the slab can carry signals in M modes and W wavelengths, then it can support MW channels. We use the doublet (ν, λ), where ν is the mode and λ is the wavelength, to represent a channel. The entire set of channels is called the *channel array*; it is an $M \times W$ array of channels (ν, λ).

Each channel (ν, λ) is available at the output of the waveguide. The channels need to be demultiplexed and separated. The number of detectors used can affect the cost of the interconnect. By using the same detector for adjacent channels their number can be reduced. Though this would mean bigger and slower detectors, the number of wires used to connect them can be significantly reduced.

2.4.2 Mapping

To perform interprocessor communication on the slab, each edge of the topology must be mapped to a channel of the channel array. Let the slab support M modes $\nu_0, \nu_1, \cdots, \nu_{M-1}$ and W wavelengths $\lambda_0, \lambda_1, \cdots, \lambda_{W-1}$. As noted earlier, each channel is a mode-wavelength pair (ν_i, λ_j). The mapping we seek here is an injective function $f : \mathcal{E} \rightarrow \mathcal{M} \times \mathcal{W}$ where $\mathcal{M} = \{\nu_i : 0 \leq i < M\}, \mathcal{W} = \{\lambda_j : 0 \leq j < W\}$ and $\mathcal{E}$ is the edge set of the topology.

This mapping can be specified as two $M \times W$ arrays Src and Dst that contain node indices. For any $0 \leq i < M$ and $0 \leq j < W$, if the directed edge $e = (p, q)$ is mapped by f to channel (ν_i, λ_j), then $Src(i, j) = p$ and $Dst(i, j) = q$.

In this thesis we consider a hypercube where an edge $e = (p, q)$ can also be specified as a dimension-b edge from source p or to destination q. That is, given b, only one of p or q need be specified. Thus we can also use another $M \times W$ array, Dim, along with Src or Dst. If the above edge (p, q) is a dimension-b edge, then $Dim(i, j) = b$. If $Src(i, j) = p, Dst(i, j) = q$ and $Dim(i, j) = b$, then $p|_b = q$ and $q|_b = p$. Thus we need only specify any two of Src, Dst and Dim to fully specify the mapping. In most of this thesis we use the arrays Dim and Dst.

2.4.3 Aggregates

In this thesis we consider weak hypercubes in which a node cannot use all its edges simultaneously. The mapping must use this fact to reduce the number of lasers and detectors. If two channels (ν_i, λ_j) and (ν_i, λ_{j+1}) (that share the same mode and adjacent wavelengths) can never be used simultaneously (possibly because they are mapped to edges incident on the same node), then a single tunable laser positioned at mode ν_i can switch between wavelengths λ_j and λ_{j+1} depending on which channel the node wishes to use. Similarly, if channels (ν_i, λ_j) and (ν_{i+1}, λ_j) cannot be used simultaneously, light from a single laser spreads over two channels and a spatial light modulator (SLM) regulates the transmission of the light in the two channels. Similarly at the destination, a single detector can be used to detect signal at such adjacent channels that cannot be used simultaneously.

Vaidyanathan and Sethuraman [33] have captured this idea in terms of the notion of aggregates. A *source* (resp. *destination*) *aggregate* is a maximal stretch of contiguous rows

or columns of array Src (resp. Dst) such that each entry in this stretch is either empty (unused) or has the same processor index. Figures 2.11, 2.12 and 2.13 show aggregates circled for three mappings. An aggregate containing less than two non-empty entries is called a *trivial aggregate*. Figures 2.11, 2.12 and 2.13 does not circle trivial aggregates. Also, these figures contain no empty entries.

The source aggregate in row 0 and columns 0 and 1 of Figure 2.13 represents a single laser, positioned at mode ν_0 that tunes to wavelengths λ_0 or λ_1. Since processor 1 uses channels (ν_0, λ_0) and (ν_0, λ_1) to communicate with processors 0 and 3, it cannot do so simultaneously (by the definition of a weak model). Therefore a single tunable laser suffices. Thus, the number of source aggregates equals the number of lasers needed. In fact, the number of source aggregates labeled with processor index p is the number of lasers needed for processor p. Similarly the number of destination aggregates with entry q is the number of detectors needed for processor q. For example, Mapping 1 of Figure 2.11 needs 24 lasers and 19 detectors, Mapping 2 of Figure 2.12 needs 24 lasers and 8 detectors and Mapping 3 of Figure 2.13 needs 12 lasers and 8 detectors. Thus, not all mappings are the same.

In this thesis we use "standard mappings" [31, 33]. A standard mapping uses an $R \times N$ channel array for an N-node topology. In addition, the destination array has wavelength aggregates that span entire columns (as in Figures 2.12 and 2.13). It can be shown that for a standard mapping of a simple graph, all source aggregates are mode aggregates. Note that the destination array of a standard mapping can be specified by specifying the N entries, one per column, as each column will form one aggregate. For example, the array Dst for Mapping 3 in Figure 2.13 could be specified as the list $\langle 0, 3, 6, 5, 1, 2, 7, 4 \rangle$.

A detailed description of aggregates and standard mapping appears in Sethuraman [31].

Src

	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
ν_0	4	6	1	0	3	2	3	7
ν_1	0	2	5	7	5	7	6	4
ν_2	6	1	4	1	2	0	5	3

Dst

0	4	3	2	1	3	2	6
1	0	7	3	1	5	7	6
2	0	5	5	6	4	4	7

Dim

2	1	1	1	1	0	0	0
0	1	1	2	2	1	0	1
2	0	0	2	2	2	0	2

FIGURE 2.11: Mapping 1

Src

	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
ν_0	4	2	7	1	3	0	3	6
ν_1	1	7	4	7	0	6	5	0
ν_2	2	1	2	4	5	3	6	5

Dst

0	3	6	5	1	2	7	4
0	3	6	5	1	2	7	4
0	3	6	5	1	2	7	4

Dim

2	0	0	2	1	1	2	1
0	2	1	1	0	2	1	2
1	1	2	0	2	0	0	0

FIGURE 2.12: Mapping 2

Src

	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
ν_0	1	1	4	4	0	0	5	5
ν_1	2	2	2	1	3	3	3	0
ν_2	4	7	7	7	5	6	6	6

Dst

0	3	6	5	1	2	7	4
0	3	6	5	1	2	7	4
0	3	6	5	1	2	7	4

Dim

0	1	1	0	0	1	1	0
1	0	2	2	1	0	2	2
2	2	0	1	2	2	0	1

FIGURE 2.13: Mapping 3

Dst list : 0 3 6 5 1 2 7 4

Dim

0	1	1	0	0	1	1	0
1	0	2	2	1	0	2	2
2	2	0	1	2	2	0	1

FIGURE 2.14: A way to specify the standard mapping, Mapping 3 of Figure 2.13

As far as we are concerned, we will assume a standard mapping and aim to reduce the number of source aggregates.

A standard mapping does not specify the number of rows. For a d-dimensional hypercube with 2^d-nodes and $d2^d$ directed edges, a standard mapping requires an $R \times 2^d$ channel array where $R \geq d$. When $R = d$, we call the mapping *dense*, as all channels of the channel array are used for hypercube edges. Otherwise it is a *sparse* mapping.

In this thesis we will map a d-dimensional hypercube on an $R \times 2^d$ channel array, where $R = d$ or $R = 2^{d-1}$. We will use a standard mapping and specify the mappings as an $X \times 2^d$ dimension array (Dim) and a 2^d-element destination list. For example, Mapping 3 of Figure 2.13 will be specified as shown in Figure 2.14. From the destination list and the dimension array, the array, Src, can be derived. To obtain the entries of Src, we complement the bit positions of the entries in Dst in the order mentioned in the corresponding entry of the array Dim. In Figure 2.14 consider $Dim(2,3) = 1$. Changing bit 1 in the binary representation of the corresponding element in (position 3) in Dst list (which is 5) gives $5|_1 = 7$. This is the value of the entry in $Src(2,3)$ for Mapping 3 (see Figure 2.13).

Chapter 3
Dense Mapping

This chapter deals with mapping a d-dimensional weak hypercube on to an optical slab waveguide. As noted in Chapter 2 this involves specifying two of the following three arrays: (1) the source array, Src, (2) the destination array, Dst and (3) the dimension array, Dim, all of which are of size $d \times N$. (Recall that for a dense mapping the size of the channel array is $d \times N$, where d is the number of dimensions in the hypercube and $N = 2^d$ is the number of nodes of the hypercube.) The array Dst uses wavelength aggregates (that span entire columns) as a result of which the array Src has mode aggregates (that run along rows). Because the destination array has aggregates spanning entire columns, one could represent Dst as an N-element list. In this chapter, we define the mapping by specifying the destination (Dst) and dimension (Dim) arrays.

We begin by defining arrays Dst and Dim in Sections 3.1 and 3.2, respectively. The proof for why these array definitions work for a weak hypercube follows in Section 3.4. Then, we determine the number of aggregates (numbers of lasers and detectors) in the mapping in Section 3.4.2.

3.1 Construction of the Destination Array, Dst

In a standard mapping, each column of the $d \times N$ destination array will form one aggregate (see Figure 3.1); i.e., all entries in a column are the same. The hypercube we consider in this thesis has N nodes and therefore the array Dst will have N columns, and hence, N aggregates. Since each column has only one distinct element, the array Dst can be specified

as an N-element list of nodes of the hypercube. We use the standard Hamiltonian path to

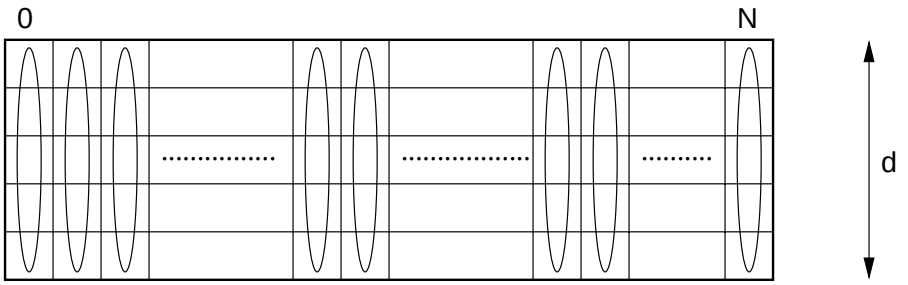


FIGURE 3.1: Structure of the destination array

construct the list that specifies the entries of the destination array. For any $d \geq 1$, let j be a d-bit number. The well-known shuffle permutation $\sigma : \{0, 1, \cdots, 2^d - 1\} \rightarrow \{0, 1, \cdots, 2^d - 1\}$ is defined as follows.

$$\sigma(j) = \begin{cases} 2j, & \text{if } 0 \leq j < 2^{d-1} \\ 2j + 1 - 2^d, & \text{if } 2^{d-1} \leq j < 2^d \end{cases}$$

If j has a binary representation $j_{d-1} j_{d-2} \cdots j_1 j_0$, where $j_\ell \in \{0, 1\}$ for each $0 \leq \ell < d$, then $\sigma(j)$ creates a number whose binary representation is a left rotate of j. For example, if $d = 3$, then $\sigma(2) = \sigma(010_2) = 100_2 = 4$ while $\sigma(5) = \sigma(101_2) = 011_2 = 3$.

Coming back to the array Dst, let $h_d = \langle h_d(j) : 0 \leq j < 2^d \rangle$ be the standard Hamiltonian path of a d-dimensional hypercube H_d. Let $\hat{h}_d = \langle h_d(\sigma(j)) : 0 \leq j < 2^d \rangle$ be the shuffled Hamiltonian path. For example with $d = 3$, $h_3 = \langle 0, 1, 3, 2, 6, 7, 5, 4 \rangle$ and $\hat{h}_3 = \langle 0, 3, 6, 5, 1, 2, 7, 4 \rangle$. Assign the list $\hat{h}$ to the columns of arry Dst. Specifically, for $0 \leq i < d$ and $0 \leq j < 2^d$, the entry $Dst(i, j) = h_d(\sigma(j))$. Figure 3.2 shows array Dst for $d = 3$.

As another example with $d = 4$, $h_4 = \langle 0, 1, 3, 2, 6, 7, 5, 4, 12, 13, 15, 14, 10, 11, 9, 8 \rangle$. The list $\hat{h}_4$ is obtained by separating the odd and even elements we have the list $\langle 0, 3, 6, 5, 12, 15, 10,$

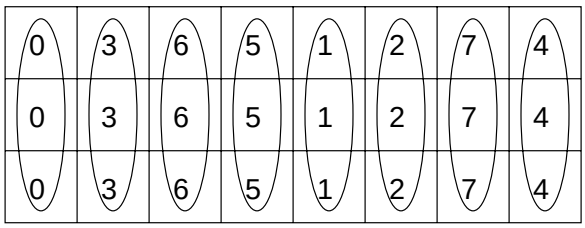


FIGURE 3.2: Array Dst for $d = 3$

$9, 1, 2, 7, 4, 13, 14, 11, 8\rangle$. This implies that array Dst has the form in Figure 3.3. The two halves of the array Dst have been separated in Figure 3.3 for reasons that will be clearer later.

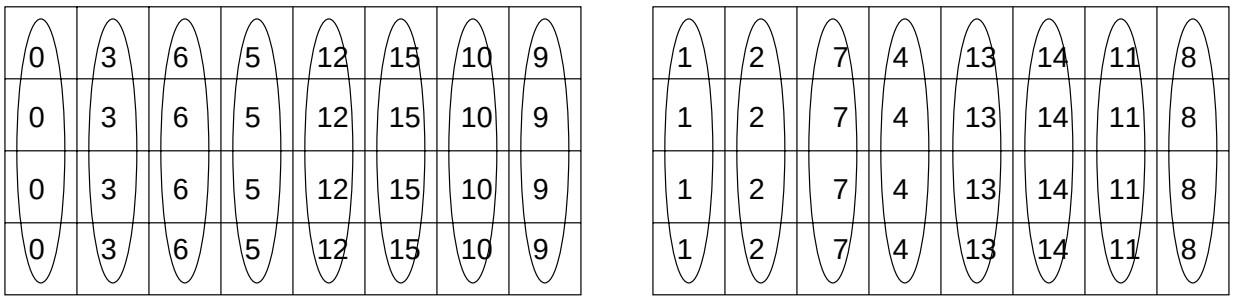


FIGURE 3.3: Array Dst for $d = 4$

We now state a technical result that is used later.

