In Proceedings of the International Workshop on System on Chip for Real Time Processing,
Calgary, Canada, June 2003.

Design Considerations for Optically Connected Systems on Chip

Neal K. Bambha and Shuvra S. Bhattacharyya
Department of Electrical and Computer Engineering, and
Institute for Advanced Computer Studies
University of Maryland, College Park USA
{nbambha,ssb}@eng.umd.edu

Gary Euliss
Applied Photonics, Inc.
Fairfax, VA USA
geuliss@ap-photon.com

Abstract

This paper addresses some fundamental issues relating to the design of systems on chip that utilize optical interconnects. We present an information theoretical model for assessing trade-offs between global and local partitions in these systems, and evaluate interconnect topology synthesis and application mapping techniques for digital signal processing (DSP) applications in these systems.

1. Introduction

As VLSI feature sizes shrink, interconnects between modules and subsystems are becoming a limiting factor for systems on chip (SoC). Narrower metallic wires placed closer together lead to increased crosstalk and larger interconnect delays. As designs become larger and more functional units are placed on the chip, greater demands are placed on the interconnects. One way to solve this problem is to utilize optical interconnects to replace the longest metallic interconnects. Such hybrid optical/electronic interconnects hold great promise for larger designs. There are still many materials, fabrication, and packaging challenges in integrating optic and electronic technologies. However, much research effort is currently taking place in these areas. The DARPA sponsored Optoelectronic Center and VLSI Photonics programs [10] are two examples of such research efforts. This paper will present some fundamental systems issues relating to a SoC utilizing hybrid optic/electronic interconnects.

2. Motivation and Previous Work

Several research groups have demonstrated optically-connected multiprocessor systems (e.g., see [2], [3], [6], [7]). Some of these systems are based on free-space optical interconnects, while others are based on wavelength division multiplexing (WDM). WDM systems typically utilize fiber or waveguide interconnects, and are advantageous for hybrid integration of independent modules. The strength of a free-space optical interconnect scheme is its potential to provide an extremely high density of interconnections, such as will be required for a single-chip system.

An example of a system utilizing free-space optical interconnects is the *FAST-Net* prototype [3]. FAST-Net is a high throughput data switching concept that uses a reflective optical system to globally interconnect a multichip array of processors. The three-dimensional optical system links each chip directly to every other with a dedicated bidirectional parallel data path. The system utilizes smart-pixel arrays (SPA), in which high density silicon electronics are integrated with two-dimensional arrays of high speed Gallium Arsenide micro-laser/detector arrays. An array of SPAs is packaged on a planar substrate and linked to itself through an optical system composed of a lens array and a mirror. This concept provides internal bisection bandwidth [5] on the order of 10^{12} bits per second.

Compiler technology and automated mapping tools for these systems have received relatively less attention than the hardware. Seo and Chatterjee [8] presented a CAD tool for physical placement of modules in SoC utilizing optical interconnects. The tool determined which interconnects should be routed electrically and which should be routed optically. They reported a 50% reduction in worst case interconnect delay over using all metallic interconnects.

3. Optically Connected System on Chip

Our general model for a system-on-chip (SoC) is one in which the chip is partitioned into regions that are connected with metallic (local) interconnects, and these local regions are then connected through optical (global) interconnects. The applications consist of *task graphs* [9], where the individual tasks must fit fully into a local region. The graph vertices (*tasks* or *nodes*) in the acyclic task graphs represent computations while the edges represent the communication of a packet of data from a source task to a sink task.

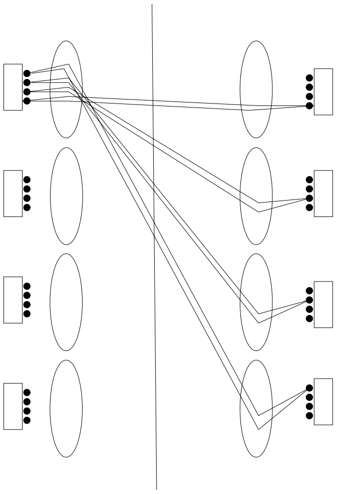

Figure 1. Schematic side view of the global optical interconnection shown folded about the mirror plane for the FAST-Net system.

