USC–SIPI REPORT #346

Optoelectronic Enhancements to Single Instruction Multiple Data Processing Architectures

by

Bogdan Hoanca

May 1999

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

Dedication

To the memory of my grandparents.

Acknowledgments

I am deeply grateful to my thesis advisor, Prof. Alexander A. Sawchuk, for his invaluable support, guidance and encouragement throughout the path of this work. Over the past few years, his insightful comments have channeled my research pursuits into interesting and fruitful territories. Also, his kind directions, endless patience and always-open door have made this time a pleasant journey of discovery, as well as self-discovery.

I also like to thank my dissertation committee members: Prof. B. Keith Jenkins, Prof. Ulrich Neumann, Prof. William Steier, and Prof. Alan E. Willner for their constructive suggestions and comments. In particular, Prof. Willner has been like a second advisor to me and has taught me many fine points on writing well and on dealing with people.

A special thanks goes to the outstanding people in the research group, Drs. Chihhao Chen, Charlie Kuznia, and Jenming Wu. We have spent many late evenings together, finishing up a design project, simulating a last minute design change or testing a chip.

Many people and friends have been of invaluable support over the course of this work. Dr. Adrian Moga welcomed me at USC and helped me get settled. Dr. Imran Hayee and Bindu Madhavan have been true friends and have acted as research advisors at times. Drs. David Norte and Eugene Park have guided my first research steps and have given me invaluable advice. Additionally, I acknowledge the support of other SIPI faculty and students, Prof. Antonio Ortega, Prof. Christos Kyriakakis, Dr. Wei-Feng Hsu, Dr. Jeng-Feng Lin, Dhawat (Eddie) Pansatiankul, Liping Zhang, and Changki Min.

I also want to thank Dr. Allan G. Weber and Seth Scafani for their invaluable help and patience with computer and network support. Gloria Halfacre, Milli Montenegro, Linda Varilla, Regina Morton, Diane Demetras and Anna Fong have been a tremendous administrative help, and have patiently endured my sometimes last minute requests.

Last but definitely not least, I am deeply grateful to my parents Vasile Hoanca and Nicoleta Hoanca, and to my sister Adriana Silvia Hoanca, who have been a constant moral support. Their love and understanding have kept me going at times when nothing seemed to work right. And more than anybody else, my wife Anne-Christine Aycaguer has been a constant presence of love and support in my life, making these years worth remembering beyond their research value.

Contents

List of figures

List of tables

Table of acronyms

n - D	n -dimensional, where n is 1, 2 or 3
ALU	arithmetic logic unit
CH	Cellular Hypercube
CMOS	complementary metal oxide silicon
CSMA/CD	carrier sense multiple access/ collision detection
DARPA	defense advanced research programs agency
DOE	diffractive optical element
ECL	emitter coupled logic
FDDI	fiber distributed data interface
FFT	fast Fourier transform
FIFO	first-in-first-out
FSM	finite state machine
GMU/CO-OP	George Mason University Consortium for Optical and Optoelectronic Technologies in Computing
IC	integrated circuit
I/O	input and output
IR	instruction ready (handshake signal for TRANSPAR)
LAN	local area network
LD	laser diode
LED	light emitting diode
M-RCH	mesh-reduced cellular hypercube
MCM	multi-chip module
MIMD	multiple instruction multiple data
MQW	multiple quantum well

OCI	Optimized Cellular Interconnection
OE	optoelectronic
OEIC	optoelectronic integrated circuit
OPDP	optical parallel data packet
OPS	optical power supply
PE	processing element
PLL	phase locked loop
QWP	quarter wave plate
RDY	chip ready (handshake signal for TRANSPAR)
SEED	self-electro-optic device
SIMD	single instruction multiple data
SNR	signal to noise ratio
SPARCL	smart pixel array cellular logic
SR	status register
SRAM	static random access memory
TRANSPAR	translucent smart pixel array
VCO	voltage controlled oscillator
VCSEL	vertical cavity surface emitting laser
VLSI	very large scale integrated circuits

Table of symbols

B	average busy period
D	expected packet delay
$D(n)$	optimality distance using the first n optical links in the interconnect
$\Delta t_{\text{network}}$	total skew per network
Δt_{node}	total skew per node
Δt_{photo}	skew due to uncertainty in the photodetected current
G	offered network load
h_i	sorting distances for the Shell sort
K	number of optical links (fan-out) for the interconnect
M	number of partitions in the time multiplexing algorithm
$N_{\text{effective}}$	ratio of the on-chip clock rate to the off-chip clock rate
N_{rate}	number of microinstructions per macroinstruction
q_0	probability of zero packets accumulated at the end of a packet transmission time
R	detector responsivity
$\bar{r}_1$	average pretransmission delay
S	maximum number of electrical hops in the OCI
S_n	normalized throughput
T	packet transmission time
T_{clock}	clock period
$X(n)$	distance of the n^{th} optical link in the interconnect
$\bar{Y}$	average packet transmission overhead

Abstract

Electronic single instruction multiple data (SIMD) architectures are used for parallel computation, for example image processing or real-time array processing. Oftentimes, such architectures are communications limited. Using optoelectronic interconnections for global connectivity can alleviate the communications bottleneck. We present optimizations of the optoelectronic architecture at four levels: interconnection topology, optical system, electrical system and inter-chip network.

In optimizing the interconnection topology we concentrate on cellular optoelectronic interconnects, because they are space-invariant. Previously, cellular interconnects have demonstrated incremental improvements, but were unable to predict the ultimate performance achievable, and their design involved a trial-and-error approach. We present a deterministic algorithm for designing optimal cellular interconnections (OCIs). According to formal proofs and numerical simulations, OCIs require the minimum number of clock cycles per data shift for a given number of optical links.

A wavelength and polarization multiplexed optical interconnect can distribute in parallel instructions and clock information in the SIMD array. The feasibility of such a multiplexed interconnection, as well as the architecture and the thermal management of the optical system, depend heavily on the choice of optical source. Optical devices allow faster switching than electronic devices, but their advantages are often masked by a bottleneck at the interface between the optical and electronic domains. An on-chip finite state machine acting as a rate-converter can eliminate this bottleneck. We also discuss the design of timing circuitry and issues on skew in parallel optical communications channels.

Stacking SIMD processing arrays in a network provides extremely high computational and communications throughput, passing 2-D packets over multiple parallel channels, distributed across the whole area of the chip. In optimizing widely parallel networks,

traffic considerations interplay with requirements on allowable skew. Optimum network performance is attained in general for a large number of channels. Clocked translucent nodes minimize the network delay.

To demonstrate most of the optimizations above, we designed and fabricated TRANSPAR, a smart pixel integrated circuit with networking and SIMD processing applications. TRANSPAR chips are interconnected into a high-throughput ring network and combine their SIMD processing power, operating as a massively parallel pipeline computational system.

Chapter 1. Introduction

Things have changed from the days when the all-optical computer was a realistic research goal. The research community now agrees that the desirable features of optics are not a replacement, but a complement for the features of electronics. While electronics is very good at switching and storing data with low access time, optics is better at transporting data over large distances and at high data rates. Attempts at using optics for switching have slowly been abandoned, while using electronics for interconnections has become more and more of a bottleneck. For these reasons, the future is not an all-optical computer, but a computer with optical interconnections carrying massive amounts of data in parallel between the electronic computing blocks.

This work presents a set of possible optimizations for computing architectures that use optical interconnections. Many such interesting architectures have been proposed and demonstrated over the past years [1, 2, 3, 4]. The system design has been facilitated by the continuous improvement of the optoelectronic devices [5, 6], as well as by the smooth integration of the optoelectronic devices with standard very large scale integration (VLSI) processing [7, 8, 9]. Nonetheless, while optoelectronic devices have steadily improved and have been attached to VLSI chips with ever increasing yield and in ever increasing array sizes, the packaging and interfacing of the system is lagging. This work will address some of the packaging and interfacing issues, and will deal mainly with single instruction multiple data (SIMD) architectures, which are particularly interesting for massively parallel optical interconnections.

SIMD machines are widely used today as a lower cost and high performance architecture for parallel programming applications. SIMD machines combine the power of processing multiple parallel data streams and the simplicity of a single instruction flow. The SIMD paradigm recognizes that a large class of applications involve identical parallel

operations on multiple sets of data, for example in image processing or in solving equations with large and sparse matrices. By distributing the storage medium (memory) to multiple processors, which perform the same operations in parallel on multiple different data elements, a serial algorithm can be used with minimum modifications in a parallel architecture. Nonetheless, the data processing still involves the overhead of communications between the processors, due to the somewhat non-local nature of the computation. In many applications, the time a processor spends communicating with other processors may exceed the time actually spent computing. In such cases, minimizing the communications overhead is necessary for achieving high-throughput parallel operation.

1.1. The communications and control bottleneck in SIMD architectures

Current research on single-instruction multiple-data machines shows that their performance is limited by the delay of the interconnections that provide the communications between the processors [10, 11, 12]. Electronic SIMD machines are available commercially and are widely used for computing-intensive applications, such as image processing [13, 14, 15], nanoelectronic device simulations [16] and atmospheric modeling [17]. Such SIMD machines are arrays of processing elements (PEs) interconnected electronically. The interconnection topology can be a mesh, an *N*-dimensional hypercube (n-cube), or a multistage interconnection (for example the Perfect Shuffle) (Fig. 1.1). These are the preferred topologies, because they have a relatively low fan-out (each PE is connects to only a small number of other PEs).

The most commonly used topology is the mesh, which is relatively simple to implement in current VLSI technologies. In such mesh topologies, only short-distance links are physically made between processors. Long distance data transfers are made in a multihop fashion, and the total interconnection delay is linearly proportional to the number of hops, so these interconnections introduce a relatively long latency. This makes the mesh rather

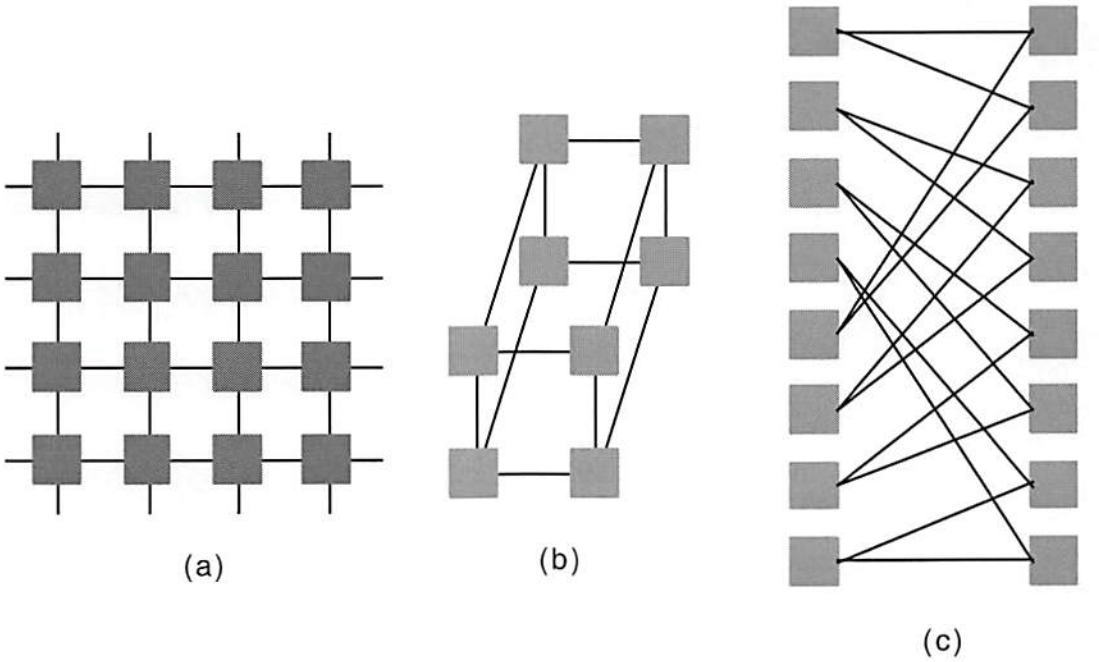

Figure 1.1 Interconnection topologies for SIMD computers. (a)- mesh (b) N–dimensional hypercube (here $N = 3$) and (c) Perfect Shuffle.

inefficient for communications over long distances in the array. While many problems that map well on SIMD machines involve only local connections and are not communications limited by the mesh interconnection, many other problems involve global, long-distance communications between processors. Image motion estimation and matrix-vector-multiplication are only two such problems, where communication distances can be as large as the diameter of the SIMD array. Both image motion estimation and matrix-vector-multiplication occur for example in multimedia applications such as video compression, virtual reality and telepresence systems.

All the alternatives to the mesh topology involve long distance links. These links are in general electronic, and they again limit the performance of the system, due to the large propagation delays associated with long and thin conductors. Moreover, at high data rates, the long electrical lines behave like transmission lines. Multiple reflections occur on the transmission lines unless the lines are impedance-matched at either or both the source and

the receiver. Impedance matching requires additional design effort, consumes large amounts of electrical power and is only a limited range solution, as the attenuation of the lines becomes prohibitive at Gbit/s rates. Optoelectronic interconnections have been proposed and demonstrated to circumvent such problems. In using optoelectronic links to interconnect the SIMD processors, care must be taken to optimize the architecture for the specific features of optical links. The rest of this work presents and assesses the type of architectural optimizations that can be made, and the performance improvement to be expected from the architectures involving optoelectronic interconnections.

1.2. Optoelectronic enhancements to the SIMD architecture

Optimization of the SIMD architectures using optoelectronic links can be done on multiple levels. At the highest abstraction level, the topology of the interconnection can be optimized to take advantage of the unique features of the optoelectronic links. At this level, we show how to design the optimum interconnect topology, for simultaneously minimizing the fan-out and the latency of the interprocessor communication.

Moving closer to the physical implementation, we analyze the optimizations that can be performed on the optical system. Optical interconnections are usually made in a third dimension, perpendicular to the plane of the electronic chip. For this reason, they offer design flexibility because they do not interfere with the electronic interconnections (in the plane of the chip). This is a big advantage, because the electronic and the optical designs are only weakly coupled. Additionally, the optical interconnections offer flexibility in the wavelength and polarization domains in addition to the space multiplexing of the third dimension interconnection. The drawback of using the third dimension is the added complexity in the design and alignment of the optical system, with novel challenges that we explore. Finally, as an added benefit of optics, in imaging systems the optical paths are

equal up to subwavelength accuracy, automatically minimizing the skew. The remaining challenge is to reduce or eliminate the skew of the electronic end-points of the link.

Additional optimizations can be made on the electronic circuitry. Over the years, the design of interfaces between the optical and the electronic domains has remained difficult because careful, mixed-signal design must be done. Similarly, the interface between the optoelectronic processor chips and the peripheral devices must be designed in such a manner as not to slow down the optoelectronic devices. Finally, careful and flexible timing must be included on the optoelectronic chips to be able to test, synchronize and use the chips at peak performance.

Optimization of the SIMD architecture may be performed over a single chip or over multiple chips. When multiple chips are interconnected with optical parallel channels in a packet switched network, this powerful architecture can combine SIMD and pipeline processing, to further increase the throughput.

After detailing the proposed optimizations, we conclude with an estimate of the performance improvement to be expected. Some of the proposed optimizations have already been implemented and tested on TRANSPAR, our demonstration system for digital optoelectronic signal processing and networking. We introduce the basic functionality of TRANSPAR in the next section. Throughout the remainder of this work, we use results obtained on TRANSPAR as a practical validation of the optimization techniques we introduce.

1.3. Original contributions in this work

This research encompasses optimizations of SIMD computing architectures at four levels: topology, optical system, electronic system and network of multiple SIMD arrays. Details on the contributions at each level are given below.

1.3.1. Optimization of the interconnection topology

- Determined the optimal cellular interconnection topology for interconnecting SIMD arrays using space invariant fan-out elements. Formulated a deterministic algorithm to design the interconnection pattern, with an option of optimizing for speed or for power efficiency.
- Demonstrated the optimality of the interconnection using theoretical means, as well as simulations on low-level and high-level applications.

1.3.2. Optimization of the optical system

- Proposed means to build complex multiplexed diffractive elements combining both wavelength and polarization, to be used for delivering multiple classes of optical signals to the SIMD array.
- Performed an engineering evaluation of the available optical sources – active and passive, and of their relative merits in building an optical system.

1.3.3. Optimization of the electronic circuitry

The contributions in this section were incorporated into the design of the demonstrator chip, TRANSPAR. A project of such magnitude as the design and testing of the TRANSPAR chips and network can only be the result of a concentrated team effort, rather than a single person's work. Drs. Chihhao Chen, Charlie Kuznia and Jenming Wu did much of the design and layout work involved. The portion that was done as part of this work includes:

- Designed the architecture of the PE for SIMD processing (section 6.2).
- Designed and programmed the finite state machine for eliminating the I/O bottleneck between the slow and the fast circuitry (section 6.3).

- Designed the circuitry for the asynchronous network interface (Section 7.2.4).
- Designed the clocking circuitry (section 6.4).
- Set up, interfaced and tested the fabricated chips.

For completeness, and to facilitate the understanding of the work reported here, we describe the entire TRANSPAR architecture, not just the original contributions. Additionally, work in progress continues on the testing of the fabricated chips, and is done by Liping Zhang and Dhawat Pansatiankul.

1.3.4. Optimizations of the network functionality

- Designed the asynchronous communication protocol that removes the bottleneck between the electronic host interface and the optical network.
- Evaluated the performance of the network across the design parameters, in particular the large number of parallel channels available through the smart pixel technology.
- Evaluated the relative performance of clocked and translucent nodes on the network.

1.4. Organization of this work

The remainder of this work is organized as follows:

- Chapter 2 reviews the basics of using optical interconnections, showing the relative advantages of optical interconnections over electronic interconnections, as well as previous work done on the topologies of the optoelectronic cellular interconnections.
- Chapter 3 presents the optimized cellular interconnection, which is formally proven to achieve the ultimate performance possible with a space-invariant connection pattern. Additionally, an algorithm to generate the optimized cellular interconnection is presented, along with simulations showing that the topology is indeed superior to the best previously published results.

- Chapter 4 introduces the demonstrator system, TRANSPAR, showing details of the architecture and the implementation.
- Chapter 5 evaluates the relative merits of the optical sources available for constructing an optoelectronic system. The influence of the optical source on the design of the overall optical system is also evaluated.
- Chapter 6 presents the optimizations of the electronic circuitry, with an emphasis on achieving high-speed operation for both computing and networking applications.
- Chapter 7 extends the considerations of the previous chapters to more complex systems, which combine multiple SIMD nodes into high throughput networks. With close references to the TRANSPAR demonstrator system, we evaluate the network performance and its limits.
- Chapter 8 concludes on the results of the research and opens up the issues of further work.

1.5. Summary

The quest for the all-optical computer has been replaced with the more realistic goal of an optically interconnected electronic computer. This work explores the possible optimization techniques that can be applied to optoelectronic computing architectures, with particular applications to SIMD computers. The communication between processors in a large SIMD array limits the performance of such architectures, either due to the short links available or due to the capacitive loading of the longer links. Optoelectronic implementations of the interconnection network can speed up the communications and remove the bottleneck. The proposed optimizations can be implemented at the topology level, at the optical system level and at the electronic level. This work explores the options and concludes on the attainable performance improvement figures that can be expected if

using optoelectronic interconnections in a SIMD architecture. Some of the proposed techniques have been demonstrated experimentally on a chip we designed and are currently testing.

References

[1] T. H. Szymanski and H.S. Hinton, "Architecture of a terabit free-space intelligent optical backplane," *Journal of Parallel and Distributed Computing*, vol. 55, no. 1, pp. 1–31, 1998.

[2] A. Louri, B. Weech, and C. Neocleous, "A spanning multichannel linked hypercube - a gradually scalable optical interconnection network for massively-parallel computing," *IEEE Transactions on Parallel Distributed Systems*, vol. 9, no. 5, pp. 497-512, 1998.

[3] A. Louri and C. Neocleous, "A spanning bus connected hypercube - a new scalable optical interconnection network for multiprocessors and massively-parallel systems," *IEEE Journal of Lightwave Technology*, vol. 15, no. 7, pp. 1241-1252, 1997.

[4] C. B. Kuznia and A. A. Sawchuk, "Time multiplexing and control for optical cellular-hypercube arrays," *Applied Optics*, vol. 35, no. 11, pp. 1836-1847, 1996.

[5] F. A. P. Tooley, "Optical interconnects do not require improved optoelectronic devices," in *Proceedings of Optics in Computing '98*, SPIE vol. 3490, pp.14-17, 1998.

[6] F. A. P. Tooley, "Challenges in optically interconnecting electronics," *IEEE Journal on Selected Topics on Quantum Electronics*, vol. 2, no.1, pp. 3-13, 1996.

[7] A. V. Krishnamoorthy, L. M. F. Chirovsky, W. S. Hobson, R. E. Leibenguth, S. P. Hui, C. J. Zydzik, K. W. Goossen, J. D. Wynn, B. J. Tseng, J. Lopata, J. A. Walker, J. E. Cunningham, and L. A. D'asaro, "Vertical-cavity surface-emitting lasers flip-chip bonded to gigabit-per-second CMOS circuits," *IEEE Photonics Technology Letters*, vol. 11, no. 1, pp. 128-130, 1999.

[8] A. Krishnamoorthy and K. W. Goossen, "Optoelectronic-VLSI: photonics integrated with VLSI circuits," *IEEE Journal on Selected Topics in Quantum Electronics*, vol. 4, no. 6, pp. 899-912, 1998.

[9] K. W. Goossen, J. A. Walker, L. A. Dasaro, S. P. Hui, B. Tseng, R. Leibenguth, D. Kossives, D. D. Bacon, D. Dahringer, et al., "GaAs MQW modulators integrated with silicon CMOS," *IEEE Photonics Technology Letters*, vol. 7, no. 4, pp. 360-362, 1995.

[10] K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, New York, NY: McGraw Hill, 1993.

[11] H. S. Stone and J. Crocke, "Computer architecture in the 1990s," *Computer*, vol. 24, pp. 30-38, 1991.

[12] H. J. Siegel, *Interconnection Networks for Large-scale Parallel Processing*, New York, NY: McGraw Hill, 1990.

[13] H. C. Shi, G. X. Ritter, and J. N. Wilson, "A fast general algorithm for extracting image features on SIMD mesh-connected computers," *Pattern Recognition*, vol. 30, no. 7, pp. 1205-1211, 1997.

[14] H. N. Kim, M. J. Irwin, and R. M. Owens, "Motion analysis on the micro grained array processor," *Real-time Imaging*, vol. 3, no. 2, pp. 101-110, 1997.

[15] D. G. Beetner and R. M. Arthur, "Generation of synthetic-focus images from pulse-echo ultrasound using difference-equations", *IEEE Transactions on Medical Imaging*, vol. 15, no. 5, pp. 665-672, 1996.

[16] X. D. Wang, V. P. Roychowdhury, and P. Balasingam, "Scaleable massively-parallel algorithms for computational nanoelectronics," *Parallel Computing*, vol. 22, no. 14, pp. 1931-1963, 1997.

[17] J. Brown, P. C. Hansen, J Wasniewski, and Z. Zlatev, "Comparing the performance of SIMD computers by running large air-pollution models," *Supercomputer*, vol. 12, no. 2, pp. 21-35, 1996.

Chapter 2. Optically interconnected cellular arrays

Moore's law has been the driving or benchmark standard for the computing industry for the past two decades. While the processing speed seems to scale well with the reduction of device sizes (as higher performance VLSI technologies with smaller feature sizes are introduced), the communication links become more and more of a bottleneck. Gone are the days when the device was the bottleneck and the interconnections behaved like ideal wires, without any parasitic elements. The combined effects of the reduction in device size and the increase in chip size have increased the loading effects of the interconnection lines on the switching devices. Smaller devices operate faster, but not when required to drive the longer and thinner lines connecting to distant destinations. In particular, SIMD architectures, which employ large arrays of processing elements (PEs) spaced over relatively large distances, have been severely plagued by the effects of slow electronic interconnections.

As SIMD machines continue to increase in size, the complexity and the challenge of designing such machines shift from the design of the PEs to the design of the interconnection network. In fact, the design of the PEs is straightforward, and largely unaffected by the array size, because all PEs are relatively small and have the same architecture. The fact that the PE is small makes the skew negligible for the signals within a single PE. At the same time, the network of links between PEs is not only increasing in size with the number of PEs, but is also becoming more challenging in terms of fan-out (number of PEs that are interconnected with a given source PE), distance of interconnect, and physical and topological design. The design of the interconnection network must allow sufficient room for the placement of the interconnection paths and sufficient separation between neighboring paths to reduce the amount of interaction (crosstalk) between them.

Optical architectures (Fig. 2.1) have shown very good promise for easing the implementation of such large-scale arrays systems.

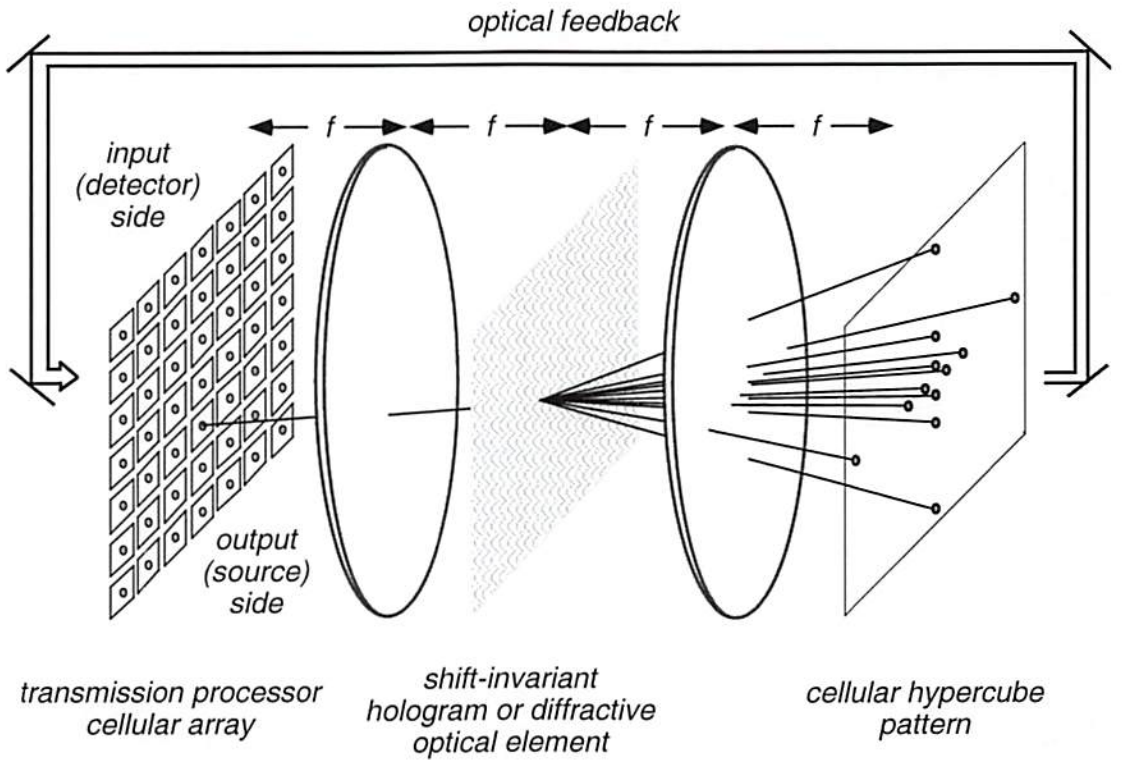

Figure 2.1 Implementation of an optical space-invariant interconnect in a two-lens telecentric (4-F) system.

Figure 2.1 shows a concept for an optical interconnection system using *transmissive* devices. Alternatively, a *reflective* cellular array and interconnection hologram or DOE could also be used to implement the system. In the implementation shown in Fig. 2.1 each transmitted beam is fanned-out onto the destination PEs using a Fourier plane hologram in a 4-F system. Macro- or micro- lenses can be used for imaging, the latter having lower aberrations but more critical alignment tolerances. The interconnection hologram can be a low space bandwidth product computer generated hologram (CGH) if the interconnection pattern is space-invariant. The advantage of using a CGH is that it can be digitally generated and reproduced, as opposed to a more expensive analog hologram that is needed if the interconnection pattern is shift variant. In addition, the shift-invariance reduces the

alignment requirements to only longitudinal and rotational degrees of freedom. The performance of the CGH depends on the size and the number of phase levels, with efficiencies of up to 90% for eight levels [1].

The optical links require optical transmitters at the source PEs that convert electronic signals to optical format before broadcasting them through the CHG. At the receiving end of the 4-F system, detectors convert the optical signal back to electronic format for the receiving PEs. Such an optoelectronic implementation preserves the PE architecture and the electronic local connections between PEs, but shifts the burden of the long-distance links to optics, which is less affected by distance.

2.1. Advantages of optoelectronic links over electrical links

Optical links can be more energy efficient than electronic links for interconnection

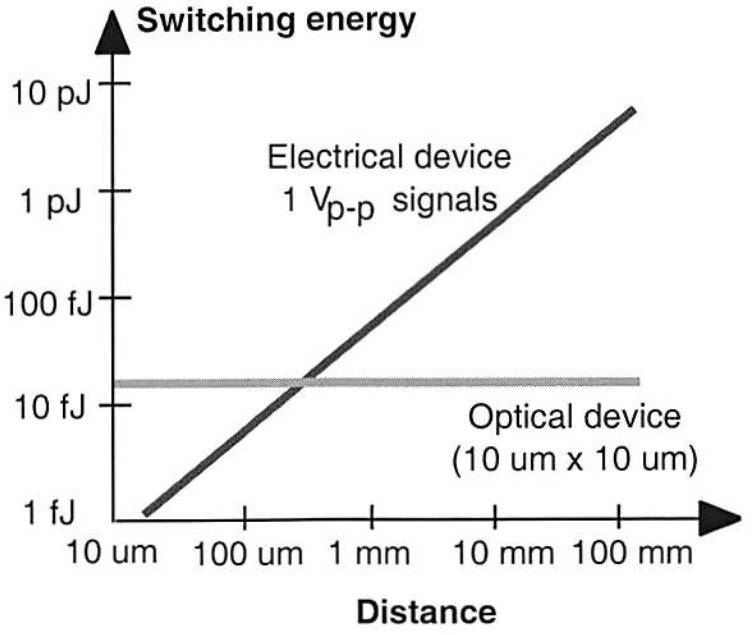

Figure 2.2. Comparison of energy required to send one bit versus the transmission distance, for the case of optical and electronic links [8].

distances exceeding even a few tenths of a millimeter (Fig. 2.2, [8] for 10 x 10 μm optical modulators and 1V electrical signals swing). In a large array of PEs, whether implemented using wafer scale integration (WSI) or using multi-chip module (MCM) techniques, the distances between the PEs are in general long enough to warrant significant advantages for optical links.

Long-distance electronic interconnections are plagued by large capacitive loads, impedance mismatch and clock skew, while optical links have a capacitance independent of the link distance, require no impedance matching and have a constant, known and

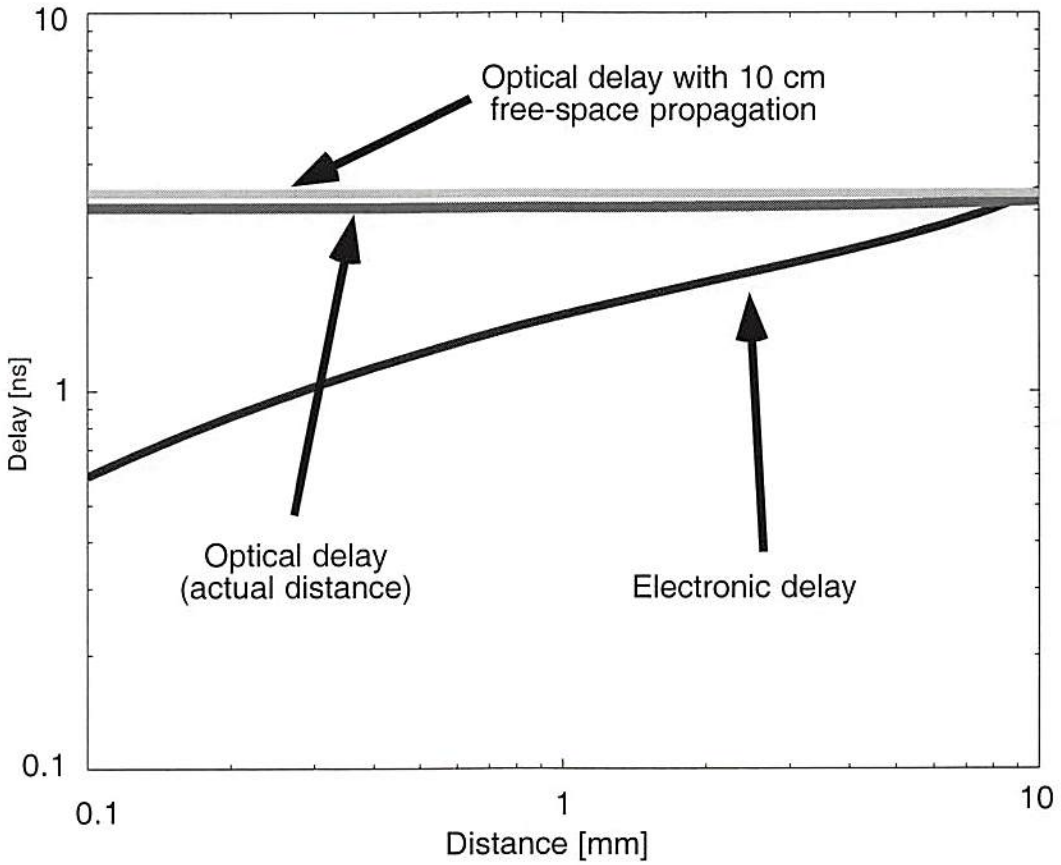

Figure 2.3. Comparison between the delay of optical and electronic links. The values are for a 0.5 μm CMOS technology. Optical delays include transmitter, time-of-flight and receiver delay. Electronic delays are based on a RC transmission line model.

controllable skew [2, 3]. Depending on the design of the link and the power budget, optical links can have a lower propagation delay than electronic links. Figure 2.3 shows a plot of the delay of optical and electronic links as a function of propagation distance. For the delay of the electronic links we assumed transmission lines propagation, while for the optoelectronic delay we used the model of the TRANSPAR OE channel (detailed in Section 7.4). The optical delay is mainly due to the latency of the receiver circuitry, and hence is only slightly dependent on the propagation distance. While this is a particular example, it is fairly general in the sense that the optical link has a latency concentrated at the end points and with low distance dependence. The electronic links on the other hand have a latency that increases as a quadratic function of the distance.