Lemma 3.1 *For $d \geq 2$, $\hat{h}_d = \hat{h}_d^0 \circ \hat{h}_d^1 \circ \hat{h}_d^2 \circ \hat{h}_d^3$, where for any $0 \leq i < 2^{d-2}$*

$$\begin{aligned}
\hat{h}_d^0(i) &= h_{d-1}(2i) \\
\hat{h}_d^1(i) &= 2^{d-1} + h_{d-1}(2^{d-1} - 1 - 2i) \\
\hat{h}_d^2(i) &= h_{d-1}(2i + 1) \\
\hat{h}_d^3(i) &= 2^{d-1} + h_{d-1}(2^{d-1} - 2 - 2i).
\end{aligned}$$

Proof: Figure 3.4 shows the relationship between $h_d, \hat{h}_d$ and $\hat{h}_d^x$ for $0 \leq x < 3$ when $d = 4$.

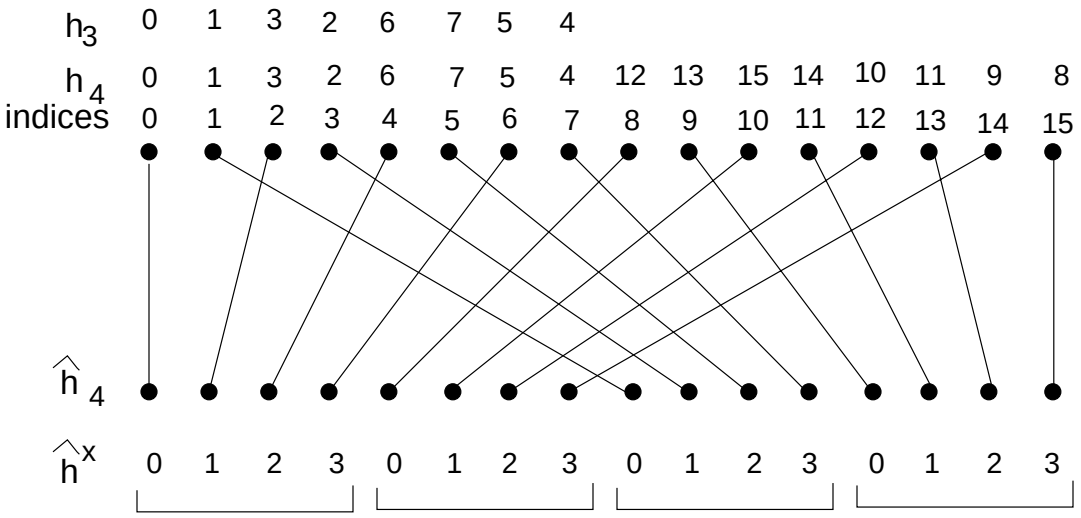


FIGURE 3.4: Relation between $h_d, \hat{h}_d$ and $\hat{h}_d^x$

Recall that $h_d = h_{d-1} \circ h'_{d-1}$, where $h'_{d-1} = \overline{(h_{d-1} + 2^{d-1})}$ and $\overline{s}$ denotes the reverse of string s (see Definition 2.3).

From Figure 3.4 it is clear that $\hat{h}_d^0$ is simply the even indexed elements of h_{d-1}. This relationship can be proved more formally by induction on d. Therefore for $0 \leq i < 2^{d-2}$, $\hat{h}_d^0(i) = h_{d-1}(2i)$. Similarly $\hat{h}_d^2$ is the odd numbered elements of h_{d-1}, which establishes that for $0 \leq i < 2^{d-2}$, $\hat{h}_d^2(i) = h_{d-1}(2i+1)$.

Again from Figure 3.4, $\hat{h}'_d$ is the even elements of $\overline{h'_{d-1}}$, as a result of which for each $0 \leq i < 2^{d-2}$ we have

$$\hat{h}_d^1(i) = \overline{h'_{d-1}}(2i). \tag{3.1}$$

The following facts stem from Definition 2.3 (8) applied to the current context.

$$\overline{h'_{d-1}}(j) = h'_{d-1}(2^{d-1} - 1 - j) \tag{3.2}$$

Note that we have used the fact that h'_{d-1} is a (2^{d-1})-element string.

$$h'_{d-1}(j) = 2^{d-1} + h_{d-1}(j) \tag{3.3}$$

From Equations 3.1, 3.2 and 3.3 we have

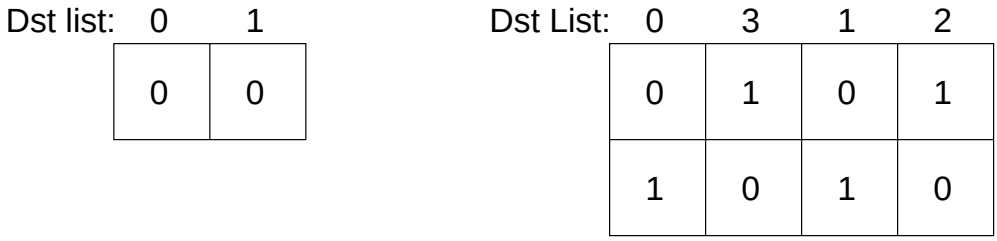
$$\hat{h}'_d = \overline{h'}(2i) = h'(2^{d-1} - 1 - 2i) = 2^{d-1} + h_{d-1}(2^{d-1} - 1 - 2i). \tag{3.4}$$

Finally for $\hat{h}^3_d$ we proceed as for $\hat{h}^1_d$ with the observation that $\hat{h}^3_d$ is the odd elements of $\overline{h'_{d-1}}$. The expression for $\hat{h}^3_d$ can be obtained by substituting $2i$ in Equation 3.4 by $2i + 1$. ■

3.2 Construction of the Dimension Array, Dim

The dimension array contains dimension numbers between 0 and $d-1$. For any $0 \leq i < d$ and $0 \leq j < 2^{d-1}$, if $Dst(i, j) = x$ and $Dim(i, j) = y$, then they denote the edge $(x|_y, x)$, where $x|_y$ is the dimension-y neighbor of x; that is, the binary representation of $x|_y$ is the binary representation of x with bit y complemented. Recall that all entries within a column of the destination array are the same. Therefore, each column j of the array Dim will specify all d dimensions in order for the processor in column j of array Dst to be able to communicate with all its d neighbors in the hypercube. Figure 3.5 shows the dimension array for $d = 1$ and $d = 2$.

To define the entries of array Dim for $d \geq 3$, we introduce three quantities α_j, β_j and γ_j, for each column $0 \leq j < N$. These three quantities fix the values of three entries in each column j of the dimension array. The remaining entries in a column will be the remaining dimensions in any order. The motivation to specify these three entries for every column will



(a) $d = 1$ (b) $d = 2$

FIGURE 3.5: Examples of the array Dim

be evident later when we derive the number of aggregates for this mapping.

For any integer $j \geq 0$, define $R_0(j)$ to be the position of the rightmost 0 in the binary representation of j; the lsb is in position 0. For example, if $j = 101_2$ then the rightmost 0 is in position 1, so $R_0(5) = 1$. If $j = 7_{10} = 111_2$ then the rightmost 0 is in position 3, so $R_0(7) = 3$. It is easy to see that for any even integer j, $R_0(j) = 0$.

Lemma 3.2 *For any b-bit number j,* $R_0(j)$

(a) $R_0(j) = b$ *iff* $j = 2^b - 1$

(b) $R_0(j) = b - 1$ *iff* $j = 2^{b-1} - 1$

(c) $R_0(j) < b - 1$ *iff* $j \neq 2^b - 1$ *and* $j \neq 2^{b-1} - 1$

Proof: We consider the three cases.

Case (a): Since $2^b - 1 = 0\underbrace{111\cdots11}_{b\ \texttt{bits}}{}_2$ the rightmost 0 is in position b i.e., $R_0(j) = b$.

Case (b): Like the previous case, the binary representation of 2^{b-1} is $0\underbrace{111\cdots11}_{b-1\ \texttt{bits}}{}_2$. Therefore, the rightmost 0 is in position $b - 1$ (the msb); i.e., $R_0(j) = b - 1$.

Case (c): From the above two cases it is clear that $R_0(j)$ can be b or $b-1$ only if $j = 2^b - 1$ or $j = 2^{b-1} - 1$. Therefore, $R_0(j)$ has to be less than $b-1$ if $j \neq 2^b - 1$ and $j \neq 2^{b-1} - 1$.

■

We now define α_j, β_j and γ_j for each column j.

Definition 3.3 For any $d \geq 3$ (where d is the number of dimensions in the hypercube) and $0 \leq j < 2^{d-1}$

$$\alpha_j^d = \alpha_{j+2^d}^d = 0$$

$$\beta_j^d = \beta_{j+2^d}^d = \begin{cases} d-1, & \text{if } j = 0 \\ \gamma_{j-1}^d, & \text{if } j > 0 \end{cases}$$

$$\gamma_j^d = \gamma_{j+2^d}^d = \begin{cases} d-1, & \text{if } j = 2^{d-1} - 1 \\ 1 + R_0(j), & \text{otherwise} \end{cases}$$

□

Remark: If the dimension d is implicit, we write α_j^d, β_j^d and γ_j^d, simply as α_j, β_j and γ_j.

Lemma 3.4 *For $d \geq 3$ and $0 \leq j < 2^d$, we have $0 \leq \alpha_j, \beta_j, \gamma_j < d$ and $\alpha_j, \beta_j, \gamma_j$ are distinct.*

Proof: Clearly, $\alpha_j = 0$ is in the right range. Also $\gamma_j > 0$ as it is $d-1$ (where $d \geq 3$ and hence $d - 1 \geq 2$) or $1 + R_0(j)$. So $\gamma_j \neq \alpha_j$. $\beta_j > 0$ as β_j equals $d-1$ or γ_{j-1}, both of which are > 0. So $\beta_j \neq \alpha_j$. In the remainder of this proof, we establish that $0 < \beta_j, \gamma_j < d$ and that $\beta_j \neq \gamma_j$.

Since $d \geq 3$, the value of $d-1$ for β_j or γ_j is in the right range. From Lemma 3.2, $R_0(j) < d-1$, except when $j = 2^d - 1$ or $j = 2^{d-1} - 1$. So here $\gamma_j = 1 + R_0(j) < d$. For the two cases noted above $\gamma_j = 2$. So for all $0 \leq j < 2^d$, we have $\gamma_j < d$. This implies for all j, $\beta_j = \gamma_{j-1} < d$.

To prove $\beta_j \neq \gamma_j$ we first observe that $\beta_0 = d - 1 \neq 1 = \gamma_0$. We now show that for all $0 < j < 2^d$, $\gamma_j \neq \gamma_{j-1} = \beta_j$.

Consider the case where $j, j-1 \notin \{2^d - 1, 2^{d-1} - 1\}$. Here j is odd iff $j-1$ is even. Also $R_0(j) = 0$ iff j is even. So $R_0(j) \neq R_0(j-1)$. This implies that $\gamma_j \neq \gamma_{j-1}$.

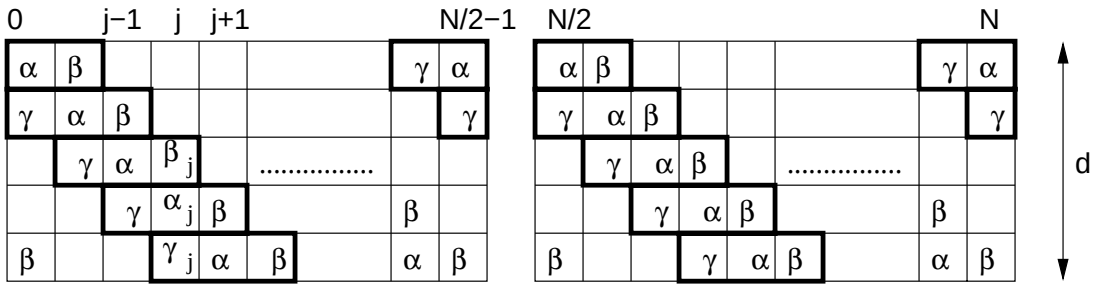


FIGURE 3.6: Positions of α, β and γ in the dimension array

If $j \in \{2^d - 1, 2^{d-1} - 1\}$ then $\gamma_j = d - 1$ but $\gamma_{j-1} = 1$ as $2^d - 2$ and $2^{d-1} - 2$ are even numbers. Similarly if $j - 1 \in \{2^d - 1, 2^{d-1} - 1\}$ then j is an even number with $\gamma_j = 1$ and $\gamma_{j-1} = 2$. ■

The above lemma ensures that each column j of array Dim can contain α_j, β_j and γ_j representing edges of H_d in three distinct dimensions. Figure 3.6 describes the arrangement of α_j, β_j and γ_j in column j. More formally, for $0 \leq j < 2^{d-1}$ consider entries $Dim(i, j)$ and $Dim(i, j + 2^d)$ of array Dim.

$$Dim(i,j) = Dim(i,j+2^d) = \begin{cases} \alpha_j & \text{iff } i \equiv j \pmod{d} \\ \gamma_j & \text{iff } i-1 \equiv j \pmod{d} \\ \beta_j & \text{iff } i+1 \equiv j \pmod{d} \\ \text{any unused dimension} & \text{otherwise.} \end{cases} \tag{3.5}$$

Notice in Figure 3.6 that $d - 3$ entries of each column j in the dimension array are filled by the remaining dimensions in any order ("otherwise" clause in the formal description above). Notice also that the arrangement repeats for columns j where $0 \leq j < 2^{d-1}$ and $2^{d-1} \leq j < 2^d$. Consequently the array Dim (and the array Dst of Figure 3.3) are shown split into two halves. Figure 3.7 is an example of the 4-dimensional array Dim.

0	1	3	3	0	1	3	3
1	0	2	2	1	0	2	2
2	2	0	1	2	2	0	1
3	3	1	0	3	3	1	0

0	1	3	3	0	1	3	3
1	0	2	2	1	0	2	2
2	2	0	1	2	2	0	1
3	3	1	0	3	3	1	0

FIGURE 3.7: Array Dim for a 4-dimensional hypercube

3.3 Construction of Source Array, Src

Given the destination list and dimension array, the construction of the source array is straightforward. This can be done by complementing the bit in the binary representation of a destination list element that is specified by the corresponding element of the array Dim. Specifically, if $h_d(j)$ is the element in $Dst(i,j)$, then $Src(i,j) = \hat{h}_d(j)|_{Dim(i,j)} = \hat{h}_d(j)$ with bit $Dim(i,j)$ complemented. Figure 3.8 shows array Src and $\hat{h}_4(j)$ for the array Dim in Figure 3.7. Every entry in any given column j of the source array is different as it is connected to node $\hat{h}_4(j)$ by a different dimension edge. Therefore, there are no wavelength aggregates in Src. (If there were, it would mean that an edge has been mapped more than once). In the next section we determine the mode aggregates of array Src.