Three fundamental design considerations for such a system are addressed in this paper:

- What is the optimum size of a local partition?
- What techniques should we use to map and schedule tasks on these partitions?
- How do we synthesize an optimum global (optical) interconnection network for the system?

These considerations are interrelated, since the size of the local partition will affect the maximum size (granularity) of the tasks, and the scheduling of tasks depends on the interconnection network.

4. Global/Local Partitioning

This section presents an information-theoretical model for trade-offs in designing the local partition of a SoC utilizing free-space optics. As mentioned earlier, free-space optical interconnects can provide higher interconnect densities than other types of optical interconnects. These trade-offs are fundamental in nature and will exist in any system utilizing these interconnects.

These systems utilize arrays of vertical cavity surface emitting laser (VCSEL) transmitters and photoreceivers to implement the interconnect. A single interconnect consists of a VCSEL/photoreceiver pair. Light from the VCSEL must be directed to and imaged on the appropriate photoreceiver. This is depicted for the FAST-Net system in Figure 1. Different systems use different imaging methods to accomplish this. The high density of interconnections arises from the use of the third dimension (free-space) and the fact that overlapping optical signals do not interfere with each other (i.e., there is no crosstalk in free space). As the dimensions of the local partition decrease, higher f-number lenses are required to collect the light from the transmitters in a constant focal-length system. (The f-number of a lens is defined as its focal length divided by its diameter). Figure 2 depicts the diffraction-limited images of an array of point sources, in a random on/off pattern, on an array of photodetectors. The data for the figure was generated using MATLAB to compute the diffraction pattern for F/1 lenses (left) and F/2 lenses (right). Using an optical system with f-number F and treating the transmitter as a point source operating at wavelength λ, the diffraction-limited image of the source on the detector is given by the expression

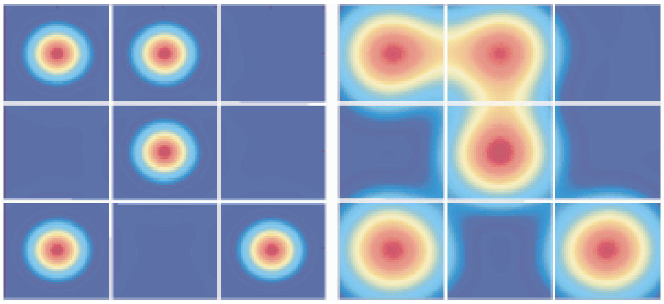

Figure 2. An array of point sources imaged using f/1 optics (left) and f/2 optics (right). The left and right pictures are different scales—the partitions on the left are twice the length of the partitions on the right.

$$Ai(\rho) = I_0 \left(\frac{2J_1 \frac{\pi\rho}{\lambda F}}{\frac{\pi\rho}{\lambda F}} \right)^2 \tag{1}$$

where ρ is the radius from the center of the image and I_0 is proportional to the source intensity. The function J is a first order Bessel function of the first kind.

From this equation, the signal received by the center channel for this pattern can be calculated by spatially integrating over the corresponding photodetector. This calculation will also take into account the inter-pixel interference (IPI). We then vary the pattern randomly to generate the conditional probability distributions for the center channel. If we assume that the IPI is only significant between adjacent channels, we can use the conditional probabilities to assess the mutual information corresponding to a channel between partitions. As partition size decreases, and the associated aperture sizes decrease (increasing the f-number), the optical signal intensity decreases and the IPI increases. Both effects reduce the mutual information. We can then characterize the mutual information as a function of partition size, and therefore, the number of partitions. The mutual information between each source and its corresponding detector is given by

$$I_{mut}(X;Y) = \sum_{i=0,1} p(y|X=i) \log_2 \left[\frac{p(y|X=i)}{p(y)} \right] dy \tag{2}$$

where $\rho(y|X=i)$ is the conditional probability that a value y is received when i is transmitted and $p(y)$ is the probability density function (PDF) of y.