In addition to the advantages above, optics uses the third dimension to interconnect PEs with out-of-plane links, and reduces the required real estate on or between PEs for electronic links. While electrical links require physical separation to avoid crosstalk, optical beams can propagate through each other in free space without deleterious effects. Electronic interconnects are confined to the plane of the PEs and require multiple layers of metal (hence multiple processing steps). Conversely, optical links allow a simple and planar layout for the electronic circuitry and move the burden of the interconnection into the readily available space above the chip. The superiority of optical interconnects for long distance interconnections has been advocated by many authors [4, 5, 6, 7, 8, 9, 10, 11].

2.2. The optoelectronic Cellular Hypercube

The optoelectronic cellular hypercube (CH) [12] was proposed to improve the interconnection performance of SIMD arrays of processors. The CH architecture is an optoelectronic overlay over a mesh-connected array of SIMD processors. It is intended to minimize the number of clock cycles required when shifting data over long distances in the PE array. The intention is not to interconnect all the PEs in a regular hypercube pattern (as

the term hypercube may suggest), but only to supplement the nearest-neighbor mesh connections with a set of direct long-distance (short-cut) optical links (Fig. 2.4). The CH makes use of a set of long distance interconnects with exponentially increasing distances (powers of two, i.e. 2, 4, ... 2^K for K links), to speed up the long distance data transfers by reducing the number of hops. Instead of a linear increase (as in the case of a mesh interconnection with multihop transfers over long distance), the number of hops per data shift in the CH increases only logarithmically with the distance.

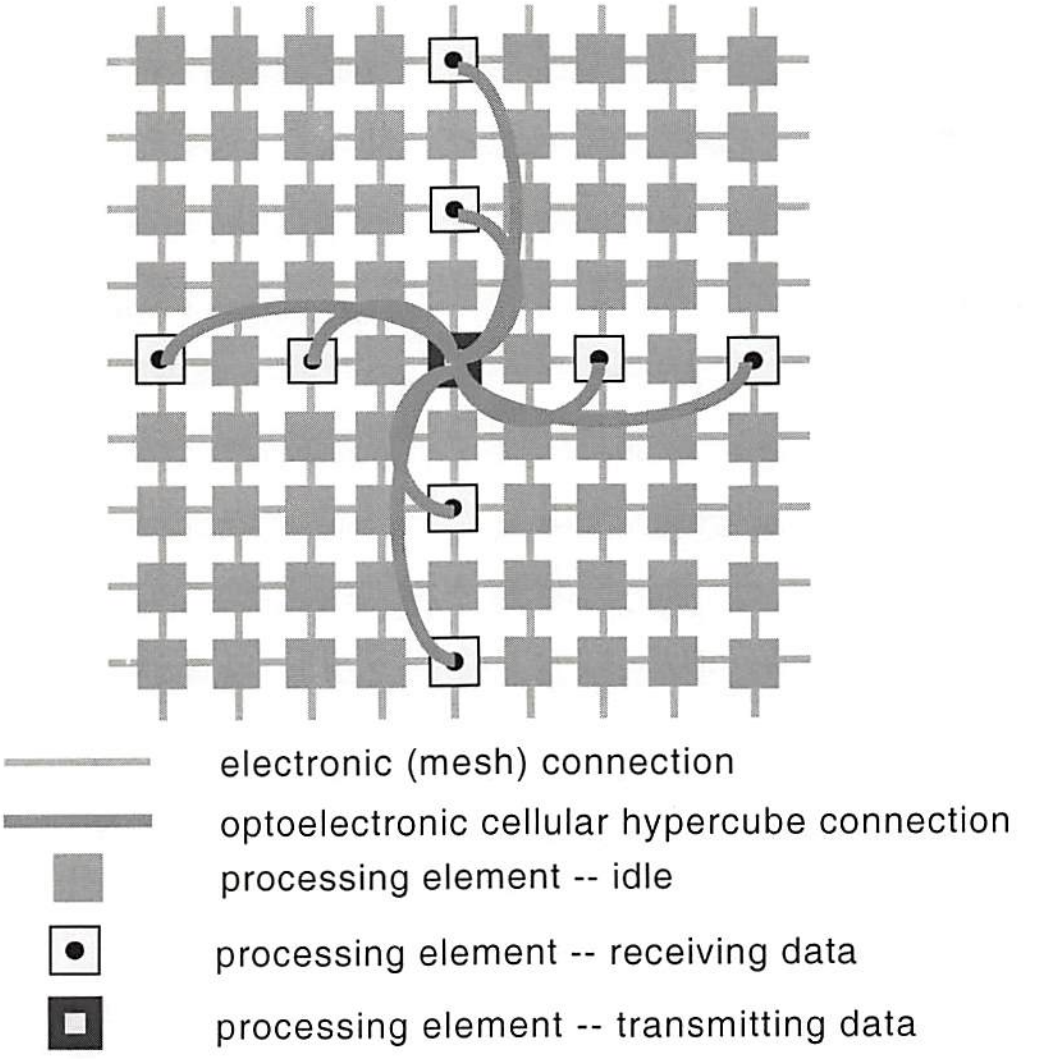

Figure 2.4. Cellular hypercube interconnection.

Among optoelectronic interconnects, the CH offers the advantage of a space-invariant interconnection pattern, in which light from each transmitting PE fans out in an identical pattern onto a set of receiving PEs. This space-invariance of the interconnect pattern makes it suitable for full-aperture optical implementations (Fig. 2.1) and does not require components with high space bandwidth product [13]. Alternative optoelectronic space–variant implementations require a pixel-based aperture and more complex designs of the optical interconnection element. In such cases, an analog hologram may need to be used in the place of the cheaper and easy to manufacture CGH [12].

Unlike other hypercube implementations, the CH allows full utilization of the real estate of the SIMD array. Sheng [14], and Louri et al. [15, 16] demonstrated space-invariant hypercube interconnection patterns, but they require empty rows and columns in the PE array, which reduces the real estate utilization. The CH does not require any empty space. On the other hand, the CH is not topologically equivalent to a hypercube and some of the links from some nodes in the CH may fall outside the array. This is not a problem, since the CH is not used as a regular interconnect intended to provide full hypercube connectivity, but it is only used as a set of shortcut links to speed up the communications in an SIMD array.

The SIMD processors which may be using the CH optoelectronic links are arranged in a one-dimensional (1-D) or two-dimensional (2-D) mesh-connected array; a 1-D array is shown in Fig. 2.5. The mesh has electrical bi-directional interconnections (not shown)

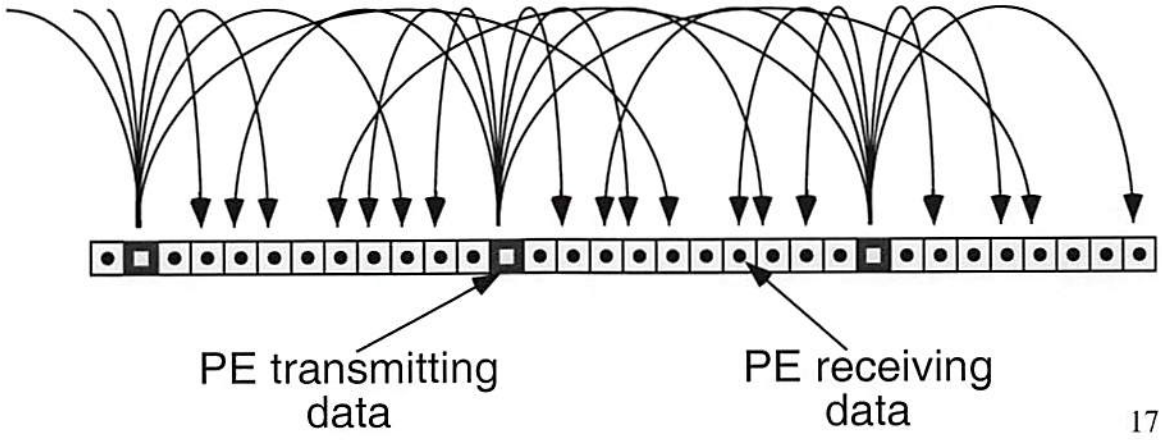

Figure 2.5. One-dimensional array of PEs showing the optical interconnections.

with neighboring PEs left, right, above and below. Optical free-space connections (shown as curved arrows in Fig. 2.5) are made symmetrically at distances which are powers of two, i.e. 2, 4, ... 2^K for K links. For a 2-D array, the same pattern is generated along rows and along columns (in a grid fashion). Positive link distances are those measured to the right (or upwards) of the transmitter node and negative distances are those measured to the left (or downwards). There are no wrap-around connections at the edges of the array. Connections falling outside the array are lost.

In an SIMD machine the same instruction is broadcast from a control unit to all PEs, but only selectively activated PEs execute synchronously the instruction on their local data. Either the electronic mesh or the optical links can be used at any time (but not both). Electronic links to any PEs neighbors are directional and selective. Exactly one of the neighbors of each PE can be selected to receive data. Since this is a one-to-one connection, all PEs can send and receive data at the same time in exactly one clock cycle. On the other hand, the optical interconnect is a pure fanout link. Data sent by each processor are broadcast through all optical links at the same time (no particular link can be selected). Thus, the optical hypercube links are one-to-many interconnects, which introduce contention (multiple PEs sending data to the same receiver at the same time).

To solve the problem of contention, a time slotted multiplexing algorithm has been demonstrated [12, 17]. The PEs in the array are partitioned into M disjoint sets, such that all the PEs in one set have addresses with the same remainder modulo M. Processors at positions $l\,M\,+n$, with l and n integers, are all in set n. Each of the M sets of PEs take turns in transmitting data, and M is chosen so that no contention occurs among PEs in the same set (Fig. 2.6, for $M = 11$). This solves the problem of contention, but increases latency. PEs now need to wait for their scheduled time slot in order to transmit data (for the case in Fig. 2.6, PE number 12 can transmit data in slot 1 and then only in slot 12). Since any PE set is scheduled to transmit data in one slot out of M, minimizing M will

ensure a minimum latency. The time slotted access is required only because of the one-to-many nature of the optical interconnects. Through the electronic mesh all PEs can send data at the same time. Thus for short distances the mesh can outperform the optical interconnect, since no time slotted access is required [18].

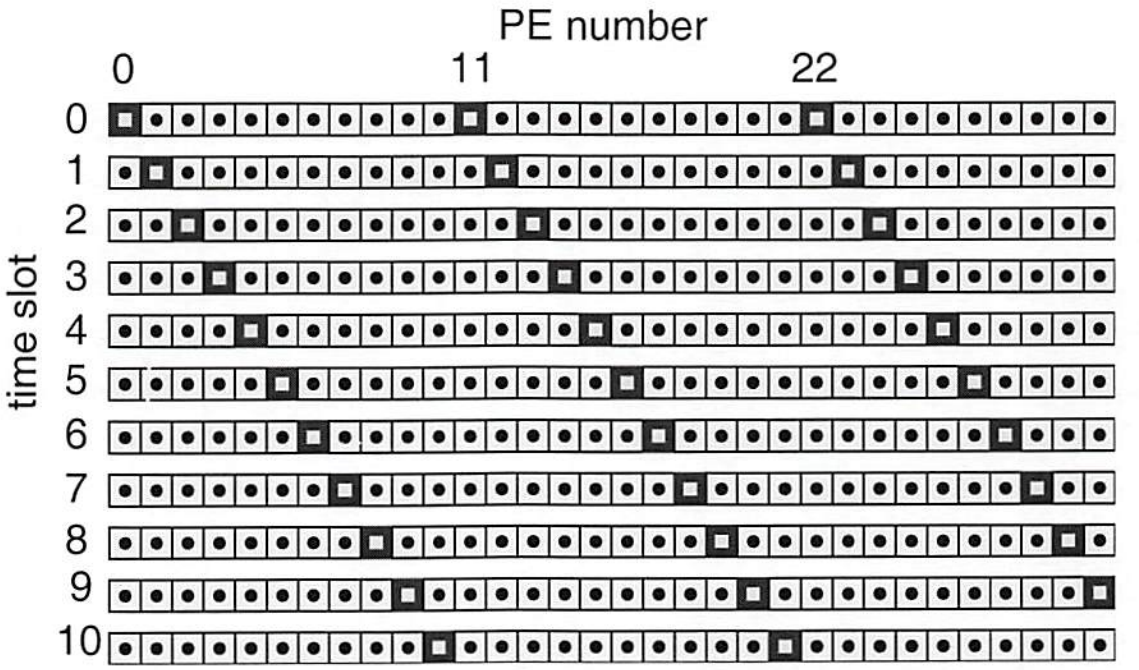

Figure 2.6. Time multiplexing for the CH. Scheduling of the transmitting PEs is illustrated for the time-slotted protocol with $M = 11$. After eleven consecutive time slots the scheduling is repeated periodically (time slot 11 is identical with time slot 0).

2.3. The mesh-reduced Cellular Hypercube

Because the mesh can be more efficient than the short distance optical links in the CH, these links can be removed, leading to savings in optical power, to simpler design of the interconnection DOE and to better performance of the interconnection. The architecture of the CH with the short links removed was called mesh-reduced CH (M-RCH) [18].

Lin and Sawchuk [18] showed that the M-RCH can significantly outperform the CH. The M-RCH uses only a subset of the optical links in the CH, eliminating the inefficient

(shorter distance) optical links, and using the electronic mesh for shorter distance connections. Overall, the M-RCH was shown to significantly reduce the number of clock cycles per data shift. This opened up the issue of how much more improvement could be expected from further modifying the CH topology.

2.4. Variations and incremental improvements of the Cellular Hypercube

In the quest for the optimum interconnection topology, Lin [19] has demonstrated various other schemes for contention avoidance, while preserving the basic M-RCH link distances. Such schemes use the same time slotted protocol and reduce the latency by augmenting the number of detectors per PE. Multiple detectors can each receive separate incoming data, reducing or eliminating contention. Based on this reduced contention, M can now in turn be reduced, which leads to lower latency. Still, the lower latency comes at the expense of increased costs and tighter fabrication constraints, both in terms of the sheer number of optical devices, as well as in the fanout and accuracy of the DOE.

In contrast with the "brute force" latency reduction by using more detectors, our work explores the advantages of optimizing the topology itself, without necessarily requiring additional detectors on the PEs.

2.5. A look ahead

Our architectural optimization starts with the topology, at the abstract level. The topology presented in this work is the most general cellular topology, in which the interconnection distances are optimized, without constraining them to be a subset of the powers of two. This design is proven to be optimal: for a given array size, it achieves the minimum fan-out of the optical interconnect, and simultaneously the minimum number of clock cycles per data shift. The optimization algorithm even allows a trade-off between fan-out and latency to be made in the design phase. More importantly, the topological

optimality of the interconnection is independent of the particular technology used, as well as independent of the size of the array. By decoupling the problem of topological optimization from the details of the implementation, we can deal with the optimization of the physical details separately from the optimization of the topology.

Additional optimization work includes the design of the optical system, considerations on the electronics and issues at the interface of optics and electronics. None of these issues have been completely solved by the previous work on the CH and M-RCH. We will explore both the theoretical and some of the practical aspects of the proposed optimizations. Our demonstrator system, TRANSPAR incorporates these ideas and serves as an experimental validation.

2.6. Summary

Optoelectronic links are much better suited for implementing the long-distance global interconnects in large SIMD arrays. Among optoelectronic implementations, the cellular hypercube offers multiple advantages, the main one being space-invariance, which allows the use of a low space-bandwidth diffractive optical element for the interconnection. The topology of the cellular hypercube has been refined and optimized, but it is not clear what the optimum topology may be. The next chapter will introduce and formally prove this optimum topology.

References

[1] K.-S. Huang, C. B. Kuznia, B. K. Jenkins, and A. A. Sawchuk, "Parallel Architectures for Digital Optical Cellular Image Processing," *Proceedings of the IEEE*, vol. 82, pp. 1711-1723, 1994.

[2] P. Sweazey, "Limits of performance of backplane buses," in *Digital Bus Handbook*, New York, NY: McGraw Hill, 1990.

[3] A. Louri and H. Sung, "3D optical interconnects for high-speed inter-chip and inter-board communications," *Computer*, vol. 27, pp. 27-37, 1994.

[4] J. W. Goodman, F. J. Leonberg, S. Y. Kung, and R. A. Athale, "Optical interconnections for VLSI systems," *Proceedings of IEEE*, vol. 72, pp. 850-866, 1984.

[5] A. A Sawchuk, C. S. Raghavandra, B. K. Jenkins, and A. Varma, "Optical crossbar networks," *IEEE Computer*, vol. 20, pp. 50-62, 1987.

[6] P. B. Berra, A. Ghafoor, M. Guizani, S. J. Marcinkowski, and P.A. Mitkas, "Optics and supercomputing," *Proceedings of IEEE*, vol. 77, pp. 1797-1815, 1989.

[7] M. R. Feldman, C. C. Guest, T. J. Drabik, and S. C. Esner, "Comparison between electrical and free space optical interconnects for fine grain processor arrays based on connection density capabilities," *Applied Optics*, vol. 28, pp. 3820-3829, 1989.

[8] D. S. Miller, "Optics for low energy communication inside digital processors: quantum detectors, sources, and modulators as efficient impedance converters," *Optics Letters*, vol. 14, pp. 146-148, 1989.

[9] D. M. Chiarulli, S. P. Levitan, R. G. Mehlen, M. Bidnurkar, R. Ditmore, G. Gravenstreter, Z. Guo, C. Qiao, M. F. Sakr, and J. P. Teza, "Optoelectronic buses for high-performance computing," *Proceedings of IEEE*, vol. 82, pp. 1701-1709, 1994.

[10] F. E. Kiamilev, P. Marchand, A. V. Krishnamoorti, S. C. Esner, and S. H. Lee, "Performance comparison between optoelectronic and vlsi multistage interconnection networks," *IEEE Journal of Lightwave Technology*, vol. 9, pp. 1674-1692, 1991.

[11] A. D. McAulay, *Optical Computer Architectures: the Application of Optical Concepts to Next Generation Computer*, New York, NY: John Wiley, 1991.

[12] C. B. Kuznia, "Cellular hypercube interconnections for optoelectronic smart pixel cellular arrays, " Ph.D. Dissertation, University of Southern California, Los Angeles, California, 1994.

[13] B. K. Jenkins, P. Chavel, R. Forchheimer, A. A. Sawchuk, and T. C. Strand, " Architectural implications of a digital optical processor," *Applied Optics*, vol. 23, pp. 3465-3474, 1984.

[14] Y. Sheng, "Space invariant multiple imaging for hypercube interconnections," *Applied Optics*, vol. 29, pp. 1101-1105, 1990.

[15] A. Louri and H. Sung, "Efficient implementation methodology for three-dimensional space-invariant hypercube based optical interconnection networks," *Applied Optics*, vol. 32, pp. 7200-7209, 1993.

[16] A. Louri and S. Furlonge, "Feasibility study of a scaleable optical interconnection network for massively parallel processing systems," *Applied Optics*, vol. 35, pp. 1296-1308, 1996.

[17] C. B. Kuznia and A. A. Sawchuk, "Time Multiplexing and Control for Optical Cellular-Hypercube Arrays", *Applied Optics*, vol. 35, pp. 1836-1847, 1996.

[18] J.-F. Lin and A. A. Sawchuk, "Optoelectronic communication speedup on mesh processors using reduced cellular hypercube interconnections," in *Optical Computing*, vol. 10, 1995 OSA Technical Digest, Optical Society of America, Washington, DC, pp. 269–271, 1995.

[19] Jen-Feng Lin, "Optoelectronic cellular array processor with reduced cellular hypercube interconnections," Ph.D. Dissertation, University of Southern California, Los Angeles, California, 1996.

Chapter 3. Optimization of the topology of the cellular interconnects

3.1. Introduction

The cellular hypercube (CH) is one of the preferred topologies for implementing the long-distance optoelectronic links among PEs in SIMD architectures. This is because the CH, like the hypercube interconnection, has exponentially increasing link distances, which make the number of hops per data shift increase only logarithmically with the array size. Moreover, for optical implementations, the CH is particularly attractive because it is based on a space-invariant interconnect pattern.

The mathematical details of our approach in designing the optimal CH topology are described in more detail in a previous publication [1] and in Appendix I. The goal is to preserve the desirable properties of the CH while improving the properties that are sub-optimal. For this, we allow the connection distances to take any values that may optimize the performance of the OCI, in terms of reduced fan-out, reduced number of clock cycles and maximum throughput. At the same time, we preserve the main strong points of the CH interconnect: the space-invariance of the optical interconnect and the logarithmic increase in the number of clock cycles with the array size.

3.2. Definitions and notations for the optimal topology

The array of processors discussed here is a one-dimensional (1-D) or two-dimensional (2-D) mesh-connected array of processing elements (PEs). An example of a 1-D array interconnection is shown in Fig. 3.1, with optical links indicated by the curved arrows. For the 2-D case, the mesh has electrical bi-directional interconnections (not shown) with neighboring PEs left, right, above and below. We initially assume that the optical

free-space connections are made symmetrically at distances $\{\pm X_{\text{rows}}(1), \ldots, \pm X_{\text{rows}}(K_{\text{rows}})\}$ along rows and $\{\pm X_{\text{columns}}(1), \ldots, \pm X_{\text{columns}}(K_{\text{columns}})\}$ along columns (in a grid fashion). In later discussions we drop the symmetry requirement. We define positive link distances as those measured to the right (or upwards) of the transmitter node, and negative distances as those measured to the left (or downwards). We denote by K_{rows} and K_{columns} the one-sided number of links, on rows or columns respectively. For a 1-D array, the fan-out is $2K$, while for a 2-D array the fan-out is $4K_{\text{rows}}K_{\text{columns}}$. For conciseness, we may also refer to K as the fan-out in a generic way, since the fan-out directly corresponds with the number of links K. For the CH the connection distances are $X(k) = 2^k$, but here we allow any values of $X(k)$ that would optimize the design. We show that the 2-D OCI design is separable on rows and columns, so we may find it advantageous to have connection distances that are different (possibly with different fan-out) for columns and rows. There are no wrap-around connections at the edges of the array. Connections falling outside the array are lost.

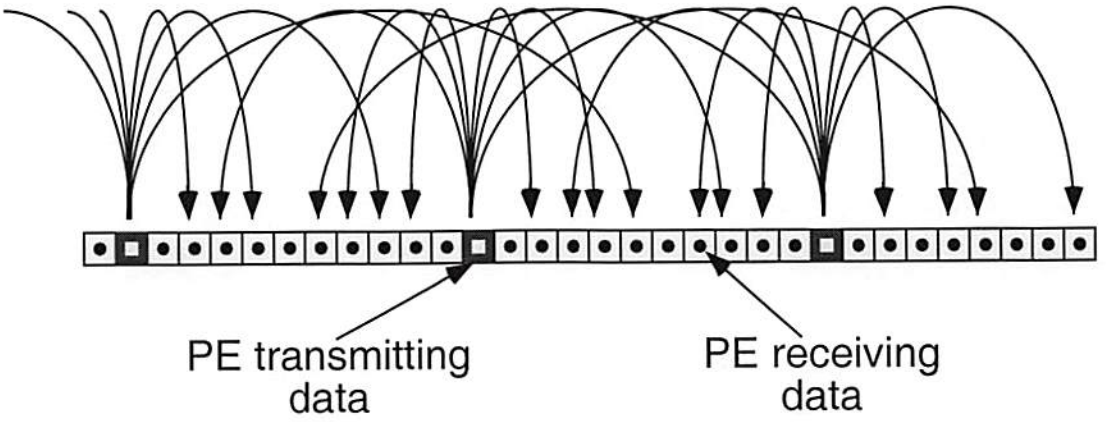

Figure 3.1. One-dimensional cellular array, showing the PEs and the optical links (shown as curved arrows). Every 11-th PE actively transmits data, and all the others receive data. Electronic links (not shown) are made between adjacent PEs. The optical connections are made to neighbors at distances $\{\pm 2, \pm 4, \pm 8, \pm 16\}$. Some of the connections may fall outside the array, but no contention occurs.

As explained in the previous chapter, a time slotted multiplexing algorithm has been demonstrated [2, 3] to solve the problem of contention. The PEs in the array are partitioned into M disjoint sets, such that all the PEs in one set have addresses with the same remainder modulo M. Processors at positions ($l\,M\,+n$), with l and n integers, are all in set n. In any given time slot, only the PEs in a single one of the M sets are active, and M is chosen so that no contention occurs among PEs in the same set. This solves the problem of contention, but increases latency. Effectively, an optical hop takes M clock cycles, as opposed to a single clock cycle for an electronic hop. Thus for short distances the mesh can outperform the optical interconnect, since no time slotted access is required [3]. Additionally, minimizing M minimizes the overall latency of the interconnection.

For ease of understanding, our design procedure, as detailed in [1], is explained as two separate optimization steps: the minimization of fan-out usage and the minimization of the number of clock cycles per data shift. The actual algorithm combines the two optimization methods in a simultaneous application, to achieve an overall optimum.

3.3. Designing for optimum fan-out

To optimize the fan-out usage we require logarithmic dependence of the number of clock cycles per data shift on the array size. In deriving the optimal link distances, the only assumption is that the processors are partitioned into M disjoint sets. The derivation is independent of the actual details of the partitioning mechanism, and is only conditioned by the time-slotted protocol, which makes the data-transfers over optical links require more clock cycles than data-transfers over electrical links.

For preserving the hypercube behavior at large distances, we require that the number of optical hops per shift distance increases no faster than logarithmically with the array size. For this, if the optical interconnect fan-out is K, we impose a limit of K on the number of optical hops. We choose to allow at most S hops through the electronic mesh in addition to

the K optical hops, where S will be chosen to optimize the performance of the OCI. Since each optical hop takes M clock cycles (because of the time slotted scheduling) and each electronic hop takes one clock cycle, the maximum number of clock cycles that a data transfer can take is $MK + S$, i.e. K optical hops and S electronic hops.

For a given set of links $\{\pm X(1), \cdots, \pm X(K)\}$, the speedup of the interconnect can only be achieved for data transfers over distances shorter than some distance $D(K)$. Beyond that point, the performance improvement will degrade, because the number of clock cycles will become again linear with the shift distance. To optimize data shifts beyond $D(K)$, another longer link must be introduced, thus increasing the fan-out (Fig. 3.2). For optimal usage, the optimality regions of two adjacent optical links must exactly touch without overlapping. From this condition, we calculate the optimal distances using an incremental procedure, similar to mathematical induction.

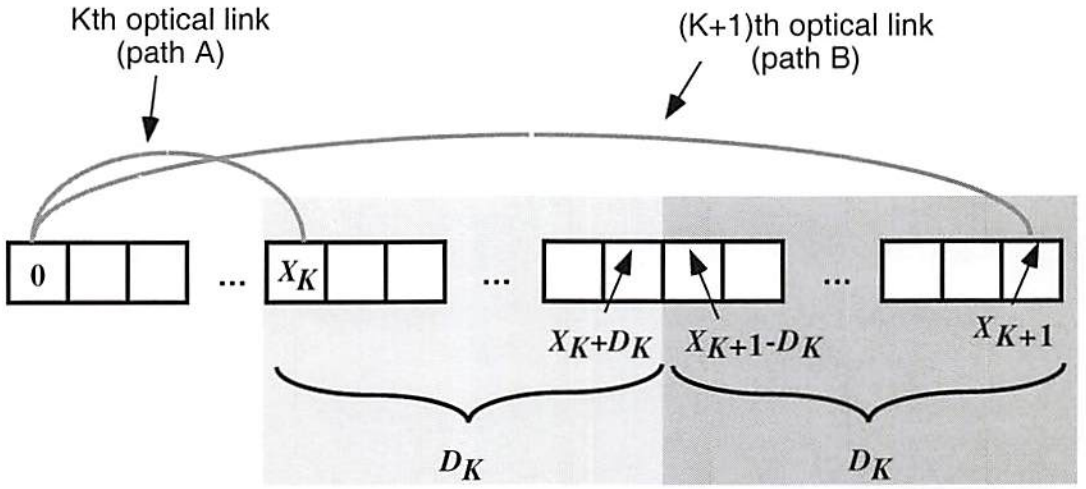

Figure 3.2. Optimality regions for two adjacent optical links. The optimality distance to the left of a newly introduced link (path B) and to the right of the next shorter link (path A) must touch without overlapping to achieve optimal coverage with the given fan-out.

We show in [1] and Appendix I that the optimum connection distances are given by the second order recurrence relation

$$X(n+1)=4X(n)-X(n-1),\ n>1 \tag{1}$$

with initial conditions

$$X(0)=M, \tag{2}$$

$$X(1)=M+2S+1 \tag{3}$$

Equation (1) can also be solved explicitly for the link distances. The n–th distant link of a node is then located at a distance

$$X(n)=\frac{1}{2\sqrt{3}}\left\{\left[M\left(\sqrt{3}-1\right)+2S+1\right]\left(2+\sqrt{3}\right)^{n}+\left[M\left(\sqrt{3}+1\right)-2S-1\right]\left(2-\sqrt{3}\right)^{n}\right\},\ n\geq 0 \tag{4}$$

which shows that the optimal connection distances increase exponentially with n, as for the case of the CH. However, the exponential increase here is faster than for the CH, because the base is $2+\sqrt{3}$, rather than 2. This is a significant factor in speeding up the communications, since links allow data to be transferred over much larger distances in fewer steps. We will show that there are other additional factors that further improve the speed.

From the derivation in [1], the maximum jump size that can be serviced using a given fan-out K and connection pattern $\{\pm X(1),\cdots,\pm X(K)\}$ is

$$D(K)=\left\lfloor\frac{3X(K)-X(K-1)-1}{2}\right\rfloor \tag{5}$$

where $\lfloor x \rfloor$ is the largest integer smaller than x. Since this maximum jump distance $D(K)$ depends linearly on $X(K)$, it will also increase exponentially with the fan-out.

Alternatively, the required fan-out increases as $K \propto \log_{2+\sqrt{3}}(D)$ for a maximum shift distance of D.

Clearly, S allows trading off fan-out for number of clock cycles. Using a non-zero S increases the number of clock cycles, but also increases the connection distances over which the OCI is optimal. We can thus fine-tune a design, trading off latency for fan-out.

So far we have determined the conditions for optimal usage of the given fan-out. The next step is to minimize the number of partitioning sets M, such that there is no contention between transmitting PEs.

3.4. Designing for minimum latency

The problem of minimizing the latency has two solutions, depending on the restrictions imposed on the connection pattern. In most optical interconnection systems, connection patterns are symmetric, due to the lower number of degrees of freedom, which makes the design of optical elements easier. Another advantage of symmetrical designs is the optimum use of resources; it is always easier to use a bi-directional interconnect if the pattern is symmetrical. Nonetheless, we found that the partitioning of the PEs in an OCI is more efficient if a non-symmetrical pattern is used.

3.4.1. Symmetrical connection patterns

To minimize the latency, i.e., find the minimum number of partitioning sets M, we define the set of connection distances as $\{\pm X(k)\}, k = 1,\cdots,K$, where $2K$ is the fan-out (for two-sided connections). Assume that the processors in the array are divided in sets, such that all processors in one set have the same remainder modulo M. Thus, processors at positions $lM+n$, with l integer will all be in set n. All the processors in one set are spaced at distances that are multiples of M. Two processors contend for the same receiver if two link distances $X(m)$ and $X(n)$ can be found that satisfy

$$X(m) \pm X(n) = lM,\ l \in Z,\ m, n \in \{1, 2, \ldots K\} \tag{6}$$

If this condition is satisfied, any two processors separated by a distance lM with l integer contend at one receiver. Thus, in order to avoid contention, we must ensure that

$$\big(X(m) \pm X(n)\big) \bmod M \neq 0,\ m, n \in \{1, 2 \ldots K\} \tag{7}$$

which can only happen if each of the $\pm X(m)$ has a different non-zero remainder modulo M. The remainder must be non-zero: if $X(m)$ has a zero remainder then $-X(m)$ will also have a zero remainder and the two will cause contention. This occurs because if $X(n) \bmod M = 0$, then

$$[-X(n)] \bmod M = \big(M - \{[X(n)] \bmod M\}\big) \bmod M = M \bmod M = 0 = [X(n)] \bmod M.$$

Thus, there must be at least $2K$ non-zero and distinct remainders modulo M. It is now clear that this cannot be satisfied unless

$$M \geq 2K + 1. \tag{8}$$

For optimal results, the inequality above becomes an equality in the form

$$M = 2K + 1. \tag{9}$$

We can always find a set of link distances that would eliminate contention, because there are 2K distinct remainders and 2K link distances $\{\pm X(k)\}, k = 1, \cdots, K$.

At this point we define an optimization problem. Each optical hop takes M clock cycles due to the time slotted protocol; hence, for a low latency, the number of partition sets M should be small. On the other hand, a large number of optical links (a large fan-out) would reduce the number of optical hops. Unfortunately, as Eq. (8) demonstrates, a small M and a large K are conflicting requirements. On the other hand, a small fan-out is desirable for

the ease of design of the diffractive optical element (DOE) for the interconnection. Conversely, as we will show later, a large M guarantees that a large fraction of the PEs in the array can be active at the same time.

One interesting consequence of this design strategy is that the maximum array size is not limited to the maximum jump size. If a larger array is used, the fan-out limits only the maximum jump size, $D(K)$, (after which the number of hops will no longer be optimal). If the desired maximum communication distance is $D(K)$, arrays larger than $D(K)$ will be able to use the same number of sets for partitioning the processors, without having to worry about contention. Considering that for an array of size N, the maximum communication distance will most likely be less than $N/2$ (maybe significantly less), the array size has been completely eliminated from among the design parameters, making the whole design problem more robust.

As a final word, the overall percentage of processors that are active is 100% (we neglect the edge effects, due to optical links that fall outside the array). In each time slot, PEs in one set transmit data (this is a fraction of $1/M$ of the processors in the array). Since $2K = M - 1$ connections are made from each processor and no contention occurs, the total fraction of processors that receive a transmission is $(M-1)/M$. Overall, all processors are busy, either transmitting or receiving. In the next section, we show that if the connection pattern is non-symmetric, the usage can be increased to the limit, in which all processors are receiving data, while some PEs are also transmitting data at the same time.

3.4.2. Non-symmetric patterns

If we allow a non-symmetric fan-out pattern, the connection distances modulo M can further be optimized. We discussed briefly the advantages of symmetrical patterns, so we try to keep at least part of the optical links symmetrical. In fact, if non-symmetrical link

distances are used, the performance of the interconnection may be very different for data shifted to the left or to the right, so we want to minimize the asymmetry. To further improve on the limit in Eq. (8), we allow one link distance to have a remainder of zero, modulo M. In this case, we can reduce M for a given K to $M = 2K$. There is a price to be paid for this: two of the connection distances will not be symmetrically paired as $\pm X(k)$. Instead, we will choose connection distances such that one of them will have a remainder of 0 modulo M, while the other one will have a remainder of K modulo M. All the other connections will have remainders that are all different from each other, with values in the set $\{1, 2, \cdots, K-1, K+1, \cdots, M-1\}$, as in the symmetric case. Overall, all the connection distances will have distinct remainders, which will ensure that no collisions can occur.