$\hat{h}_4(j)$

0	3	6	5	12	15	10	9
1	1	14	13	13	13	2	1
2	2	2	1	14	14	14	13
4	7	7	7	8	11	11	11
8	11	4	4	4	7	8	8

1	2	7	4	13	14	11	8
0	0	15	12	12	12	3	0
3	3	3	0	15	15	15	12
5	6	6	6	9	10	10	10
9	10	5	5	5	6	9	9

FIGURE 3.8: Src array for a 4-dimensional hypercube

3.4 Analyzing the Mapping

In Sections 3.1 and 3.2 we specified the mapping of the hypercube to the slab by constructing arrays Dst and Dim. In this section we first prove the mapping correct (i.e., all edges of hypercube H_d have been mapped). Then we derive the number of aggregates due to this mapping.

3.4.1 Correctness

Observe that all entries in a column of array Dst are the same and entries in different columns are different (this is obvious from the construction). Next note that for array Src all entries in a column are different (this was established by Lemma 3.4 and Equation (3.5)). Therefore, each pair $\langle Src(i,j), Dst(i,j)\rangle$ represents a different directed edge of H_d. Since the $d \times 2^d$ channel array has $d2^d$ entries (which is also the number of directed edges in H_d), and since all these entries represent different edges, all edges of H_d are covered.

Lemma 3.5 *For any $d \geq 1$, the mapping specified in Sections 3.1-3.3 correctly map's each edge of a d-dimensional hypercube H_d on to a distinct channel of a $d \times 2^d$ channel array.* ■

We now develop some technical results before we proceed to derive the number of aggregates (Section 3.4.2).

Lemma 3.6 *Let $0 \leq i < d$ and $0 \leq j < 2^{d-1}$. If $Dim(i,j) = \alpha_j$, then $Dim(i,j-1) = \gamma_{j-1}$ and $Dim(i,j+1) = \beta_{j+1}$ (if these cells exist in one half of the source array.)*

Proof: From Equation (3.5), α_j is in entry $Dim(i,j)$ iff $i \equiv j \bmod d$. Let α_j be in row r_j, i.e., $i = r_j$. Then from Equation (3.5) it is clear that β_j is in row $(r_j - 1) \pmod d$ and γ_j is in row $(r_j + 1) \pmod d$. Similarly if $j > 0$, then in column $j-1$ we have α_{j-1} in row

$r_{j-1} \equiv (j-1) \pmod{d}$ where $r_{j-1} = (r_j - 1) \pmod{d}$. Similarly β_{j-1} is in row $(r_j - 2) \pmod{d}$ and γ_{j-1} is in row $r_j \pmod{d}$. This establishes one half of the lemma namely α_j and β_{j-1} are in the same row. The other half establishing α_j and γ_{j+1} to be in the same row follows similarly. ■

Lemma 3.7 *For any $0 \leq j < 2^{d-1}$, if $Dim(i,j) = \alpha_j$, then $Src(i,j-1) = Src(i,j) = Src(i,j+1)$, if they exist.*

Proof: By Lemma 3.6, $Dim(i,j-1) = \gamma_{j-1}$ and $Dim(i,j+1) = \beta_{j+1}$. Therefore,

$$Src(i,j) = Dst(i,j)|_{Dim(i,j)} = \hat{h}(j)|_{\alpha_j} = h(2j)|_0 \tag{3.6}$$

$$Src(i,j-1) = Dst(i,j-1)|_{Dim(i,j-1)} = \hat{h}(j-1)|_{\gamma_{j-1}} = h(2j-2)|_{\gamma_{j-1}} \tag{3.7}$$

$$Src(i,j+1) = Dst(i,j+1)|_{Dim(i,j+1)} = \hat{h}(j+1)|_{\beta_{j+1}} = h(2j+2)|_{\beta_{j+1}} \tag{3.8}$$

We need to show that for any $0 \leq j < 2^{d-1}$, $h(2j)|_0 = h(2j-2)|_{\gamma_{j-1}} = h(2j+2)|_{\beta_{j+1}}$ whenever these entries exist.

We know from the construction of array Dst that $\hat{h}(j) = h(2j)$, for $0 \leq j < 2^{d-1}$. Therefore columns $j-1, j$ and $j+1$ of array Dst will have entries $h(2j-2), h(2j), h(2j+2)$ respectively. Let $h(2j) = a$, $h(2j-2) = c$ and $h(2j+2) = b$ and represent them in the following manner.

$$\begin{aligned} h(2j) &= a = a_{d-1}a_{d-2}\cdots a_1a_0 \\ h(2j-2) &= b = b_{d-1}b_{d-2}\cdots b_1b_0 \\ h(2j+2) &= c = c_{d-1}c_{d-2}\cdots c_1c_0 \end{aligned}$$

Let $j = 0j_{d-2}j_{d-3}\cdots j_1j_0$; recall that $0 \leq j < 2^{d-1}$. Let k and ℓ (where $0 \leq k, \ell < d$) be the positions of the rightmost 0 and the rightmost 1, respectively, in the binary representation

of j. Then Figures 3.9 and 3.10 show the bit values in $j, j+1, j-1, 2j, 2(j+1)$ and $2(j-1)$. In those figures $A = \lfloor j/2^{k+1} \rfloor$ and $B = \lfloor j/2^{\ell+1} \rfloor$. Note that A and B are $d-(k+2)$ and $d-(\ell+2)$ bit quantities. In fact, $A = j_{d-2}j_{d-3}\cdots j_{k+1}$ and $B = j_{d-2}j_{d-3}\cdots j_{\ell+1}$. Define $\hat{A} = j_{d-2}\hat{j}_{d-3}\cdots\hat{j}_{k+2}j_{k+1}$ and $\hat{B} = j_{d-2}\hat{j}_{d-3}\cdots\hat{j}_{\ell+2}j_{\ell+1}$ with $\hat{j}_p = j_p \oplus j_{p+1}$.

Observe now that any two of $h(2j), h(2j-2)$ and $h(2j+2)$ differ in exactly 2 bits (regardless of the case). Figure 3.11 gives the bits where they differ. Since $\alpha_j = 0$, $h(2j)|_{\alpha_j} = h(2j)|_0$ differs from $h(2j-2)$ in bit 1 or $\ell+1$ (depending on whether $k > 0$ or $k = 0$). Similarly, $h(2j)|_{\alpha_j}$ differs from $h(2j+2)$ in bits $k+1$ or 1.

Thus for $k > 0$

$$h(2j)|_0 = h(2j-2)|_1 = h(2j+2)|_{k+1}, \tag{3.9}$$

and for $k = 0$

$$h(2j)|_0 = h(2j-2)|_{\ell+1} = h(2j+2)|_1. \tag{3.10}$$

We now show that the bit numbers for $h(2j-2)$ and $h(2j+2)$ in Equations 3.9 and 3.10 are precisely γ_{j-1} and β_{j+1}. We consider two cases.

Case $k > 0$: From Figure 3.9, $R_0(j-1) = 0$, so $\gamma_{j-1} = 1$. Since $j < 2^{d-1}$, we have $0 \le k \le d-1$. If $k = d-1$, then $j = 2^{d-1} - 1$, and $Src(i, j+1)$ would be outside the half of the array Src that we are considering. So $k < d-1$. This implies that $\gamma_j = 1 + k = \beta_{j+1}$. Thus Equation 3.9 can be written as:

$$h(2j)|_{\alpha_j} = h(2j-2)|_{\gamma_{j-1}} = h(2j+2)|_{\beta_{j+1}}.$$

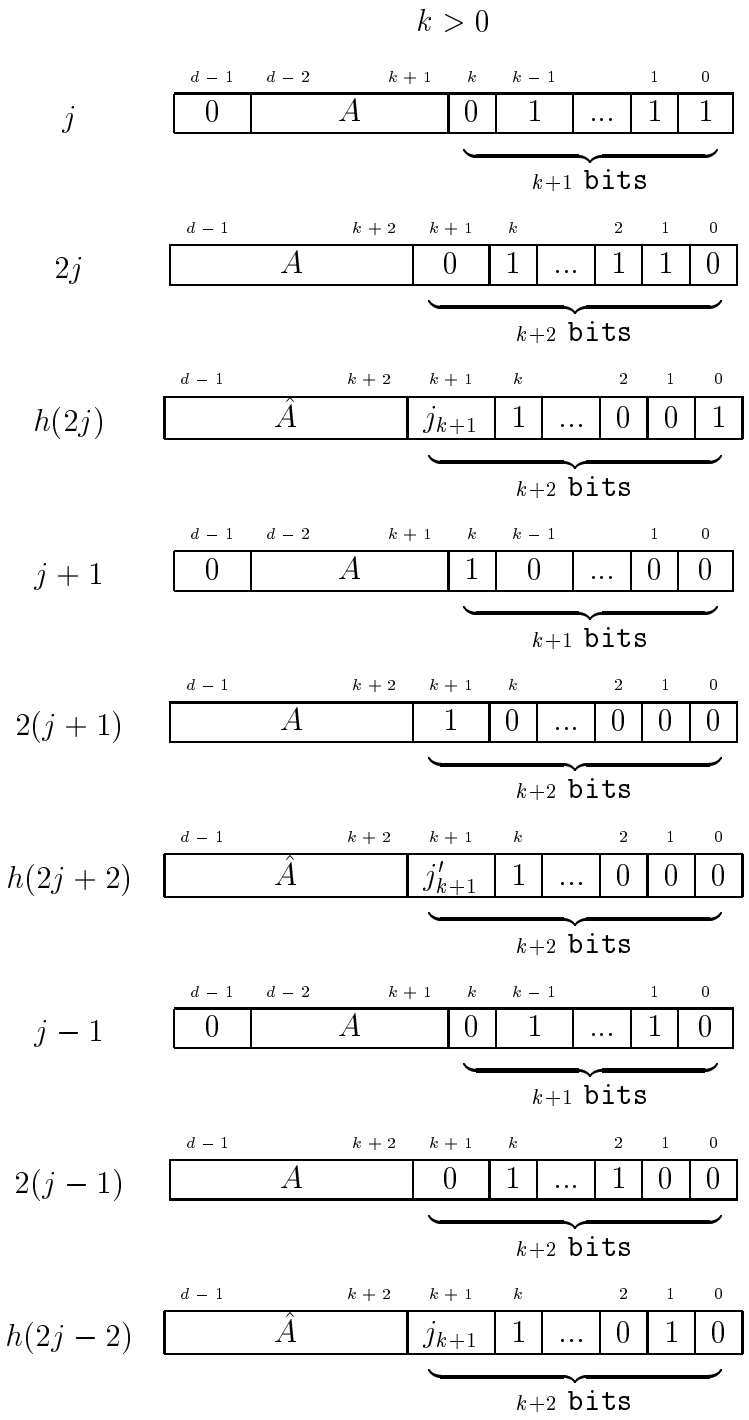


FIGURE 3.9: Bit values for the proof of Lemma 3.7 when $k > 0$

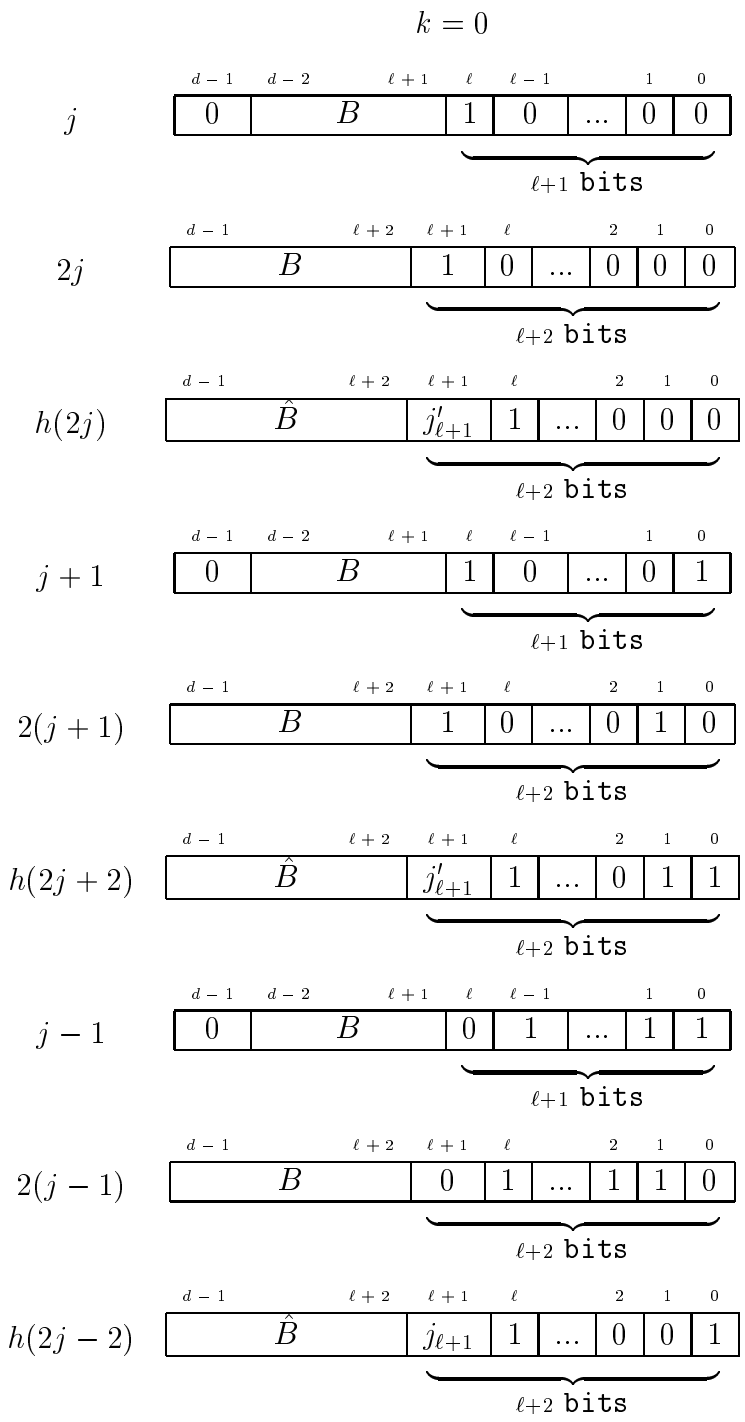


FIGURE 3.10: Bit values for the proof of Lemma 3.7 when $k = 0$

	$h(2j-2)$		$h(2j+2)$	
	$k>0$	$k=0$	$k>0$	$k=0$
$h(2j)$ $h(2j-2)$	$0,1$	$0,\ell+1$	$0,k+1$ $1,k+1$	$0,1$ $1,\ell+1$

FIGURE 3.11: Differing bits in the proof of Lemma 3.7

Case $k=0$: From Figures 3.9 and 3.10, $j-1$ has the leftmost 0 in position ℓ. That is, $R_0(j-1)=\ell$. From Equation 3.10, $\gamma_{j-1}=\ell+1$. Also $R_0(j)=k=0$ (see Figure 3.10). Then $\gamma_j=\beta_{j+1}=1$. Thus Equation 3.10 can be written as:

$$h(2j)|_{\alpha_j}=h(2j-2)|_{\gamma_{j-1}}=h(2j+2)|_{\beta_{j+1}}.$$

With both cases we may write for all elements with columns $0,1,\cdots,2^{d-1}-1$ of array Src

$$h(2j)|_{\alpha_j}=Src(i,j)=h(2j-2)|_{\gamma_{j-1}}=Src(i,j-1)=h(2j+2)|_{\beta_{j+1}}=Src(i,j+1).$$

This completes the proof. ■

Lemma 3.8 *For any $2^{d-1}\le j<2^d$, if $Dim(i,j)=\alpha_j$, then $Src(i,j-1)=Src(i,j)=Src(i,j+1)$, if they exist.*

The proof for this case follows along the same lines as the $0\le j<2^{d-1}$ case except that bit $d-1$ will be a 1 instead of a 0. The argument provided still holds good and therefore we skip the proof for this case. ■

3.4.2 Number of Aggregates

In this section we derive the number of aggregates in the source and destination arrays.