Restoring the mutual information required for the application can be achieved by decreasing the bit rate and integrating over a longer clock cycle in order to increase the signal-to-noise ratio. We define the information capacity, or data rate, as the product of the mutual information and the bit rate. Therefore, it can be generally shown that increasing the number of partitions on a chip will lead to lower global data rate across the chip. At the same time, smaller partitions will reduce the length requirements on local interconnections (intra-partition) performed electrically. Therefore, local interconnect data rates can benefit from reduced partition size. We assume that the data rate is inversely proportional to the RC time constant, which in turn is proportional to the square of the interconnection length. A simple approximation then results in a factor $\sqrt{N}$ decrease in local interconnect length, therefore, a factor N increase in the local data rate, where N is the number of partitions. These opposing effects of partition size suggest a tradeoff between the local and global data rates, which is illustrated hypothetically in Figure 3, and thus an optimum partitioning of the SoC. This is the crossing point of the two curves in Figure 3.

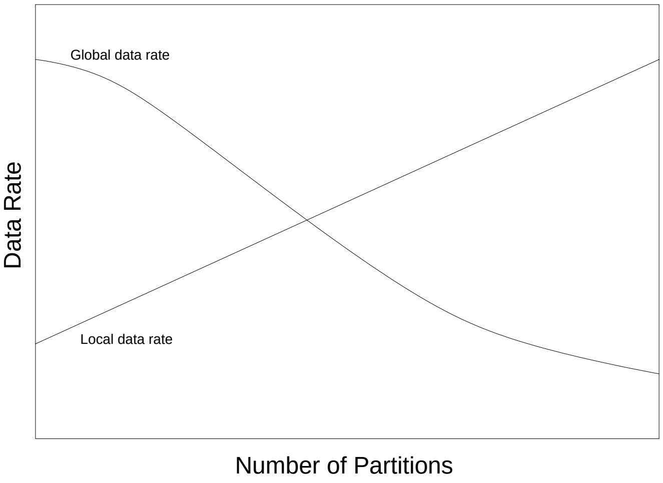

Figure 3. Tradeoff between partition size, global data rate, and local data rate.

4.1. Typical Numbers

We next give some estimates of system parameters based on today's components. The optical channel density on the chip will impose a fundamental upper limit on the number of partitions, M, for the SoC. For a chip with dimensions LxL, the number of optical channels N will be given by $N \leq L^2/2d^2$ where d is the VCSEL and detector pitch. For a full crossbar connection, $N = M(M-1)$. For a "typical" VCSEL pitch of 125 microns, this implies that we would be limited to 57 partitions for a one square centimeter chip. The power requirements depend on the architecture, but some insight can be gained by considering examples. Let P_0 represent the power required to drive a VCSEL-detector pair. If every partiton is transmitting and receiving data, the total optical power is given by the number of partition times the number of VCSEL-detector pairs per cluster times P_0, $P = L^2 P_0 / 2d^2(M-1)$. Then since $L^2/2d^2 \geq M(M-1)$, $P \geq P_0 M(M-1)/M-1 = MP_0$. The lower limit represents the case in which the SoC contains the maximum number of partitions, and a cluster contains a single VCSEL-detector pair. This also represents the lowest utilization of the optical interconnections. Under these assumptions, the most power will be consumed for a two-partition architecture in which case all VCSEL-detector pairs will be operating. Therefore, $P \leq L^2/2d^2 P_0$. If we assume $P_0 = 10$ mW, then the total power consumption would be 32W for the one square centimeter chip in the most demanding case and 570mW for the least demanding case.

The one-way data rate between two partitions is given by the data rate per VCSEL-detector pair, D_0, times the number of pairs: $D_{\text{partition}} = L^2/2d^2 M(M-1) D_0$. For $D_0 = 2.5$ Gbps, $D_{\text{partition}} = 4$Tbps in a two-partition architecture. In the case of a single VCSEL-detector pair per cluster, the partition data rate is equal to the channel data rate at 2.5 Gbps, with an aggregate data rate of 142.5 Gbps for 57 partitions.