It is clear now that the overall percentage of processors that are receiving data is 100%, because in any given time slot, every one of M processors transmits to $2K = M$ others and no contention occurs. As for the symmetric case, the total fraction of processors that transmit data is $\frac{1}{M}$. This means that $\frac{1}{M}$ of the processors are transmitting and receiving data at the same time. This may pose a problem for the power dissipation of the optoelectronic PE array, and possibly affects the architecture of the PE itself, but would certainly increase the throughput of the network.

3.5. Designing for overall optimization: OCI

We presented in the previous two sections two separate algorithms: one for optimizing the fan-out and one for optimizing the partitioning of the PEs in a SIMD machine with cellular interconnects. The fan-out optimization assumes that the PEs are partitioned into M sets such that there is no contention among PEs in the same set. Conversely, the partitioning optimization shows how to achieve the partitioning without contention. Analysis of various OCI patterns shows that the class of interconnection sets that are

simultaneously optimal in terms of fan-out usage and of minimum latency is very limited. A hybrid approach needs to be used to achieve the overall optimum.

As a hybrid algorithm that simultaneously optimizes both the fan out and the latency, we build the connection set incrementally, in a manner consistent simultaneously with fan-out optimization and with contention avoidance. The resulting connection set is slightly sub-optimal in terms of fan-out usage, but ensures that no contention can occur.

3.5.1. Symmetric connection patterns

The flow diagram for the algorithm used to calculate the optimal distances for symmetrical connection sets is shown in Fig. 3.3. For a given number of optical hops K, we let the number of partitioning sets be $M = 2K + 1$ and start assigning connection distances to $X(n)$ from n=1. The first link distance is given by Eq. (3). We then calculate the following link distances using Eq. (1). At each step we ensure that the newly introduced link distance has a remainder which is non-zero, as well as distinct from the remainders modulo M of all the previous link distances. If this condition for the remainders is not satisfied with the value for $X(n)$ from Eq. (1), we decrement $X(n)$ till the condition is satisfied. The fan-out usage will thus be slightly sub-optimal, but the procedure will allow a global optimization: the resulting connection set will be the set that is closest to optimal fan-out usage and at the same time contention-free.

3.5.2. Non-symmetric connection patterns

For non-symmetrical connection patterns, a further improvement in the OCI performance can be achieved, both in reducing the fan-out and in reducing the number of clock cycles per data shift. To minimize the effects of the non-symmetry, we found out that the best connection patterns are symmetrical in the first K-1 links and only non-symmetrical in the K-th optical link distance. If the asymmetry is introduced at lower order

links, then Eq. (1) would make the higher order link distances extremely asymmetric, and hence would severely degrade the performance of the interconnect.

The flow diagram for non-symmetrical connection sets is shown in Fig. 3.4. For non-symmetric patterns we can use the lower value of partitioning sets, $M = 2K$ for K optical hops. As in the case of the symmetric patterns, we start assigning connection distances to $X(n)$ from n=1. The first link distance will again be given by Eq. (3). As before, we calculate the following link distances using Eq. (4), but now we need to ensure that the

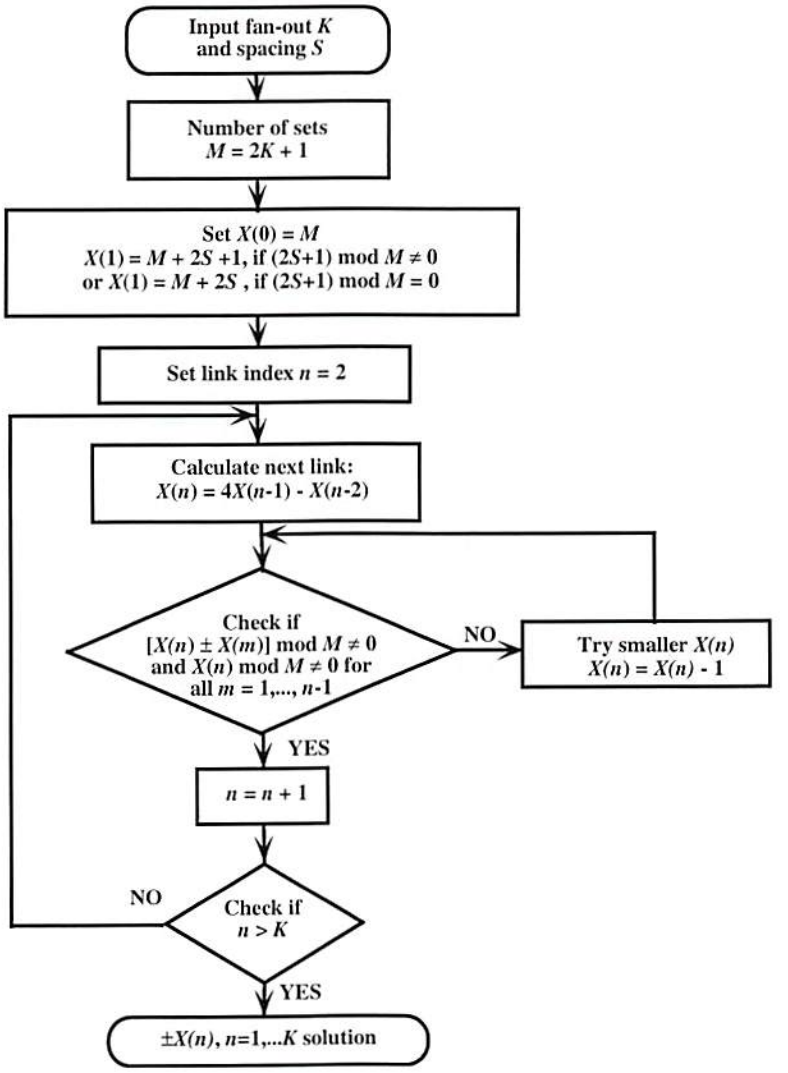

Figure 3.3 Flow-chart for the algorithm for symmetrical OCI interconnections.

newly introduced link distance has a remainder which is both non-zero modulo K and distinct from the remainders modulo M of all the previous link distances. Only the K-th link distance will be allowed to have a remainder whose value is non-zero modulo K. The condition for the non-zero remainders modulo K is equivalent to non-zero remainders modulo M and modulo $M/2$. This ensures that the symmetrical part of the optical interconnect (links 1 through K-1) can in fact have symmetrical values, while obeying the collision avoidance condition Eq. (7). The non-symmetrical part of the connection pattern will only be at the K-th link distance. One side of the link (say the positive $X_+(K)$) will have a remainder which is zero modulo M, while the other side (say the negative link $-X_-(K)$) will have a remainder which is K modulo M. Thus all the link distances have distinct remainders modulo M, including the values 0 and K. Overall, the contention avoidance condition Eq. (7) holds, so the pattern is contention-free. As in the symmetrical case, if the condition for the remainders is not satisfied with the value for $X(n)$ from Eq. (4), we decrement $X(n)$ till the condition is satisfied. The fan-out usage will again be slightly sub-optimal, but globally optimal.

3.5.3. Using the OCI design algorithm

Based on the algorithms above, we generated a table of connection patterns given an allowed number of optical hops (K) and electronic hops (S). For each pair of K and S, the maximum jump size can also be determined and tabulated. To design an OCI for a given array size, the user looks up the K and S pair with a corresponding jump size entry exceeding the desired one. The user can then simply look up in the table the connection set that meets the design constraints. We found that using only 6 optical hops and 12 electronic hops is sufficient to cover distances exceeding 2^{15}, which should be more than enough for practical purposes. The fan-out values for the OCI are significantly lower than for an M-RCH of the same size, because the exponential increase in the OCI has a base

$2+\sqrt{3}$, as opposed to the base 2 for the M-RCH. The number of clock cycles required to shift data is also reduced, due to the optimized partitioning described above. More detailed performance benchmarks are presented in Section 3.9.

As a side note, the optimal designs for the M-RCH also conform to the conditions we derived. The M-RCH connection patterns satisfy the same contention avoidance conditions (7) and (9), but even the best ones are severely sub-optimal in terms of fan-out usage.

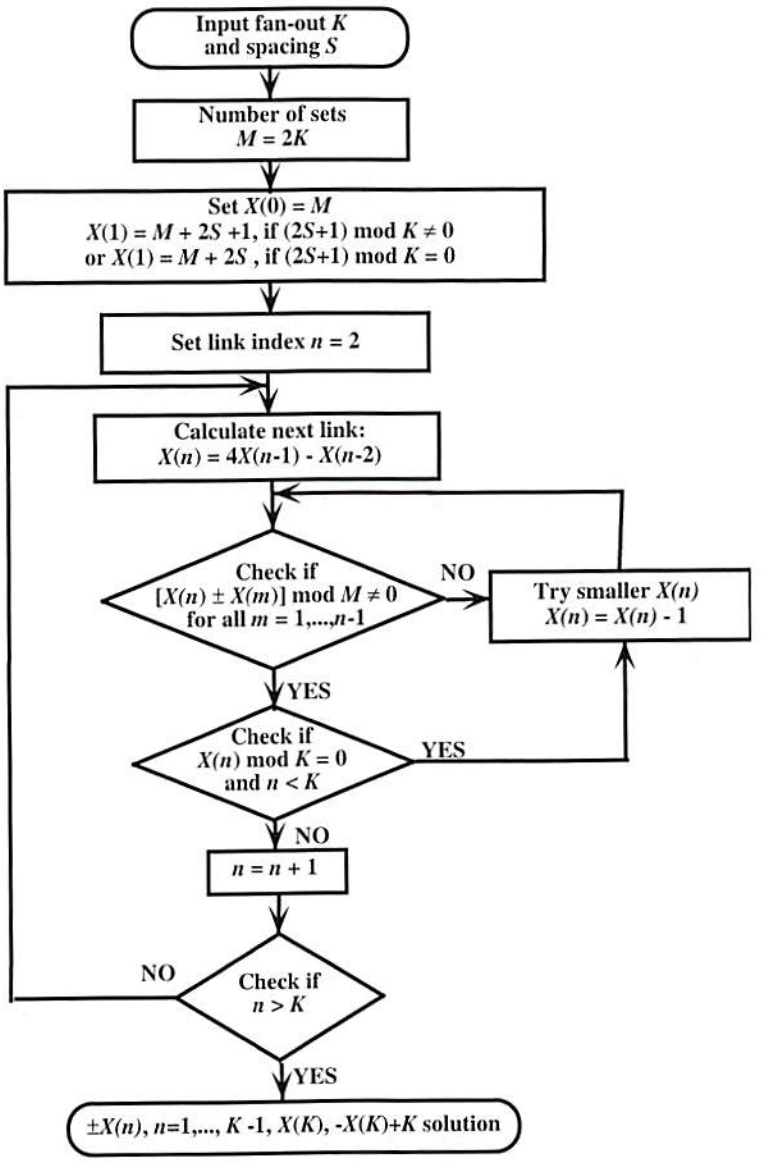

Figure 3.4. Flow-chart for the algorithm for non-symmetrical OCI interconnections.

Finally, the algorithm for decoding the incoming transmission is identical to the algorithm for the CH [3]. If the PE at address P must receive a data shift made over a link distance $X(n)$, the data will arrive in a time slot t given by

$$t = (P - X(n)) \bmod M \tag{10}$$

This relation is simple enough to be implemented efficiently and with a low usage of device area in a very large scale integrated (VLSI) SIMD machine.

3.5.4. Comparison of symmetric and non-symmetric patterns

We plot in Fig. 3.5 the dependence of the number of clock cycles needed to transfer data in an SIMD array, as a function of the maximum shift distance, $D(K)$, and the number of

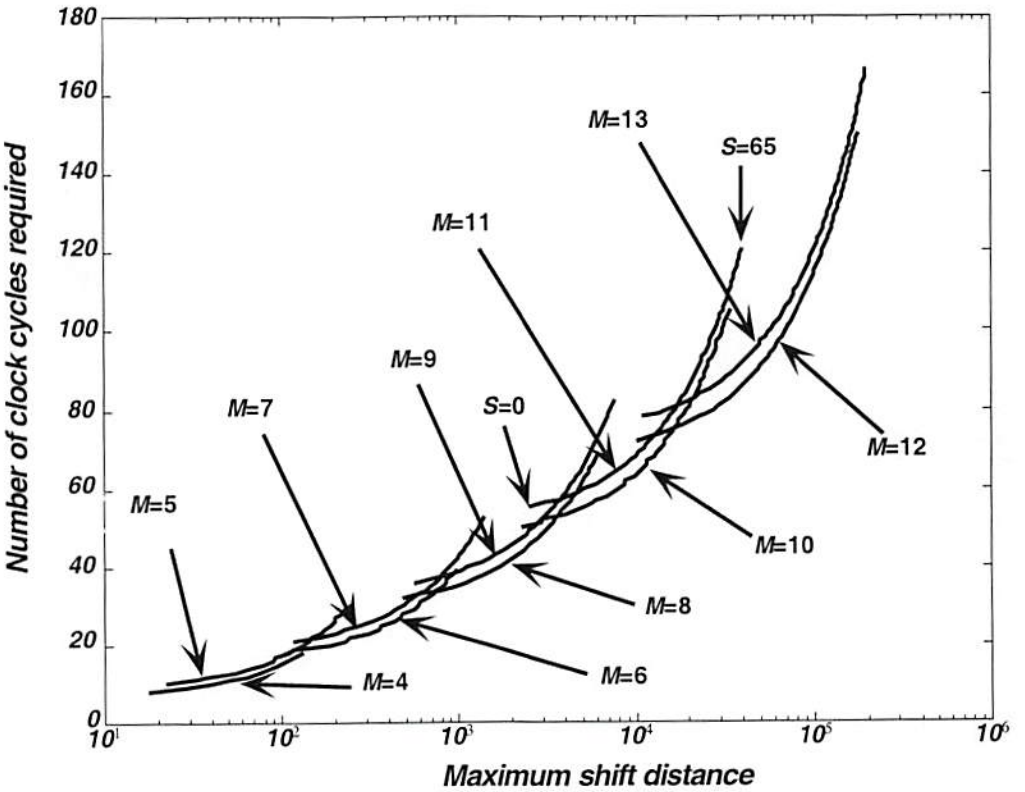

Figure 3.5. Number of clock cycles versus maximum shift distance (for optimum usage of available fan-out), for OCI of different fan-outs.

partitioning slots, M. For a constant M, the maximum shift distance is an implicit function of S and is increasing with increasing S. Thus, the lowest value of S for each constant-M curve is $S = 0$. The highest plotted value of S is chosen to provide a continuous and optimal coverage for the full range of clock cycles and shift distances on the axes of the figure. Clearly, as the maximum shift distance increases, a fixed number of partitioning sets M can be used, but the interconnect becomes less and less effective (the number of hops increases linearly with the distance beyond a certain shift distance). Upon incrementing the number of partitioning sets, the rate of increase of the number of clock cycles is reduced, but it increases again as the maximum shift distance increases further. As an example, in Fig. 3.5, for M=11 we indicate both the starting value $S = 0$, and the ending value plotted, $S = 65$, even though there are distances for which M=13 is clearly a better choice for reducing the maximum number of clock cycles.

It is also clear that an even value and the next higher odd value of M cover relatively the same range of connection distances, but the even M (non-symmetrical connection pattern) has a slightly lower number of clock cycles than the odd M (symmetrical connection pattern). The difference in number of clock cycles between the symmetrical and the non-symmetrical patterns with the same fan-out is approximately equal to K. The non-symmetrical pattern offers lower clock cycles and lower fan-out, but it may be less desirable in practice, where symmetrical systems are easier to design and comprehend. Note that with a fan-out of only $K = 6$ ($M = 12$ or $M = 13$ in Fig. 3.5), arrays of size up to 10^5 (1-D) or 10^{10}(2-D) could in principle be interconnected.

3.6. Two-dimensional arrays

Two-dimensional arrays can be viewed as separable outer products of two one-dimensional arrays (Fig. 3.6). The connection patterns developed for each of the dimensions of the array (viewed as a one-dimensional array) can now be combined to

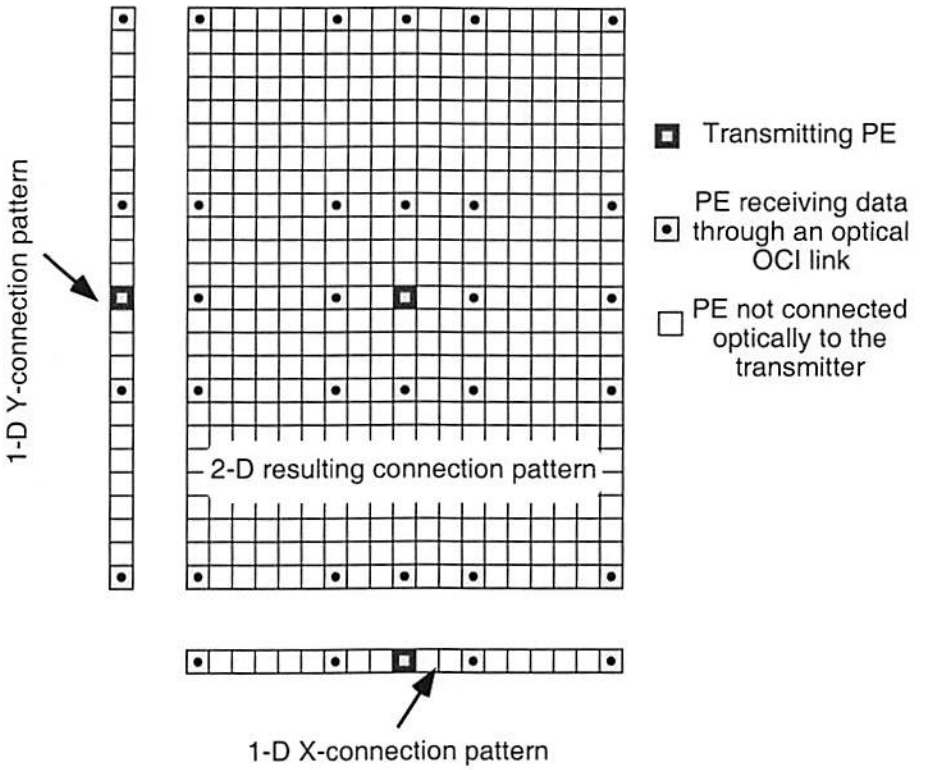

Figure 3.6. Two-dimensional interconnection pattern, shown as a product of two one-dimensional patterns.

obtain the two-dimensional result. This way, each transmitting processor can broadcast to a 2-D array of processors, rather than only to a row and a column, as in the case of the CH. This is because the coordinates of each processor are separable; a one-dimensional contention-free solution in each dimension ensures lack of contention for the combined solution. Moreover, if the usage is 100% in each dimension, the 2-D usage is 100% as well (this is possible in the case of non-symmetrical patterns). Extension of this formalism to higher dimensions is trivial, while also rather academic, since there is no known way to interconnect a three-dimensional array efficiently and with a large fan-out.

3.7. Extensions of the OCI

If more than one detector per pixel is used, the design can achieve even more flexibility in avoiding contention. In this case, if a pixel has d detectors, it is able to receive

simultaneously d incoming data streams, without contention. This higher number of detectors per pixel reduces the number of partitioning sets needed. Clearly, for the symmetric case M-1 detectors are enough to ensure contention-free reception with all PEs in the same partitioning set. This is because in the M-slotted protocol each processor acts as a transmitter in one out of M time slots and as a receiver in the rest of M-1 slots. If we make all these receptions simultaneous, we would need M-1 detectors to differentiate the incoming signals. Similarly, in the case of the non-symmetric connections, each processor receives data in each of the M time slots, so at most M detectors are needed to ensure contention-free operation without a time slotted protocol.

Based on the number of detectors per pixel, we can envision another level of tradeoff in the OCI design, where a higher number of detectors ($d \geq 1$) and a lower number of time slots can be traded off, depending on the system requirements. We assume that the PEs are partitioned into M sets and that each pixel has d detectors, and thus can receive d simultaneous data streams. Each partition set would now change dynamically to include in time slot m the processors with addresses of the form $lM + md + n$, with l, m, n integers and $n = 0, \cdots, d-1$. The period of the time slots is the least common multiple of the number of partitioning sets (M) and the number of detectors (d). In each time slot a fraction d/M of processors are active, so that each PE will be active every M/d clock cycles. The overall latency is thus reduced from M to M/d. Depending on the values of M and d, a lower latency can be traded off for a higher number of detectors per receiver.

3.8. Edge effects

The discussions above have assumed that the array size is much larger than the length of an optical hop. Alternatively, the assumption is that the optical links that fall outside the array do not have an important effect over the performance of the interconnection. In truth,

many of the shift operations rely on a hop in one direction (for example towards west) followed by a shorter hop in the opposite direction (for example towards east). If the first hop lands outside the array, the overall transfer is no longer feasible.

As detailed in Appendix I, a simple modification of the algorithms can be made to take these effects into consideration. Starting with the optimal fan-out considerations, the optimality regions to be considered must be only in the direction of the overall transfer (for example, if data is to be transferred to the west, no east shifts are allowed). With this small modification, if a link lands outside the array, it is because a shift over that distance would have lost the data anyway.

As shown in Appendix I, the recurrence relationship in Eq. (1) changes to

$$X(n+1) = 3X(n) - X(n-1),\ n > 1, \tag{11}$$

but the initial conditions remain identical to Eq. (2)-(3). The solution of the recurrence relation is of the form

$$X(n) = \left\{ \left[\frac{M}{2} + \frac{2S+1-\frac{M}{2}}{\sqrt{5}} \right] \left(\frac{3+\sqrt{5}}{2} \right)^n + \left[\frac{M}{2} - \frac{2S+1-\frac{M}{2}}{\sqrt{5}} \right] \left(\frac{3-\sqrt{5}}{2} \right)^n \right\},\ n \geq 0 \tag{12}$$

showing that the link distances increase indeed slower than in Eq. (4). The latency optimization is not changed by the assumption of unidirectional data transfer in the array, which makes the algorithms for overall optimality remain identical, except for the new form of the recurrence relation.

The comparison we show in the sections on the simulation results is always done considering the recurrence relation in Eq. (4), because all the M-RCH simulations in the literature neglect the edge effects.

3.9. Simulation results for the OCI performance on basic operations

Using the optimal connection sets defined in the previous section, we determined the number of clock cycles needed to shift data in 1-D arrays of various sizes, for both symmetrical and non-symmetrical connection sets. The results show the advantage of OCI interconnects, both in terms of fan-out reduction and in terms of communication speed up.

3.9.1. Simulation results for large arrays

We first simulated the performance of large arrays of PEs, where the advantages of the OCI are expected to be more evident. Indeed, for arrays of size 4096 (Table 3.1), the fan-out required is reduced by up to 50%, while the maximum number of clock cycles per data transfer is reduced by up to 30% and the mean number of clock cycles is reduced by 25%. The optimal design is for $K = 4$, where both the fan-out and the number of clock cycles are reduced by 30%.

Topology	K	S	M	Connection set	# of clock cycles	
					Maximum	Mean
M-RCH	6	n. a.	13	$\pm 2^6$, $\pm 2^7$, ..., $\pm 2^{11}$	84	49.5
OCI-SF	3	39	7	±86, ±337, ±1257	74	43.7
OCI-NF	3	38	6	±83, ±326, 1221, -1218	68	40.0
OCI-SC	4	23	9	±56, ±215, ±804, ±3001	59	40.2
OCI-NC	4	24	8	±57, ±219, ±818, 3052, -3048	56	37.4

Table 3.1. Performance comparison of OCI and M-RCH for arrays of 4096 PEs. The flavors of the OCI interconnect are: S - symmetric, N - non-symmetric, F - optimized for minimum fan-out, C - optimized for minimum number of cycles.

The data in Table 3.1 demonstrate that the non-symmetric OCI (OCI-N) fares better than the symmetrical OCI (OCI-S), if only slightly. For this case, the performance advantages of the OCI-N are not clear enough to warrant its use, given the complications of non-symmetric interconnect patterns. Figure 3.7 shows a comparison of the histograms

for number of clock cycles per data shift for the M-RCH and the OCI-N with $K = 4$ for an array of 4096 PEs, showing one data point for each shift distance from 1 to 4096. The OCI-N histogram is clustered at much lower values than for the M-RCH. Also, most shifts require 30 to 40 clock cycles, as opposed to the M-RCH, which most often requires 40 to 60 clock cycles.

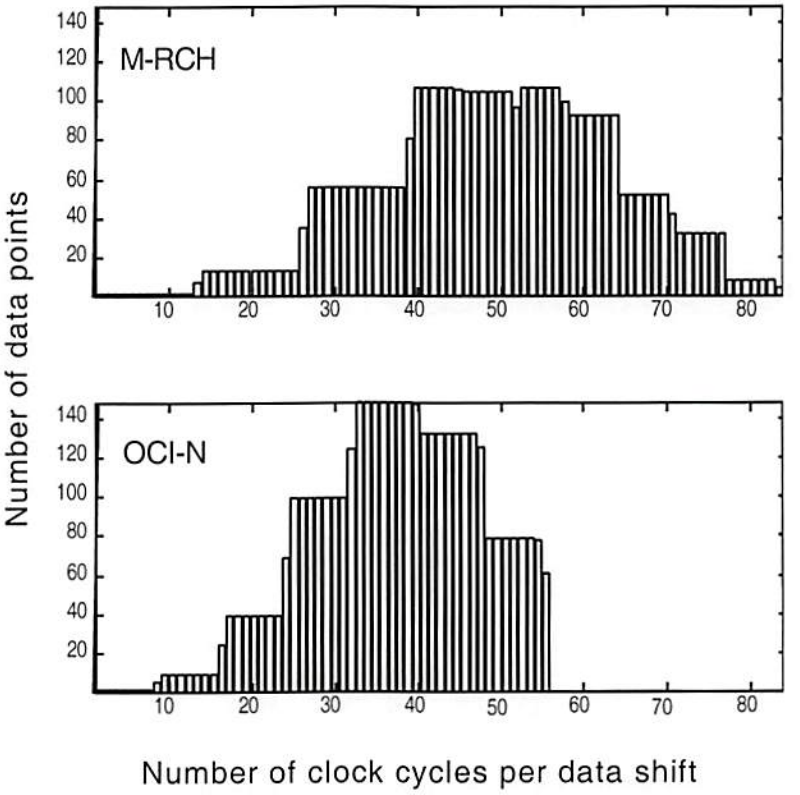

Figure 3.7. Histogram of the number of clock cycles per data shift in a 4096 processor array, with one data point for each shift distance from 1 to 4096. OCI optimized for reduced number of clock cycles. Connection sets as in Table 3.1 for M-RCH and OCI-N (OCI non-symmetric) with $K = 4$.

A different view of the same results is shown in Fig. 3.8. Here we plot the actual number of clock cycles per data shift, function of the shift distance, for the M-RCH and for the OCI optimized for reducing the number of clock cycles. It is clear that the OCI performs better than the M-RCH for most of the shift distances.

Table 3.1 also shows another solution, with an even lower fan-out, $K = 3$. This design is sub-optimal, in that the number of optical hops is not equal to K. In fact, for this design, the optimality distance is much smaller than the array size ($D(K) = 1716$ for OCI-S and $D(K) = 1663$ for OCI-N, while the array size is 4096). For this design, up to three hops will be taken over the longest optical links, for a total of five optical hops, in order to shift data over the array length; for comparison, an optimal OCI with this fan-out would only allow a total of three optical hops. Even though the design is non-optimal for OCI standards, it clearly outperforms the M-RCH. As seen in the histograms in Fig. 3.9, this sub-optimal OCI-N is still more efficient than the M-RCH in reducing the maximum number of clock cycles, even though it requires only *half* the fan-out of the M-RCH.

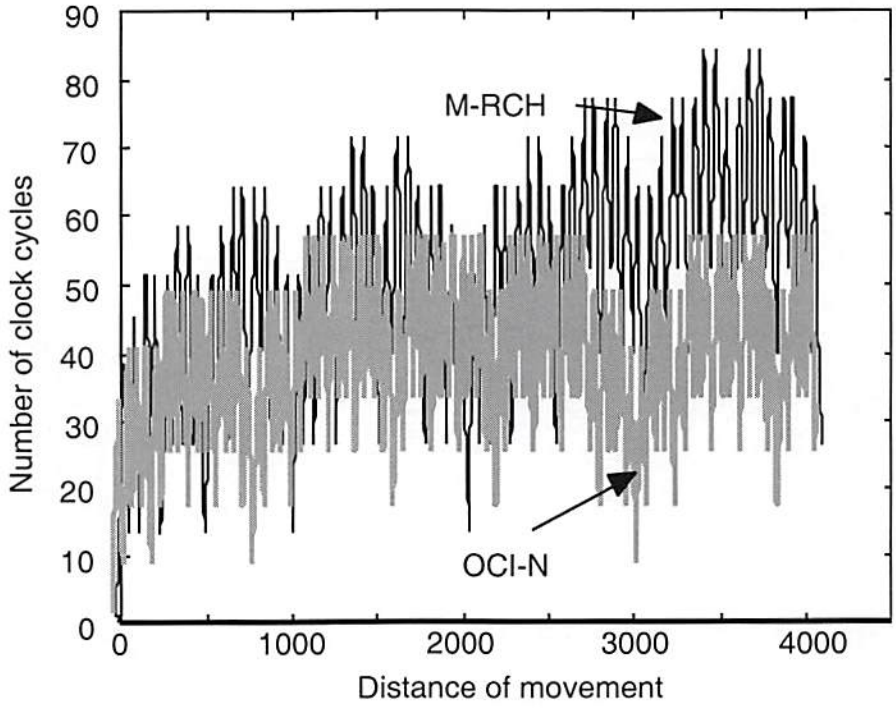

Figure 3.8. Comparison of the number of clock cycles per data shift function of the shift distance in a 4096 processor array. OCI optimized for reduced number of clock cycles. Connection sets as in Table 3.1 for M-RCH and OCI-N with $K = 4$.

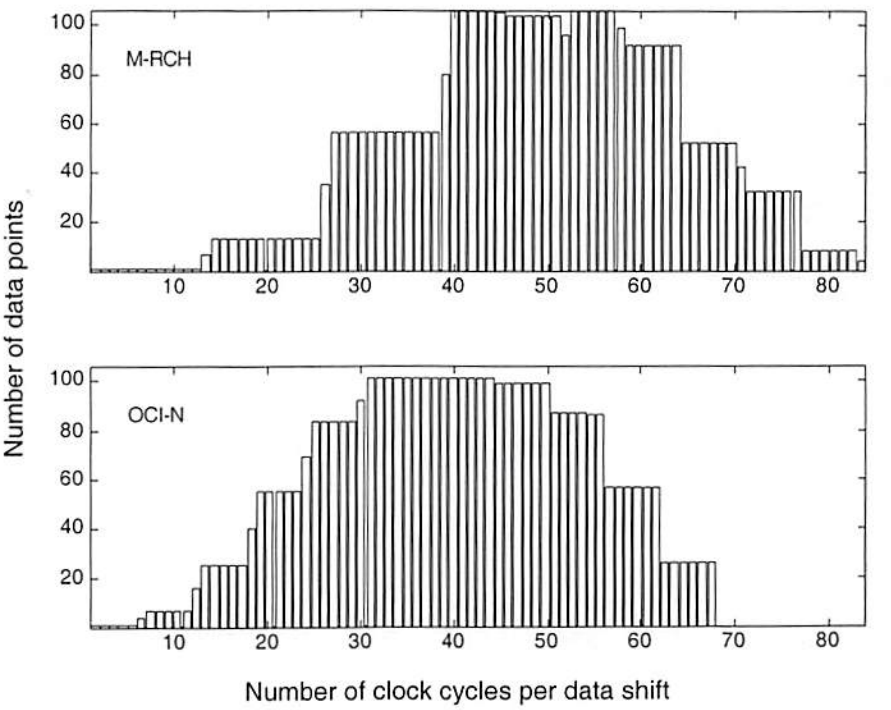

Figure 3.9. Histogram of the number of clock cycles per data shift in a 4096 processor array, with one data point for each shift distance from 1 to 4096. OCI sub-optimal, optimized for reduced fan-out. Connection sets as in Table 3.1 for M-RCH and OCI-N (OCI non-symmetric) with $K = 3$.

3.9.2. Simulation results for smaller arrays

For smaller array sizes (Table 3.2), the reductions in fan-out and number of clock cycles are somewhat smaller, but still better than 10%. The histograms in Figures 3.10 and 3.11 show as data points the number of clock cycles per data shift, for data shifts over distances from 1 to 256. Depending on the optimization constraints, the histograms show a possible trade off between mainly reducing the fan-out (Fig. 3.10), or mainly reducing the number of clock cycles (Fig. 3.11). Overall, both clock cycles and fan-out values are reduced for the OCI as compared to the M-RCH. An interesting observation is that the *mean* number of clock cycles is fairly similar for both M-RCH and OCI, even though the *maximum* number of clock cycles for the OCI is significantly lower. This shows that the OCI mainly

redistributes the fan-out, so that the data transfers are done in a more uniform manner (with a number of clock cycles more balanced across various shift lengths).

Topology	K	S	M	Connection set	Number of clock cycles	
					Maximum	Mean
M-RCH	3	n. a.	7	$\pm 2^5$, $\pm 2^6$, $\pm 2^7$	33	19.1
OCI-SF	2	22	5	±49, ±188	32	18.8
OCI-NF	2	22	4	±49, 192, -190	30	17.5
OCI-SC	3	4	7	±16, ±57, ±207	25	17.9
OCI-NC	3	5	6	±17, ±62, 231, -228	23	16.0

Table 3.2. Performance comparison of OCI and M-RCH for arrays of 256 PEs. The flavors of the OCI interconnect are: S - symmetric, N - non-symmetric, F - optimized for minimum fan-out, C - optimized for minimum number of cycles. For example, OCI-SF is a symmetric interconnect, optimized for reduced fan-out.