Destination Array: Since all aggregates in the array Dst span entire columns, the number of aggregates is N (see Figure 3.3 for an example).

Source Array: An aggregate is trivial iff it has a size of 1 (as there are no empty entries). Here we will establish that array Src has aggregates of size 1 (trivial), 2 and 3. Each non-trivial aggregate will be associated with an α_j for some j.

Lemma 3.9 *For each $0 \leq j < 2^{d-1}$, there is a non-trivial aggregate containing α_j.*

Proof: If $j = 0$, then $Dim(0,0) = \alpha_0$ and $Dim(0,1) = \beta_1$. By Lemma 3.7, $Src(0,0) = Src(0,1)$, forming a 2-element aggregate. If $j = 2^{d-1} - 1$, then for $i = (2^{d-1} - 1) \pmod{d}$, $Dim(i,j) = \alpha_j$ and $Dim(i,j-1) = \gamma_{j-1}$. Again by Lemma 3.7, $Src(i,j) = Src(i,j-1)$, forming another 2-element aggregate. For any $0 \leq j < 2^{d-1}$ and $i \equiv j \pmod{d}$, we have $Dim(i,j-1) = \gamma_{j-1}, Dim(i,j) = \alpha_j$ and $Dim(i,j+1) = \beta_{j+1}$. Once again Lemma 3.7 ensures that $Src(i,j-1) = Src(i,j) = Src(i,j+1)$ to form a 3-element aggregate. ■

Theorem 3.10 *For any $d \geq 2$ the array Src has $2^d(d-2)+4$ aggregates.*

Proof: From the proof of Lemma 3.9, one half of array Src xhas 2-element aggregates and $2^{d-1} - 2$ 3-element aggregates. The remaining $d2^{d-1} - (2 \times 2 + 3(2^{d-1} - 2)) = (d-3)2^{d-1} + 2$ elements are trivial aggregates. Thus, the number of aggregates (in both halves of Src) is $2(2 + 2^{d-1} - 2 + (d-3)2^{d-1} + 2) = 2((d-2)2^{d-1} + 2) = (d-2)2^d + 4$. ■

Remark: In Chapter 5 we show that this is the optimal number of aggregates.

Theorem 3.11 *For any $d \geq 2$, a d-dimensional hypercube has the following mappings on a slab waveguide.*

(a) With 2^d detectors and $(d-2)2^d + 4$ tunable lasers.

(b) With $(d-2)2^d + 4$ *detectors and* 2^d *tunable lasers.*

(c) With 2^d *detectors and* $(d-2)2^d + 4$ *lasers with SLMs.*

(d) With $(d-2)2^d + 4$ *detectors and* 2^d *lasers with SLMs.* ■

Chapter 4
Sparse Mapping

In the previous chapter we have shown that it is possible to map a d-dimensional hypercube onto a sawtooth slab waveguide by a dense mapping using $2^{d-1}(d-2)+2$ lasers and $N = 2^d$ detectors (or vice versa). In this chapter we will consider larger channel arrays, specifically $\frac{N}{2} \times N$ arrays. Since the number of edges in the hypercube is dN, not all entries of this channel array are used when $d \geq 3$. The unused channels are said to be "empty." We call such a mapping as a sparse mapping. Note that a sparse mapping can also be a standard mapping; in fact, we will use a standard sparse mapping with one wavelength aggregate per column of the destination array. As in the case of dense mapping, we specify the mapping through arrays Dst and Dim. (Clearly these arrays are of size $2^{d-1} \times 2^d$ or $\frac{N}{2} \times N$.)

In the next two sections we specify the dimension and destination arrays. The number of aggregates for this mapping will be discussed in Section 4.4.

4.1 Construction of the Dimension Array, Dim

Here we define the dimension array, Dim, recursively. As before, we define just one $2^{d-1} \times 2^{d-1}$ half of the array. The other half has the same entries as the first half. Before we proceed, we define some additional terms.

Definition 4.1 For any $d \geq 0$, *a reverse diagonal array* R is a $2^d \times 2^d$ array with rows numbered $0, 1, \cdots, 2^d - 1$ and columns numbered $0, 1, \cdots, 2^d - 1$. Entry $R(i, 2^d - 1 - i)$ has some non-empty value, for each $0 \leq i < 2^d$. All other entries are "empty" (or without value).

□

$$\begin{bmatrix} 1 \end{bmatrix} \qquad \begin{bmatrix} & 2 \\ 2 & \end{bmatrix} \qquad \begin{bmatrix} & & & 3 \\ & & 3 & \\ & 3 & & \\ 3 & & & \end{bmatrix}$$

(a) (b) (c)

FIGURE 4.1: Examples of reverse diagonal arrays

$$\begin{bmatrix} & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & & \end{bmatrix}$$

FIGURE 4.2: Unit reverse diagonal array, U_2

Figure 4.1 shows examples of reverse diagonal arrays.

Definition 4.2 For $d \geq 0$, a reverse diagonal array of size $2^d \times 2^d$, in which all non-empty entries have value 1 is called a *unit reverse diagonal array* and is denoted by U_d (see Figure 4.2).

□

For any array $A = \begin{bmatrix} A_2 & A_1 \\ A_3 & A_4 \end{bmatrix}$, decomposed into 4 quadrants, A_1, A_2, A_3, A_4, we will refer to he first, second, third and fourth quadrants as indicated by the subscripts (following the convention for angles from 0° to 360°).

Coming back to the definition of the array Dim, we first note that the array is for a d-dimensional hypercube. We use Dim_d to indicate this. Also since our definition is for a half of the array, we use $Dim_{d,1}$ and $Dim_{d,2}$ to denote the two halves. As noted earlier $Dim_{d,1} = Dim_{d,2}$. The entries of Dim_d can be defined recursively as follows.

$$Dim_{1,1} = \quad [0]$$

$$Dim_{2,1} = \quad \begin{bmatrix} Dim_{1,1} & 1U_0 \\ 1U_0 & Dim_{1,1} \end{bmatrix} = \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Dim_{3,1} = \quad \begin{bmatrix} Dim_{2,1} & 2U_1 \\ 2U_1 & Dim_{2,1} \end{bmatrix} = \quad \begin{bmatrix} 0 & 1 & & 2 \\ 1 & 0 & 2 & \\ & 2 & 0 & 1 \\ 2 & & 1 & 0 \end{bmatrix}$$

FIGURE 4.3: Recursive construction of $Dim_{3,1}$

Definition 4.3 $Dim_{1,1} = Dim_{1,2} = [0]$.

For all $d > 1$, $Dim_{d,1} = Dim_{d,2} = \begin{bmatrix} Dim_{d-1,1} & (d-1)U_{d-2} \\ (d-1)U_{d-2} & Dim_{d-1,1} \end{bmatrix}$. □

For example, Figure 4.3 shows the recursive construction of $Dim_{3,1}$. Recall that we defined three quantities α_j, β_j and γ_j for each column j of array Dim for dense mapping (Chapter 3). In the case of sparse mapping also the recursive definition provided above can be structured around the values of α_j, β_j and γ_j. Their positions, however, are defined in a larger array of size $2^{d-1} \times 2^{d-1}$. The following lemma places α_j^d, β_j^d and γ_j^d in Dim_d.

Lemma 4.4 *For any* $d \geq 3$ *and any* $0 \leq j < 2^{d-1}$

(*i*) $Dim(j, j) = \alpha_j^d = \alpha_{j+2^{d-1}}^d$

(*ii*) $Dim((j-1) \bmod (2^{d-1}), j) = \beta_j^d = \beta_{j+2^{d-1}}^d$

(*iii*) $Dim((j+1) \bmod (2^{d-1}), j) = \gamma_j^d = \gamma_{j+2^{d-1}}^d$

Proof: Since $Dim_{d,1} = Dim_{d,2}$, clearly $\alpha_j^d = \alpha_{j+2^{d-1}}^d, \beta_j^d = \beta_{j+2^{d-1}}^d$ and $\gamma_j^d = \gamma_{j+2^{d-1}}^d$. So we restrict our discussion to $0 \leq j < 2^{d-1}$. We proceed by induction on $d \geq 3$.

For the base case ($d = 3$) and from Figure 4.3 we find that $\alpha_j^3 = 0$ (for $0 \leq j < 4$), $\beta_0^3 = 2, \gamma_0^3 = \beta_1^3 = 1, \gamma_1^3 = \beta_2^3 = 2, \gamma_2^3 = \beta_3^3 = 1, \gamma_3^3 = 2$. By comparing these values with

$Dim_{3,1}$ shown in Figure 4.3, it is easy to verify that the lemma holds for $d = 3$.

Assume the lemma to hold for any $d \geq 3$, and consider

$$Dim_{d+1,1} = \begin{bmatrix} Dim_{d,1} & dU_{d-1} \\ dU_{d-1} & Dim_{d,1} \end{bmatrix}.$$

Proof of part (i): If $0 \leq j < 2^{d-1}$, then $Dim_{d+1,1}(j,j)$ is a main diagonal term of the first $Dim_{d,1}$ in the recursive expression for $Dim_{d+1,1}$. By the induction hypothesis and Definition 3.3, this element is $\alpha_j^d = 0 = \alpha_j^{d+1}$. Similarly if $2^{d-1} \leq j < 2^d$, then $Dim_{d+1,1}(j,j)$ is the diagonal element of its second $Dim_{d,1}$. Again, $Dim_{d+1,1}(j,j) = \alpha_{j-2^d-1}^d = 0 = \alpha_j^{d+1}$. Therefore part (i) of the lemma holds.

Proof of part (iii): If $0 \leq j < 2^{d-1} - 1$, then $Dim_{d+1,1}((j+1) \bmod (2^d), j)$ is element $((j+1), j)$ of the first $Dim_{d,1}$ in the recursive expression for $Dim_{d+1,1}$. By the induction hypothesis, an element of $Dim_{d,1}$ is $\gamma_j^d = 1 + R_0(j)$ (see Definition 3.3), where $R_0(j)$ is the position of the rightmost 0 in the binary representation of j. Observe that since $0 \leq j < 2^{d-1} - 1$, $R_0(j)$ is the same, regardless of whether j is represented by d bits or $d+1$ bits. That is, the rightmost 0 of j is not in bit position $d-1$. Thus $\alpha_j^d = 1 + R_0(j) = \alpha_j^{d+1}$. Element $Dim_{d+1,1}(2^{d-1}, 2^{d-1} - 1)$ is not in the above $Dim_{d,1}$, rather it is in the $(0, 2^{d-1} - 1)^{th}$ element of U_d. From Definition 3.3, $\gamma_j^d = d - 1$, when $j = 2^{d-1} - 1$. Since $Dim_{d+1,1}(2^{d-1}, 2^{d-1} - 1)$ is element $U_d(0, 2^{d-1} - 1) = d$, we have $Dim_{d+1,1}(2^{d-1}, 2^{d-1} - 1) = d = \gamma_{2^{d-1}-1}^{d+1}$. Thus part (iii) holds when $0 \leq j < 2^{d-1}$.

0	1		2				3
1	0	2				3	
	2	0	1		3		
2		1	0	3			
			3	0	1		2
		3		1	0	2	
	3				2	0	1
3				2		1	0

0	1		2				3
1	0	2				3	
	2	0	1		3		
2		1	0	3			
			3	0	1		2
		3		1	0	2	
	3				2	0	1
3				2		1	0

FIGURE 4.4: 4-dimensional sparse Dim array

If $2^{d-1} \leq j < 2^d - 1$, then $Dim_{d+1,1}((j+1) \bmod (2^d), j) = Dim_{d+1,1}((j+1), j)$ is the element $(j + 1 - 2^{d-1}, j - 2^{d-1})$ of the second $Dim_{d,1}$ in the recursive expression for $Dim_{d+1,1}$. By the induction hypothesis this element is $\gamma_j^d = 1 + R_0(j) = \gamma_j^{d+1}$ (as argued above). The proof for this part now follows along the same lines as the $0 \leq j < 2^{d-1}$ case. For $j = 2^d - 1$, $Dim_{d+1}((j+1) \bmod 2^d, j) = Dim_{d+1}(0, 2^d - 1) = d$. Thus part (iii) of the lemma is established.

Proof of part (ii): We have shown that for all $0 \leq j < 2^d$, $\gamma_j^{d+1} = Dim_{j+1,1}((j+1) \bmod (2^{d-1}), j)$. Then for $j > 0$, $\beta_j^{d+1} = \gamma_{j-1}^{d+1} = Dim_{d+1,1}(j \bmod 2^{d-1}, j-1)$. Also for β_0^{d+1}, $d = D_d(2^{d-1} - 1, 0) = Dim_{d+1,1}(2^d - 1, 0) = Dim_{d+1,1}((0-1) \bmod (2^d), 0)$. Thus in general, $\beta_j^{d+1} = ((j-1) \bmod (2^d), j)$. ■

At this point we have specified the array Dim. An element of Dim_d is either a dimension from $\{0, 1, \cdots, d-1\}$ or it is empty. If $Dim_d(i, j)$ is empty, then the corresponding elements $Src_d(i, j)$ and $Dst_d(i, j)$ of the source and destination arrays are also empty. An example of a 4-dimensional array Dim is shown in Figure 4.4. The corresponding array Dst is in Figure 4.5. Both halves of the arrays have been illustrated.

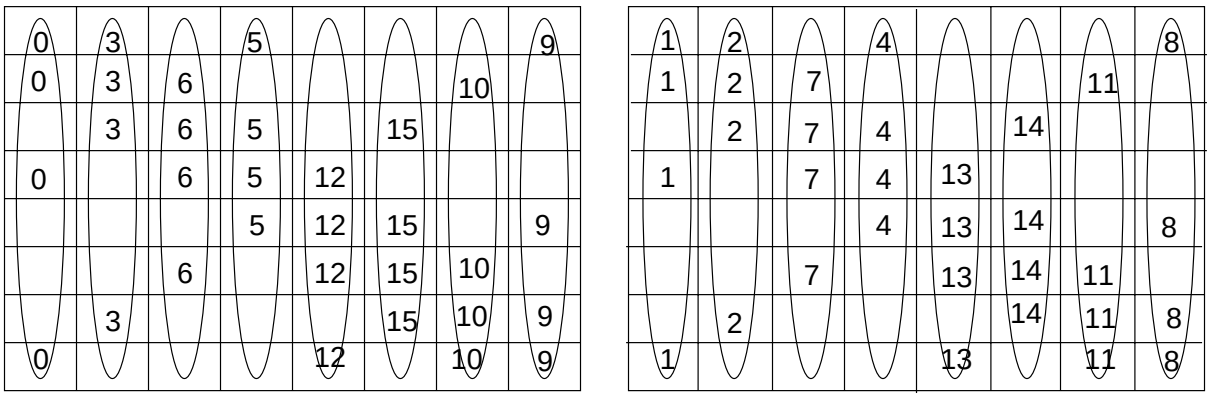


FIGURE 4.5: Sparse destination array for $d = 4$

4.2 Construction of the Destination Array, Dst

Since this section does not use a recursive definition, we will return to the usual notation of Src, Dst and Dim, rather than Src_d, Dst_d and Dim_d. For the sparse case, there are "empty" entries in each column of arrays Dst and Src that (as noted above) correspond to the empty entries of array Dim. For each column j of array Dst, all non-empty entries have the same value, namely $\hat{h}(j)$, the j^{th} element of the shuffled standard Hamiltonian path (Section 3.1). The array Dst for the sparse mapping is similar to that of the dense mapping. Recall that it is sufficient to specify array Dst as the N-element list of nodes $\hat{h} = \langle \hat{h}(j) : 0 \leq j < 2^{d-1} \rangle$. However, the size of this array for a sparse mapping is $2^{d-1} \times N$ (instead of the $d \times N$ size array in the case of dense mapping). An example of a sparse destination array is shown in Figure 4.5. As is evident from the figure, not all cells in the array are filled. Notice from Figure 4.4 and Figure 4.5 that empty cells coincide.

4.3 Construction of the Source Array, Src

With the destination and dimension arrays already defined, the construction of the source array is similar to that of the dense mapping. Again empty entries of arrays Dim and Dst also correspond to empty entries of array Src. More specifically, for any $0 \leq i < 2^{d-1}$ and $0 \leq j < 2^d$,

$$Src(i,j) = \begin{cases} \text{empty}, & \text{if } Dst(i,j) \text{ is empty} \\ Dst(i,j)|_{Dim(i,j)}, & \text{otherwise.} \end{cases}$$

An example of the array Src for a 4-dimensional hypercube is shown in Figure 4.6. Notice from the figure that each row of this array has two aggregates, one in each half. The larger number of rows for a sparse mapping allows room for just one processor to occupy an entire row of a half of the array Src. We formally prove this later in Section 4.4.

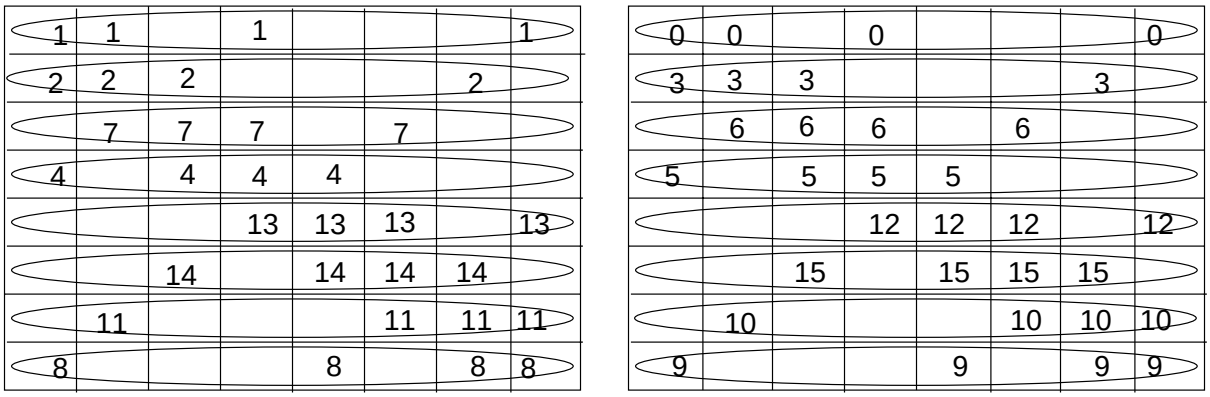


FIGURE 4.6: A 4-dimensional Sparse Src

Consider the following example with a 7-bit string $A = 0110101$. With $S_1 = \{0, 1, 5\}$, $A|_{S_1} = 0\underline{0}101\underline{10}$ (the complemented bits are underlined). If $S_2 = \{2, 4, 5\}$, then $[A|_{S_1}]|_{S_2} = 0\underline{10}0\underline{0}10$. Also observe that $A|_{S_2} = 0\underline{000}\underline{0}01$ and $[A|_{S_2}]|_{S_1} = 0\underline{1}000\underline{10} = [A|_{S_1}]|_{S_2}$. In fact, it is easy to see that if $S_1 \oplus S_2 = \{x : (x \in S_1 \text{ and } x \notin S_2) \text{or} (x \in S_2 \text{ and } x \notin S_1\}$, then

$[A|_{S_1}]|_{S_2} = [A|_{S-2}]|_{S_1} = A|_{S_1 \oplus S_2}$.

The following lemma states this observation without proof.

Lemma 4.5 *For any array A and any sets S_1, S_2 of bit indices, $[A|_{S_1}]|_{S_2} = [A|_{S_2}]|_{S_1} = A|_{S_1 \oplus S_2}$ where $S_1 \oplus S_2$ is the set of indices that are either in S_1 or in S_2, but not in both.* ■

Lemma 4.6 *For any $d \geq 2$, array Src_d can be expressed recursively as follows.*

$$Src_{d,1} = \begin{bmatrix} Src_{d-1,1} & S_{11} \\ S_{13} & Src_{d-1,1}|_{\{d-1,d-2\}} \end{bmatrix} \text{ and } Src_{d,2} = \begin{bmatrix} Src_{d-1,2} & S_{21} \\ S_{23} & Src_{d-1,2}|_{\{d-1,d-2\}} \end{bmatrix},$$

where S_{11}, S_{13}, S_{21} and S_{23} are $2^{d-2} \times 2^{d-2}$ reverse diagonal arrays.

Proof: From the fact that

$$Dim_{d,1} = Dim_{d,2} = \begin{bmatrix} Dim_{d-1,1} & (d-1)U_{d-2} \\ (d-1)U_{d-2} & Dim_{d-1,1} \end{bmatrix}$$

(4.1)

(see Definition 4.3) and because $Src_d(i,j) = Dst_d(i,j)|_{Dim_d(i,j)}$, for any $0 \leq i < 2^{d-1}$ and $0 \leq j < 2^d$ (see Section 4.3), the arrays S_{11}, S_{13}, S_{21} and S_{23} are established to be reverse diagonal.

Let

$$Src_{d,1} = \begin{bmatrix} A & S_{11} \\ S_{13} & B \end{bmatrix}.$$

(4.2)

From Equation 4.1 and Lemma 3.1 we have for all $0 \leq i, j < 2^{d-2}$,

$$A(i,j) = h_{d-1}(2j)|_{Dim_{d-1}(i,j)}. \tag{4.3}$$

To prove that $A(i,j) = Src_{d-1,1}(i,j)$ we need to show that

$$A(i,j) = h_{d-1}(2j)|_{Dim_{d-1}(i,j)} = \hat{h}_{d-1}(j)|_{Dim_{d-1}(i,j)} = Src_{d-1,1}(i,j). \tag{4.4}$$

By Definition 3.1, for $0 \leq j < 2^{d-2}$, $\hat{h}_{d-1}(j) = h_{d-1}(2j)$. This establishes Equation 4.4 and hence the second quadrant of $Src_{d,1}$ is indeed $Src_{d-1,1}$.

From Equation 4.2 we have, for all $0 \leq i, j < 2^{d-2}$,

$$B(i,j) = [2^{d-1} + h_{d-1}(2^{d-1} - 1 - 2j)]|_{Dim_{d-1,1}(i,j)}. \tag{4.5}$$

We need to show that

$$B(i,j) = Src_{d-1,1}(i,j)|_{\{d-1,d-2\}} = [\hat{h}(j)|_{Dim_{d-1,1}(i,j)}]|_{\{d-1,d-2\}} = [\hat{h}(j)|_{\{d-1,d-2\}}]|_{Dim_{d-1,1}(i,j)}. \tag{4.6}$$

From Equations 4.5 and 4.6 we only need to show

$$2^{d-1} + h_{d-1}(2^{d-1} - 1 - 2j) = \hat{h}_{d-1}(j)|_{\{d-1,d-2\}}. \tag{4.7}$$

Note that $0 \leq j < 2^{d-2}$, so j is a $d-2$ bit number. Let $j = (\underbrace{j_{d-3}j_{d-4}\cdots j_1 j_0}_{d-2 \text{ bits}})_2$. Then

$$2j = (\underbrace{j_{d-3}j_{d-4}\cdots j_1 j_0 0}_{d-1 \text{ bits}}). \quad (4.8)$$

Since $2^{d-1} - 1 = \underbrace{11\cdots 11}_{d-1 \text{ bits}}$ we have

$$(2^{d-1} - 1) - 2j = (\underbrace{j'_{d-3}j'_{d-4}\cdots j'_1 j'_0 1}_{d-1 \text{ bits}}). \quad (4.9)$$

Let $h_{d-1}(2^{d-1} - 1 - 2j) = x = (x_{d-2}x_{d-3}\cdots x_1 x_0)_2$. Then by Lemma 2.4 and Equation 4.9, we have

$$\begin{array}{lcll} x_0 & = & 1 \oplus j'_0 & = & j_0 \\ x_k & = & j'_k \oplus j'_{k-1} & = & j_k \oplus j_{k-1}, \text{for } 0 < k < d-2 \\ x_{d-2} & = & j'_{d-3} \oplus 0 & = & j'_{d-3} \end{array} \quad (4.10)$$

Thus,

$$2^{d-1} + h_{d-1}(2^{d-1} - 1 - 2j) = 2^{d-1} + x = (\underbrace{1x_{d-2}\cdots x_1 x_0}_{d-1 \text{ bits}})_2, \quad (4.11)$$

where the bits of x are given by Equation 4.10.

Returning to the right hand side of Equation 4.7,

let $\hat{h}_{d-1}(j) = h_{d-1}(2j) = y = (\underbrace{y_{d-1}y_{d-2}\cdots y_1 y_0}_{d-1 \text{ bits}})_2$. Then from Equation 4.8 and Lemma 2.4 we have the following

$$\begin{array}{lcll} y_0 & = & 0 \oplus j_0 & = & j_0 \\ y_k & = & j_k \oplus j_{k-1}, & & \text{for } 0 < k < d-2 \\ y_{d-2} & = & j_{d-3} \oplus 0 & = & j_{d-3} \\ y_{d-1} & = & 0 \oplus 0 & = & 0 \end{array} \quad (4.12)$$

From Equations 4.10, 4.11 and 4.12, we see that $2^{d-1} + h_{d-1}(2^{d-1} - 1 - 2j)$ and $\hat{h}_{d-1}(j)$ differ only in bits $d-1$ and $d-2$. Thus Equation 4.7, and hence Equation 4.6, holds. This establishes the recursive structure for $Src_{d,1}$.

The proof for $Src_{d,2}$ is similar and is omitted for brevity. ■

4.4 Number of Aggregates

In the case of a sparse mapping, both the destination array and the source array have N aggregates each. The fact that the destination array has N aggregates follows from the fact that the mapping is a standard mapping. For the source array, each half of the $2^{d-1} \times N$ array has 2^{d-1} aggregates. The Src therefore has N aggregates. We prove this now.

Lemma 4.7 *For any $d \geq 1$ and any $0 \leq i, j, k < 2^{d-1}$, if $Src_d(i,j)$ and $Src_d(i,k)$ are non-empty, then $Src_d(i,j) = Src_d(i,k)$.*

Proof: We proceed by induction on d. The lemma holds for the case $d = 1$ as there is only one entry in $Src_{1,1}$ and $Src_{1,2}$.

Assume the lemma to hold for any $d \geq 1$ and consider Src_{d+1}. By Lemma 4.6

$$Src_{d+1,1} = \begin{bmatrix} Src_{d,1} & S_{11} \\ Src_{13} & Src_{d,1}|_{d,d-1} \end{bmatrix}.$$

Clearly the induction hypothesis applies to the $Src_{d,1}$ in the second quadrant of $Src_{d+1,1}$. The induction hypothesis also applies to the $Src_{d,1}|_{d,d-1}$ of the fourth quadrant as all elements of $Src_{d,1}$ are altered in an identical manner. The strategy we use is as follows. Pick any row i. If this row traverses $Src_{d,1}$ (or $Src_{d,1}|_{d,d-1}$), then consider a particular element ρ of

Src_d (or $Src_{d,1}|_{d,d-1}$). All other elements in row i of $Src_{d,1}$ (or $Src_{d,1}|_{d,d-1}$) are identical to ρ, by the induction hypothesis. Now we note that S_{11} and S_{13} are reverse diagonal arrays (Lemma 4.6) and contain only one non-empty element in row i. All we need to establish is that this non-empty element equals ρ.