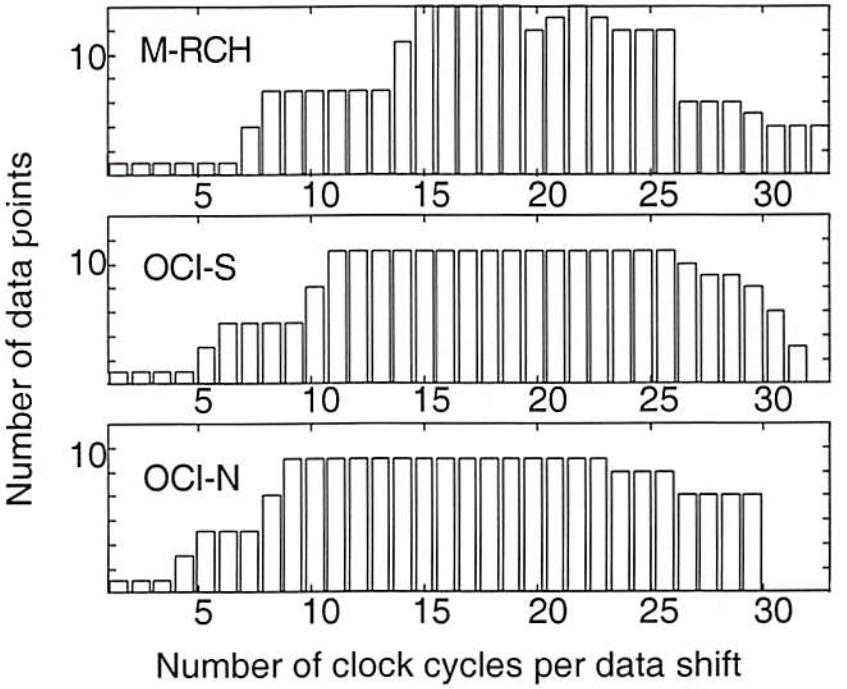

Figure 3.10. Histogram of the number of clock cycles per data shift in a 256-processor array, with one data point for each shift distance from 1 to 256. OCI optimized for reduced fan-out. Connection sets as in Table 3.2 for M-RCH, OCI-S (OCI symmetric) and OCI-N (OCI non-symmetric) with $K = 2$.

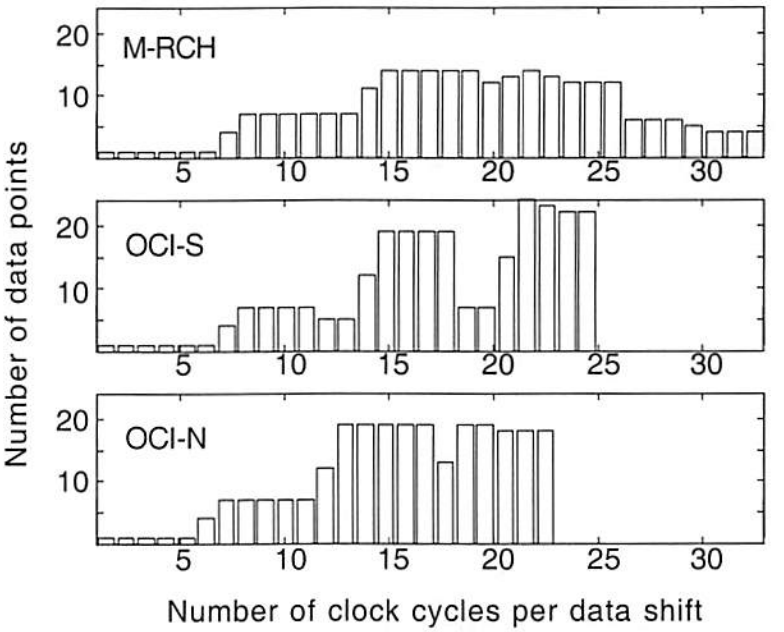

Figure 3.11. Histogram of the number of clock cycles per data shift in a 256-processor array, with one data point for each shift distance from 1 to 256. OCI optimized for reduced number of clock cycles. Connection sets as in Table 3.2 for M-RCH, OCI-S (OCI symmetric) and OCI-N (OCI non-symmetric) with $K = 3$.

3.9.3. Array sizes currently feasible

We show for completeness the simulations results for an array of a size that is practical even with today's technologies (Fig. 3.12). Even though arrays of 256 x 256 PEs have already been demonstrated [4, 5], we limit this 1-D case to a more practical 32 PE array. For this array size, only a single set of parameters K and S can be used. Unlike the cases simulated above, no choice can be made to optimize for fan-out or latency. Using other values for the parameters K and S leads to sub-optimal designs. The simulation results show that the OCI is again better than the M-RCH, with a maximum number of clock cycles 36% lower at equal fan-out ($K = 2$). A summary of the connection parameters and of the performance comparison is shown in Table 3.3.

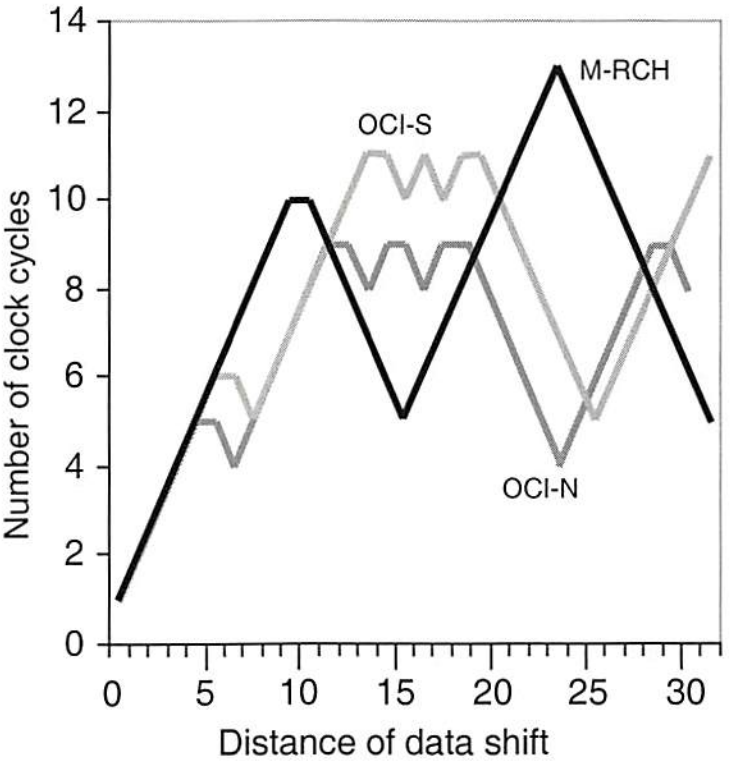

Figure 3.12. Comparison of the number of clock cycles per data shift function of the shift distance in a 1-D 32-processor array. For such small array sizes, there is no possibility to specify the optimization for fan-out or for reduced number of clock cycles). Connection sets are as in Table 3.3.

Topology	*K*	*S*	*M*	Connection set	Number of clock cycles	
					Maximum	Mean
M-RCH	2	n. a.	5	$\pm 2^4, 2^{32}$	13	7.84
OCI-S	2	1	5	±8, ±26	11	7.81
OCI-N	2	2	4	±9, 32, -30	9	6.77

Table 3.3. Performance comparison of OCI and M-RCH for arrays of 32 PEs. The flavors of the OCI interconnect are: S - symmetric, N – non-symmetric.

3.10. Performance quantification of the OCI performance on real-life applications

The M-RCH and the CH have been evaluated on both general-purpose operations and on particular problems, like sorting, FFT, data exchanges, matrix vector multiplication. We showed in the previous sections that the OCI is superior to the other topologies in the general case. In particular applications, the power of the OCI may sometimes be lost. In performing FFT operations or particular types of data exchanges (for example the Perfect Shuffle or the Banyan permutations), data exchanges occur over distances that are powers of two. For such cases, it is clear that the M-RCH will have a strong advantage over the OCI, because shifts over a power-of-two distance can be done using a single optical hop using the M-RCH, but may require multiple optical and electronic hops using the OCI.

Still, in addition to its general-purpose qualities, the OCI may be better suited than the M-RCH for some particular classes of applications. For parallel sorting applications for example, the most used sorting method is the Batcher sort [6], which relies on exchanging elements over a set of binary distances. Less known and used is the optimal sorting algorithms, the Shell sort using the Pratt sequence [6], which exchanges numbers over the set of distances of the form 2^p3^q, with p and q positive integers. Such a sequence could be designed into the OCI interconnection, but not in the M-RCH.

3.10.1. Shell sorting on optically interconnected parallel computers

The attractive feature of the Shell sort for optoelectronically interconnected computer architectures is that it involves multiple exchanges among elements situated relatively far away. Also, unlike the similar Batcher sort, the Shell sort can be stopped after the elements in the array have been sorted, rather than after a prescribed number of iterations.

For the Shell sort, comparisons are made between elements at distances h_i, assumed to be ordered in decreasing sequence, and with $h_1 = 1$. Using larger h_i values allows the

propagation of the comparison results over large distances, speeding up the completion. Starting with the largest distance, comparisons are made and elements are exchanged until no further exchanges are needed. At that point, the next lowest value for h_i is used, until the last one, $h_1 = 1$. At that point, the array has been sorted. Because of the optoelectronic interconnection, all the shifts over the distances h_i can be done using the optoelectronic speedup links, whether in the OCI or in the M-RCH. The comparison is done inside the PE. Each PE receives all copies of the numbers to be compared and only preserves the copy that results from the exchange.

For example, when two PEs are comparing elements, they each broadcast their keys to the PEs with which they compare. If comparing with another PE to the right, a PE will keep the smallest value. Conversely, when comparing with another PE to the left, a PE will keep the largest value. This is equivalent to an exchange of the values if the elements are not sorted.

We propose to perform the Shell sort using a sorting cell with three inputs and three outputs (Fig. 3.13). Because the PEs operate in SIMD fashion, we differentiate between

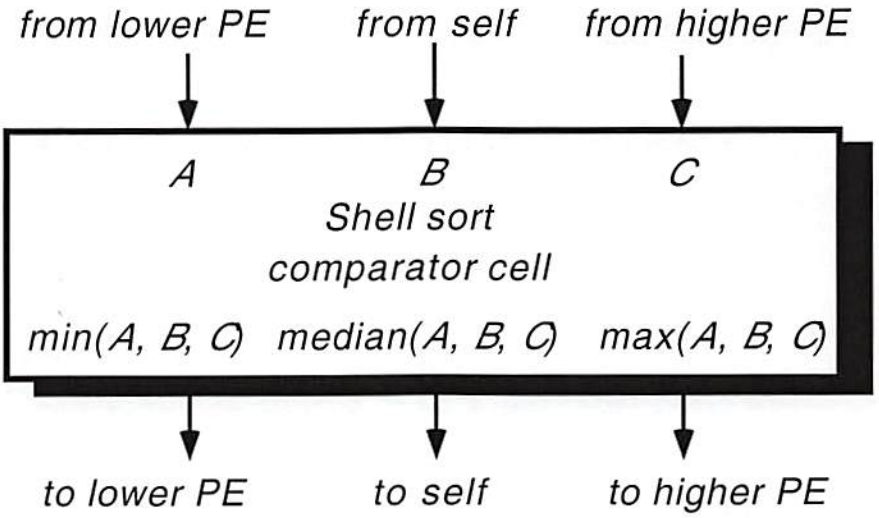

Figure 3.13. Functional diagram of the Shell sort cell. In each sorting step, such cells will have their inputs and outputs connected for PEs separated by the current sorting distance, h_i.

the left and the right elements by storing them in different memory locations. If the array is distributed "horizontally," with the first element at the leftmost end, we can perform a data shift to the right over a distance h_i and store the shifted value in a location called "low." This will be used as the input from the "lower" PE in the array. Similarly, we perform a data shift to the left over a distance h_i and store the shifted value in a location called "high." This will be used as the input from the "higher" PE in the array. In each iteration, the low, high and the PE's own element will be compared. The highest value will be the one shifted to "higher" PEs later, the smallest value to the "lowest" PE and the middle value will be stored as the PEs own value. When no more exchanges are needed, the array is h_i-sorted, and the next lower h_i is used. Ideally, all the shifts and exchanges over the distances h_i take place in parallel. In fact, because they use the optoelectronic links, the exchanges occur in M successive sets, as explained in the section on time multiplexing. For the last distance in the Shell sort, $h_1 = 1$, the electronic mesh is used without the need for time multiplexing. For simplicity, we count the two physical shifts needed for an exchange (to the left and to the right) as a single logical data shift.

3.10.2. Performance comparison of Shell sorting for OCI and M-RCH patterns

We have used the OCI and the M-RCH connection patterns described above as the Shell sort sequence, h_i, for 1-D arrays. For the OCI we used the symmetrical patterns, which are easier to implement, and perform almost as well as nonsymmetrical patterns.

We compared the number of shifts required to sort 1000 different arrays, each one containing 512 random numbers, and we plot histograms of the results in Fig. 3.14 (patterns as in sections 3.9.2). The OCI clearly outperforms the M-RCH in a statistical average. The median of the histogram is reduced from 320 to 250 when using OCI instead of the M-RCH, a speedup of 26%.

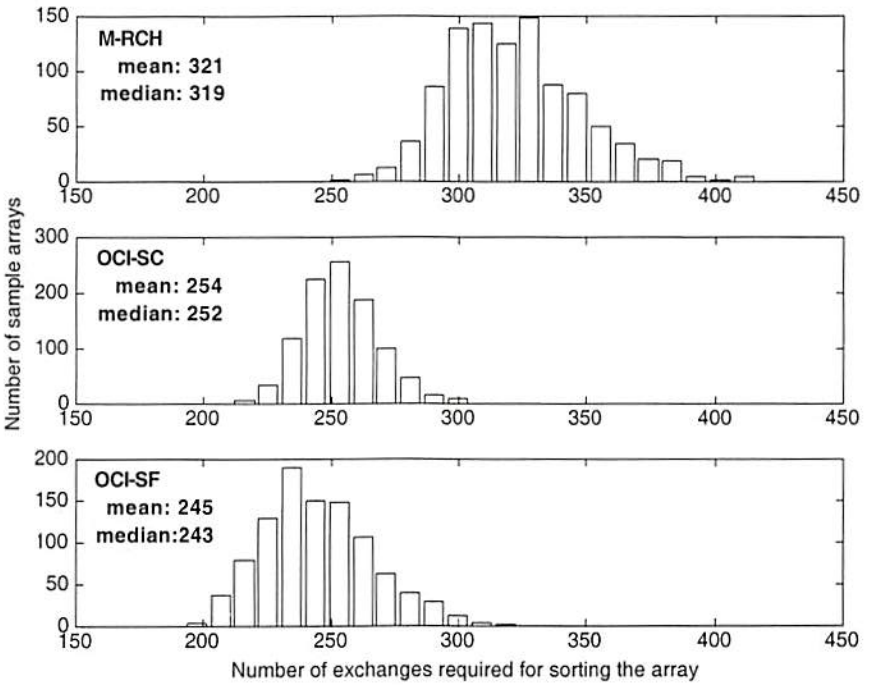

Figure 3.14 Histograms of the number of shifting operations for 1000 arrays of 512 random numbers, using the Shell sort and the distances given by the M-RCH pattern (top), OCI-SC (middle) and OCI-SF (bottom) as in Table. 3.2.

Even on a run-by-run basis, the OCI outperforms the M-RCH, as shown in Fig. 3.15. Here, we plot the histogram of the ratio of the number of shifts for OCI and M-RCH for the same input, over the 1000 random arrays. For the array with 512 elements, only a small number of runs (around 1%) fare better when using the M-RCH.

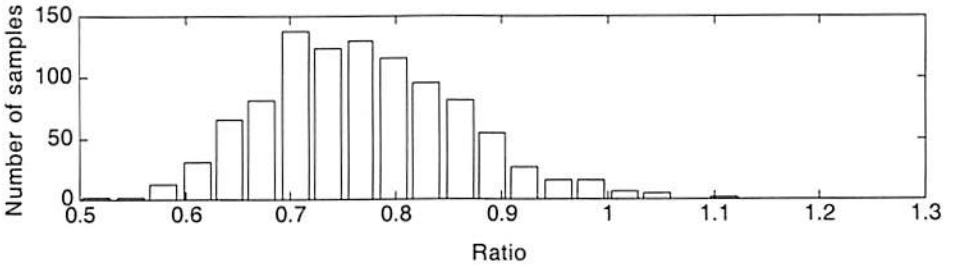

Figure 3.15. Histogram of the ratio of number of exchanges for M-RCH and OCI-SF (as in Table 3.2) on a run-by-run basis for sorting 1000 arrays of 512 random elements

For longer arrays, the OCI advantage is once again even more clear: a 50% speedup. Figure 3.16 shows the histograms for sorting 1000 arrays of 8192 random numbers. OCI patterns as in section 3.9.1 are used as the Shell sequence, for which OCI outperforms the M-RCH in a categorical way.

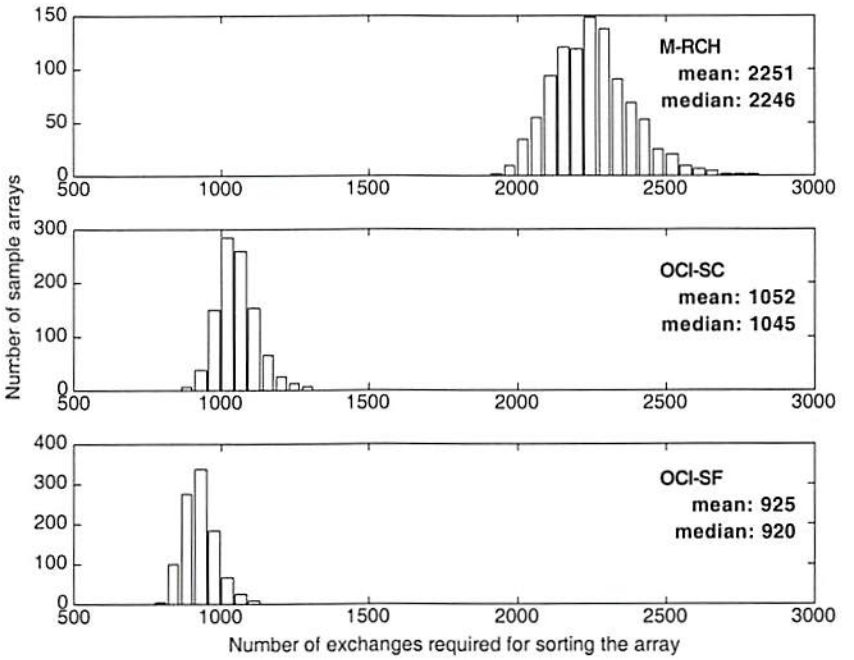

Figure 3.16 Histograms of the number of shifting operations for 1000 arrays of 8192 random numbers, using the Shell sort and the distances given by the M-RCH pattern (top), OCI-SC (middle) and OCI-SF (bottom) as in Table. 3.2.

3.10.3. Batcher sorting on cellular architectures

Batcher sorting is another sort method that is widely used on parallel machines, especially on SIMD architectures, because it involves identical steps performed on different data [6]. Unlike the Shell sort, which operates on a given subset until the subset is ordered, the Batcher sort is deterministic and involves a prescribed set of operations that is guaranteed to sort the input set. Because the number steps in a Batcher sort is known *a priori*, it is much easier to quantify the performance of the algorithm. Using a classical sorting procedure [6], we count the number of clock cycles required when the long data

distance exchanges are performed using either the M–RCH or the OCI. The results are summarized in Table 3.4. Clearly, the OCI is again superior to the M-RCH.

Array size	M-RCH	OCI-SF	OCI-SC
512	1851	1371	1181
8192	9309	6260	4837

Table 3.4. Performance comparison between the number of clock cycles required to sort arrays of different sizes using a Batcher sort in a cellular architecture using an optoelectronic interconnection with M-RCH and OCI patterns.

3.11. Conclusions

From a practical point of view, the most impressive performance gains of the OCI occur for arrays of large sizes. For smaller size arrays, the OCI still outperforms the M-RCH, but incremental advantages are less impressive. This is actually an advantage, because in real life applications the cellular interconnections are targeted to large arrays, where the OCI advantage is more apparent. Nonsymmetrical OCI patterns are slightly faster than symmetrical patterns, but require relatively more complex scheduling algorithms. The performance gain of using non-symmetrical connection patterns may or may not be sufficient reason to motivate the designer to use them.

3.12. Summary

We have presented a simple algorithm that can be used to design the optimal cellular interconnection topology for any array size and fan-out. While previous implementations of the cellular hypercube were designed using a trial-and-error approach, our algorithm is deterministic, and it allows the designer to set *a priori* constraints on the topology. Moreover, we theoretically demonstrate that the resulting interconnection topology is optimal in terms of achieving the minimum number of clock cycles per data shift at a given fan-out per pixel. The design allows a tradeoff between reducing the fan-out or reducing

the number of hops. Connection patterns for realistic array sizes can be designed with a fan-out of at most 6.

Simulation results validate the theoretical derivation and indicate the superiority of the OCI over even the best cellular interconnects demonstrated previously. The improvement comes both from a longer-distance connection set and from a reduced latency. The advantages are more evident for large arrays, but are present even for smaller sizes. The simulations presented involve data translations as well as two types of sorting algorithms, the Shell sort and the Batcher sort. In all cases, the OCI outperforms the M-RCH.

For communication-intensive SIMD applications (one-to-all broadcast, matrix-vector-multiplication) the OCI allows the ultimate speedup. There are exceptions to be found, for example the Fast Fourier Transform, for which the power-of-two link distances offer a powerful advantage to the M-RCH. A more exhaustive test of the OCI topology would be on an actual SIMD machine. As the biggest performance gain occurs for large arrays, still far from being feasible with current technologies, the experimental verification of the results awaits the implementation of such a system containing very large arrays of PEs.

References

[1] B. Hoanca and A. A. Sawchuk, "Optimized cellular interconnects for single instruction multiple data arrays," *Applied Optics*, vol. 37, no. 5, pp. 871-883, 1998.

[2] C. B. Kuznia, "Cellular hypercube interconnections for optoelectronic smart pixel cellular arrays, " Ph.D. Dissertation, University of Southern California, Los Angeles, California, 1994.

[3] C. B. Kuznia and A. A. Sawchuk, "Time Multiplexing and Control for Optical Cellular-Hypercube Arrays", *Applied Optics*, vol. 35, pp. 1836-1847, 1996.

[4] F. A. P. Tooley, "Optical interconnects do not require improved optoelectronic devices," *Proceedings of Optics in Computing '98*, Brugge, Belgium, pp. 14-17, 1998.

[5] J. A. Trezza, J. S. Powell, C. Garvin, K. Kang, and R. Stack, "Creation and application of very large format, high fill factor GaAs-on-CMOS binary and gray scale modulator and emitter arrays," *Proceedings of Optics in Computing '98*, Brugge, Belgium, pp. 78-81, 1998.

[6] D. E. Knuth, *The art of computer programming*, vol. 3, Sorting and Searching, Addison-Wesley Publishing Company, Reading, MA, 1973.

Chapter 4. TRANSPAR architecture and demonstration system

Part of the work in this dissertation is based on TRANSPAR, a system demonstrator we are building and testing. This chapter will briefly describe the functionality of the demonstrator. Additional details on the electronic circuitry will be given in Chapter 6, then further details on building a multiple-chip system in Chapter 7.

The TRANSPAR architecture and the physical smart pixel chip with the same name are designed to optimize two functions: 1) network interface for 3-D data packet transfer between computing nodes using a carrier-sense-multiple-access/collision-detection (CSMA/CD) protocol; and 2) high-throughput SIMD-type processing of 2-D data fields. By combining these two functions, a set of TRANSPAR chips can interconnected and be used for parallel pipeline processing (Fig. 4.1).

4.1. Operation of the TRANSPAR network

The operation of the TRANSPAR (TRANslucent Smart Pixel ARray) network is based on an optical carrier-sense-multiple-access/collision-detection (CSMA/CD) protocol, like in the case of the ubiquitous Ethernet. Unlike the Ethernet, the network is a ring with unidirectional propagation. Also, unlike the electronic and serial Ethernet, it is built as an optical and parallel packet network, using as nodes our smart pixel chip TRANSPAR.

The nodes on the network operate asynchronously, and the packet transfer between two nodes is performed using an asynchronous protocol and a first-in-first-out (FIFO) elastic buffer. This scheme avoids the need for a global clock, which greatly relaxes the design and implementation of the network. Additionally, to increase the throughput, the network operates at a high clock rate, comparable to the on–chip clock rate, while the electrical interface between the TRANSPAR node and the host computer operates at only a fraction

of this clock rate. This allows the electrical interface to be very simple, while allowing the powerful optical network to operate at high data rates. Further details on TRANSPAR network operation will be presented when we discuss the optimizations of the electronic circuitry, in Chapter 7.

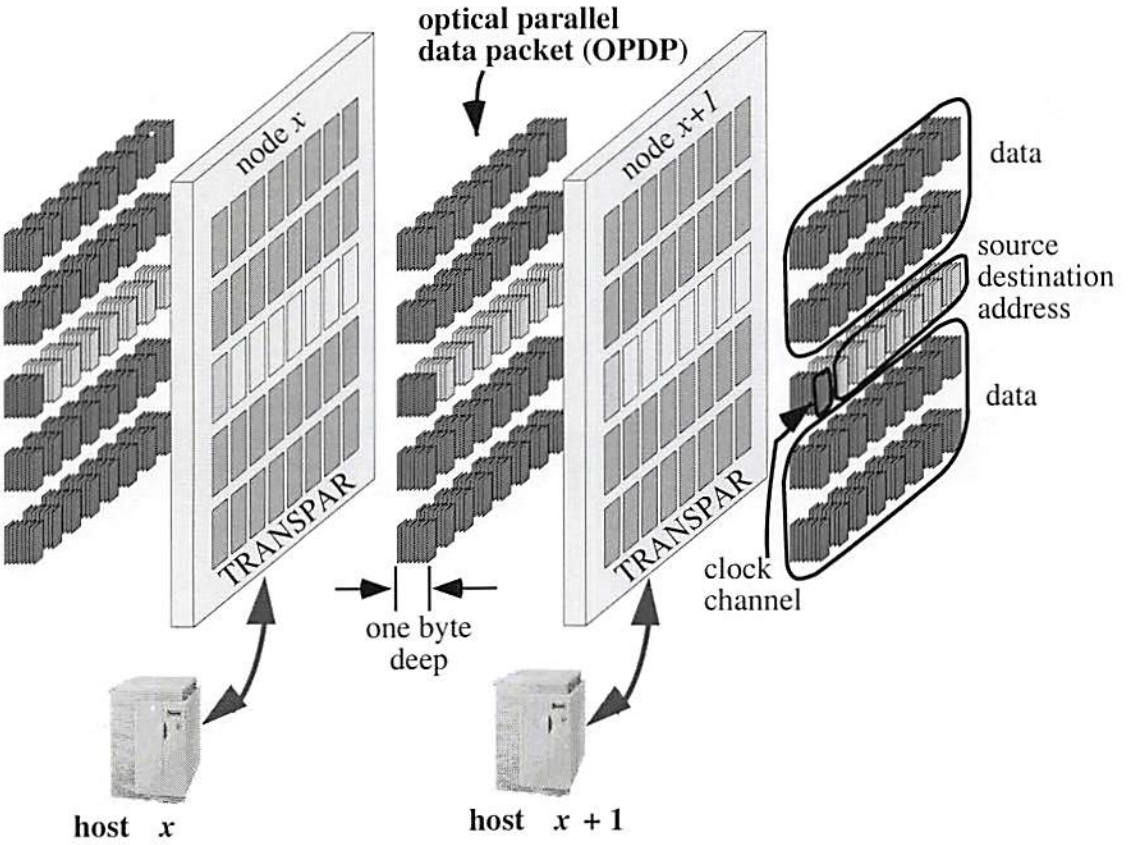

Figure 4.1. TRANSPAR network showing nodes connected to host computers. Optical parallel packets on the network allow pipeline operation of the SIMD nodes.

The TRANSPAR chip operates as an array of parallel optical channels. Each optical channel consists of two pairs of diodes, one pair connected as modulators and the other as detectors. There are a total of 39 such channels: an 8 x 4 array of data channels, three destination address channels, three source address channels and one clock channel. Behind each data channel there is a general-purpose processing element (PE). PE's are interconnected electrically using a mesh topology, and operate in SIMD fashion. Each PE

contains memory and logic for performing arithmetic and logic operations. Additional circuitry handles the network protocol, the interfacing of the chip with a host computer and the clocking of the chip.

4.2. SIMD functionality of the TRANSPAR chips

Multiple nodes on the network operate as SIMD computation engines. Each node performs a small set of operations on each data block and then transfers the data to the next stage in the pipeline for further processing. For SIMD computation, each TRANSPAR chip contains an array of mesh-connected processing elements (PEs) implemented by smart pixels. Although the area dedicated to each smart pixel is only 125 x 250 square microns, it is sufficient to implement an ALU-based fine-grain SIMD processing element (PE) (Fig. 4.2, right). Nearly all SIMD machines use this type of PE architecture, including the popular Connection Machine (CM) and Massively Parallel Processor (MPP) [1, 2]. In the TRANSPAR, these smart pixel PEs are replicated into a 4 x 8 mesh-connected array and can perform SIMD processing using similar programming methods used in current SIMD machines.

The one-bit ALU performs data processing on data stored within the PE's 32-bit SRAM or within a neighboring PE's SRAM. Although the ALU operates on single bits internally, the TRANSPAR node can be viewed as a computing system with a programmable word length. An on-chip finite-state-machine (FSM) receives *macrocode* instructions from an off-chip controller (or host computer) and converts them into series of micro-codes that are broadcast across the PE array. The host interface is thus insulated from seeing the bit-serial nature of the PE architecture.

Each PE has three registers (not shown in Fig. 4.2) used to store intermediate values of the computations. Registers RA and RB provide interface between the ALU and the memory blocks and register RC holds the carry bit for arithmetic operations or the enable

bit. The enable bit allows the partitioning of the PE array into enabled and disabled PEs, selecting a subset of PEs to perform the instructions broadcasted to the whole array.

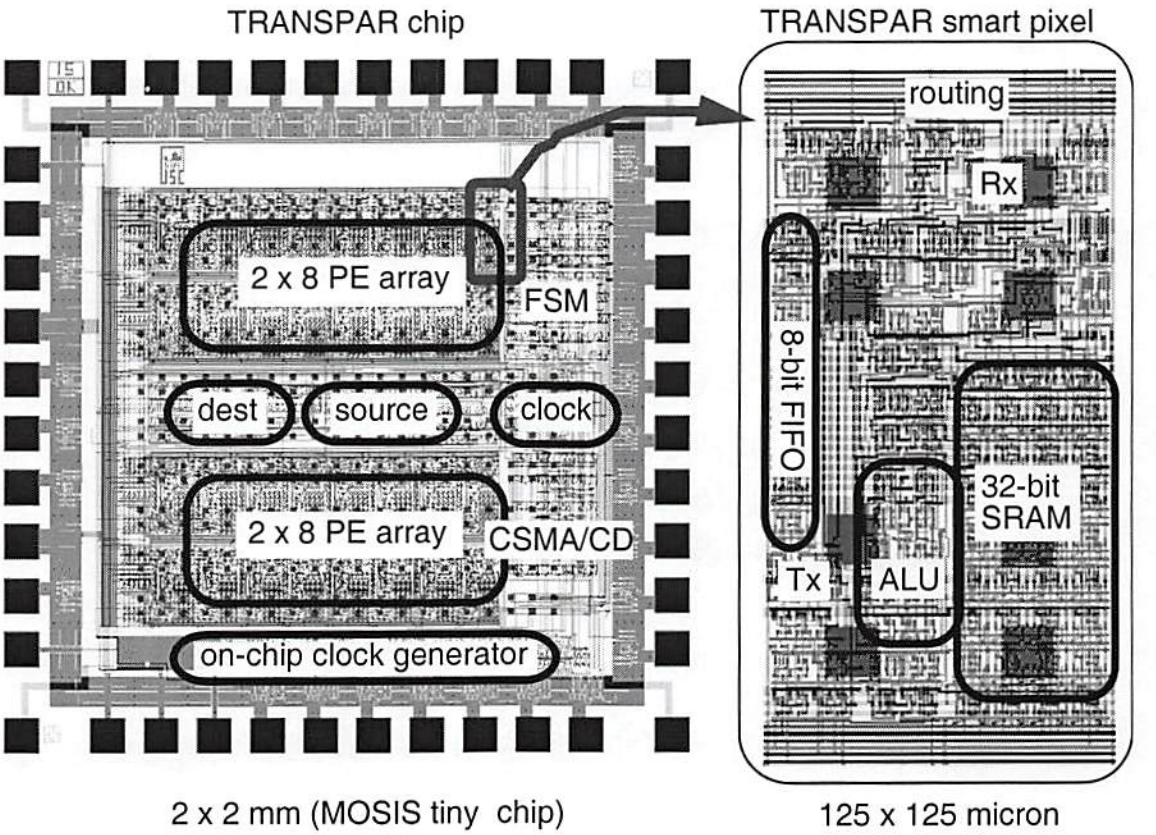

Figure 4.2. Layout of the TRANSPAR chip showing details of a processing element.

For communication among the PEs, data fields can either be shifted into the array electrically, via the mesh network (in a 1-D row parallel format) or optically via the detectors integrated into the PEs (as a 2-D array). Likewise, data is unloaded from the array through electronic channels in a row parallel format or optically transmitted in 2-D array format.

4.3. Multiple-chip pipeline architecture

The pipelined SIMD processing of each TRANSPAR chip and OPDP transfer between TRANSPAR chips can be combined to create a multi-node SIMD processor. Each node is

a mesh connected SIMD processing array and data transfers between nodes are made using the 8-bit deep, 8 x 4 bit wide data packets, based on the asynchronous CSMA/CD protocol. Because the network can accommodate up to six nodes, it is equivalent to a 6 x 8 x 4, 3-D array of SIMD processors. This system can perform very fast parallel processing of 2-D data fields, as required in image/video processing, packet header recognition or routing.

4.4. Design and fabrication of the TRANSPAR chip

The layout of the TRANSPAR chip is shown in Fig. 4.2, with the most important functional blocks indicated at their approximate locations. The TRANSPAR chip was partly synthesized from VHDL and partly custom designed using Magic. The resulting 1.5 mm x 1.5 mm chip was fabricated using the HP14TB 0.5 μm technology through MOSIS. After fabrication, a 20 x 10 array of GaAs multiple quantum well (MQW) diodes were flip-chip bonded to the CMOS logic. Pairs of diodes operate as dual-rail optical I/O, used either as modulators or as detectors, depending on the biasing. Not all diodes are actually being used in our implementation. The flip-chip bonding was done at Lucent Technologies and was sponsored by DARPA through the GMU/CO-OP foundry program [3]. Because of the rich functionality and the complexity of the design, testing is still underway. This work presents the preliminary results on the system demonstrator we are building with the TRANSPAR chips.