Depending on which quadrants row i traverses, we consider two cases.

Case $0 \leq i < 2^{d-1}$: Here row i traverses the $Src_{d,1}$ in the second quadrant of $Src_{d+1,1}$.

Consider

$$\rho = Src_{d,1}(i,i) = \hat{h}_d(i)|_{D_d(i,i)} = h_d(2i)|_{\alpha_i} = h_d(2i)|_0. \tag{4.13}$$

The only non-empty element in row i of S_{11} is in column $2^{d-1} - 1 - i$ of S_{11} or in column $2^{d-1} + 2^{d-1} - 1 - i = 2^d - 1 - i$ of $Src_{d+1,1}$. This element has the value

$$\hat{h}_{d+1}(2^d - 1 - i)|_{Dim_{d+1}(i,2^d-1-i)} = h_{d+1}(2(2^d - 1 - i))|_d. \tag{4.14}$$

The last part of this equation follows from the fact that $Dim_{d+1}(i, 2^d - 1 - i) = dU_{d-1}(i, 2^d - 1 - i) = d$. With $0 \leq i < 2^{d-1}$, let

$$i = (\underbrace{i_{d-2} i_{d-3} \cdots i_1 i_0}_{d-1 \text{ bits}})_2. \tag{4.15}$$

Then $2^d - 1 - i = (\underbrace{11 \cdots 11}_{d \text{ bits}})_2 - (\underbrace{0 i_{d-2} i_{d-3} \cdots i_1 i_0}_{d-1 \text{ bits}})_2 = \underbrace{1 i'_{d-2} i'_{d-3} \cdots i'_1 i'_0}_{d \text{ bits}}$.

So, $2(2^d - 1 - i) = \underbrace{1 i'_{d-2} i'_{d-3} \cdots i'_1 i'_0 0}_{d+1 \text{ bits}}$.
Let $h_{d+1}(2(2^d - 1 - i)) = x = x_d x_{d-1} \cdots x_1 x_0$, where

$$\begin{array}{lcll} x_0 & = & i'_0 \oplus 0 & = & i_0 \\ x_d & = & 1 \oplus 0 & = & 1 \\ x_{d-1} & = & 1 \oplus i'_{d-2} & = & i_{d-2} \text{ and for } 0 < l < d-1 \\ x_\ell & = & i'_\ell \oplus i'_{\ell-1} & = & i_\ell \oplus i_{\ell-1}, for 0 \leq \ell < d-1. \end{array} \tag{4.16}$$

Then the only non-empty element in row i of S_{11} has a value of

$$h_{d+1}(2(2^d - 1 - i))|_d = 0 x_{d-1} x_{d-2} \cdots x_0, \tag{4.17}$$

where x_ℓ is given by Equation 4.16.

As noted earlier (Equation 4.13), all elements in row i of the $Src_{d,1}$, in the the second quadrant of $Src_{d+1,1}$ have value $\rho = h_d(2i)|_0$.

Let $\rho = \rho_d \rho_{d-1} \cdots \rho_1 \rho_0$. Then $h_d(2i) = \rho|_i = \rho_d \rho_{d-1} \cdots \rho_1 \rho'_0$. From Equation 4.15 $2i = i_{d-2} i_{d-3} \cdots i_1 i_0 0$, and we have

$$\begin{array}{lcll} \rho'_0 & = & 0 \oplus i_0 & = & i_0 \text{ or } \rho_0 = i'_0 \\ \rho_\ell & = & i_\ell \oplus i_{\ell-1} & \text{for } 0 < \ell < d-1 & \\ \rho_{d-1} & = & i_{d-2} \oplus 0 & = & i_{d-2} \\ \rho_d & = & 0 \oplus 0 & = & 0. \end{array} \tag{4.18}$$

From Equation 4.14 and 5.1 we find that the only non-empty element of S_{11} has value ρ, as required.

Case $2^{d-1} \leq i < 2^d$: The only difference between this and the previous case is that bit $d-1$ of i is a 1 (rather than 0 in the previous case). The argument remains the same, however. We, therefore, skip the details.

Overall the proof for $Src_{d+1,2}$ follows along the same lines as that for $Src_{d+1,1}$. ■

Therefore we have the following main result of this section.

Theorem 4.8 *For any $d \geq 1$, a d-dimensional hypercube can be mapped on a slab waveguide using 2^d lasers and 2^d detectors.* ■

Remark: We prove this mapping optimal in Chapter 5.

4.5 Relation to the Extended Hypercube

The sparse construction so far has several empty elements in the arrays Src, Dst and Dim. That is, not all channels of the channel array are used. If these empty elements of the source and destination were filled with values already present in the aggregates (so that the aggregates are not broken), then we get a supergraph of the hypercube. Figure 4.7 shows this for H_3. In this section we formally identify this supergraph of the hypercube H_d and call it the extended hypercube Q_d. The extended hypercube represents the largest graph that the sparse mapping will support.

Definition 4.9 For any $d \geq 1$, let $Q_d = (V, \mathcal{E})$ be the *extended hypercube*, where $V = \{0, 1, \cdots, 2^{d-1}\}$. An edge $(p, q) \in \mathcal{E}$ iff p and q differ in an odd number of bits. □

Clearly, a regular hypercube has edges corresponding to 1 different bit (rather than an odd number of differing bits). An example of a 5-dimensional extended hypercube is illustrated in Figure 4.8. In this figure only extra edges out of node 0 have been shown.

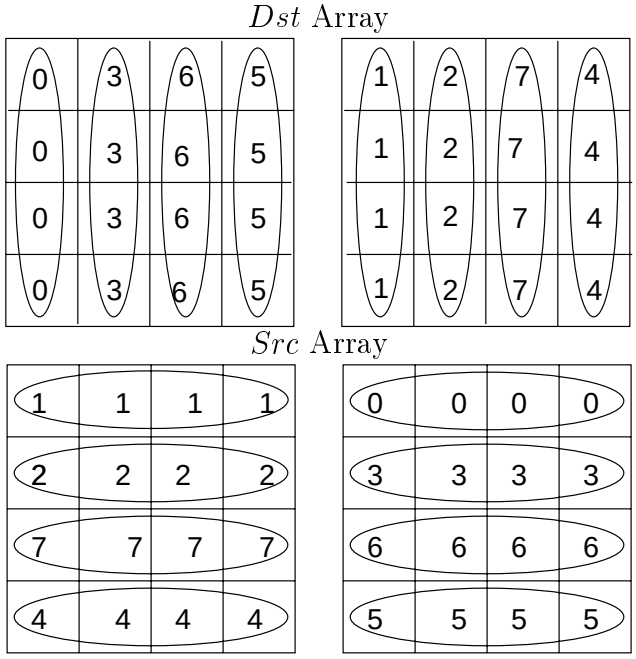


FIGURE 4.7: Mapping for an extended hypercube Q_3

Lemma 4.10 *For any $m \geq 0$, standard Hamiltonian paths $h_d(j)$ and $h_d((j+m)(\bmod (2^d)))$ differ in an even number of bits, if m is even, and an odd number of bits, if m is odd.*

Proof outline: Since h_d represents a Hamiltonian path, for $0 \leq j < 2^d$, the pair $\langle h_d(j), h_d((j+1)(\mathrm{mod}(2^d)))\rangle$ represents an edge of the hypercube H_d. So $h_d(j)$ and $h_d((j+1)(\mathrm{mod}(2^d)))$ differ in exactly 1 bit. Therefore, $h_d(j)$ and $h_d((j+2)(\mathrm{mod}(2^d)))$ differ in 2 bits. Let $a_0, a_1, a_2, \cdots, a_k$ represent the string of nodes that form a Hamiltonian cycle for any hypercube. Observe that a_0 and a_1 have to differ in 1 bit, a_0 and a_2 differ in 0 or 2 bits, a_0 and a_3 differ in either 1 or 3 bits and so on. ■

Lemma 4.11 *For $d \geq 1$ let $\hat{h}_d = \hat{h}_d^a \circ \hat{h}_d^b$. For $x \in \{a, b\}$ the binary representation of any pair of distinct elements in $\hat{h}_d^x$ differ in an even number of bits.*

Proof: The elements of $\hat{h}_d^a$ are from the set $\{h_d(2i) : 0 \leq i < 2^{d-1}\} = S_a$ (say) and the elements of $\hat{h}_d^b$ are from $S_b = \{h_d(2i+1) : 0 \leq i < 2^{d-1}\}$. Thus all elements of S_a (or S_b)

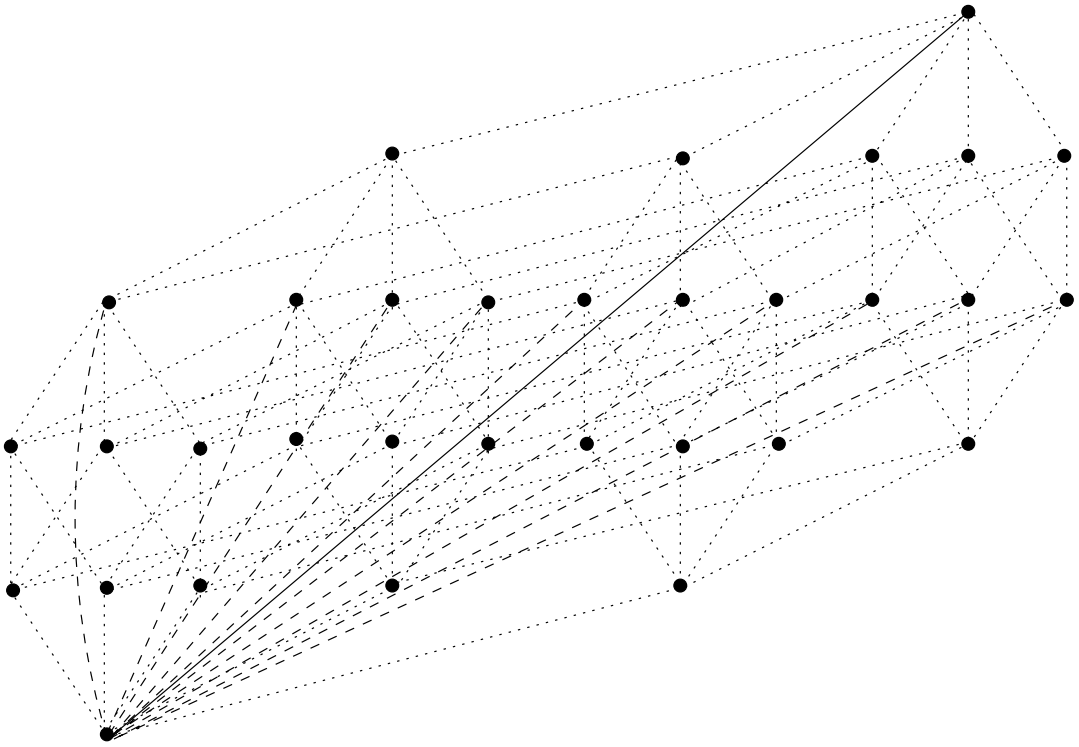


FIGURE 4.8: A 5-dimensional extended hypercube where the regular edges are shown dotted and the extended hypercube edges are shown dashed for one node

have indicies in h_d that differ by an even number of bits. That is, if $h_d(j), h_d(k) \in S_a$ (or S_b) then $(j-k)(\mathrm{mod}(2)) = 0$. By Lemma 4.10, $h_d(j)$ and $h_d(k)$ differ in an even number of bits. ■

In the following we denote the source, destination and dimension array of the extended hypercube by $\mathsf{Src_d}, \mathsf{Dst_d}$ and $\mathsf{Dim_d}$ to indicate that they have no empty elements.

Lemma 4.12 *For any $d \geq 1$; any $0 \leq i < 2^{d-1}$ and any $0 \leq j < 2^d$ the binary representation of $\mathsf{Src_d(i,j)}$ and $\mathsf{Dst_d(i,j)}$ differ in an odd number of bits.*

Proof: We consider one half at a time. Let $0 \leq i, j < 2^{d-1}$. We know that $Dim_d(i,j) = \alpha_i = 0$, so $Src_d(i,i)$ is non-empty and $Src_d(i,i) = h(2i)|_0 = \hat{h}(i)|_0$.

Clearly $Src_d(i,i) = \mathsf{Src_d(i,j)}$ (for all $0 \leq j < 2^{d-1}$) differs from $\mathsf{Dst_d(i,i)} = \hat{\mathsf{h}}_\mathsf{d}(\mathsf{i})$ in an odd number of bits. By Lemma 4.10, $\mathsf{Src_d(i,j)}$ differs from every $\hat{h}_d(j) = \mathsf{Dst_d(i,j)}$ (where $0 \leq j < 2^{d-1}$) in an odd number of bits. ■

The following are well known results.

Lemma 4.13 *For any $d, x \geq 1$, $\binom{d+1}{k} = \binom{d}{k} + \binom{d}{k-1}$.* ■

Lemma 4.14 *[Binomial Theorem] For any integer $n \geq 0$ and reals a, b,*

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^i b^{n-i}$$ ■

With $a = b = 1$ in Lemma 4.14, one obtains the following.

Corollary 4.15 *For any $n \geq 0$, $\sum_{i=0}^{n} \binom{n}{i} = 2^n$.* ■

Lemma 4.16 *Each node of the extended hypercube Q_d (for any $d \geq 1$) has $\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{d}{2i+1} = 2^{d-1}$ neighbors.*

Proof: We proceed by induction on $d \geq 1$. For $d = 1$ we have,
$\sum_{i=0}^{0} \binom{d}{2i+1} = \binom{1}{1} = 1 = 2^{1-1}$.

Assume the lemma to hold for any $d \geq 1$ and consider two cases for $\sum_{i=0}^{\lfloor \frac{d}{2} \rfloor} \binom{d+1}{2i+1}$.