4.5. Summary

Some of the proposed optimization techniques have been demonstrated experimentally on TRANSPAR, a chip we designed and are currently testing. TRANSPAR chips are designed to combine high-throughput networking with powerful SIMD computations in massively parallel pipeline architecture.

References

[1] L.W. Tucker and G. G. Robertson, "Architecture and Applications of the Connection Machine," *Computer*, vol. 21, no. 8, pp. 26-38, 1988.

[2] J. L. Potter, "Image Processing on the Massively Parallel Processor," *Computer*, vol. 16, no. 1, pp. 62-67, 1983.

[3] See http://co-op.gmu/edu/

Chapter 5. Optimization of the optical system

5.1. Introduction

The CH is intended to speed up the long distance data transfers in SIMD arrays, by providing optical short-cut links. If electronic links are used for such long-distance links, the operation speed is limited by a combination of electrical crosstalk, delay, power dissipation, and frequency dependent loss (distortion). Optical links are much better suited for implementing the short cuts, because they do not suffer from any of the above effects. On the other hand, optical links introduce complications due to their hybrid optic and electronic nature. In designing the optical links, the choices to be made are between passive or active optical sources, micro-optics or macro-macro-optics, and among a set of wavelength and polarization multiplexing schemes. This chapter will review the available choices and will outline the most efficient ones for interconnecting SIMD processors.

5.2. The choice of the optical source

The two possible technologies for implementing free-space optical links are either vertical cavity surface emitting lasers (VCSELs), active sources, or modulators, passive sources [1]. Regardless of the type of source used at the transmitter, the receiving end is an optical receiver that incorporates a photodetector and a transimpedance amplifier. Thus, the design choice is only about the type of optical source to be used. Some of the tradeoffs in using active or passive sources are detailed below.

5.2.1. VCSEL based systems

The VCSEL is a relatively new device, developed less than a decade ago as a laser that could couple better into optical fibers [2]. This is because the VCSEL has a laser cavity

perpendicular to the substrate, which allows it to have a circular shape (or elliptic with low ellipticity). Moreover, because the light is emitted perpendicular to the substrate, the VCSEL is much more appropriate for array fabrication than the edge-emitting laser. Arrays of VCSELs can now be fabricated with relatively high yields and may soon become cost effective in replacing electronic-only interconnections [3].

Three main issues face a designer of VCSEL based systems: the power management, the turn-on delay and the divergence angle of the VCSEL.

The most striking problem is the power dissipation. Even at high wall-plug efficiencies, the VCSELs generate significant amounts of heat and pose serious problems to the thermal management of the system. For Gb/s data rates the optical power required at the receiver is of the order of 10 μW [4]. The typical free space optical system has losses of 10 dB between two chips. This would only require a low power at the transmitter. At the same time, commonly available VCSELs have threshold currents of about 1 mA and operate with a forward voltage drop of 1-2 V. Such VCSELs dissipate at least 1-2 mW only to reach threshold! Moreover, for good mode quality, the VCSEL must be operated at a current at least 10% above the threshold [5]. Low threshold devices are currently being researched, to reduce the power dissipated in the VCSEL at or below threshold, but such low power VCSELs have low wall plug efficiency and low optical power output in general.

A second problem with VCSELs, related, yet somewhat different from the threshold current, is the issue of the turn on delay, which is dependent on the quiescent VCSEL current. This is the current driven into the VCSEL when transmitting a "zero." Intuitively, this current would be zero, to ensure maximum extinction ratio between "zero" and "one" states in a digital system and to minimize the power dissipation. Not so intuitive is the fact that a low quiescent current imposes a relatively large turn-on delay, which increases the latency of the link [6]. For this reason, VCSELs are usually biased close to threshold to

minimize the turn-on delay. This requires complex solutions for thermal management [7], for example using return-to-zero data or limiting the pitch of the active devices.

Finally, a third problem with VCSELs is their relatively large numerical aperture (divergence angle). To be able to capture most of the light emitted from a VCSEL (for reasons of maximizing efficiency and minimizing crosstalk into additional optical channels), the lenses of the optical system must have relatively low F/#'s. For this reason, microlenses tend to be the best solution, as detailed in Section 5.3.

5.2.2. Modulator based systems

As an alternative to VCSELs, arrays of multiple quantum well (MQW) modulators [8] can be used. These MQWs are in general flip-chip bonded onto the PE VLSI chip. An external continuous wave laser beam is converted to a spot array using a Dammann grating [9] or a computer generated hologram (CGH) [10] for readout of the data from each individual modulator (Fig. 5.1). Using MQWs moves the burden of power management off chip, where a high-power laser beam poses no major limitations. Conversely, the

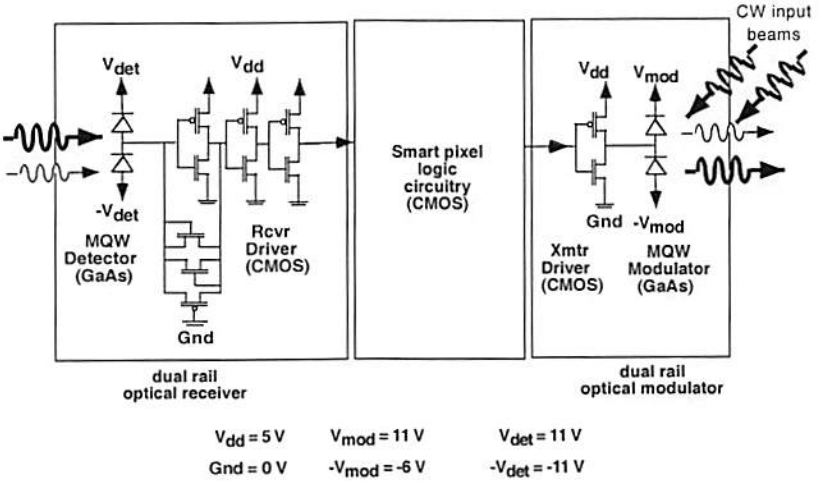

Figure 5.1. Dual rail optical I/O channel, showing the input CW beams that are used as an optical power supply for the modulators.

optical system is more complicated than in the case of active sources. A modulator based optical system needs to allow a double pass. First the light from the laser is incident onto the MQW modulators, then the reflected light is read off on a second path, usually in reflection off a modulator array (Fig. 5.1, right). Complex designs must be employed to separate the two optical paths, usually based on the polarization of the light.

Another drawback of the modulator based optical links is the relatively low contrast ratio. While the best devices can have contrast ratios of up to 80:1 [11], arrays of devices have usually much lower contrasts, usually in the range 1.5-2:1 [12]. This could be sufficient for implementing an optical link, were not for additional uncertainties and nonuniformity in the optical system. When using arrays of modulators, a discrete laser and a CGH with a large fan-out are used to generate an array of spots. Because of imperfections in the design and the fabrication of the CGH, the power of the different spots in the array can vary by as much as 20%. Combining this uncertainty with the low contrast ratio may make it impossible to find a similar threshold for all the receivers in the receiver array. For this reason, a dual rail approach can be used to enhance the effective modulator contrast ratio.

Dual rail signaling is similar to differential signaling in electronics. Two equal-power CW beams are used as input to a pair of modulators (Fig. 5.1). Instead of providing an absolute modulation on a single beam, the dual-rail modulator pair needs to provide only a relative modulation of one beam with respect to the other. This can be achieved much easier with the limited contrast ratio available. Additionally, a special design of the CGH may further reduce the requirement for uniformity.

5.2.3. Comparison between active and passive source optical systems

The main reason active optical transmitters have become so popular is because they simplify considerably the design and the use of the optical system. Using VCSELs, the

optical system is a single pass configuration. The light from the source must be routed to the detector. In contrast, when using modulators, the light from the "optical power supply" (OPS) [13, 14] must be first routed to the modulators, then the reflected light routed to the detectors. This complicates the design of the optical system and makes the alignment much more difficult.

Active sources ease the alignment process. First, they provide light that can be used as a reference in aligning the transmitter and the receiver chips. Second, only a single alignment step is required, between the two transmitter and the receiver chips. This alignment could be further fine-tuned by running multiple channels in parallel and minimizing the crosstalk between additional channels (active alignment). In contrast, passive sources require two alignment steps. The OPS and the transmitter chip must be aligned first, then the transmitter and receiver chips require a second alignment step. Moreover, the alignment between the OPS and the transmitter chip cannot be made actively, because the modulated steps must be imaged onto an array of detectors for active alignment (which is done in step two). Thus passive sources require two coupled alignment steps, which makes the setup of the system significantly more difficult.

On the other hand, VCSELs have high numerical aperture (high divergence), especially if they are designed to be fast and low power (which requires a small aperture). This complicates the design of the optical system, requiring either macrolenses with low F/# or arrays of microlenses. Most often the second solution is preferred, and microlenses have been integrated with VCSELs using self-aligned processes [15]. Even so, the relay distance for microlens based systems is relatively low, and additional macrolenses are required to extend the distance. While such hybrid microlens and microlens systems have much more relaxed requirements on the aberrations and the F/# of the macrolenses, they have additional loss due to the extra optical surfaces in the system.

Another major disadvantage of the VCSEL based systems is the power budget on chip. As we described above, the power dissipation of a typical VCSEL is 1-2 mW. In contrast, modulators require two orders of magnitude less power on chip (the power dissipation in the laser that generates the OPS is not usually an issue, because it is far away from the chip, well separated, and it can be easily handled). On the other hand, the more complex optical system and the need for an external laser for the OPS makes the modulator systems more bulky than the VCSEL systems.

The future of the two technologies we presented is still under investigation and it is not clear which one of them will become more widespread. While the VCSEL technology is less mature, the optics for the interconnection is simpler than for the modulators, making VCSELs the preferred choice for other optical interconnect applications (for example in smart pixel arrays). At high densities of optical channels, where the power dissipation is the limit, passive devices (modulators) still have an advantage, but technological advances in VCSEL design and fabrication may change this in the future.

5.3. Choice of optical components

Figure 2.1 shows a concept for a full-aperture optical interconnection system using *transmissive* devices. Alternatively, a *reflective* cellular array and interconnection hologram or diffractive optical element could also be used to implement the system. The advantage of the reflective CGH is that the source PEs can be imaged back onto the destination PEs on the same chip. The reflective configuration introduces a mirroring, which must be compensated with an additional reflection [16]. The transmissive implementation on the other hand requires a feedback mechanism for bringing the images from the output plane of the 4-F system back onto the destination PEs on the chip, which is a lot more complicated to realize in practice, albeit easier to understand in a drawing.

In the implementation in Fig. 2.1 each transmitted beam fans out onto the destination PEs using a Fourier plane hologram in a 4-F system. Macro- or micro- lenses can be used for imaging, the later having lower aberrations but more critical alignment tolerances. A hybrid system employing both macro and micro lenses may provide the best combination of features. Using microlenses integrated with the optical devices and macrolenses for relay reduces significantly the alignment requirements for the macrolenses [15].

A macro-lens based optical system (as shown in Fig. 2.1) is used for full-aperture imaging, where the optical channels are separated spatially only at the source and destination. An array of optical sources can share the same lens for imaging to the detector plane. This makes the design simple, because the lens need not be aligned with the source or detector arrays. Additionally, the lens is spaced a few centimeters away from either of the arrays, allowing easy access for the opto-mechanical alignment system. On the other hand, the lens must be of very good quality, and must have a field of view wide enough, but with low aberrations. To allow easy set-up of the optical components and lenses, a base-plate can be used with guides that restrict some of the degrees of freedom of the optical elements.

A micro-lens optical system (Fig. 5.2) has relatively complementary features, as compared with the macro-optic system. Individual micro-lenses are used to deliver each optical channel from one source to one or more detectors. The challenge in such a system is to align the micro-lenses precisely with the source or detector, and attach them to the optoelectronic devices, into a subsystem. Once this is achieved, the relative alignment of the resulting subsystems with respect to each other is greatly relaxed. The drawback of the micro-lens systems is the relatively short relay distance that can be achieved with microlenses. This is because the microlens size is limited by the pitch of the optical channels and the F/# is limited by technological constraints. These two constraints limit the focal length of the lens and through it the relay distance.

A critical problem with micro-lens based systems is the difficulty of testing them. The system is much smaller in size than a macro-lens based system, which makes the job of aligning and assembling it extremely difficult. For this reason and to achieve the best possible alignment performance, micro-lenses are usually aligned using either lithographic [15] or interferometric [17] techniques.

A hybrid system combines micro-lenses at the sources and detectors with macro-lenses that greatly enhance the relay distance. However, hybrid systems still require precise alignment of the micro-lenses with the sources and detectors, as well as macro-lenses with good quality, low aberration and a large field of view.

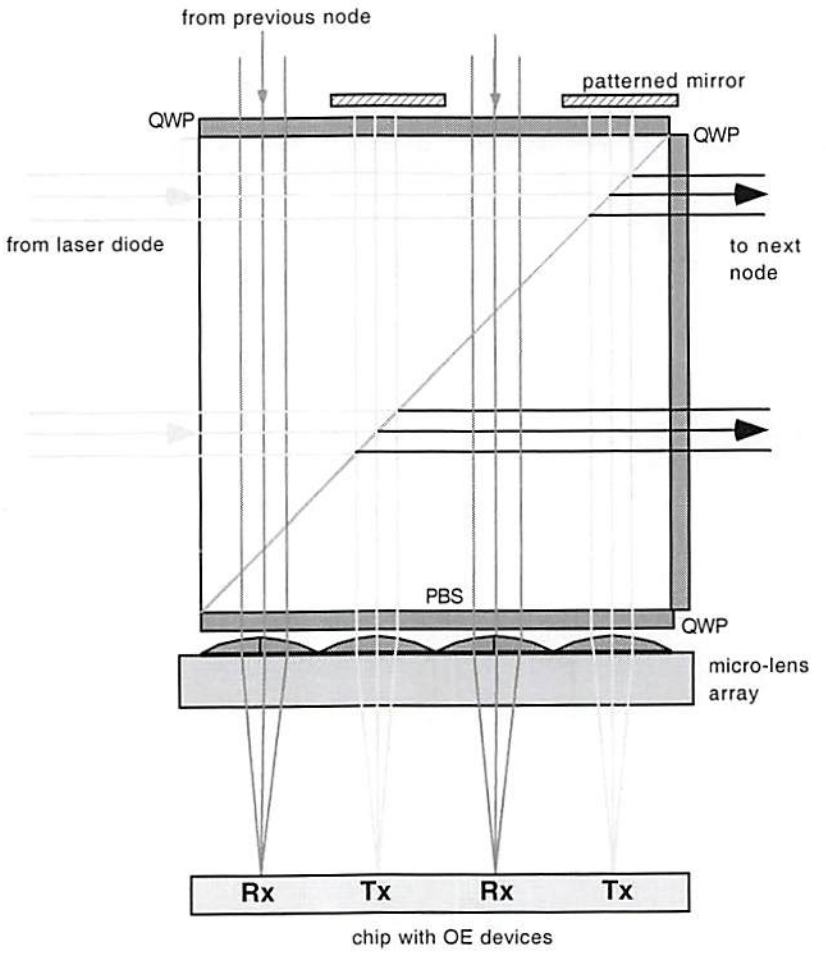

Figure 5.2. Microlens based optical system. QWP is a quarter wave plate.

5.4. A practical example – the TRANSPAR optical system

We show in Fig. 5.3. a photo of the TRANSPAR set-up with two chips interconnected using dual-rail modulator optical I/O. The system is relatively bulky, because it requires the optical power supply laser (not shown, located towards the bottom of the figure). Additionally, the number of optical components is relatively high, even though the connection between chips is only unidirectional. A schematic layout for four chips interconnected with each other in a ring is shown in Fig. 5.4. This setup requires four discrete laser sources, but could be optimized to use a single high-power laser. At the top of Fig. 5.4, we indicate for comparison the size of an optical system using active sources and microlenses (the beam splitter assembly in Fig. 5.2). Four such assemblies could be used to interconnect four chips in a ring, replacing the modulator-based baseplate at the bottom of Fig. 5.4.

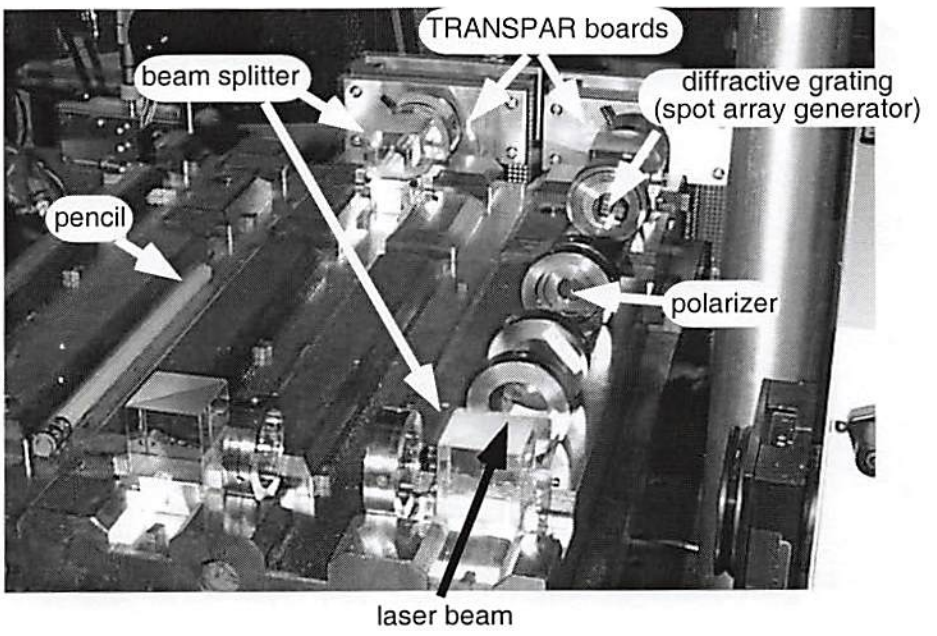

Figure 5.3. Photograph of the TRANSPAR baseplate showing two electronic boards and the components of the modulator based optical system. The laser for the modulator power supply is not shown (it falls outside the picture frame).

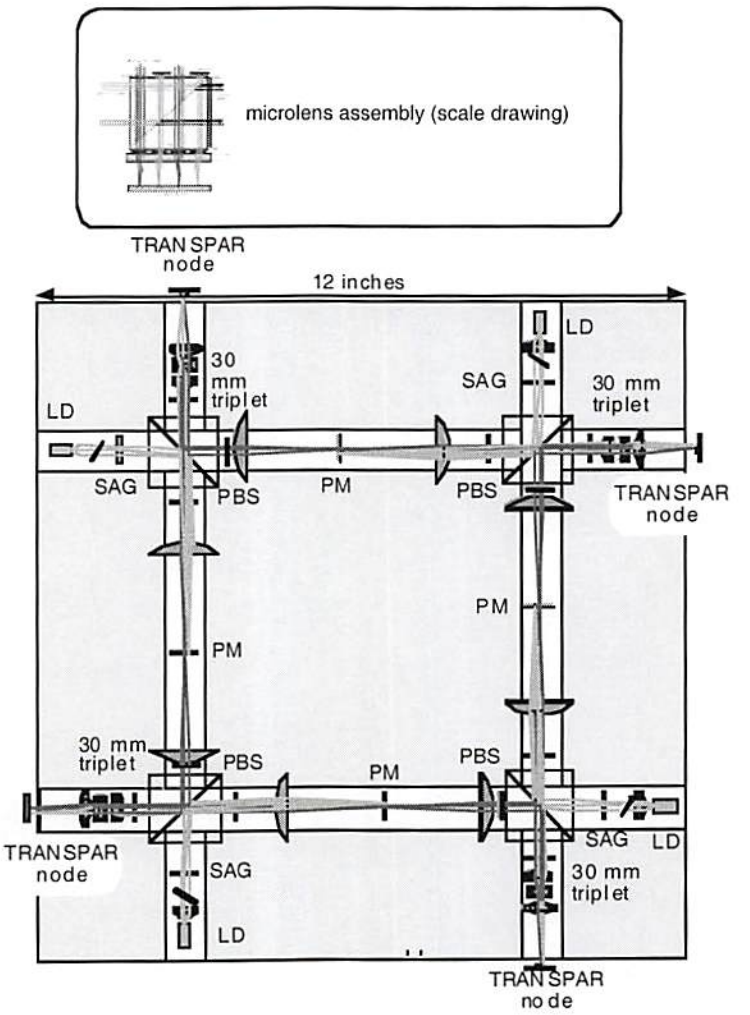

Figure 5.4. Schematic layout of four smart pixel chips using modulators as optical sources and interconnected with each other in a ring. The acronyms stand for laser diode (LD), patterned mirror (PM), polarization beam splitter (PBS) and spot array generator (SAG).

5.5. Wavelength and polarization multiplexing

One additional feature of optical links is that they allow multiplexing not only spatially, as electrical links allow (by using parallel wires), but also using the polarization and the wavelength domain. Separate multiplexing techniques are used for wavelength and for polarization multiplexing. We will show how they can be combined to enhance even more the flexibility of the design of the interconnection elements.

5.5.1. Design of computer generated holograms

A computer-generated hologram (CGH) is a phase only element that reconstructs a given output pattern from an input pattern, based on the diffraction and interference of multiple beam elements. The input to a CGH is usually a collimated laser beam. The area of the CGH is divided into an array of pixels, each introducing a different delay on the incoming wavefront. At the output of the CGH, usually in the Fourier plane of a lens, the interference of the different portions of the wavefront create the desired pattern. The name of the CGH was given because the design of the phase element is done using computer calculations of the optimum phase profile, rather than using a procedure involving analog exposure of a photographic film, as in classical holography.

5.5.2. Wavelength multiplexing of computer generated holograms

Wavelength multiplexing allows multiple optical channels to coexist in the same spatial location. A CGH design is normally specified for a particular wavelength, for which the output pattern is synthesized from the diffraction and interference of multiple beam elements. Operating at another wavelength introduces chromatic aberrations, unless a special design procedure is used. Such a special design procedure has been proposed, based on the periodicity of the phase of the optical beams [18]. Additional degrees of freedom in the design of the CGH occur because phase delays multiples of 2π can be added to any or all of the CGH pixels without changing the output field. This way, a given pixel can be specified to introduce two different and controlled delays at two different wavelengths. By using a stack of multiple layers with different refractive indices and thickness profiles, multiple wavelengths CGH could in principle be fabricated [19]. In practice, alignment and choice of materials may limit the number of wavelengths multiplexed to a small value, possibly not more than three.

5.5.3. Polarization multiplexing of computer generated holograms

Another approach to multiplexing more patterns in a single CGH is to use polarization multiplexing. Two types of materials can be used for polarization multiplexing: birefringent materials (which exhibit natural birefringence) and form-birefringent materials (in which the birefringence is induced by subwavelength structures etched on the surface of a material).

Naturally birefringent materials have been known and used for many years, and are readily available in crystal form. Such materials have two special directions, aligned to the crystal structure. If an incoming light beam is polarized with its polarization plane aligned to one of the two directions, the birefringent material acts like an isotropic material with a given refractive index. Conversely, a similar behavior is exhibited along the second direction (which is orthogonal to the first one), but now the refractive index is different. A beam that is not polarized along one of the two directions can be analyzed by decomposing it into two components, one aligned with each direction. To fabricate polarization multiplexed holograms, the material can be etched such that the two indices of refraction seen along the two directions generate different types of output patterns [20]. This way, depending whether the input beam is polarized along one or the other directions, the output pattern of the CGH will be different.

Form birefringent materials are used on exactly the same principle, but here the birefringence is induced artificially, by etching structures with subwavelength dimensions into an isotropic material [21]. The advantage of such materials is that they have a much larger birefringence, which can also be tailored to some extent from the design of the etched structures. In contrast, naturally birefringent materials have a much smaller birefringence, with refractive indices limited by existing materials.

5.5.4. Combined wavelength and polarization multiplexing of computer generated holograms

Hybrid multiplexing schemes can be devised by combining wavelength and polarization. For example, using a substrate of form birefringent material with adjustable birefringence, a polarization and wavelength multiplexed CGH may be produced. The design parameters are the birefringence (dictated by the geometry of the etched structures) and the thickness profile of the substrate. The number of substrates to be used and the number of etching depths required depend on the complexity of the output pattern.

5.6. Summary

The design of the optical system is an engineering decision-making task in which multiple tradeoffs must be considered. The decision of whether to use an active (VCSEL) or a passive (modulator) optical source has implications on the power budget of the system, but also on the packaging and the ease of alignment. Similarly, the use of microoptics or macrooptics affects the scale of the system as well as the ease of alignment. Using optics allows a high degree of flexibility in the design of the interconnection pattern if combining polarization and wavelength multiplexing in a computer generated hologram.

References

[1] T. Nakahara, S. Matsuo, S. Fukushima, and T. Kurokawa, "Performance comparison between multiple-quantum-well modulator-based and vertical-cavity-surface-emitting laser-based smart pixels," *Applied Optics*, vol. 35, no. 5, pp. 860-871, 1996.

[2] L. A. Coldren and S. Corzine, *Laser diodes and photonic integrate d circuits*, New York: Wiley Interscience, 1995.

[3] C. W. Stirk and J. Neff, "The cost of optical interconnects vs. MCMs", *Proceedings of Optics in Computing '97*, OSA Technical Digest, Optical Society of America, Washington, DC, pp. 21-23, 1995.

[4] T. V. Muoi, "Receiver design for high-speed optical-fiber systems," *IEEE Journal of Lightwave Technology*, vol. LT-2, no. 3, pp. 243-267, 1984.

[5] S. Esener and P. Marchand, "3D optoelectronic stacked processors: design and analysis," *Proceedings of Optics in Computing '98*, Brugge, Belgium, pp. 541-545, 1998.

[6] L. Zei, K. Obermann, T. Czogalla, and K Petermann, "Turn-on jitter of zero-biased nearly single-mode vcsel's," *IEEE Photonics Technology Letters*, vol. 11, no. 1, pp. 6-8, 1999.

[7] M. Osinski and W. Nakwaski, "Thermal-analysis of closely-packed 2-dimensional etched-well surface-emitting laser arrays," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 1, no. 2, pp. 681-696, 1995.

[8] A. V. Krishnamoorthy and D. A. B. Miller, "Scaling optoelectronic-VLSI circuits into the 21-st century: a technology roadmap," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 2, no. 1, pp. 55-75, 1996.

[9] J. Jahns, M. M. Downs, and M. E. Prise, "Damman gratings for laser beam shaping," *Optical Engineering*, vol. 28, no. 12, pp. 1267-75, 1989.

[10] M. R. Feldman and C. C. Guest, "Holograms for optical interconnects for very large scale integrated circuits fabricated by electron-beam lithography," *Optical Engineering*, vol. 28, no. 8, pp. 915-21, 1989.

[11] J. A. Trezza, J. S. Powell, C. Garvin, K. Kang, and R. Stack, "Creation and application of very large format, high fill factor GaAs-on-CMOS binary and gray scale modulator and emitter arrays," *Proceedings of Optics in Computing '98*, Brugge, Belgium, pp. 78-81, 1998.

[12] J.-M. Wu, C.-H. Chen, C.B. Kuznia, B. Hoanca, A.A. Sawchuk, "Demonstration and Architecture Analysis of CMOS/MQW Smart Pixel Array Cellular Logic (SPARCL) Processors for SIMD Parallel Pipeline Processing," accepted for publication in *Applied Optics*, 1999.

[13] A. G. Kirk, D. F. Brosseau, F. K. Lacroix, E. Bernier, M. H. Ayliffe, B. Robertson, F. A. P. Tooley, and D. V. Plant, "Design and implementation of a two-stage optical power supply spot array generator for a modulator-based free-space interconnect," *Proceedings of Optics in Computing '98*, Brugge, Belgium, pp. 48-50, 1998.

[14] R. Iyer, Y.S. Liu, G.C. Boisset, D.J. Goodwill, M.H. Ayliffe, B. Robertson, W.M. Robertson, D. Kabal, F. Lacroix, and D. V. Plant, "Design, implementation, and characterization of an optical power-supply spot-array generator for a 4-stage free-space optical backplane," *Applied Optics*, vol. 36, no. 35, pp. 9230-9242, 1997.

[15] D. A. Louderback, O. Sjölund, E. R. Hegblom, J. Ko, and L. A. Coldren, "Novel technique for monolithic integration of microlensed resonant detectors and vertical cavity lasers," *IEEE/LEOS Proceedings of the Summer Topical Meetings on Smart Pixels*, Monterey, CA, pp. 11-12, 1998.

[16] C. B. Kuznia, "Cellular hypercube interconnections for optoelectronic smart pixel cellular arrays, " Ph.D. Dissertation, University of Southern California, Los Angeles, California, 1994.

[17] Y. Liu, B. Robertson, G. C. Boisset, M. H. Ayliffe, R. Iyer, and D. Plant, "Design, implementation, and characterization of a hybrid optical interconnect for a four-stage free-space optical backplane demonstrator," *Applied Optics*, no. 37, pp. 2895-2914, 1998.

[18] J. E. Ford, F. Xu, and Y. Fainman, "Wavelength-selective planar holograms," *Optics Letters*, vol. 21, no. 1, pp. 80-82, 1996.

[19] Y. Arieli, S. Noach, S. Ozeri, and N. Eisenberg, "Design of diffractive optical elements for multiple wavelengths," *Applied Optics*, vol. 37, no. 26, pp. 6174-6177, 1998.

[20] F. Xu, R.-C. Tyan, Y. Fainman, and J. E. Ford, "Single-substrate birefringent computer-generated holograms," *Optics Letters*, vol. 21, no. 7, pp. 516-518, 1996.

[21] F. Xu, R.-C. Tyan, P.-C. Sun, Y. Fainman, and J. E. Ford, "Fabrication, modeling and characterization of form-birefringent nanostructures," *Optics Letters*, vol. 20, no. 24, pp. 2457-2459, 1995.

Chapter 6. Optimization of the electronic circuitry

Along with the optoelectronic enhancements, we have also explored purely electronic techniques for optimizing the performance of the computing architecture. We present here four research directions, which have proven the most crucial for the overall performance results. The first one deals with the receiver design, the second with the interfacing of the on-chip and off-chip circuitry (which have widely different capabilities), the third one with clocking strategies, and the fourth one with the PE architectures.

6.1. Receiver design

A low-noise, high-sensitivity receiver is essential for reliable transmission of optical channels. Most of the work done on receiver design has dealt with serial links. This work, in contrast, considers the issues that are particular to massively parallel architectures, where large amounts of power supply noise and the crosstalk are present due to the large number of switching circuits. Special design techniques are required to deal with these deleterious effects.

The optical receiver consists of a photodetector, which converts the incoming optical power into electrical current, followed by a high-gain, low noise transimpedance amplifier that converts the small photocurrent to a large voltage swing. Typical input currents are in the 1-100 μA range. Typical output voltages depend on the logic family used, and are in the range of 1V for ECL to 5V to CMOS. This shows that the transimpedance gain of the receiver amplifier must be in excess of 50 kΩ.

With such a high-gain stage, there is always the danger of oscillations. The receiver designer must ensure stable operation, by leaving sufficient design margins to avoid oscillations. In architectures involving massively parallel arrays of receivers, an added constraint comes from the coupling between adjacent and even distant receivers. Two such

coupling mechanisms may be present simultaneously: power-supply noise and substrate coupling effects.

6.1.1. Power supply noise

When a channel is switching, the output voltage swing across its (capacitive) load draws relatively large current spikes from the power supply lines. If the parasitics of the lines have an inductive component, usually due to the inductance of the bonding wires of the integrated circuit (IC) package, the current spike induces a voltage pulse on the power line. This pulse may reach the input of an adjacent (idle) channel where due to the high sensitivity of the receiver it may trigger false switching. The problem is compounded when multiple channels may switch at the same time and the effects cumulate.

Two approaches can be taken to combat the effects of power supply noise. The first one is to use a differential design, which has a strong common mode rejection. The input signal for a differential amplifier is given by the difference between two currents or voltages. If the two currents or voltages change in the same direction, the output of the amplifier is (ideally) unaffected. This way, any pulses on the power supplies will affect both inputs equally, and will not produce any changes at the output of the receiver. In practice, depending on the design, the effects of the pulse may be attenuated by a factor of 10-1000.

The second approach is to use separate power supply rails for each of the gain stages in the amplifier. Each stage has a relatively low gain as compared with the total gain of the amplifier. For this reason, the output of the stage is not too large relative to the input. Any coupling through the power rail of a single stage will have a smaller influence than for the case of a single power rail shared by multiple stages. Additionally, using separate power rails for intermediate stages reduces the overall inductance per rail, further reducing the voltage step on the power lines.

6.1.2. Substrate coupling

Another mechanism for coupling between different channels and between the outputs and inputs of the same channel is coupling through the substrate of the IC. As switching occurs in the circuitry, electrons and holes may be injected into the substrate of the IC. From there, they may resurface in the input section of the amplifier, whether in the same channel or in a different channel. At the input of the amplifier, there is no difference between the photodetected carriers and the ones injected from the substrate: both types of carriers will be amplified and will generate a voltage swing at the output. Because the carriers injected from the substrate are not related to the input signal, they may cause false switching, the same way as the power supply noise.

Two measures can be taken against substrate coupling. One is to use guard rings around the input areas of the receiver. These guard rings screen the sensitive input circuitry by catching the carriers that flow in the substrate. The rings must be connected to a "clean" power supply or ground, otherwise the power supply noise explained in the previous section may cause more damage than good. The second approach is to use "slow" output circuitry, with a rise time just good enough for the data rates but not any faster. This approach only reduces the excess hot carriers, but still leaves some of the carriers in the substrate.

6.2. Considerations on the pixel architecture

Very often, the designer of smart pixel systems encounters new challenges, which are not usually encountered in designing conventional architectures. One such challenge is that the PE area is dictated by the pitch of the optical channels, which, in turn, is dictated by the available technology. The pitch of the optical channels is often chosen to allow easy coupling of the free-space channels into an array of optical fibers, on 125 μm centers.

Because of this limit on the PE area we chose a PE architecture that can fit in a small space, yet is flexible enough for general purpose SIMD computation.

For the TRANSPAR system design, the pitch of the optical channels imposed a limited area of 125 μm x 250 μm per PE. In utilizing the available area, we had to find an optimal tradeoff between the amount of processing power and the amount of memory in the PE. We chose to use a powerful, yet small-area architecture for the processing logic, to leave more space for memory. We used a bit-serial arithmetic and logic unit (ALU) [1], in which the processing of a multiple-bit word is done bit by bit, in a serial fashion. This approach requires many clock cycles per operation, compared to a single clock cycle per operation for a full bit-parallel implementation, but the simplicity of the ALU allows a higher clock rate. In a full bit-parallel architecture, processing, a long word or a short word take the same amount of time, while in our case the short word can be processed faster. This is because the clock period for a parallel ALU needs to be reduced to allow for the delay through the relatively complex structure, even if only a small number of bits are processed in a given instruction.

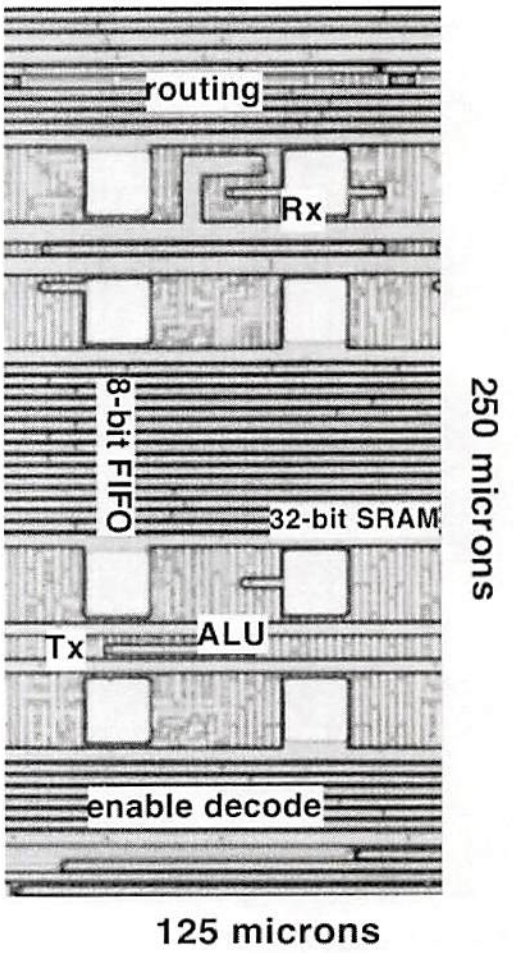

Figure 6.1. Microphotography of a pixel, showing the details of the design.

The TRANSPAR contains a 4 x 8 array of smart pixels (PEs). Figure 6.1 shows the physical layout of a single PE. Each pixel is

mesh connected to its north, east, west and south neighbors. To allow row-parallel I/O with the PE array, the PEs on the east border of the array receive electrical data from wire-bond pads, and the PEs on the west border send electrical data out to wire-bond pads. All other border data inputs of PEs are grounded. Each PE also contains a dual-rail optical receiver and transmitter for loading and unloading the array of data using 2-D parallel optical channels. The PEs in the array operate on data synchronized with a common chip clock and through instructions broadcasted to all PEs in a SIMD fashion.

The logical schematic of the PE smart pixel is shown in Fig. 6.2. Each PE contains: 32 bits of SRAM; an 8-bit FIFO buffer; three registers (RA, RB and RC); the bit-serial ALU;; conditional execution logic (validated by the bit in RC); multiplexers for neighborhood routing; an optical transmitter and an optical receiver. The PE operates on 36-bit microinstructions from a finite state machine. Including the neighborhood connections, each PE has a total of 42 electrical inputs, one electrical output, one optical input and one optical output. The following describes each of the PE sub-circuits.

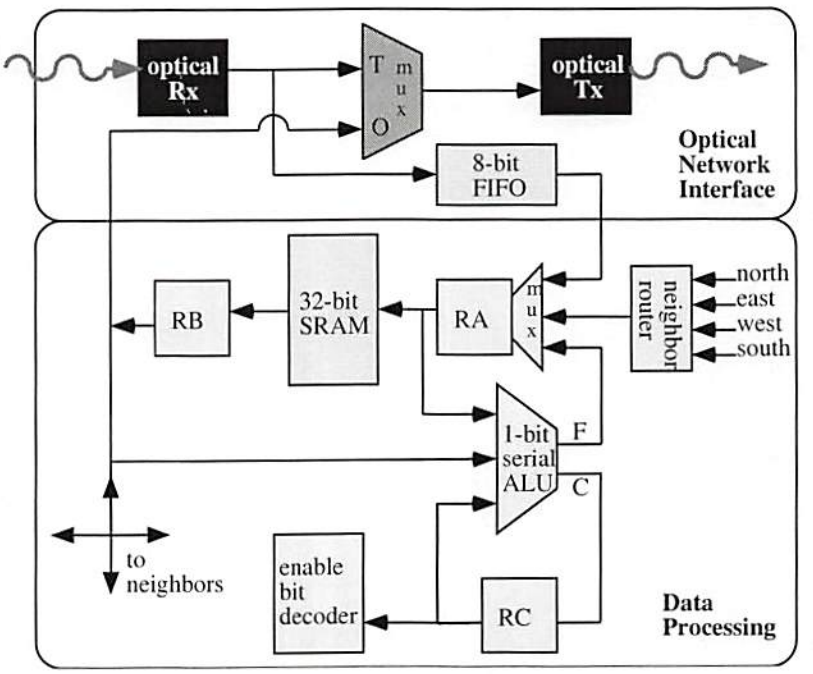

Figure 6.2. Block diagram showing the functional blocks inside a processing element.

The processing engine of each PE is the bit-serial ALU that can implement any binary function of two bits. ALU operations use the two registers RA and RB, while the register RC is used to store the carry bit for multiple bit arithmetic operations. The bit in register RC can also be used to execute conditional operations with the PE. The PE performs the indicated operation only if the bit in RC is zero and the MASK bit is set. This allows only a subset of the PEs to execute an operation – for example in case of arithmetic overflow, only the PEs in which overflow has occurred will perform a cleanup sequence.

The data storage in each PE includes two memory blocks. One 32-bit SRAM memory is used as general-purpose storage for the source and the destination of the ALU operations. A second SRAM block, only eight bits deep, is used as elastic buffer for the asynchronous packet reception from the optical network. Both memory modules are based on a standard six-transistor SRAM cell design with pre-charge and sense read-out. Each SRAM block has a single bi-directional port, used for both input and output.

To minimize the number of input signals, the address locations of memory are not directly visible outside the TRANSPAR node. The only way to access the memory is through the finite state machine (FSM), using a pointer-based access. Three registers store two source addresses (RS1 and RS2) and one destination address (RD), which are used when performing ALU operations. A fourth register stores the length of the words to be operated on. Thus an ALU operation deals with an array of bits in the memory starting at the address pointed to by either the source or the destination addresses and of length specified in the word length register (Fig. 6.3). Effectively, this allows the operation on words of variable length, up to the length of the memory (32 bits), when coupled with a bit-serial ALU and a finite state machine for instruction serializing.

Many all-electronic SIMD architectures use a serial ALU [1], for the advantages outlined above. Our design has the advantage that it hides the serial nature of the processing. Using the FSM and the programmed word length (stored in SR) operation over multiple

bits can be accomplished in a single macroinstruction, allowing extremely high on-chip throughput. Moreover, operating on adjacent memory locations can further increase the throughput. If four images are stored in the memory as successive bitmaps (for example each bitmap eight bits deep), a simultaneous operation can be performed over all the bitmaps by setting the word length to be 32 bits. This assumes that no overflows occur (which would propagate as carry between different bitmaps); this assumption is in general satisfied, because an overflow is not acceptable for a single bitmap taken separately.

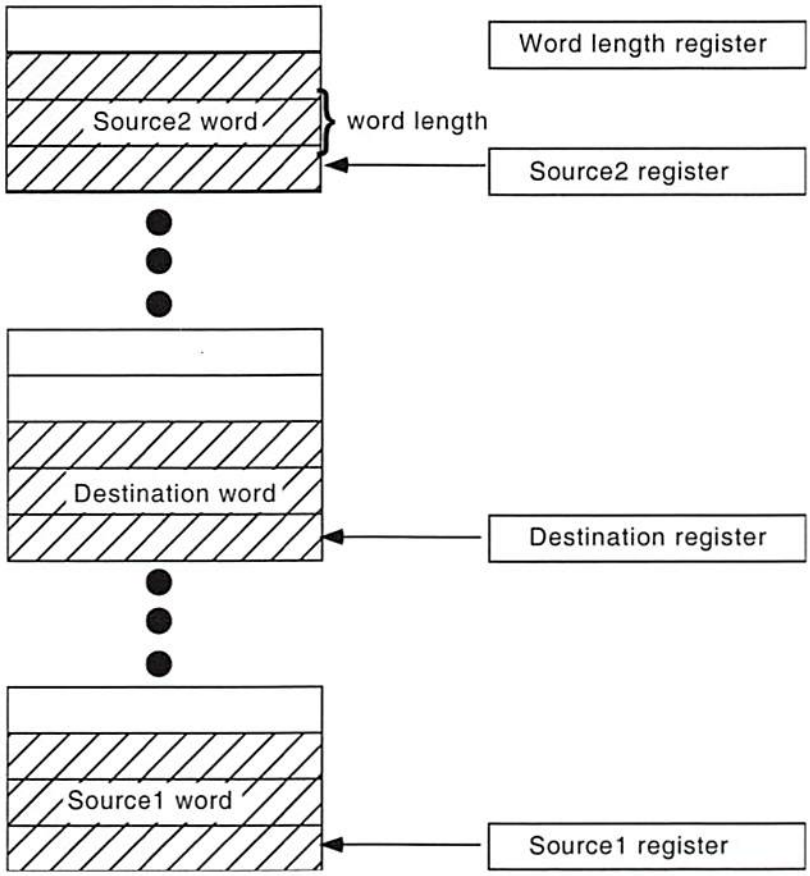

Figure 6.3. Pointer based memory access. Three words are pointed to by two source and one destination register. The shaded regions indicate the memory space of the words to be operated on.

6.3. Electronic interface to the host computer

The second direction for optimizing the architecture is the electronic interface between the SIMD array and the host computer. This interface is required to pass a relatively large number (tens, up to a hundred) of parallel channels at (ideally) large data rates. Unfortunately, this interface has always been a bottleneck. On-chip clock speeds have steadily increased, following Moore's law, while the off-chip bus speeds are trailing almost an order of magnitude behind [2]. The reason for this is the difficulty of sending high-speed electronic signals in parallel over relatively long distances (of the size of a computer box) without skew and distortions.

6.3.1. Using a FIFO at the interface -- off-line operation

Even when using optoelectronic links, most authors use an on-chip FIFO to demonstrate high-speed operation. This is because the host computer is unable to drive the off-chip lines at the speed at which the on-chip optical lines can operate. The approach is to fill up the FIFO at the (low) speed of the host interface, then fire up a fast on-chip clock and send a burst of data from the FIFO through the optical channels. Using a FIFO is acceptable in proof-of-principle demonstrations and possibly also in some practical applications, but it is not a generally acceptable solution. The data transfers can only occur in bursts, when the full FIFO is discharged at high speed. Moreover, no communication with the chip can be implemented while the FIFO is being discharged (for example no branching can occur based on events that are triggered by the contents of the FIFO). In our demonstrator system, we use a finite state machine (FSM) at the interface, which allowed us to achieve high-speed operation running continuously, rather than in bursts, and allowing sustained communication between the chip and the host computer.

6.3.2. Finite state machine as a real-time interface buffer

The TRANSPAR chip uses an on-chip FSM as a clock-speed transformer at the interface [1]. The FSM reads one macroinstruction from the (slow) host and sends multiple microinstructions to the (fast) on-chip logic (Fig. 6.4). The FSM is able to process instructions in real time, which allows for much more flexibility than if using an off-line (FIFO-based) approach. Additionally, because of the encoding of multiple microinstructions in a single macroinstruction, this is equivalent to data compression on the host-to-chip links.

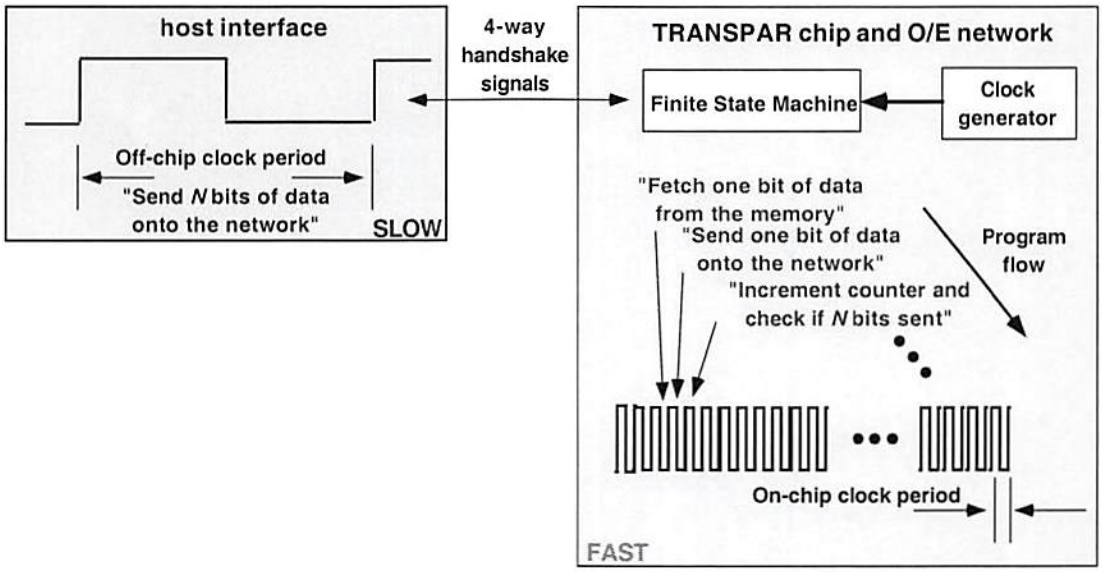

Figure 6.4. The finite state machine is used to match the on-chip and off-chip clock rates and to allow the optical network to operate at the on-chip clock rate.

6.3.3. Speed partitioning of the computer

As we described, the bottleneck between the on-chip and off-chip worlds occurs only in the electrical domain. We can eliminate this bottleneck and at the same time make best use of the fast optical interconnects if we partition the fast and slow domains and use the FSM as the only link between them. The fast domain includes the on-chip lines (which can be driven fast, because they are short) as well as the optical interconnects, driven by on-chip

drivers. Unlike electrical links that are rate limited when exceeding a certain length, the optical links are rate insensitive over the scale of a computer box [3]. The slow domain includes the off-chip logic, which may include long wires that would limit the transfer rate.

Using a FSM at the interface removes the bottleneck, because these two domains do not need to operate at the same speed. The FSM converts from one complex macroinstruction to a series of simple microinstructions, operating at the fast on-chip clock rate, generating a firehose of data and instructions. Moreover, in contrast with all-electronic SIMD architectures like the Connection Machine which allow fast processing but are I/O limited,, the smart pixel implementation has the advantage of allowing fast parallel data transfers on the optical network (Chapter 6) at a rate that matches the on-chip processing power.

6.3.4. Instruction firehose and the optimum rate conversion

The rate conversion performed by the FSM is dictated by the ratio of microinstructions per macroinstruction that the architecture allows. If each macroinstruction can be decomposed into N_{rate} microinstructions, then the on-chip clock could be N_{rate} times faster than the off-chip clock. Statistically, the ratio of the clocks will be designed to a particular value, $N_{effective}$. Because some macroinstructions may be equivalent to a large number of microinstructions while others may be equivalent to a small number of microinstructions, either the chip or the host may have to wait. For example, for complex or long macroinstructions, the actual ratio between the clock rates, $N_{effective}$, may be lower than the ratio of microinstructions per macroinstruction, N_{rate}. In this case the chip will take longer than one external clock cycle to execute the macroinstruction. Conversely, for short macroinstructions the chip will finish early and wait for the next macroinstruction. Clearly, for optimum results, the lengths of the macroinstructions should be similar and the ratio of the clock rates should be slightly larger than the largest N_{rate}.

6.3.5. Experimental demonstration: the TRANSPAR-host interface

The TRANSPAR chip includes an FSM that acts as a buffer between the fast circuitry (the on-chip electronic logic and optical networking circuitry) on one hand and the slower circuitry (the off-chip interface) on the other hand [1]. The 0.5 μm technology used for TRANSPAR is extremely fast, with the gate delay of approximately 150 ps. Also, the optical I/O devices (MQW modulators and MQW receivers) have been shown to operate at data rates exceeding 1 Gb/s [4]. However, the electronic transfer rate at the pins of the chip is much lower. Connecting the TRANSPAR through a printed circuit board and an interface to a general purpose commercially available computer limits the communications speed to tens of Mwords/s. Using the FSM as a buffer allows the use of an on-chip clock rate of 125-250 MHz even if the interface transfer rate is much slower. Because the optical network operates at the on-chip clock rate, data transfers can be much faster than if using a direct connection to the host-computer.

The FSM operates on-chip, at the on-chip clock rate of 125-250 MHz. It reads macroinstructions from the off-chip interface using a slow asynchronous four-way handshaking protocol. The FSM then converts each macroinstruction into multiple microinstructions that are applied to the on-chip logic and executed at the on-chip clock rate. For this, the FSM connects to the control lines of the PEs, which are not directly accessible from the chip pins. Due to the bit-serial pixel architecture and due to the local data access, each microinstruction can be performed very fast, allowing the use of a fast on-chip clock. The FSM executes long sequences of such microinstructions in parallel on all PEs, achieving a very high computational throughput for the on-chip data.

We chose a four-way handshaking between the chip and the host interface to allow reliable bit-parallel communication between the on-chip and the off-chip circuitry. The FSM uses two lines for handshaking: the 'instruction ready' (IR) line from the host and the

'chip ready' (RDY) line to the host. When the RDY line is high, the TRANSPAR is requesting a new macroinstruction. In response, the host interface asserts the IR line to signal when the macroinstruction is available for the chip to read. When the chip has read the macroinstruction, it resets the RDY line. In response to RDY going low, the interface resets the IR line. The whole cycle repeats when the execution of the current macroinstruction is completed. This way, TRANSPAR receives and executes exactly one macroinstruction per handshake cycle.

A macroinstruction can be of one of four types: "load source/destination address registers," "load state register," "bypass FSM," and "execute" microinstructions (Fig. 6.5). The "load" instructions, used for loading data from the electronic pins of the

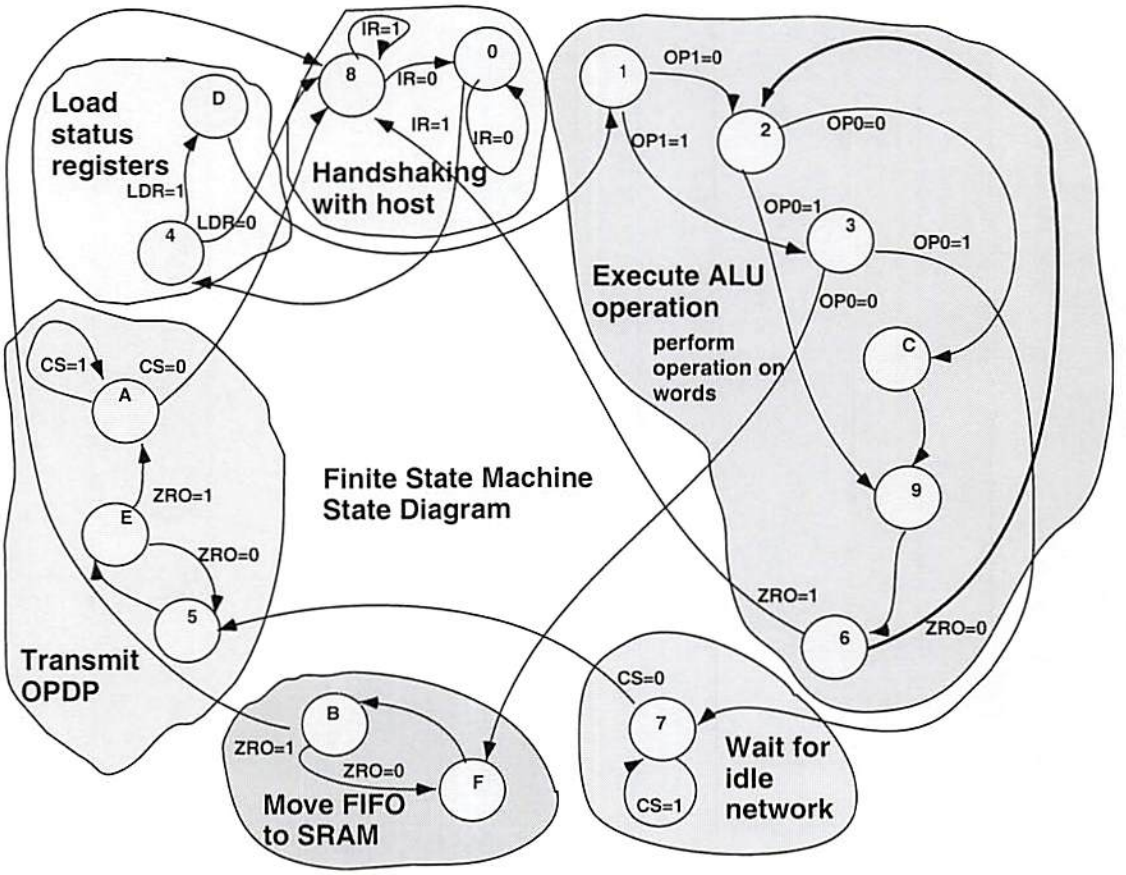

Figure 6.5. Transition diagram for the finite state machine, showing the internal flags tested at each transition, as well as the main functional groups of states.

chip, are less efficient, because they convert to a small number of microinstructions. To load the whole array, a large number of macroinstructions need to be sent from the host to the chip. Fortunately, for a well-written program, load operations should only occur infrequently, when a change of context is recorded in the SR. The "bypass FSM" instructions are used for testing and debugging. The "execute" macroinstructions are the most used and are optimized for achieving high computational on-chip bandwidth. "Execute" macroinstructions perform powerful high-throughput SIMD processing on chip, or send parallel optical packets between chips on the optical network.

6.4. Clocking strategies

Because of the FSM, the on-chip logic can be run at clock rates an order of magnitude above the off-chip clock. For testing purposes, the on-chip clock may need to be slowed down, so that the on-chip lines can be monitored at selected test points. Additionally, multiple chips may need to be synchronized with an external clock much slower than the chip clock [5]. To cover all these possible cases, a variety of clock rates are required and smooth switching between the different rates must be done on the fly. In this subsection, we exemplify a possible clocking design with our TRANSPAR chip.

We designed a variety of clocking circuits on the TRANSPAR chip (Fig. 6.6). The timing of the chip can be obtained either from an internally generated clock (based on a ring voltage controlled oscillator - VCO) or from an external source. The internal clock is tunable using an analog input voltage. According to the HSPICE simulation, the frequency range is 250–500 MHz for a tuning voltage of 2–5 V. The on-chip frequency divider can provide a clock range of 125–250 MHz. The external clock source can cover the lower range, 0-125 MHz. This external clock can be applied directly as the chip clock or can be frequency multiplied on-chip using a phase locked loop (PLL) and digital frequency dividers (T flip-flops); the frequency multiplication factor is 16. Additionally, a frequency

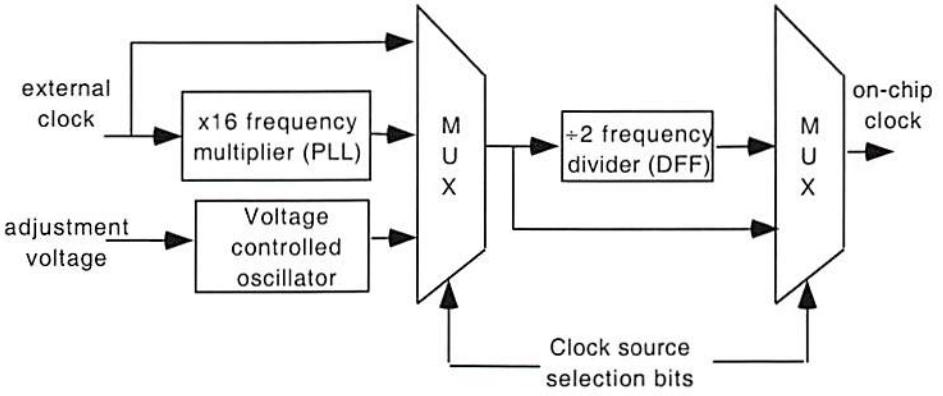

Figure 6.6. Architecture of the clocking circuitry.

divider can be used for both the internal and the external clock sources to halve the clock rate, for even more flexibility. This allows the chip to operate under a wide range of clock rates, for testing, as well as for interfacing with the off-chip logic. A comparison between the simulated clock rates and the experimentally measured values is shown in Fig. 6.7.

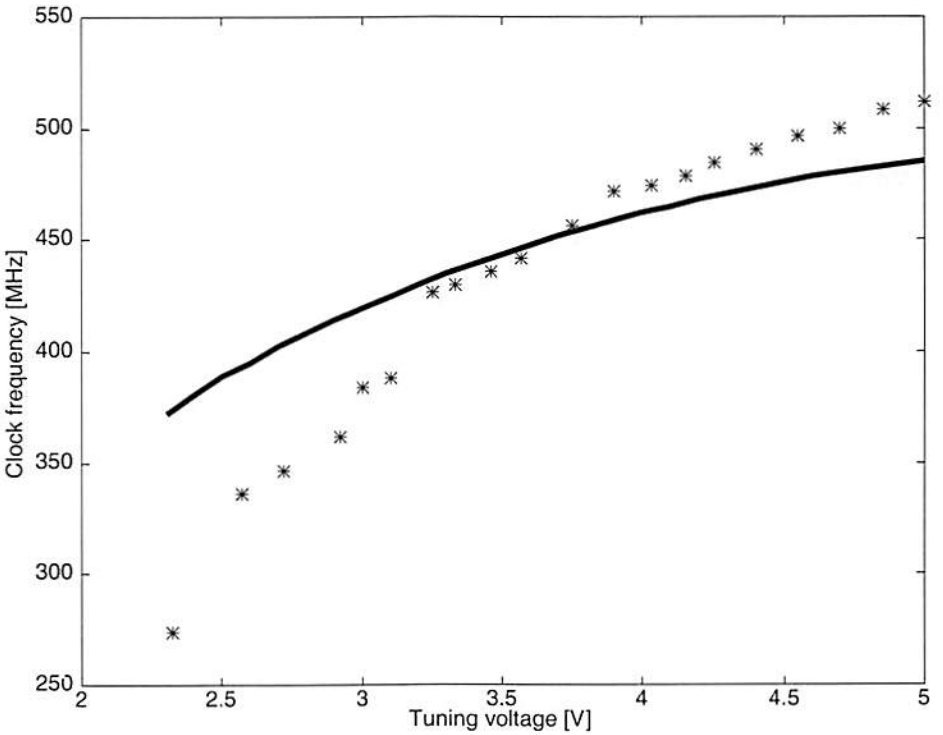

Figure 6.7. Comparison between H-Spice simulation (solid) and experimental clock rate (data points) as a function of tuning voltage of the tunable on-chip clock generator.

The programming of the timing block is done using the status register (SR). After an asynchronous reset the clocking is based on the external clock, without the frequency multiplier, to ensure a reliable start of the on-chip logic. In order to use the other clocking options, the SR must be subsequently updated with the desired values. To select the clock speed and the clock source (internal or external), the SR value may be changed at any time during the operation of the chip. The clock distribution circuitry ensures that the clock rate is changed smoothly, i.e., no clock pulses are clipped during the transition period. During normal operation, one of the chip pins is dedicated for extracting the on-chip clock for measurement purposes or for synchronizing off-chip circuitry.

6.5. Summary

In optimizing the optoelectronic architecture, the electronic circuitry must receive considerable attention. When using large arrays of PEs, the receiver design should consider the larger amounts of noise and crosstalk that are generated by the many circuits switching simultaneously. A good design for the interface between the on-chip and the off-chip domains will not create a bottleneck between them. When using such an architecture, multiple clocking mechanisms must be provided, to allow multiple options in testing as well as in operating the chip. Finally, a bit serial architecture is the most appropriate for fine-grain computing architectures, allowing at the same time smaller real estate and faster operation.

References

[1] D. H. Hillis, *The Connection Machine*, The MIT Press, Cambridge, MA, 1985.

[2] Semiconductor Industry Association, *The national technology roadmap for semiconductors*, 1993-1994.

[3] D. A. B. Miller, "Physical reasons for optical interconnection," *International Journal of Optoelectronics*, vol. 11, no. 3, pp. 155-168, 1997.

[4] T.K. Woodward, A.V. Krishnamoorthy, A.L. Lentine, and L.M.F. Chirovsky, "Optical receivers for optoelectronic VLSI," *IEEE Journal on Selected Topicsin Quantum Electronics*, vol. 2, no. 1, pp. 106–116, 1996.

[5] H.B. Bakoglu, *Circuits, interconnections, and packaging for VLSI*, Reading, Massachusetts, Addison Wesley Publishing Company, 1990.

Chapter 7. Extension to multiple pipelined SIMD planes: the TRANSPAR network

One of the strengths of smart pixel technology is the capability to optically cascade multiple SIMD chips, interconnected with arrays of optoelectronic devices. This approach can create massively parallel pipeline architectures, with extremely high throughputs. In this chapter we consider the implications of cascading such SIMD arrays into a three-dimensional (3-D) computing mesh.

7.1. Combining SIMD with optical parallel packet-switched networks

For the TRANSPAR demonstrator system, the functionality of the pipelined SIMD processing and optical parallel data packet (OPDP) transfer can be combined to create a multi-node SIMD processor. This architecture can be viewed either as a 3-D mesh or as a pipeline processing system.

For the 3-D mesh architecture, each node is a mesh-connected SIMD processing array and data transfers between nodes are made using the 8-bit deep, 8 x 4 bit wide data packets, based on the asynchronous carrier sense multiple access with collision detection (CSMA/CD) protocol. The optical network connecting each PE with its corresponding neighbors on adjacent chips implements the third dimension of the mesh. Because the network can accommodate up to six nodes, it is equivalent to a 6 x 8 x 4, 3-D array of SIMD processors. This system can perform very fast parallel processing of 3-D data fields, as required in image/video processing or packet header recognition and routing.

Alternatively, the cascaded PE arrays can be viewed as stages in an array pipeline system. Each stage processes a parallel array of optical bits then sends the packet to the next stage in the pipeline for further processing. Such an architecture that combines the low latency of pipeline processing with the high throughput of parallel processing can have

interesting applications in real-time image processing or in very computationally intensive applications.

7.2. Architecture of the network interface

The TRANSPAR nodes on a network are physically arranged in a ring (Fig. 7.1). Host processors are attached to the nodes and data transfers occur asynchronously via OPDPs under the control of a CSMA/CD protocol. The TRANSPAR CSMA/CD protocol is modified from the usual Ethernet and operates over a ring network that passes spatially parallel packets. The latency per node, including both the propagation time through the node and the optical propagation between nodes is less than 4 ns.

Network nodes operate asynchronously, at on-chip rates of hundreds of MHz. The interface with the host computer operates also asynchronously, but much more slowly, allowing for an inexpensive packaging and a simple design for the printed circuit board for the TRANSPAR nodes. For asynchronous operation, the packet, which includes a clock

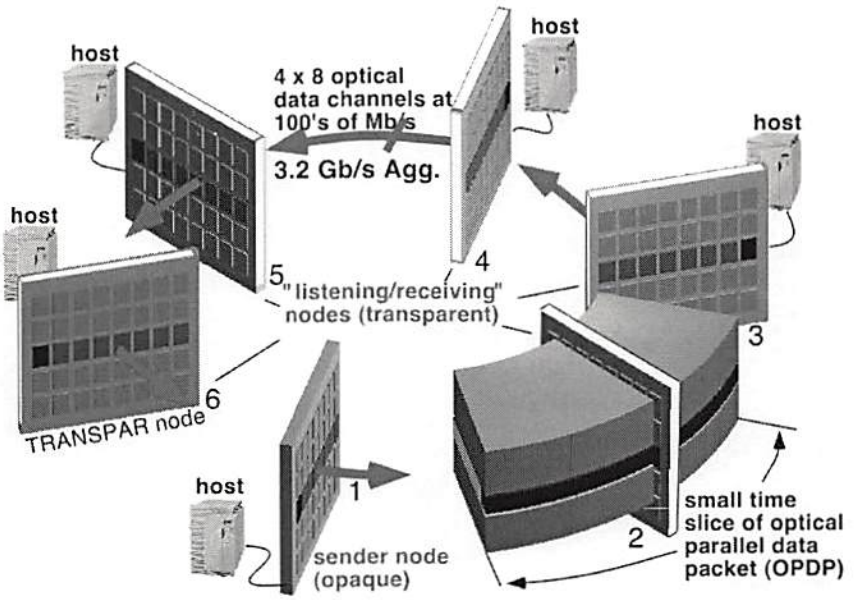

Figure 7.1. Concept drawing of a TRANSPAR network with six nodes, showing a portion of an optical parallel packet propagating translucently through a node.

channel, is received into an 8 bit deep elastic storage first-in-first-out (FIFO) buffer. The FIFO buffer is usually employed at the interface between two systems which operate at different clock rates (here, the sender and the receiver chips).

Each PE contains a network interface (Fig. 6.2), including an optical receiver (Rx), an optical transmitter (Tx), and an 8-bit FIFO buffer. This interface section of the PE uploads or downloads OPDPs to or from the optical network using the modified CSMA/CD protocol. The packets consist of eight frames, transmitted on eight separate clock cycles. Each frame contains an array of 4 x 8 bits of data, three source and three destination address bits (sent continuously over the duration of the packet) and a clock channel (used to read in the data at the destination node).

The PEs operate in one of two modes. Unless the PE is uploading an optical data packet to the network, the PE is in *transparent* mode. In transparent mode, any signal entering the Rx passes directly on to the Tx, through only a few gates, incurring only a small delay (~3 ns). When uploading an optical data packet onto the network, the PE is in *opaque* mode. In this mode, signals entering the Rx are blocked, and the PE sends data from the 32-bit SRAM onto the network. Ideally, only a single TRANSPAR node on the network is uploading a packet at any given time. In this case, the transmitted packet travels through all other network nodes almost instantaneously, since they are transparent. Only the node whose address matches the destination address (encoded in the OPDP) downloads the OPDP, using the clock channel included with the packet.

A node with a packet to send must wait until the optical network is idle before uploading an OPDP. However, there is a chance that two or more TRANSPAR nodes will detect an idle network at nearly the same moment and upload OPDPs that will cause contention (or a *collision,* in Ethernet terms). The TRANSPAR node contains circuitry to detect such a collision and reset the network.

The physical layout of TRANSPAR network interface contains: carrier sense module, collision detection module, source address pixel, destination address pixel, optical clock pixel and FIFO control. The architecture of the carrier sense module and the collision detection module will be described in Section 7.3, when the network functionality of these modules is discussed. We now present the functionality of the other network modules in the TRANSPAR node.