Case d is odd: From Lemma 4.14 we have

$$\sum_{i=0}^{\lfloor \frac{d}{2} \rfloor} \binom{d+1}{2i+1} = \sum_{i=0}^{\frac{d-1}{2}} \binom{d}{2i+1} + \binom{d}{2i}. \tag{4.19}$$

Notice that $2i + 1 \leq 2.\frac{d-1}{2} + 1 = d$. So $\binom{d}{2i}$ is a valid binomial coefficient. Also since the set $\{2i, 2i+1 : 0 \leq i \leq \frac{d-2}{2}\} = \{0, 1, \cdots, d\}$, we have from Equation 4.19 and from Lemma 4.14 $\sum_{i=0}^{\lfloor \frac{d}{2} \rfloor} \binom{d+1}{2i+1} = \sum_{i=0}^{d} \binom{d}{i} = 2^d$.

Case d is even: Again from Lemma 4.14 we have

$$\begin{aligned}
\sum_{i=0}^{\lfloor \frac{d}{2} \rfloor} \binom{d+1}{2i+1} &= \sum_{i=0}^{\frac{d}{2}} \binom{d+1}{2i+1} &&= \binom{d+1}{d+1} + \sum_{i=0}^{\frac{d}{2}-1} \binom{d+1}{2i+1} \\
&= \binom{d}{d} + \sum_{i=0}^{\frac{d}{2}-1} \binom{d}{2i+1} + \binom{d}{2i} &&= \sum_{i=0}^{d} \binom{d}{i} \\
&= 2^d.
\end{aligned}$$

The last but one step follows from the fact that $\binom{d+1}{d+1} = \binom{d}{d} = 1$ and $\{2i, 2i+1 : 0 \leq i < \frac{d}{2}\} = \{0, 1, \cdots, d-1\}$ and the last step is from Corollary 4.15.

In any case, $\sum_{i=0}^{\lfloor \frac{d}{2} \rfloor} \binom{d+1}{2i+1} = 2^d$ as required. ■

Theorem 4.17 *For any $d \geq 1$, $\mathsf{Src_d}$ and $\mathsf{Dst_d}$ as defined in Lemma 4.9 map the extended hypercube Q_d on a slab waveguide using 2^d lasers and 2^d detectors.*

Proof: Lemma 4.16 shows that every edge mapped between $\mathsf{Src_d(i,j)}$ and $\mathsf{Dst_d(i,j)}$ is an edge of Q_d. Also for any $0 \leq i, i' < 2^{d-1}$ and $0 \leq j, j' < 2^d$ such that $i \neq i'$ or $j \neq j'$, the doublets $\langle \mathsf{Src_d(i,j)}, \mathsf{Dst_d(i,j)} \rangle$ and $\langle \mathsf{Src_d(i',j')}, \mathsf{Dst_d(i',j')} \rangle$ are different. If $\mathsf{Src_d(i,j)} = \mathsf{Src_d(i',j')}$, then $i = i'$. In addition, if $\mathsf{Dst_d(i,j)} = \mathsf{Dst_d(i',j')}$, then $j = j'$. So the two doublets can be identical iff $i = i'$ and $j = j'$. Therefore each entry of the channel array represents a different edge of Q_d. Finally by Lemma 4.16, Q_d has $2^d \times 2^{d-1}$ edges, which is also the number of elements in the channel array.

Therefore, every edge of Q_d is mapped. Clearly the mapping has 2^d source aggregates and 2^d destination aggregates. ■

We prove in the next chapter that the sparse mapping has an optimal number of aggregates.

Chapter 5
Lower Bounds

We have so far derived upper bounds on the number of aggregates (lasers and detectors) in the dense and sparse mapping cases. How good are these mappings? In this chapter we derive matching lower bounds for these mappings and show that they are optimal (in terms of the number of aggregates).

5.1 A General Lower Bound

In this section we derive a general (trivial) lower bound that applies for any weak topology with an underlying strongly connected graph.

Lemma 5.1 *Every strongly connected weak topology requires at least* 1 *laser and* 1 *detector per node.*

Proof: Because the graph is strongly connected each node has an indegree and an outdegree ≥ 1. This implies that it needs at least one laser and one detector. ■

Recall that the sparse mapping uses N lasers and N detectors (one each, per node) to map an extended hypercube on to an optical slab waveguide. Since an (extended) hypercube is strongly connected, Lemma 5.1 establishes our mapping to be optimal.

Theorem 5.2 *The sparse mappings of the hypercube and extended hypercube (Theorems 4.8 and 4.17) have an optimal number of source and destination aggregates.* ■

5.2 Dense Mapping Lower Bound

Since the dense mapping is a standard mapping, the array Dst has N aggregates, and hence N detectors. This is required (and optimal) for any standard mapping. Here we derive a lower bound for the number of source aggregates in any standard dense mapping.

A simple directed graph is one in which for any pair of nodes u, v, there is at most one directed edge $\langle u, v \rangle$. That is, a simple graph does not allow multiple edges from u to v.

Lemma 5.3 *Consider any weak topology based on a simple directed graph that is mapped to a slab using a standard mapping in which the array Dst has one column aggregate per column. If no edge of the graph is mapped to more than one channel, the array, Src, cannot have any non-trivial column aggregates.*

Proof: Consider any column j of Dst. Let this column have index u in all entries. If the corresponding column in Src has a non-trivial aggregate in rows i and i' (that is, $Src(i, j) = Src(i', j) = v$), then edge $\langle u, v \rangle$ is mapped to channels (i, j) and (i', j). (Figure 5.1 illustrates this.) ■

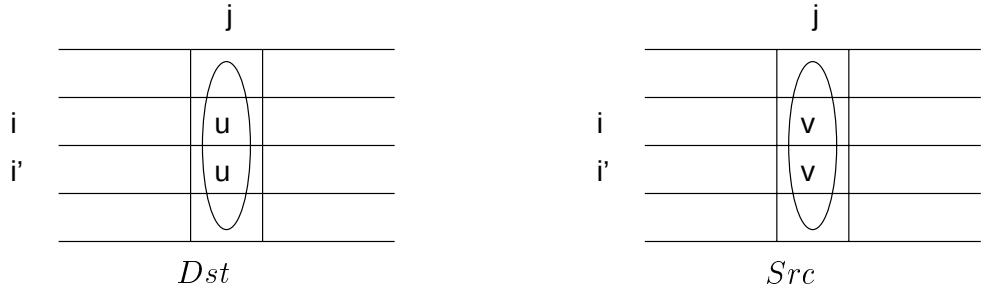


FIGURE 5.1: An illustration of the proof of Lemma 5.3

Since the dense mapping of Chapter 3 satisfies all the conditions of Lemma 5.3 we have the following result.

Corollary 5.4 *The standard dense mapping of a hypercube has no column aggregates in Src.* ■

Lemma 5.3 shows that no column of Src can have a non-trivial aggregate. We now limit the number of non-trivial aggregates in Src.

Lemma 5.5 *In a standard mapping of a strongly connected weak topology, for any two columns j, j' of Dst, the indices corresponding to column aggregates are distinct.*

Proof: Let column j and j' have aggregates with indicies u and v respectively. Since Dst has N columns, where N is the number of nodes and each node has an incoming edge (for a strongly connected topology), each column of Dst must be associated with a distinct node (aggregate index). ■

Before we proceed, recall that the notation $a|_b$ is used to denote a binary number a with bit b complimented.

Lemma 5.6 *In a standard dense mapping, two adjacent columns of array, Src, cannot have more than two non-trivial aggregates.*

Proof: Suppose columns j and $j+1$ have three non-trivial row aggregates in rows i_1, i_2 and i_3 (see Figure 5.2). Note that these aggregates may extend in either direction beyond columns j and $j+1$. Also note that since a column in Src cannot have a column aggregate the value a, b and c in Figure 5.2 are distinct. Let u and v be the indicies in column j and $j+1$ respectively of Dst (as shown in Figure 5.2(b)). Thus $\langle a, u\rangle, \langle b, u\rangle, \langle c, u\rangle, \langle a, v\rangle, \langle b, v\rangle$ and $\langle c, v\rangle$ are hypercube edges. Since each hypercube edge is between processor pairs whose binary representation differ in exactly 1 position, let, $u = a|_k = b|_\ell = c|_m$ and $v = a|_{k'} = b|_{\ell'} = c|_{m'}$. Since $u \neq v$, we have $k \neq k', \ell \neq \ell'$ and $m \neq m'$. Consider a and b first. Since $a \neq b$, we

(a) Src (b) Dst

FIGURE 5.2: An illustration of the proof of Lemma 5.6

have $k \neq \ell$ and $k' \neq \ell'$, and

$$u = a|_k = v|_{k'|k} = v|_{k,k'} = b|_{\ell'|_{k,k'}} = b|_{\{\ell'\}\oplus\{k,k'\}} = u|_{\ell|_{\{\ell'\}\oplus\{k,k'\}}} = u|_{\{\ell,\ell'\}\oplus\{k,k'\}}.$$

In summary, $u = u|_{\{\ell,\ell'\}\oplus\{k,k'\}}$. That is $\{\ell,\ell'\} \oplus \{k,k'\} = \emptyset$. Since $\ell \neq k$ and $\ell' \neq k'$, $\ell = k'$ and $\ell' = k$. Without loss of generality let $\ell = k' = 0$ and $k = \ell' = 1$ and a, b, u and v have the form shown in Figure 5.3. Now we bring in c. Since $u|_m = v|_{m'} = c$ the only way is

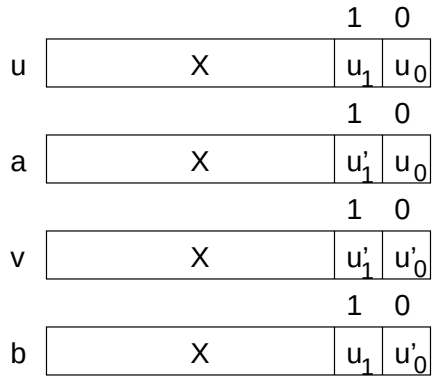


FIGURE 5.3: Illustrating the structure for numbers u, a, v, b

to have $\{m, m'\} = \{k, k'\}$ (see Figure 5.3) which would make $c = a$ or $c = b$, providing the necessary contradiction. ■

Consider any standard dense mapping of a d-dimensional hypercube resulting in an array

Src_d. Let $Src[0, j]$ denote the part of the array including the first $j+1$ columns $0, 1, \cdots, j$ and in general let $Src[j, j']$ denote the portion of Src from column j to column j'.

Definition 5.7 Let A be an aggregate of Src_d. For any $0 \leq j < 2^d$, aggregate A is *incomplete* in $Src_d[0, j]$ iff some portion of A lies outside $Src_d[0, j]$. Otherwise it is called a *complete* aggregate *of* $Src_d[0, j]$. □

Figure 5.4 shows examples of a complete and an incomplete aggregate.

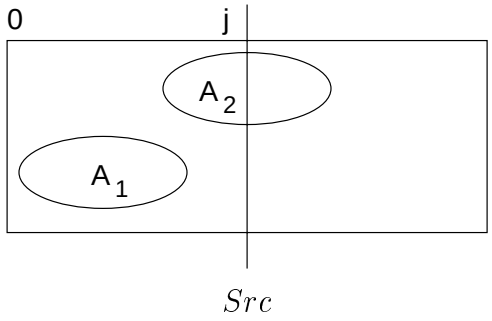


FIGURE 5.4: Examples of a complete aggregate A_1 and an incomplete aggregate A_2 of $Src_d[0, j]$.

Lemma 5.8 *For any* $d \geq 1$ *and* $0 \leq j < 2^d$, $Src_d[0, j]$ *has at most two incomplete aggregates.*

Proof: Every incomplete aggregate of $Src_d[0, j]$ must traverse column j. By Definition 5.7 if $j = 2^{d-1}$, then there are no incomplete aggregates. Let $j < 2^d - 1$ and consider columns j and $j+1$. All incomplete aggregates of $Src_d[0, j]$ traverse columns j and $j+1$. By Lemma 5.6 there can be at most two such aggregates. ■

We now develop an important result which lays the foundation for the main result of this section.

Lemma 5.9 *For any* $0 \leq j < 2^d$, $Src_d[0, j]$ *has at least* $(j+1)(d-2)+2$ *aggregates.*

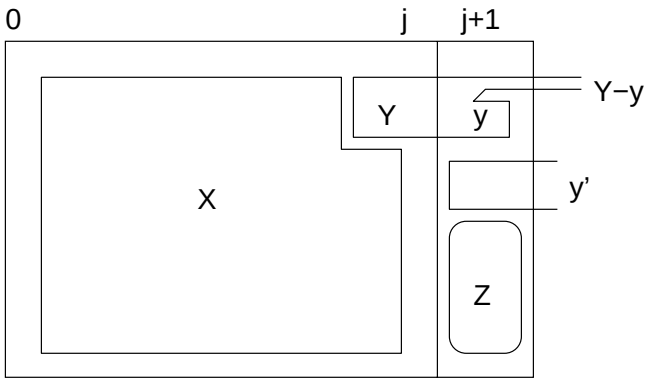


FIGURE 5.5: An illustration of the proof of Lemma 5.9

Proof: We proceed by induction on $j \geq 0$. Note first that all aggregates are row aggregates (Lemma 5.3). Then for the first column ($j = 0$) each element is part of a different aggregate, so there are $d = (0+1)(d-2)+2$ aggregates. Assume the lemma to hold for any $j \geq 0$ and consider $Src_d[0, j+1]$ (see Figure 5.5). Let $Src_d[0, j]$ have X complete aggregates and Y incomplete aggregates. Of these Y incomplete aggregates, let y become complete in column $j+1$. Let column $j+1$ start y' new incomplete aggregates. Let Z be the number of trivial aggregates in column $j+1$.

The number of incomplete aggregates in $Src_d[0, j+1]$ is $Y - y + y'$ by Lemma 5.8. The number of aggregates in $Src_d[0, j]$ is

$$X + Y \geq (j+1)(d-2) + 2 \tag{5.1}$$

by the induction hypothesis. The number of trivial aggregates on column $j+1$ is

$$Z + d - (Y + y') \tag{5.2}$$

(see Figure 5.5). The total number of aggregates in $Src_d[0, j+1]$ is

$$\begin{aligned}
(\underbrace{X + Z + Y}_{complete}) + (\underbrace{Y - y + y'}_{incomplete}) &= X + d && \text{from Equation 5.2} \\
&\geq (j+1)(d-2) + 2 - Y + d && \text{from Equation 5.1} \\
&\geq (j+1)(d-2) + 2 - 2 + d && \text{as } Y \leq 2 \\
&= (j+2)(d-2) + 2.
\end{aligned}$$

This completes the proof. ■

If Lemma 5.9 is used over the entire array Src_d, we have a lower bound of $(d-2)2^d + 2$ aggregates. Our method (Theorem 4.17) uses $(d-2)2^d + 4$ aggregates. In the remainder of this section we further tighten the lower bound of Lemma 5.9 to establish that the method of Theorem 3.10 is optimal.