7.2.1. Source address pixel

Three source-address pixels in each frame identify the sender node. These source-address pixels have four states: idle, send, silent, and jam. When *idle*, they simply remain transparent and transmit the incoming source address from the network. When the node wishes to send a packet, the source-address pixels become opaque and *send* out the address of the node. When finished sending the packet, the pixels remain opaque during a silent period, when they send out zeros to remove the packet data from the network. Finally, in the case of collision, the source address pixels send out a *jam* signal to the network, to ensure that all nodes have sensed the collision. As explained in Section 7.3.4, this is needed because only the sender nodes can identify a collision reliably.

7.2.2. Destination address pixel

Three destination address pixels in each frame identify the destination node. Every node on the network compares the incoming destination address with its own address to see whether there is a match. The match is detected on the destination address bits in parallel, within the first cycle of the clock transmitted with the data packet. This contrasts with serial network methods that require several clock cycles to determine the packet address. When an address match occurs, the node activates its FIFO control and downloads the incoming data packet.

7.2.3. Optical clock pixel

The TRANSPAR frame contains an optical clock channel that defines the bit rate of the data packet. The optical clock pixel has three states - idle, send and match. When *idle*, the pixel is transparent and retransmits the optical clock from the previous node to next node. When in the state of *send*, it sends out an optical clock defined by the on-chip clock. The clock is straddled with the data, ensuring that the receiver node will sample the data when it is most stable (Fig. 7.2); this clocking scheme is also more tolerant to clock-data skew. Finally, the clock pixel is in the state of *match*, when there is an address match. In this

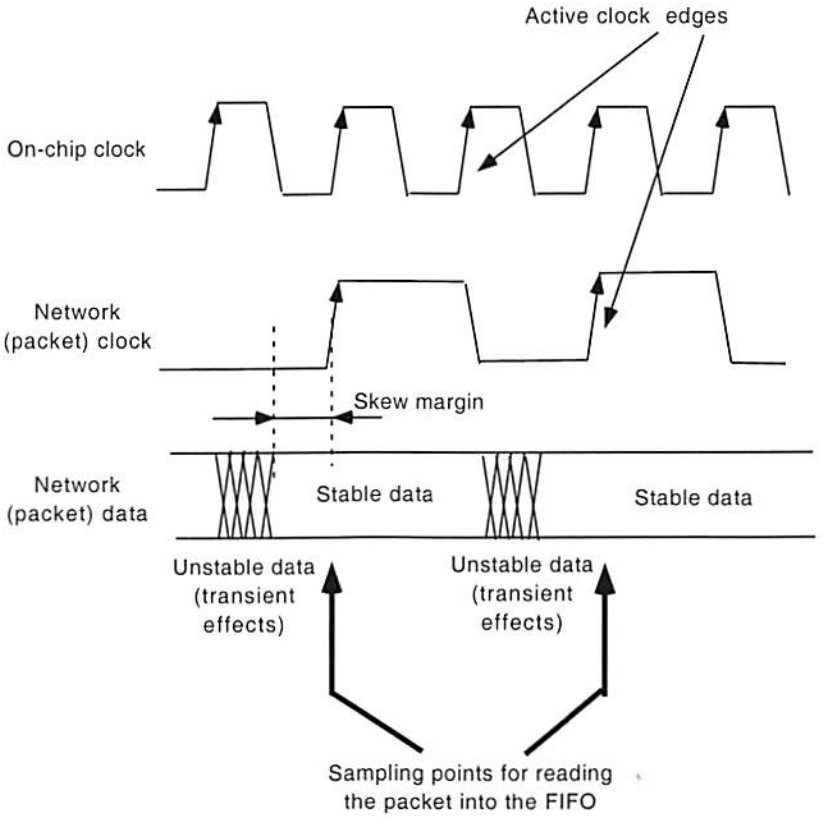

Figure 7.2. Data sampling at the middle of the bit period ensures that the data received is detected at an optimal signal level, away from the transitions at the beginning and the end of the bit period. This offers maximum immunity to clock jitter and skew.

case, the optical clock pixel directs the incoming clock to the FIFO control for data download.

7.2.4. FIFO control

The PE in the TRANSPAR node has an 8-bit deep FIFO buffer for data packet download. This FIFO is controlled by the FIFO control, ticking at the rate of the incoming optical clock. The received packet is written in the FIFO buffer with the incoming packet clock and then read out by the PE's with the local, on-chip clock. Therefore no global clock synchronization is required across network nodes.

7.3. Ring Interconnected Network with CSMA/CD

The TRANSPAR networking is designed for interconnecting nodes with SIMD arrays on a high-speed local area network (LAN). The LANs are, by definition, networks covering a small area a few feet to a few hundred of meters in diameter. Such networks are used principally to interconnect terminals, computer nodes, and various intelligent systems within a building or a campus. The TRANSPAR network is configured as a ring connected topology embedding the Carrier Sense Multiple Access with collision detection (CSMA/CD) protocol.

7.3.1. Ring interconnected network

The configuration of a LAN is generally limited to either the bus or the ring topologies. In the case of the bus, a node transmits the signal over the network in both directions simultaneously. Terminations at each end of the bus absorb the signal, avoiding multiple reflections on the network. For the case of the ring topology, one-way transmission or two-way transmission (as in FDDI) can be used. The physical medium of the network can be coaxial cable, twisted pair wire, optical fiber or any medium that can carry signals. In

the case of TRANSPAR, we use CMOS multiple quantum well (MQW) smart pixels technologies to generate, modulate, and receive digital signals that propagate in free space.

7.3.2. Random Access

The random access scheme CSMA/CD, which is the basis of Ethernet, is one of the most popular physical layer LAN protocols used today, because it strikes a good balance between speed, cost and ease of installation. These strong points combine with wide acceptance in the computer marketplace to make Ethernet an ideal networking technology for most computer users today.

For optical networks, we believe that the main advantage of CSMA/CD is the random access. Optical networks are capable of sending data at GHz speeds, when global synchronization would be impractical. With random access, networks can use nodes that operate asynchronously and possibly with widely different clock rates.

7.3.3. Carrier Sense Multiple Access

The CSMA/CD protocol we are using is a slight modification of the Ethernet protocol. The random access to the shared medium is based on a carrier sense mechanism. TRANSPAR senses the carrier by detecting the three-bit wide source address. The "all zero" source address signals an idle network; otherwise the network is busy. The design is simple, allowing the carrier detection circuitry to simply use the logical OR of the incoming source address bits as carrier sense. Because each packet contains an optical clock, the source address bits are sampled with this packet clock at the receiver chips, allowing asynchronous operation of the source and destination chips.

When a node has data to send, it first listens to the channel to see if any other node is transmitting at that moment. If the channel is idle, the node sends the data right away. If the channel is busy, the node waits and tries to send again when the channel becomes

available, as in the 1-persistent CSMA/CD [1]. Each network packet contains eight frames. Each frame contains a 3–bit wide source address header and a 3–bit wide destination address header. In our particular implementation, the network can accommodate up to six independent nodes with addresses from "001" to "110". Each node has the same priority to access the network. Each node listens to the network by sensing the source address bits. The node considers the channel idle if all zero source address bits are detected; otherwise, the channel is busy.

7.3.4. Collision detection

Collisions happen when two or more nodes try to transmit data at the same time. Because of the non-zero propagation delay between nodes, two or more nodes could sense the channel as idle and begin sending data at the same time, causing a collision. When a collision happens, all colliding packets are garbled. The nodes involved must step back for a random length of time and re-transmit the collided packets later. The challenge of collision detection is to have every node detect the collision automatically by listening to the channel.

Normally, only one node on the network is opaque (sending) at any given moment, and all other nodes are in transparent mode. Because of this, the sender expects to receive its own packet after a round-trip delay. If a collision has occurred, a second node is opaque and transmitting a packet on the network, so the first node will receive a packet with a source address different from its own. This is a very reliable mechanism for detecting a collision, but it is only accessible to the sender.

To ensure that all other nodes are aware of the collision, the sender node broadcasts a jamming signal. The physical reason of this signal is very different from the Ethernet implementation of CSMA/CD, where the network signals are analog and the jamming signal ensures collision detection in the presence of widely different power levels in the

colliding packets. Because of the unidirectional ring in our implementation, signals from two colliding packets do not interfere in the channel medium, so it is not a question of signal dynamic range; only one (digital) packet can propagate between two transmitters on the TRANSPAR network, so the dynamic range is very low. The transparent nodes in-between have no way of deciding whether this packet was entirely sent by a single node (a valid transmission) or made up of frames from more than one transmitting node (collision). It is the task of the sender nodes to inform all the nodes on the network about collisions.

Figure 7.3 shows an example of collision detection, which assumes propagation delay of 5 ns per node and packet length of 40 ns. Node 1 tries to send a packet to node 3. After 10 ns, node 5 senses the channel to be idle and tries to send a packet to node 2. Node 1 detects the collision at 20 ns when it receives packet from node 5. Also node 5 detects the collision at time 20 ns when it receives packet from node 1.

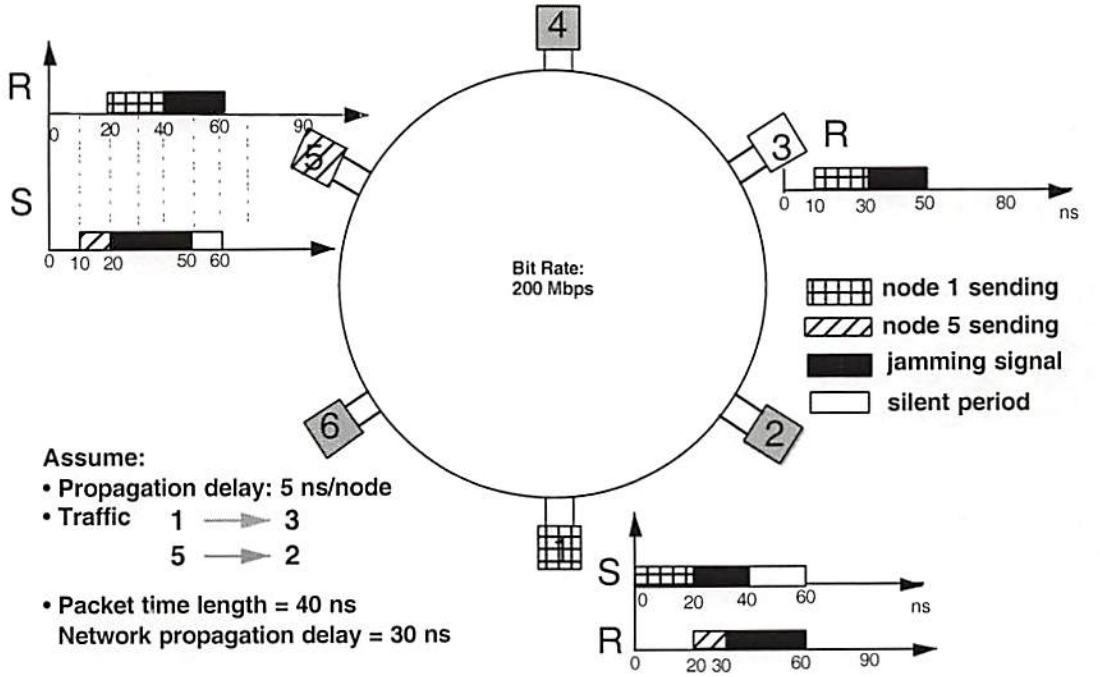

Figure 7.3. Scenario of collision detection on the TRANSPAR network. The propagation delay is 5 ns per node and the packet length us 40 ns.

7.3.5. Packet removal

Each node that transmits a packet on the network is responsible for removing the packet after a round-trip propagation, as in the case of FDDI. For this, the node remains opaque for a period after the end of the packet. We call this silent period, because the node sends out an empty packet, with all data and address bits zero. This silent period is very important. For a node down the line, this empty packet is interpreted as an idle channel. Another node may start transmitting as soon as it senses the silent period. On the other hand, if no other node has data to send, the silent period of the transmitter ensures the removal of the tail of the packet. If the transmitter would become transparent immediately after sending the end of the packet and all other nodes would be already transparent, the tail of the packet (which is still propagating on the network), would continue to propagate forever, keeping the carrier sense always high.

7.4. Transparency as a way of reducing latency – rearrangeable pipelines

In our previous network designs we used a synchronous approach [2, 3], with all nodes operating at the same clock rate, synchronized with a global clock. Thus, if a packet had to be sent from a source node to a destination node, the two nodes would have to be synchronized and programmed, one to send and the other one to receive at the same time. Moreover, if the packet had to propagate through multiple intermediate nodes between the source and the destination, the synchronization process would have to be repeated for each pair of point-to-point connected intermediate source and destination nodes. This way, the packet would be delayed at each intermediate node because it would have to be clocked in and clocked out, as in intermediate stages of a pipeline. Last, but not least, interface with the computer limited the on-chip clock of the network to about 10 Mb/s. All these limitations weigh down the resulting throughput, to very low numbers.

As an example, our SPARCL system [2] uses a 20 Mb/s on-chip clock (limited by the interface with the host computer) and sends one bit frame (a parallel packet of bits) per clock cycle. To send an eight-bit deep packet (as in the TRANSPAR network) through 6 intermediate nodes requires 48 clock cycles, or 2.5 μs per packet at 20 Mb/s. In contrast, the TRANSPAR network operates with the same host interface, at on-chip clock speeds above 100 Mb/s and transmits an eight-bit deep packet in only 12 clock cycles. The transmission time is 120 ns, or a factor of 24 speedup, and even more for larger networks and when considering the possibility of collisions and the synchronization overhead.

Minimizing the network latency involves minimizing two components: the delay of the physical link and the delay due to the network protocol. To minimize the delay of the physical link, the intermediate network nodes between a source and destination should ideally be transparent, allowing a packet to pass through them with minimum latency. While other researchers have demonstrated such architectures using physically transparent nodes [4, 5, 6], the technology available for the TRANSPAR chips did not allow such an implementation. We then chose to design our nodes to be *translucent*, i.e., allowing the packet to propagate with minimum delay, through only an optical receiver, a minimum amount of logic and an optical transmitter. The latency is thus minimized, and is due mainly to the optical receiver (Fig. 7.4).

Figure 7.4 shows an H-Spice simulation of the latency of the optical link as a function of the detected photocurrent at the receiver for a translucent TRANSPAR node. Here, latency is the maximum delay for an incoming optical pulse, from the input of the receiver circuitry, through the digital electronics, and to the output of the modulator driver. The extra delay due to the optical device is negligible for both the transmitter and the detector.

For the plot in Fig 7.4, the capacitance of a single MQW diode is assumed to be 100 fF, as in [7]. With an input power of 100 μW per modulator diode (limited by device

saturation), a modulator reflectivity of 0.1...0.3 and a detector responsivity R = 0.3 A/W, the detected photocurrent is about 6 μA, even considering zero coupling

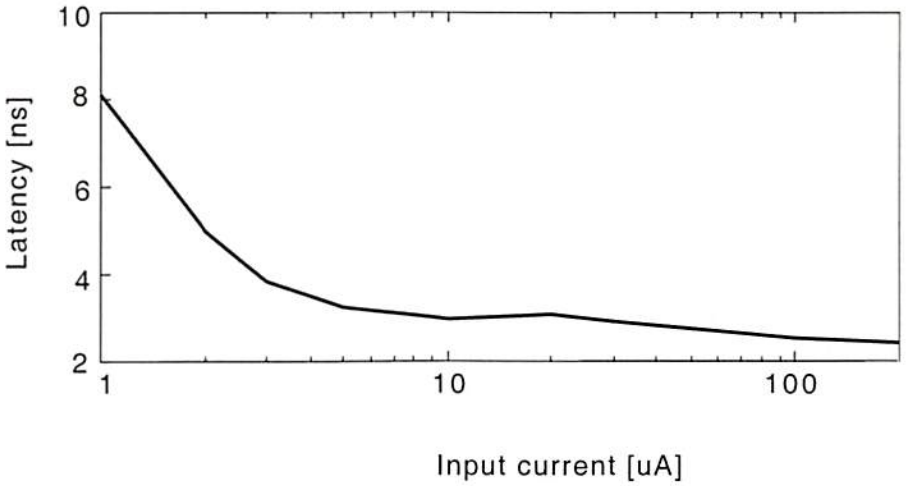

Figure 7.4. H-Spice simulation of the latency of the TRANSPAR optical link versus the photodetected current.

losses between the modulator and detector. The latency is thus close to 3 ns per node.

To reduce the latency due to the protocol, we use asynchronous transfers, which do not require that the sender and receiver nodes be synchronized. The sender must check if the network is idle before sending out a packet, but neither the intermediate nodes nor the destination node must be synchronized with the sender. In fact, all these nodes can operate on their own SIMD processing in the meantime, while the packet propagates asynchronously through the intermediate nodes and is written asynchronously in the FIFO of the receiver node. Moreover, even the host computer of the sender node is not required to wait for the completion of the network transmission. The TRANSPAR chip includes smart logic that can wait for the idle network before sending the packet.

7.5. Comparison between clocked and translucent nodes

For asynchronous operation, there are two variations of timing that can be implemented. The intermediate nodes can clock in the packets (using a clock supplied with the optical packet) before retransmitting the packet; alternatively, the intermediate nodes can be translucent, and simply pass on the packet to the next node. In both cases the logic levels of the packets are regenerated, but only for clocked nodes is the timing of the packet regenerated as well.

Whether at the last node or at intermediate nodes, for best performance, the data bits must be sampled at the middle of the bit period (Fig. 7.2). This ensures the best immunity of the received bits to jitter, because the sampling point is as far as possible from the transitions at the beginning and the end of the bits. The two types of architecture, clocked and translucent have different advantages and disadvantages. A hybrid architecture may combine the advantages of both clocked and translucent nodes and may offer the optimum performance.

7.5.1. Clocked packet transmission

The main advantage of clocked packet transmission is that the timing of the packet is regenerated at each intermediate node. This does not allow skew (differential delays among the channels in the packet) to accumulate across nodes; across the whole network, the total skew accumulated among the channels in a packet is no larger than the skew between two adjacent nodes on the network. At the same time, retiming at every intermediate node is expensive in terms of processing logic, as well as in terms of delay. At each node the packet must be clocked in, then clocked out. The packet clock is used to sample the incoming bits. Following this sampling, the bit period will now start on the active edge of the clock, which is in the middle of the received bit, i.e., half a clock period delayed as compared to the received bit (Fig. 7.2).

The same procedure must be applied at each node on the network, so a packet will accumulate a delay of half a clock period per network node. We showed in Fig. 7.4 the latency for a translucent node. In the case of a clocked node, this latency increases by half the clock period. For an on-chip clock rate of 100 MHz, the latency of a clocked node is more than double the latency of a translucent node. In a large network, after many such clocking stages, the overall delay may be substantial. Additionally, the logic for the clocked nodes is more complex than the design of the translucent nodes.

7.5.2. Translucent packet transmission

For translucent nodes, as the packet propagates through the intermediate network nodes between the source and destination, it is detected, passed on through combinational logic and retransmitted, but not through clocking circuitry. This allows the skew among the different channels to increase and accumulate from node to node. Moreover, if the chips are part of the same fabrication run it is likely that all the PEs on a given position in the array will either be slower or faster than the average. Skew accumulation will occur as in a worst case scenario, rather than in an average scenario. For example, if the capacitance of a detector is largest in the corners of the chip and smallest in the center, the corner nodes are always connected to corner nodes, so the largest delays will accumulate with respect to the center of the chip. For this reason, a translucent architecture may not be efficient for large networks with many nodes, where large amounts of skew may accumulate. A hybrid approach, alternating translucent and clocked nodes may be a much better solution. We will explore such hybrid architectures in a following section.

7.5.3. Comparison of skew for pipeline and translucent nodes

Latency itself is not a major concern on a network, where the propagation delays are not always known in advance. As mentioned above, for a parallel network, skew is the most

critical parameter. Skew between adjacent channels occurs because the physical channels are not perfectly identical. Even though they are designed to have identical parameters, the uncertainty in the fabrication makes them different. Differential delays are caused by non-uniformity across array components in the OE link: the spot array generator optical power, the modulator contrast ratio, the detector responsivity, and finally the receiver threshold.

Figure 7.5 shows an H-Spice simulation of the variation in delay as the detector capacitance or the input photocurrent change. The delay is referred to the ideal design, where the capacitance is 100 pF and the photocurrent is 6 A. This differential delay is what we call the channel skew. To estimate skew, we assume that the design parameters of the devices can vary within a 10% range (conservative estimate!). The skew due to the variation in the diode capacitance is $\Delta t_{cap} \leq 0.05$ns. Assuming the same 10% variation in the uniformity of the spot array generator, the modulator reflectivity and the detector

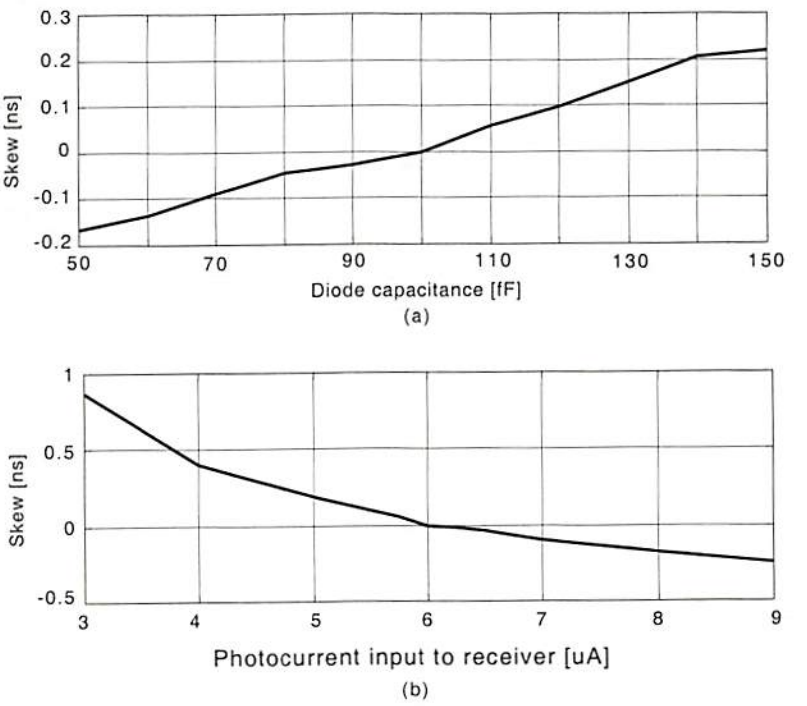

Figure 7.5. Channel skew as a function of the error in the modulator capacitance (a) and in the detected photocurrent (b).

responsivity, the uncertainty in the photocurrent is close to 30%! The skew due to such a large uncertainty in the detected photocurrent is $\Delta t_{photo} \leq 0.4$ns.

The total (worst case) skew *per TRANSPAR node* is $\Delta t_{node} \leq 0.45$ns. For the case of translucent nodes, in a six-node network, the overall skew may reach $\Delta t_{network} \leq 2.7$ns. To minimize the deleterious effects due to the skew, the clock period must exceed about ten times the amount of skew, $T_{clock} \geq 10\ \Delta t_{network}$. This will limit the clock rate to 30MHz! At the same time, in a clocked network only the skew between two nodes must be considered. Thus for $\Delta t_{node} \leq 0.45$ns and $T_{clock} \geq 10\ \Delta t_{network}$, so the clock rate can be as high as 180 MHz, *and is independent of the number of network nodes.* The price to pay for this is the extra delay of half a clock cycle per node, as explained in Section 7.5.1.

To further address the issue of skew, a look at Fig. 7.5 shows that by operating at larger photocurrent values the slope of the latency curve (and hence the skew) will be significantly reduced. This can be achieved by using a higher modulator reflectivity, a higher modulator saturation power and a higher detector responsivity. Using a 20 μA photocurrent as the nominal operating point would reduce the skew due to the photocurrent to $\Delta t_{photo} \leq 0.06$ns. Ultimately, this would allow a clock rate of 150MHz for a translucent network or 900 MHz for a clocked network (this last figure would most likely be limited by the on-chip circuitry to a lower value). On the other hand, if the number of nodes on the network is increased to 100, the clock frequency for the translucent network must again be reduced to about 10 MHz.

Clearly, the translucent architecture can provide very high throughput, but due to the lack of retiming it is limited to a small number of nodes in order to operate at high clock rates. At the same time, the clocked architecture can operate at higher clock rates (where the clock rate is essentially independent of the number of nodes), but with a delay that increases linearly with the number of nodes.

7.5.4. Extensions to hybrid clocked-translucent architectures

We have identified a throughput versus delay tradeoff in choosing the network architecture. A clocked architecture improves the throughput (by allowing a faster clock rate), but also increases the amount of delay. It may happen that using a clocked architecture may allow a clock rate too high for the on-chip logic. In the example above, if the on-chip logic can only operate at 133 MHz and the clocked architecture allows operation up to 180 MHz, the chips will operate at the lower speed. The total delay over N nodes on the network will be $\Delta t = \frac{N}{\frac{133\ \text{MHz}}{2}} = 15.0N[\mu s]$. A combination of the two architectures may significantly improve the network performance.

Indeed, using a hybrid translucent-clocked architecture with alternating translucent and clocked nodes can actually reduce the delay. This hybrid architecture doubles the overall skew, because the skew of a translucent node and of the next clocked node add, then the skew is reset upon retiming in the clocked node. This reduces the maximum data rate by half, down to 90 MHz. At the same time, the number of clocked nodes has decreased to half. The overall delay is $\Delta t = \frac{\frac{N}{2}}{\frac{90\ \text{MHz}}{2}} = 11.1N[\mu s]$, a decrease of 26 %. This result does not include the full effect of reducing the clock rate; the next section includes the effects of the lower clock rate on the offered network load, and on the resulting overall delay.

7.6. Optimal degree of parallelism for maximizing the network throughput

The number of design variables that may be adjusted to optimize the performance is much larger for parallel networks than for serial networks. Some of the design parameters

are particular only to parallel networks, while others may be used for serial networks as well. Four of the most important such parameters are the channel rate, the throughput, the packet size and the number of network nodes.

The *channel rate* is the data rate of a single channel. This is related to the clock rate of the transmitter chip. For the TRANSPAR design, the channel rate is equal to half the on-chip clock rate (as shown in Fig. 7.2). From the point of view of the sender nodes, the *throughput* is the maximum average data rate that can be pumped into the network and delivered reliably, per node and per unit time. The throughput includes the delays due to collisions and retransmission, as well as the propagation time and protocol overhead. As more channels are used in parallel, the data rate increases due to the parallel transmission. The *packet size* dictates the length of time of a packet transmission, which in turn affects the amount of traffic on the network. Finally, the *number of nodes* on the network affects both the roundtrip packet delay and the amount of traffic (for a fixed data rate per node).

In designing a parallel network, all these parameters are dictated by external factors. The channel rate can be changed by adjusting the on-chip clock speed, but is limited to a maximum operating speed. From a practical standpoint, the data rate per node is set by the application. The packet size is dictated by the amount of memory allocated per channel (for example the FIFO buffer of the TRANSPAR chip). The number of nodes may also be constrained by available hardware and by the required connectivity. It is then interesting to consider the optimum number of parallel channels that minimizes the network delay when the network operates with a certain number of nodes, a given data rate per node and at the maximum on-chip clock rate.

In optimizing the network design we use theoretical formulas already published in the literature. Kleinrock and Tobagi have derived the formulas for a serial Ethernet network [1]. They relate the delay (D) to the offered network load (G), for a fixed data rate and packet size. To relate the delay to the throughput, they solve an implicit equation that gives

the throughput S as a function of the offered network load G. We took this analysis one step further, considering the delay for a parallel network, not only as a function of the throughput, but also as a function of the channel rate, number of parallel channels, and number of network nodes. Details of the equations are included in Appendix II.

We fix the rate at which new packets are generated at the source nodes (the throughput), the number of nodes and the channel rate, and we plot the packet delay as a function of the number of parallel channels in the network. Unless otherwise specified, the aggregate throughput is 10 Gb/s (for a six-node network, and scaled correspondingly for more nodes), the channel rate 100 Mb/s, and the packet size 1 Mb. The number of nodes is six, the largest value possible on a TRANSPAR network with the chips we fabricated. This number could be further increased with only minor changes in the chip design.

For a small number of parallel channels, the traffic may exceed the capabilities of the network, and the delay may become infinite. For this reason, some of the curves we plot have no data points in the left portion of the figure. As the number of channels is increased, the packet delay decreases, because the packet transmission time becomes shorter, as more bits are transmitted in parallel on the added channels. Beyond a certain value, the packet becomes too short, and the delay increases again, as the finite overhead (due to physical latency) dominates when compared to the duration of the very short packets.

For a complete picture, we plot both the absolute delay and the normalized delay of a packet. The normalized delay is normalized to the packet transmission time. A minimum *absolute* delay is desirable, because it indicates fast delivery. Conversely, a minimum *relative* delay indicates a minimum overhead. For short packets, the absolute delay may be small, because it is proportional to the packet transmission time. If, on the other hand, the relative delay is large, this indicates that too much time is spent retransmitting packets or signaling.

We show in Fig. 7.6 the absolute and the normalized delay as a function of the number of channels, when the channel data rate (or the on-chip clock rate) is varied. The absolute delay decreases monotonically with the number of channels, and no optimum value can be found. Yet, the normalized delay is minimized for a certain number of channels. As discussed above, a minimum normalized delay is desirable, and it indicates a minimum overhead for an optimum number of parallel channels. This optimum increases almost linearly with the channel data rate. From a practical standpoint, the number of channels is fixed once the chip is fabricated, so the clock rate can be fine-tuned to minimize the delay for the given number of channels. Another interesting observation is that the magnitudes of

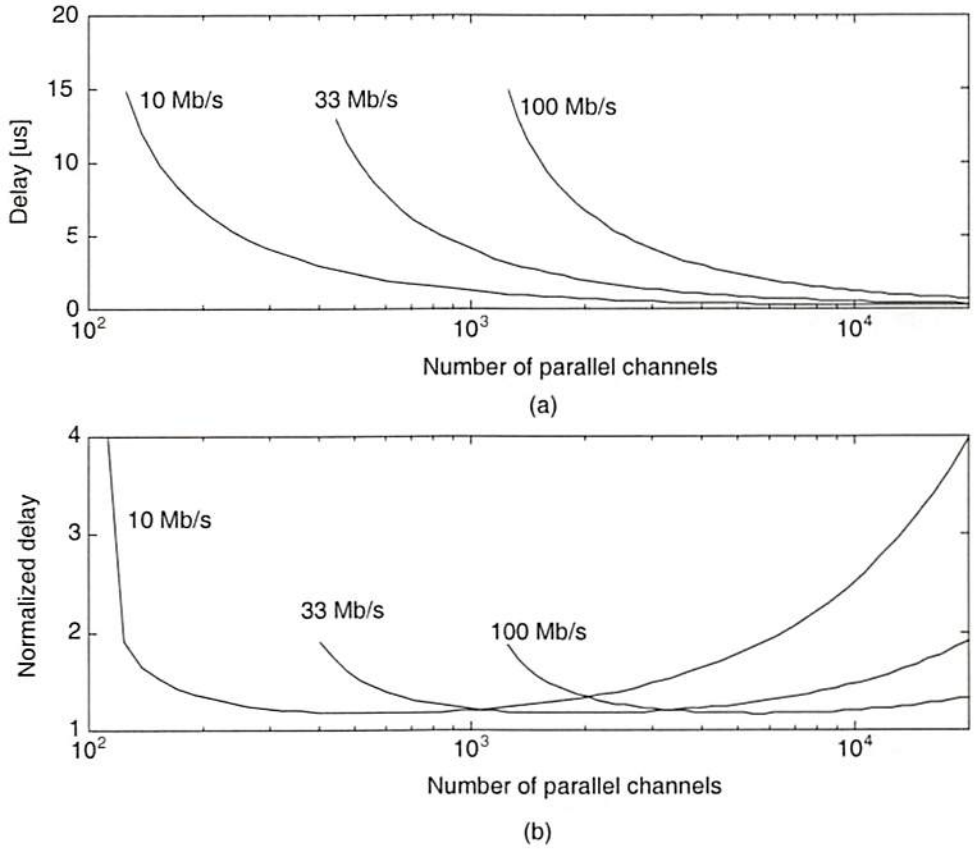

Figure 7.6. Normalized delay (a) and absolute delay (b) versus number of channels for a parallel packet Ethernet. Multiple curves correspond to various chip clock rates (channel data rates). The delay normalization is to the packet transmission time.

both the absolute and the relative delays (at the minimum normalized delay) remain relatively constant for different channel rates.

A rather different scenario occurs if increasing the number of nodes (Fig. 7.7), which increases the roundtrip propagation delay (affecting the traffic on the network indirectly) and increases the traffic itself, because more nodes are present and may need to transmit packets. Figure 7.7 shows that even a modest increase in the number of nodes can cause a large increase in the packet delay and can require many more parallel channels for acceptable performance. For small numbers of channels, an increase in the number of nodes may render the network unusable, while for larger numbers of channels, the performance degrades more gracefully.

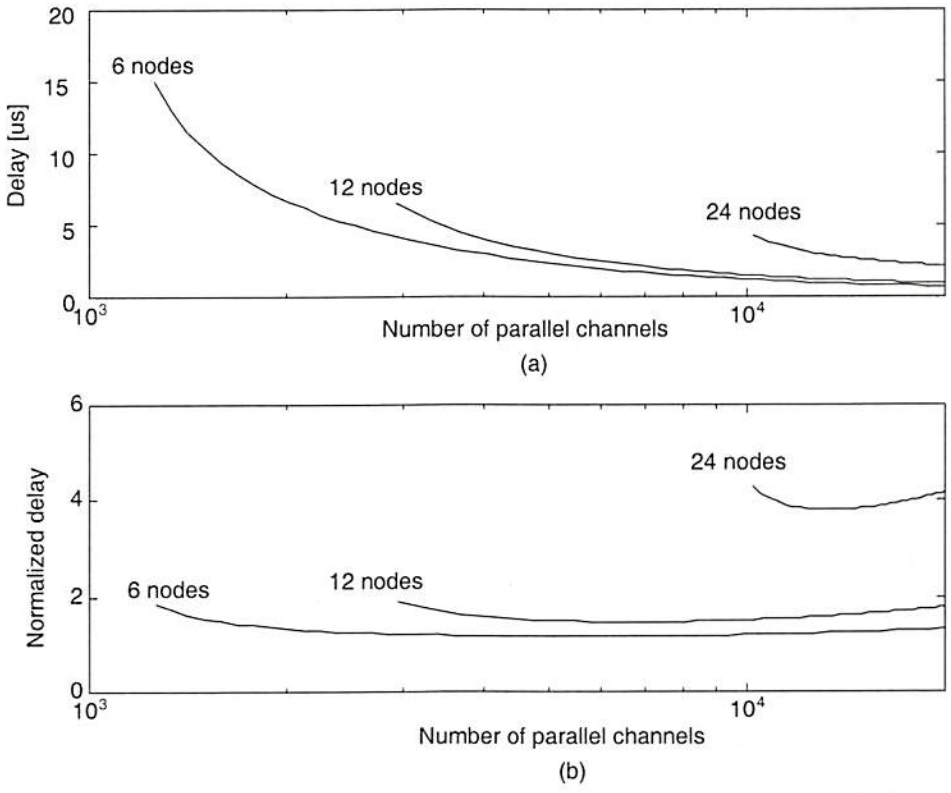

Figure 7.7. Normalized delay (a) and absolute delay (b) versus number of channels for a parallel packet Ethernet. Multiple curves correspond to various numbers of nodes on the network. The delay normalization is to the packet transmission time.