Lemma 5.10 *For any standard, dense mapping of a d-dimensional hypercube, Src_d has at least a pair of adjacent columns j and $j+1$ such that no single aggregate traverses both j and $j+1$.*

Proof: Suppose the lemma was not true. Let Src_d have k row aggregates (see Figure 5.6) $A_1, A_2, \cdots, A_k$ such that each A_i, $1 \leq i < k$, starts at column j'_i. These aggregates satisfy $j_i \leq j'_{i-1}$ (for $1 < i \leq k$); that is, A_i starts at a column number higher than one in which A_{i-1} ends. This assures that A_i and A_{i-1} overlap in at least one column. Let A_i have contain index a_i. Let column j_i of Dst_d have index b_i (see Figure 5.6). Then we have the following path in the hypercube.

$$b_1 \rightarrow a_1 \rightarrow b_2 \rightarrow a_2 \rightarrow \cdots \rightarrow b_k \rightarrow a_k.$$

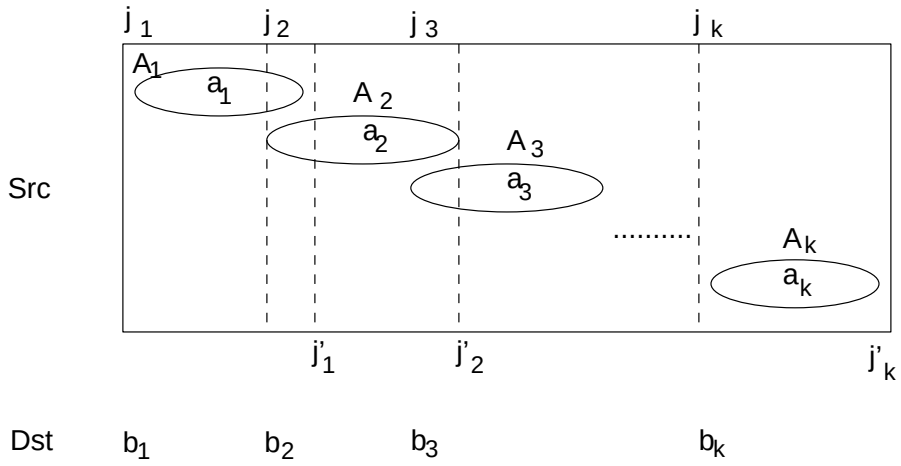


FIGURE 5.6: An illustration for the proof of Lemma 5.10

Let $A = \{a_i : 1 \leq i \leq k\}$ and $B = \{b_i : 1 \leq i \leq k\}$. Clearly A and B are non-empty. We claim that $A \cap B = \emptyset$.

Let $a_i = b_{i'} = a \in A \cap B$. Without loss of generality, let $i < i'$ then we have the following cycle

$$a = a_i \rightarrow b_{i+1} \rightarrow a_{i+1} \rightarrow \cdots \rightarrow b_{i'-1} \rightarrow a_{i'-1} \rightarrow b_{i'} = a.$$

Clearly this is a cycle of odd length. That is not possible on a hypercube (needs even number of bit complements to get back to same node). Since A and B are disjoint and non-empty, not all elements of the hypercube can be included in B. Since each column of Dst_d has a distinct index (Lemma 5.5), B cannot cover the entire array Dst_d. That is, A cannot cover the entire array Src_d. ■

We now derive the main result of this section.

Theorem 5.11 *Every standard dense mapping of a d-dimensional hypercube on a slab requires* 2^d *destination aggregates and at least* $(d-2)2^d + 4$ *source aggregates.*

Proof: The bound on the destination aggregates follows from the fact that it is a standard

mapping.

By Lemma 5.10, there exists columns j and $j+1$ of Src_d such that no aggregates overlap over these columns. So Lemma 5.9 applies separately to both $Src_d[0, j]$ and $Src_d[(j+1), 2^d - 1]$. This gives a total of $[(j+1)(d-2)+2] + [(2^d - (j+1)(d-2)+2] = (d-2)2^d + 4$ aggregates. ■

Our method in Chapter 3 (Theorem 3.10) is therefore optimal in the number of aggregates.

Chapter 6
Conclusion

It is widely acknowledged that optical interconnects are the best candidates to meet the communication needs of future computing systems. This thesis deals with optical interconnects among spatially proximate components. While a lot of research has addressed the technological improvements for such interconnects, very little research has gone into approaching the design of optical interconnects in a manner that exploits knowledge of the computation, as is done in this thesis.

We have developed a few methods to map a d-dimensional weak hypercube onto an optical slab waveguide. A weak hypercube allows each node to communicate with at most one neighbor at a time. Our approach uses this under-utilization of the topology's edges to pack as many channels as possible into a single laser/detector, while reducing the number of optical components used and consequently their cost.

We have also derived non-trivial lower bounds for any standard mapping of a d -dimensional weak hypercube and shown that the mappings we have proposed match these lower bounds and are, hence, optimal.

We have proposed results for two cases: (1) when the size of the channel array is $d \times 2^d$ (dense mapping) and (2) when the size of the channel array is $2^{d-1} \times 2^d$ (sparse mapping). For each of these cases we have defined the construction of the channel array in terms of three 2-dimensional arrays, namely, the source, destination and dimension arrays. We have also derived matching lower bounds for the number of aggregates for each of the mappings. The dense mapping uses $(d-2)2^{d-1}+4$ lasers and 2^d detectors, while the sparse mapping

uses 2^d lasers and 2^d detectors.

Though the sparse mapping has a clean recursive structure definition we have also shown that it can be viewed as an extension of the dense mapping approach; i.e., the Hamiltonian path for Dst array and α_j, β_j and γ_j along the diagonal is the same for both. It is possible to generalize these two cases (sparse and dense) in a mapping on any $R \times 2^d$ channel array for $d \leq R < 2^{d-1}$.

Other directions include the mapping of more complex higher dimensional topologies and ways to pack more elements per aggregate. The cost of the mappings that we consider depends only on the number of aggregates per mapping. Analyzing the placement, shape and size of these aggregates may be possible.

Our results can also help in designing new optical interconnect models.

Bibliography

[1] "Light and Optics," ACEPT, Department of Physics and Astronomy, Arizona State University, 1999. *http : //acept.asu.edu/PiN/rdg/color.color.shtml*

[2] G. P. Agarwal, "Fiber-Optic Communication Systems," John Wiley, June 2002.

[3] S. Berdague and P. Facq, "Mode Division Multiplexing in Optical Fibers," *Applied Optics*, vol. 21, June 1, 1982, pp. 1950-1955.

[4] M. Chateauneuf, A. G. Kirk, D. V. Plant, T. Yamamoto, and J. D. Ahearn, "512-Channel Vertical-Cavity Surface-Emitting Laser Based Free-Space Optical Link," *Applied Optics*, vol.41, (2002), pp. 5552-5561.

[5] G. Chen, "A Tutorial on Interconnection Networks," Nanjing University, 1999.

[6] I. Chlamtac and A. Ganz, and G. Karmi, "Lightnets: Topologies for High Speed Optical Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 11, May/June 1993.

[7] R. T. Chen, D. Robinson, Z. Sun, T. Jannson, and D. V. Plant, "60 GHz Board-to-Board Optical Interconnection Using Polymer Optical Buses in Conjunction with Microprism Couplers," *Applied Physics Letters*, vol. 60(5), February 1992, pp. 536-538.

[8] S. Y. Cho, M. A. Brooke, and N. M. Jokerst, "Optical Interconnections on Electrical Boards Using Embedded Active Optoelectronic Components," *IEEE Journal of Selected Topics in Quantum Electronic*, vol. 9,(2), March/April 2003, pp. 465-476.

[9] Introduction to DWDM, Cisco Systems, 2000,
http : //www.cisco.com/univercd/cc/td/doc/product/mels/dwdm/dwdm_fns.htm

[10] J. Duato, S. Yalmanchili, and L. Ni, "*Interconnection Networks: An Engineering Approach*," Morgan Kaufman, 2002.

[11] M. Feldman, A. El-Amawy, and R. Vaidyanathan, "High Speed, High Capacity Bused Interconnects Using Optical Slab Waveguides," Proc. Workshop on Optics and Computer Science (Springer-Verlag Lecture Notes in Computer Science, Vol. 1586, 1999, pp.924-937.

[12] M. Feldman, A. El-Amawy, and R. Vaidyanathan, "Optical Slab Waveguide for Massive, High-Speed Interconnects," U.S.Patent 6332050, December 2001.

[13] M. R. Feldman, S. C. Esener, C. C. Guest, and S. H. Lee, "Comparison Between Optical and Electrical Interconnects Based on Power and Speed Considerations," *Applied Optics*, vo. 27, no. 9, 1988, pp. 1742-1751.

[14] G. R. Fowles, "Introduction to Modern Optics," Dover Publications, 2nd edition, 1989.

[15] X. Han, G. Kim, J. Lipovski, and R. T. Chen, "An Optical Centralized Shared-Bus Architecture Demonstrator for Microprocessor-to-Memory Interconnects," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 9, no. 2, March/April 2003, pp. 512-517.

[16] L. J. Irakliotis and P. A. Mitkas, "Optics: A Maturing Technology for Better Computing," *IEEE Computer Magazine*, vol. 31, no. 2, February 1998, pp. 36-37.

[17] G. Kim, X. Han, and R. T. Chen, "A Method for Rebroadcasting Signals in an Optical Backplane Bus System," *Journal of Lightwave Technology*, vol. 22, no. 3, March 2004, pp. 840-844.

[18] G. Kim, X. Han, and R. T. Chen, "Crosstalk and Interconnection Distance Considerations for Board-to-Board Optical Interconnects using 2-D VCSEL and Microlens array," *IEEE Photonics Technology Letters*, vol. 12, no. 6, June 2000, pp. 743-745.

[19] T. Leighton, "*Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*," Morgan Kaufmann Publishers, San Mateo, California, 1992.

[20] R. Lytel, H. Davidson, N. Nettleson, and T. Sze, "Optical Interconnections within Modern High-Performance Computing Systems," Technical report, Sun Microsystems, May 2000.

[21] "Ultrahigh Speed Multi-Gigabit Wireless Laser Communication System with Fully Integrated High-Speed Microwave Radio Backup," MDA Technologies, 2002. *http* : $//www.mdatechnology.net/techresearch.asp?articleid = 536$

[22] E. Mohammed, A. Alduino, T. Thomas, and H. Braunisch, *et al.*,, "Optical Interconnect System Integration for Ultra Short Reach Applications," *Intel Technology Journal*, vol.8, issue 2, May 2004, pp. 115-128.

[23] A. V. Mule, S. Schultz, T. K. Gaylord, and J. D. Meindl, "Input Coupling and Guided Wave Distribution Schemes for Board-Level Intra Chip Optical Clock Distribution Network Using Volume Grating Coupler Technology," *Proc. IEEE International Interconnect Technology Conference*, 2001, pp. 128-130.

[24] B. E. Nelson, G. A. Keeler, D. Agarwal, N. C. Helman, and D. A. B. Miller, "Wavelength Division Multiplexed Optical Interconnect Using Short Pulses," *IEEE Journal of Selected Topics in Quantum Electronic*, vol. 9,(2), March/April 2003, pp. 486-491.

[25] E. G. Paek, "High Speed Characterization of Spatial Light Modulators and its Applications to Optical Information Processing," *Lasers and Electro-Optics Society*, vol. 1, 1999, pp. 323-324.

[26] C. G. Plaxton, "Load Balancing, Selection and Sorting on the Hypercube," *Proc. Symposium on Parallel Algorithms and Architectures*, 1989, pp. 64-73.

[27] M. Raksapatcharawong and T. M. Pinkston, "Modelling Free-Space Optical k-ary n-cube wormhole networks," *Journal of Parallel and Distributed Computing*, vol. 55(1), 1998, pp. 60-93.

[28] R. Ramaswami and K. Sivarajan, "Optical Networks: A Practical Perspective, 2nd edition," Morgan-Kauffman, October 2001.

[29] C. S. Ram Murthy and M. Guruswamy, "*WDM Optical Networks: Concepts, Designs and Algorithms*," Prentice Hall, 2002.

[30] T. Saito, T. Ota, T. Toratani, and Y. Ono, "16-ch Arrayed Waveguide Grating Module with 100-GHz spacing," *Furukawa Review*, Vol. 19, 2000.

[31] K. Sethuraman, "Mapping Weak Multidimensional Torus Communications on Optical Slab Waveguides," M.S.Thesis, Dept. of Electrical Computer Engineering, Louisiana State University, 2005.

[32] R. Vaidyanathan and A. Padmanabhan, "Bus-Based Networks for Fan-in and Uniform Hypercube Algorithms," *Parallel Computing*, vol. 21, pp. 1807-1821, 1995.

[33] R. Vaidyanathan and K. Sethuraman, "On Mapping Multidimensional Weak Tori on Optical Slab Waveguides," *Proc. International Conference on Parallel Processing*, 2005.

[34] Y. Yang, and J. Wang, "Designing WDM Optical Interconnects with Full Connectivity by Using Limited Wavelength Conversion," *Proc. International Parallel and Distributed Processing Symposium*, 2004.

[35] A. Yenjay, R. Gao, K. Takayama, and A. F. Garito, "Ultra-Low-Loss Polymer Waveguides," *Journal of Lightwave Technology*, vol. 22, no. 1, January 2004, pp. 154-158.

[36] I. A. Young, "Introducing Intel's Chip-to-Chip Optical I/O Technology," Technology Intel Magazine, April 2004. www.intel.com/update/departments/initech/ito4o41.pdf

Vita

Divya Rengan completed her schooling in April 2000, at Holy Angels Anglo Indian Higher Secondary School, Chennai, India.

She pursued her bachelor's degree at Sri SaiRam Engineering College, Chennai (affiliated to the University of Madras), majoring in information technology. She graduated with distinction in May 2004.

She then joined the Department of Electrical and Computer Engineering at Louisiana State University, Baton Rouge, to do her master's, in the Fall of 2004. She will be graduating in December 2006 with the degree of Master of Science in Electrical Engineering.