Figure 7.8 shows what happens if only the traffic increases on the network (if nodes have more packets to transmit), but the number of nodes remains the same. The delay penalty is significantly reduced as compared to Fig. 7.7. Once again, the network is more robust if more parallel channels are used. This is because the traffic is far enough from the maximum capabilities of the network, and is able to accommodate the increase in demand.

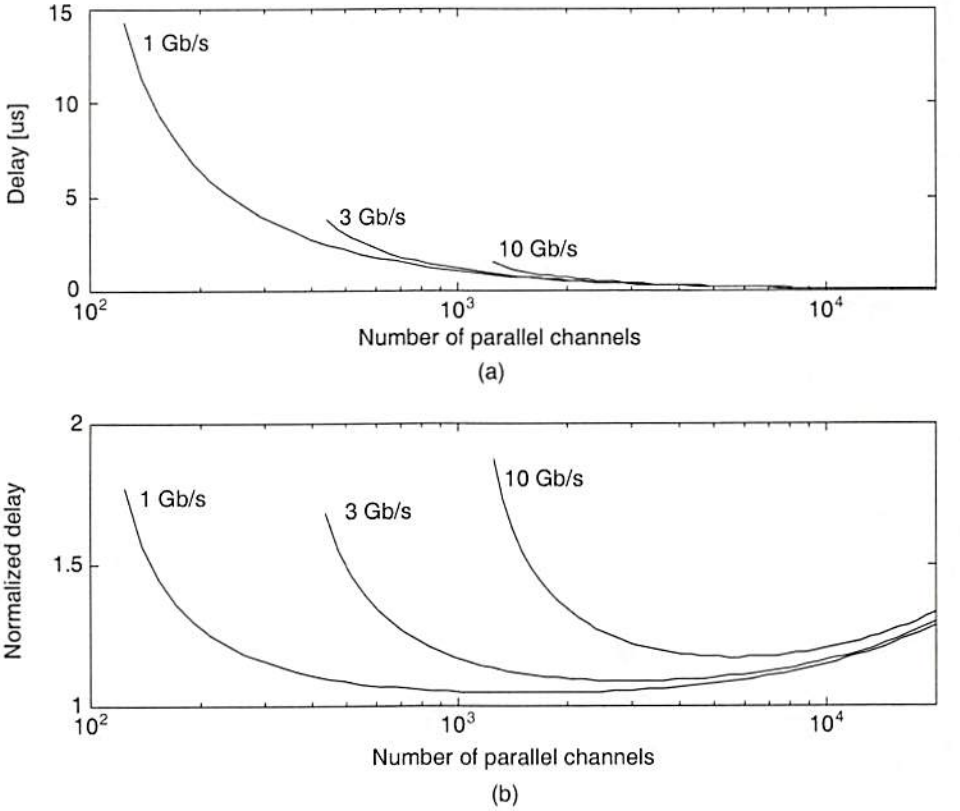

Figure 7.8. Normalized delay (a) and absolute delay (b) versus number of channels for a parallel packet Ethernet. Multiple curves correspond to various aggregate data rates. The delay normalization is to the packet transmission time.

Similar curves can be obtained if we fix the number of channels (unless otherwise specified, we assume 4000 parallel channels, corresponding to a 64 x 64 array, which is feasible with current technologies). The variation of the delay with the channel rate is shown in Fig. 7.9, for various numbers of channels. As expected from Fig. 7.6, there is an optimum data rate, at which the normalized delay is minimum. The minimum value of

the normalized delay increases slightly with the number of channels. Nonetheless, for a 30 times increase in the number of channels, the normalized delay increases by less than 20%.

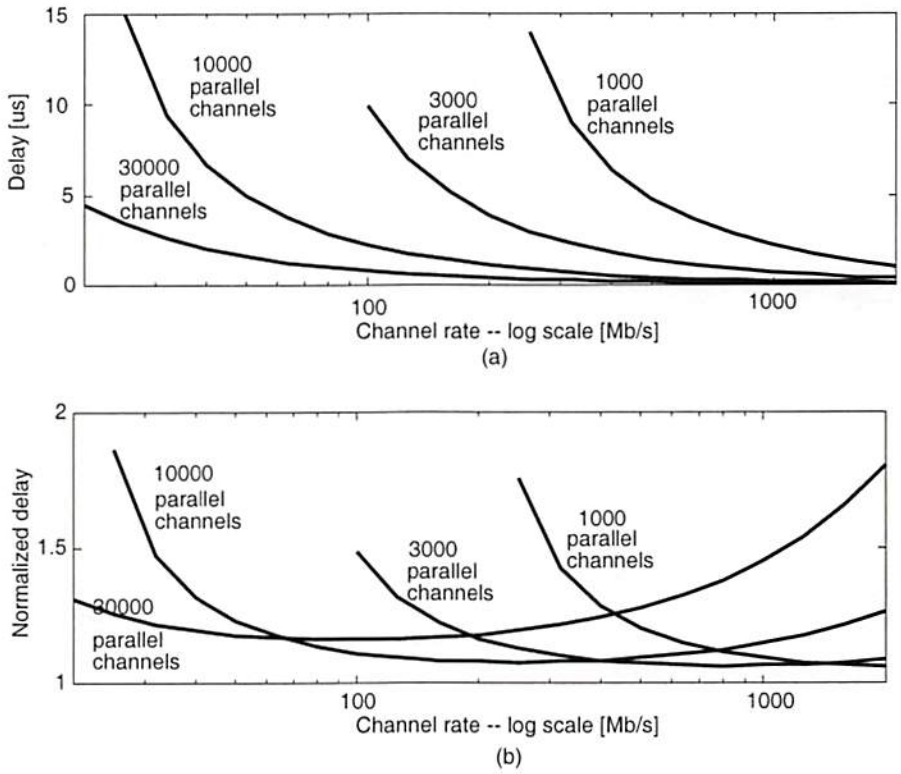

Figure 7.9. Normalized delay (a) and absolute delay (b) versus channel rate for a parallel packet Ethernet. Multiple curves correspond to various numbers of parallel channels. The delay normalization is to the packet transmission time.

Ideally, for a stable network the delay should increase linearly with the number of nodes on the network. In reality this is true only for low throughput rates. As seen in Fig. 7.10,

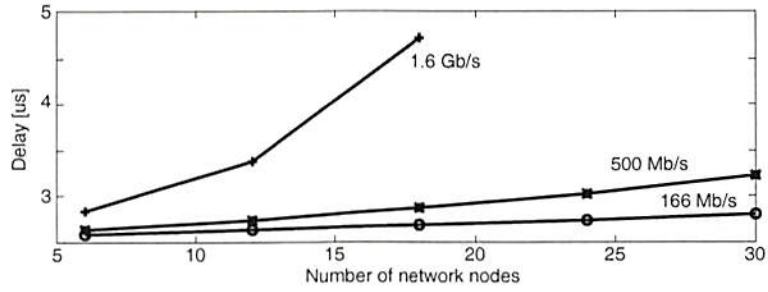

Figure 7.10. Delay versus number of nodes for a parallel packet Ethernet. Multiple curves correspond to various throughput rates per node.

increasing the number of nodes for a fixed number of parallel channels and for low throughput rates increases the delay almost linearly, but as higher throughput rates are required the delay increases exponentially. To ensure a linear increase of the delay, the exponential increase can be flattened out if more parallel channels are used (Fig. 7.8).

If the throughput is the design parameter, and a small number of parallel channels are used, the delay increases exponentially at a relatively low data rate (Fig. 7.11). If a large number of parallel channels are used, the relative delay is relatively unchanged for a wide range of throughput rates. This shows that the wide parallel network is robust and can operate under a wide variety of traffic conditions. Again, this is because the network is operating at a low offered load, where it can easily accommodate an increase in traffic. Still, using a large number of parallel channels incurs a relatively large, albeit constant delay at low data rates. If the maximum expected data rate is known in advance, the design

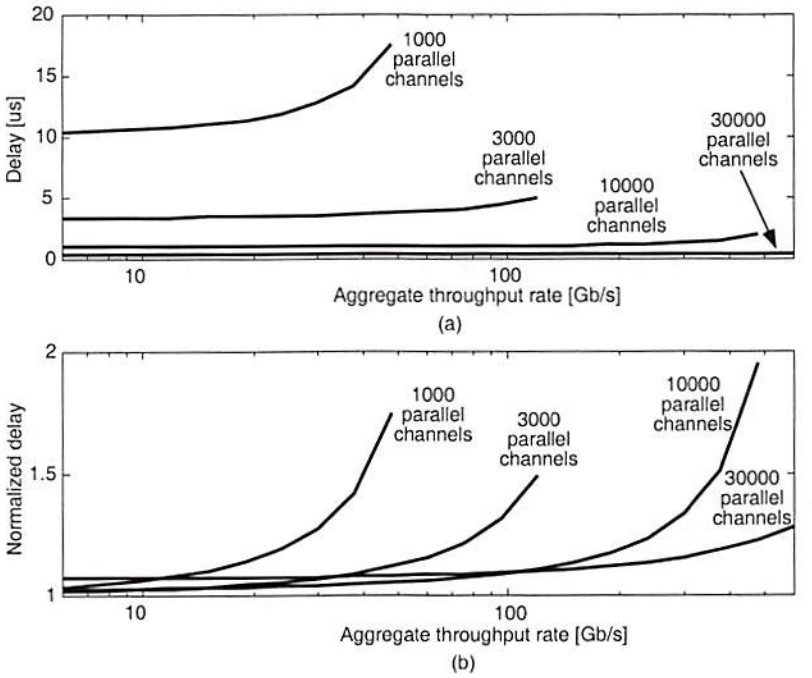

Figure 7.11. Normalized delay (a) and absolute delay (b) versus throughput rate for a parallel packet Ethernet. Curves correspond to various numbers of parallel channels. The delay normalization is to the packet transmission time.

of the network should use the smallest number of parallel channels that can offer the minimum delay, yet offer robust operation up to the desired maximum data rate.

7.7. Effects of skew on network optimization

When optimizing the parallel network operation above we have not included the effects of skew. Indeed, as the clock rate is increased, the performance of the network is more and more limited by skew. Additionally, increasing the number of network nodes increases the skew, unless clocked nodes are used. On the other hand, using clocked nodes increases the delay per node, hence the network delay. As we conjured in Section 7.5.4, using alternating clocked and translucent nodes may prove the best compromise for attaining minimum delay.

These considerations are illustrated in Fig. 7.12, where we plot the delay of a network operating at 1.66 Gb/s per node, over 4000 parallel channels. The skew is 0.45 ns per node, (as in Section 7.5.3) and the maximum clock rate used is one-tenth of the inverse of the skew (thus the maximum skew is limited to one tenth of the bit period). The clocked nodes are distributed uniformly around the network ring, so that for example when using 24 nodes and 3 clocked nodes, there are at most seven consecutive translucent nodes. Figure 7.12 shows that using a large number of clocked nodes is best in most cases. The

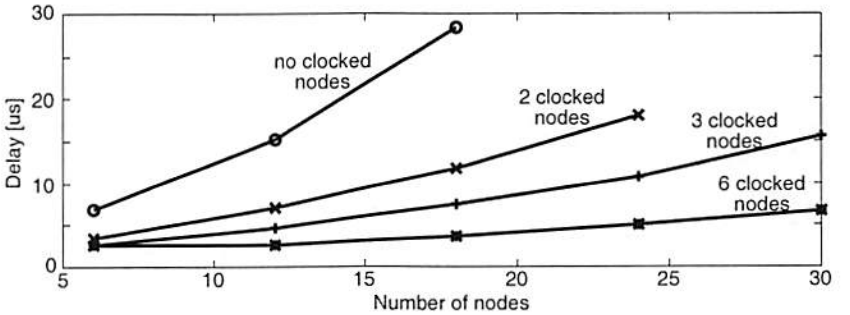

Figure 7.12. Delay versus the number of nodes for a parallel packet Ethernet. Multiple curves correspond to the numbers of clocked nodes. Each node carries 166 Mb/s over 4000 parallel channels. The maximum skew per node is 0.45 ns.

only exception is for small number of nodes (six), where using only three clocked nodes offers the same delay as when using all nodes clocked.

Because the network delay is huge compared with the extra delay of half a clock period per node, using clocked nodes rather than translucent nodes introduces a minimum delay. On the other hand, using clocked nodes does not allow the skew to accumulate, allowing a higher channel rate, which in turn reduces the network delay. This is because at a higher channel rate fewer packets must be transmitted to achieve the throughput rate per node. Overall, using clocked nodes introduces a small delay through the clocking, but allows a much higher clock rate, which compensates for the extra delay.

Similar results are shown in Fig. 7.13 where now the clock skew is assumed much smaller, only 0.1 ns per node. In this case, the delay for both clocked and translucent nodes is identical for small number of network nodes. Indeed, because skew is so small, the clock rate is not limited by skew even if all nodes are translucent. Again, this is because the extra delay of the clocking is negligible compared with the network delay, so using either clocked or translucent nodes offers similar performance. The only advantage is on the part of the translucent nodes, which have simpler architecture.

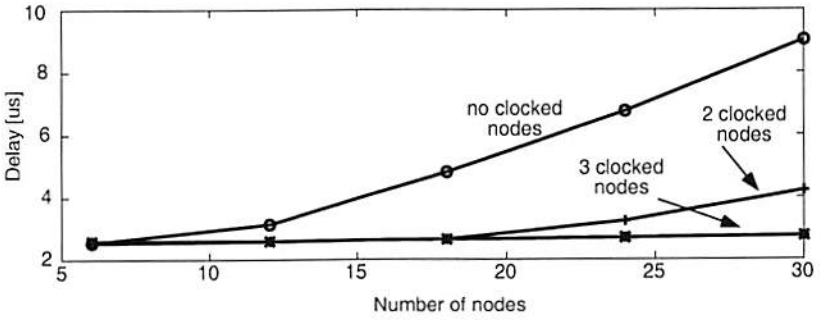

Figure 7.13. Delay versus the number of nodes for a parallel packet Ethernet. Multiple curves correspond to the numbers of clocked nodes. Each node carries 166 Mb/s over 4000 parallel channels. The maximum skew per node is 0.1 ns.

As more and more translucent nodes are added to the network, the delay increases exponentially. On the other hand, using clocked nodes ensures that the delay is almost constant, even if a relatively large number of nodes are added or removed from the network. As for the hybrid approach, there is little advantage in using both clocked and translucent nodes on the network. The delay of a network with 18 nodes is almost the same whether using 2, 3 or even 18 clocked nodes, because skew is no longer a limitation if using at least 2 clocked nodes. Given this, using only clocked nodes makes more sense from an architectural point of view, because it allows an identical design for all nodes, rather than two different designs for translucent and clocked nodes.

7.8. Summary

Combining the SIMD processing of multiple arrays by connecting them through 2-D arrays of optical interconnections can lead to very high-throughput and low latency architectures, which may have applications in real-time image processing. Increasing the number of parallel channels is a good way to reducing the network delay, up to an optimum number of channels. Beyond this value, the absolute delay still decreases but the overhead of data transmission actually increases. Skew is a major factor in the design of parallel networks. Clocked nodes are much better for achieving low delays, because the packets are retimed at each node, not allowing skew to accumulate.

References

[1] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part I – Carrier sense multiple access modes and their throughput delay characteristics," *IEEE Transactions on Communications*, vol. COM-23, no. 12, pp. 1400-1416, December 1975.

[2] J.-M. Wu, C.-H. Chen, C.B. Kuznia, B. Hoanca, A.A. Sawchuk, "Demonstration and Architecture Analysis of CMOS/MQW Smart Pixel Array Cellular Logic (SPARCL) Processors for SIMD Parallel Pipeline Processing," accepted for publication in *Applied Optics*, 1999.

[3] C.-H Chen, B. Hoanca, C.B Kuznia, J.-M. Wu, and A.A. Sawchuk, "Smart Pixel Array Network Interface (SAPIENT) for 2-D Parallel Data Packet Networks," *Optics in Computing,* OSA Technical Digest Series, vol. 8, OSA Spring Topical Meeting, paper OThD11, pp. 218-220, 1997.

[4] P. May, M.H. Lee, S.T. Wilkinson, O. Vendier, Z. Ho, S.W. Bond, D.S. Wills, M. Brooke, N.M. Jokerst, *et al.*, "A 100 Mb/s led through-wafer optoelectronic link for multi-computer interconnection networks," *Journal on Parallel and Distributed Computing*, vol. 41, no. 1, pp. 3–19, 1997.

[5] N.M. Jokerst and D.S. Wills, "A 3-dimensional high-throughput architecture using through-wafer optical interconnect," *Journalof Lightwave Technology*, vol. 13, no. 9, pp. 1935-1935, 1995.

[6] J.E. Leight, J. Yoo, and A.E. Willner, "System-design and performance of reconfigurable and simultaneous 2-D multiple-plane WDM optical interconnects," *Electronics Letters*, vol. 33, no. 7, pp. 613-614, 1997.

[7] A. V. Krishnamoorthy and D. A. B. Miller, "Scaling optoelectronic-VLSI circuits into the 21st-century - a technology roadmap," *IEEE Journal on Selected Topicsin Quantum Electronics*, vol. 2, no. 1, pp. 55-76, 1996.

Chapter 8. Conclusions and future work

The research presented in this work is not a definitive answer to optimizing the SIMD architectures. We have introduced a host of optimization techniques, some of which have been tested and demonstrated experimentally. There are certainly other means for further advancing the performance, beyond what was proposed here. Moreover, many of the proposed optimizations are far from being practical yet. There is a long way to go from our laboratory demonstrations to achieving commercial availability. This will make the object of further research work, as outlined in the remainder of the chapter.

8.1. The size issue

More than anywhere, in parallel computing applications, size matters. Most often, the best results can be obtained when using massively parallel operation, as in the case of the OCI speedup or in the case of the throughput analysis for the parallel network. To achieve such high degrees of parallelism, large arrays of processors will have to be built and interfaced with both an electronic host computer and an optoelectronic network. The most critical issues for the system architect are yield, optical alignment of large arrays of optoelectronic devices, achieving a large optical field of view, the thermal management of such large arrays, and last but not least cost.

8.1.1. Yield

Current array sizes using sub-micron VLSI technologies are limited not so much by material cost (which grows with the area of the array), but by the yield [1]. In other words, in applications where cost is not an issue, yield is the ultimate limitation. The yield for CMOS circuits decreases with area (due to clustered point defects) following the negative binomial distribution. The yield for VCSELs decreases exponentially with area

due to the presence of dislocation defects in the lattice. To improve the yield, large systems may be designed as a multi chip module (MCM) that integrates multiple chips together, with reduced parasitics in the interconnect and with better mechanical stability.

8.1.2. Optical alignment

In constructing a large optical scale system, a very difficult problem is to achieve and to maintain the required alignment. Packaging of optical systems requires accurate positioning of multiple images over a predetermined grid (of sources or detectors). The alignment of a VCSEL array and an array of detectors may be done by monitoring the optical power coupled onto the detectors (active alignment). Similarly, the alignment between the VCSELs and microlens arrays is rather straightforward using again active alignment and an array of detectors. For aligning optical modulators and detectors, the modulators (viewed as p-n diodes) can be forward biased (in which case they act as low efficiency LEDs) and then imaged onto the detectors. Lastly, to align detectors and microlenses, the methods used can be quite involved, using an external optical source and on-chip alignment elements [2].

Limited effort has been devoted to achieving dynamic alignment, to ensure the robustness of the system to external changes (changes in the temperature, vibrations, or changes in the properties of the optoelectronic devices) [3, 4].

8.1.3. The optical field of view

The optical field of view of the system, proportional to the number of PEs and the dimension of the PE is another limiting factor for the size of the SIMD array. For large arrays, the field of view of the transmitter beam can be relatively large. Such wide-angle components introduce optical aberrations and raise the costs of the optical system. To reduce this field of view, the distance between the planes of the 4-F system needs to be

large. This in turn introduces latency as well as crosstalk and power loss due to the finite divergence angle of the transmitter beam [5]. Alternatively, a hybrid system with multiple apertures could divide the field of view across different parallel optical channels.

8.1.4. Thermal management

Further limitations on the system packing density arise from the requirements for the thermal management. With the system operating in steady state, the temperature difference between the system and the environment rises until all the heat dissipated by the optical and the electronic devices in the system can be removed, i.e. the system is in thermal equilibrium with the environment. If this happens at a relatively high temperature, it may lead to overheating of the system, with consequences on alignment as well as on optical and electronic performance. Ultimately, the system may even destroy as a result of overheating. As mentioned in Section 4.2.1, the issues of power management are particularly stringent in VCSEL based systems.

8.1.5. Cost of the optical system

Reducing the cost of the system can sometimes be achieved through miniaturization. Unfortunately, this cannot easily be done for the optical system, because of external constraints on the pitch of the optical channels (for example if coupling into optical fiber). Additionally, reducing the size of the optoelectronic devices is feasible in principle. From a technology standpoint optoelectronic devices could be made even smaller than the now current 5 to 10 μm sizes, but at very high expense on the optical system design. Smaller device sizes require smaller spot sizes. This would require very low aberrations, diffraction limited optics, stable and accurate positioning devices, athermal design of both optics and mechanics, as well as tighter tolerances of device positioning. All these requirements would again drive the cost of the optics to unacceptably high values.

8.1.6. Conclusions

We reviewed some of the issues that limit the size of the optical system. Most of these issues (for example the yield) are limitations of the current technological level, and they pose no fundamental limitations. They do limit the practical sizes of arrays achievable today and hence make an experimental proof of large-scale cellular designs difficult to implement.

8.2. Schemes for automated alignment of large scale arrays

In designing the optoelectronic interface, significant consideration must be given to the alignment issues. Even a slight misalignment of the optical beams can lead to serious power loss and crosstalk, ultimately reducing the achievable channel data rate. It is our experience that the optical alignment is the most frequent, most tedious and the most important operation performed during our experimental demonstrations. For this reason, automated alignment procedures are extremely desirable and will be essential when building large and reliable arrays of optically interconnected devices.

There is currently a wide interest in automated schemes. Two approaches are usually taken. The first one is to use the accuracy of lithographic and interferometric techniques to align features. This first approach is already routinely used, although the applications are limited, for example to when multiple devices are fabricated using the same masking step.

The second approach is more powerful, but requires more design effort, as well as more real estate. It requires building additional hardware on each array to detect the amount of misalignment and then drive an active alignment element. This second approach, usually employing a feedback loop, is extremely powerful and can ensure the reliable functioning of the system under vibrations and temperature changes. The large-scale implementation of the active alignment is slowed down by the lack of active alignment elements, as well as by

the complexity of the required alignment operations, but will certainly be a requirement for reliable operation under realistic conditions.

8.3. Low power considerations

Another very complex issue to be solved is the thermal dissipation. Classical computing architectures use one side of the chip package for interconnections (through an array of electrical pins) and the other side of the chip package for cooling. This makes the architecture equivalent to a 2-D array, even though chips communicate through the third dimension, perpendicular to the plane of the chip. Indeed, because one side of the chip is always dedicated to the thermal dissipation block, the chips could topologically be distributed across a single plane, so they do not form a true 3-D structure [6]. Using low power design techniques could remove the requirement of cooling the chips, thus allowing true 3-D stacking.

8.4. Summary

The optimizations proposed and demonstrated in this work go a long way in making possible the realization of large-scale SIMD machines with optoelectronic interconnections. There are still issues to be resolved, not limited to thermal management, optical alignment and cost. Judging by the recent progress in the device technologies [7], the rapid progress in the design of optoelectronic SIMD architectures should follow shortly.

References

[1] C. W. Stirk and J. Neff, "The cost of optical interconnects vs. MCMs", in *Proceedings of Optics in Computing '97*, OSA Technical Digest, Optical Society of America, Washington, DC, pp. 21-23, 1995.

[2] G. C. Boisset, B. Robertson, W. S. Hsiao, M. R. Taghizadeh, J. Simmons, K. Song, M. Matin, D. A. Thompson, and D. V. Plant, "On-die diffractive alignment structures for packaging of microlens arrays with 2-D optoelectronic device arrays," *IEEE Photonic Technology Letters*, vol. 8, no. 7, pp. 918-920, 1996.

[3] J. Gourlay, T.-Y.Yang, and A. C. Walker, "Low-order adaptive optics for free-space optically connected components," in *Proceedings of Optics in Computing '98*, OSA Technical Digest, Optical Society of America, Washington, DC, pp. 52-55, 1998.

[4] G. C. Boisset, B. Robertson, and S. H. Hinton, "Design and construction of an active alignment demonstrator for a free-space optical interconnect," *IEEE Photonic Technology Letters*, vol. 7, no. 6, pp. 676-678, 1995.

[5] F. Sauer, J. Jahns, and C. R. Nijander, "Refractive-diffractive micro-optics for permutation interconnects," *Optical Engineering*, vol. 33, no. 5, pp. 1550-1560, 1994.

[6] H. M. Ozaktas, "Toward an optimal foundation architecture for optoelectronic computing . Part 1. Regularly interconnected device planes," *Applied Optics*, vol. 36, no. 23, pp. 5682-5696, 1997.

[7] F. A. P. Tooley, "Optical interconnects do not require improved optoelectronic devices," *Proceedings of Optics in Computing '98*, Brugge, Belgium, pp. 14-17, 1998.

Appendix I. Proof of the recurrence relation for optimal link distances

To preserve the logarithmic increase of number of clock cycles with array size, we require that no more than K optical links be used to travel over any distance if the OCI fan-out is K. Clearly, for a fan-out of $K+1$, there are two ways to reach PEs at distances exceeding $D(K)$. After taking a single optical jump through the OCI to $X(K)$ (path A), a multihop path can then proceed to use any of the links $\{\pm X(1),\cdots,\pm X(K)\}$ for longer distances, thus covering in an optimal fashion a distance of $D(K)$, from $X(K)$ to $X(K)+D(K)$, and using at most $K+1$ optical hops (one optical jump to $X(K)$, then at most K optical jumps and S electronic jumps to cover the distance $D(K)$). On the other hand, using a $K+1$ optical link, $X(K+1)$ (path B), longer jumps would cover a distance $D(K)$ to the left of $X(K+1)$ using at most $K+1$ optical links. This covers the PEs $X(K+1)-D(K)$ through $X(K+1)$ in the array. To ensure that the two regions of coverage are adjacent but not overlapping (for optimality of the fan-out usage), we need to enforce $X(K+1)-D(K)=X(K)+D(K)+1$, or

$$X(K+1)=X(K)+2D(K)+1. \tag{A1.1}$$

Thus, reaching to the right of $X(K+1)$, the distance over which the connection is optimal using at most $K+1$ optical hops is now

$$D(K+1)=X(K+1)+D(K). \tag{A1.2}$$

Combining the recursion relations (A1.1) and (A1.2), we solve for the link distances as

$$X(K+1)=4X(K)-X(K-1). \tag{A1.3}$$

The recurrence relation (A1.3) can now be solved explicitly for $X(K)$. The recurrence has a second order dependence ($X(K)$ depends on terms up to $X(K\text{-}2)$). Thus, the solution of the recurrence relation is of the form

$$X(K) = AR_1^K + BR_2^K, \tag{A1.4}$$

with A, B, R_1 and R_2 constants to be determined. With (A1.4) in (A1.3), we obtain

$$R_i^2 - 4R_i + 1 = 0, i = 1,2, \tag{A1.5}$$

which can now be solved to yield $R_1 = 2+\sqrt{3}$ and $R_2 = 2-\sqrt{3}$. The constants A and B can now be found from the initial conditions, $X(0)$ and $X(1)$.

We showed that the first link is $X(1) = M + 2S + 1$. We can now calculate $X(0)$, which is not in fact a physical link, but an abstract distance needed as initial condition for the recurrence relation (A1.3). Clearly, $D(0) = \left(MK + S\right)\Big|_{K=0} = S$. Using this value for $D(0)$ in (A1.1) for $K = 0$, we conclude that $X(0) = M$, again only for the sake of the recurrence relation (A1.3), since $X(0)$ is not in fact a physical link.

For initial conditions $X(0) = M$ and $X(1) = M + 2S + 1$, we obtain

$$A = \frac{1}{2\sqrt{3}}\left[2S + 1 + M(\sqrt{3} - 1)\right]$$

$$B = \frac{1}{2\sqrt{3}}\left[M(1 + \sqrt{3}) - 2S - 1\right] \tag{A1.6}$$

which lead to the explicit dependence

$$X(K)=\frac{1}{2\sqrt{3}}\left\{\left[M\left(\sqrt{3}-1\right)+2S+1\right]\left(2+\sqrt{3}\right)^{K}+\left[M\left(\sqrt{3}+1\right)-2S-1\right]\left(2-\sqrt{3}\right)^{K}\right\}, \tag{A1.7}$$

$$K\geq 1$$

Similarly, we can obtain the explicit dependence for $D(K)$

$$D(K)=\left\lfloor\frac{X(K+1)-X(K)-1}{2}\right\rfloor=$$

$$\left\lfloor\frac{3X(K)-X(K-1)-1}{2}\right\rfloor= \tag{A1.8}$$

$$\left\lfloor\frac{\left[(2S+1)(\sqrt{3}+1)+2M\right](2+\sqrt{3})^{K}+\left[(2S+1)(\sqrt{3}-1)-2M\right](2-\sqrt{3})^{K}}{4\sqrt{3}}\right\rfloor$$

where $\lfloor x\rfloor$ is the largest integer smaller than x. This shows that $K\propto\log_{2+\sqrt{3}}(D)$ for a maximum shift distance of D.

We derived the recurrence relations and the explicit dependencies above for the last optical link, i.e., $X(K)$ for a fan-out of K. Considering that the link set for a given fan-out includes all the optical links of the sets with lower fan-outs, a complete connection set will now be given by the recurrence relation

$$X(n+1)=4X(n)-X(n-1),\ n>1, \tag{A1.9}$$

with initial conditions

$$X(0)=M,\ X(1)=M+2S+1, \tag{A1.10}$$

and the explicit dependence

$$X(n)=\frac{1}{2\sqrt{3}}\left\{\left[M\left(\sqrt{3}-1\right)+2S+1\right]\left(2+\sqrt{3}\right)^{n}+\left[M\left(\sqrt{3}+1\right)-2S-1\right]\left(2-\sqrt{3}\right)^{n}\right\}, \quad \text{(A1.11)}$$

$$n\geq 1$$

When the edge effects are taken into considerations, the only change to be made is in the optimal link usage equation, Eq. (A1.1). Only the coverage in the direction of data transfer can be used. Thus, using the optical links up to a link order K can cover a certain optimality distance $D(K)$ to the left of a PE. The next optical link must be located exactly at the border of this optimality distance, hence

$$X(K+1)=X(K)+D(K)+1. \quad \text{(A1.12)}$$

Using this into Eq. (A1.2) the new recurrence relation is

$$X(K+1)=3X(K)-X(K-1), \quad \text{(A1.13)}$$

which can be solved with the same initial conditions $X(0)=M$ and $X(1)=M+2S+1$ to yield the explicit dependence

$$X(n)=\left\{\left[\frac{M}{2}+\frac{2S+1-\frac{M}{2}}{\sqrt{5}}\right]\left(\frac{3+\sqrt{5}}{2}\right)^{n}+\left[\frac{M}{2}-\frac{2S+1-\frac{M}{2}}{\sqrt{5}}\right]\left(\frac{3-\sqrt{5}}{2}\right)^{n}\right\}, n\geq 0. \quad \text{(A1.14)}$$

Appendix II. Network traffic formulae

The formulae below are for a 1-persistent CSMA/CD network, for which, after sensing the network idle at the end of a busy period, a node retries to send a packet with probability $p = 1$. All quantities are normalized with respect to the packet transmission time, T.

The throughput S_n as a function of the offered load G is given as an implicit function

$$S_n = \frac{G\left[1+G+aG\left(1+G+\frac{aG}{2}\right)\right]e^{-G(1+2a)}}{G(1+2a)-\left(1-e^{-aG}\right)+(1+aG)e^{-G(1+a)}}, \tag{A2.1}$$

which must be solved numerically to obtain the offered load G. The expected packet delay is function of the offered load as well,

$$D=\left(\frac{G}{S}-1\right)\left(1+2a+\alpha+\delta+\bar{r}_1\right)+\bar{r}_1+1+a, \tag{A2.2}$$

where $\bar{r}_1$ is the average pretransmission delay (due to waiting), given by

$$\bar{r}_1 = \frac{1+a^2+2\left(1-\frac{1}{G}\right)\bar{Y}}{2q_0\left(\bar{B}+\bar{Y}\right)}. \tag{A2.3}$$

The expected duration of a busy period is

$$\bar{B} = \frac{1+a+\bar{Y}}{q_0}, \tag{A2.4}$$

where

$$\bar{Y} = a - \frac{1}{G}(1 - e^{-aG}), \tag{A2.5}$$

and the probability of zero packets accumulated at the end of a period T is

$$q_0 = (1 + aG)e^{-G(1+a)}. \tag{A2.6}$$

The expected duration of an idle period is

$$\bar{I} = \frac{1}{G}. \tag{A2.7}$$

The other quantities above are the normalized roundtrip delay, a, the average normalized retransmission delay, δ, and the normalized duration of the acknowledgment, α.

With this, given the aggregate data rate (for all nodes on the network), R_{data}, the number of parallel channels, H, and the channel rate (on-chip clock rate), C, the throughput is

$$S = \frac{R_{data}\left(1 + hC/P\right)}{HC}, \tag{A2.8}$$

where P is the packet size (the payload, in bits) and h the number of header bits. This is because the aggregate throughput must include the overhead of sending the header bits (to be sent ahead of the payload, on all channels in parallel – for the TRANSPAR network, these empty bits are due to the finite state machine decoding its initial states).

Knowing throughput S, we solve (A2.1) for the offered load G and then use the calculated values in (A2.2) to find the delay D. The average retransmission delay is 100 ns and the duration of the acknowledgment is negligible (0.1 ns). The propagation delay time includes the delay per node and an additional half a clock period of delay per clocked node.