5. Flexible Interconnect Topologies

Electrically connected systems generally have a regular interconnection pattern, due to the physical constraints imposed by two-dimensional circuit board layout. Some examples include ring, mesh, bus, and hypercube interconnect topologies. Using these topologies, communication between remote processors requires multiple hops, which increases both latency and power, and increases contention throughout the network.

In contrast, optically connected multiprocessor systems, particularly those utilizing free space optics and three dimensions, are free to utilize arbitrarily irregular interconnection networks. Once the signal is in the optical domain, there is very little attenuation, so the energy required to transmit a unit of data is essentially independent of distance. The required energy instead is a function of the number of electrical-to-optical conversions that must be performed [4], which in turn is determined by the number of hops. Furthermore, due to the flexibility of the communication medium, it is generally possible to avoid multi-hop communication operation by simply activating direct communication channels between the source and destination processors. It is shown in [1] that restricting the schedule to single-hop communication can produce significant power savings. Together, these properties make it desirable to limit the number of hops per communication operation when exploring configurations (interconnection patterns and task graph mappings) for an optically connected, embedded multiprocessor.

The scheduling and mapping algorithms described in the following sections apply to both free-space and WDM based optical systems. When developing automated mapping tools for optically connected systems, we have several design constraints. It is desirable to map the application onto the architecture without requiring multi-hop communication, while satisfying constraints on system throughput and latency. Area and routing constraints limit the number of VCSELs and detectors surrounding a local partition. This limits the maximum I/O fanout (degree) of a single local partition. In order to conserve area and power, we would also like to minimize the total number of communication links.

6. Scheduling for Arbitrary Interconnections

In systems that are not fully connected (i.e., full crossbar connection with every processor directly connected to every other processor), one consequence of limited hop communication is that the scheduling algorithms must take into account the connectivity constraint in order to avoid deadlock [1]. Much research has been devoted to scheduling techniques for fixed interconnection networks in which these connectivity constraints are not considered. We have previously reported a polynomial complexity *feasibility* algorithm which enables standard list scheduling techniques to be modified to efficiently avoid deadlock with arbitrary interconnect topologies. It is also shown that utilizing a *flexibility* metric calculated in conjunction with the feasibility algorithm further improves scheduling performance [1].

7. Interconnect Synthesis

Embedded systems typically run a limited and fixed set of applications. We can use this application-specific information to optimize the interconnection network. For our purposes, an optimal network is defined in the context of a set of applications and constraints. The constraints may include the latency, throughput, and power consumption for the given applications, along with cost and area constraints of the overall system. Cost and area constraints dictate the total number of transmitters and receivers in the system (i.e., total number of optical links). Routing constraints from local partitions to their associated VCSEL transmitters and detectors dictate a maximum fanout for each local partition. An optimum interconnect is then one that minimizes the number of links while enabling the application to meet the power, latency, and throughput constraints.

The freedom to optimize interconnection patterns opens up a vast design space, and thus the design of an optimal interconnect structure for a given application or set of applications is a significant challenge. In this section, we illustrate an interconnection synthesis algorithm based on a genetic algorithm (GA), and compare it with a previously developed deterministic algorithm.

We developed a GA-based interconnect synthesis algorithm. This algorithm employs the dynamic level scheduling (DLS) algorithm [1] modified for arbitrary interconnection networks as the underlying list scheduling strategy, although any list scheduling algorithm could have been used. The algorithm takes into account constraints on the total number of links $l_{\max}$ and a maximum fanout for each processor $f_{\max}$, as described earlier and motivated by area and cost constraints for the system.

In our algorithm, the individuals are bit vectors corresponding to a given interconnect topology. The fitness function for a chromosome in our interconnect synthesis algorithm is described by

$$\text{fitness} = M(1 + P_f + P_l) \qquad (3)$$

where M is the makespan (latency) calculated by the modified DLS algorithm for the interconnect topology of the chromosome, P_f is a penalty based on violating the fanout constraint, and P_l is a penalty based on violating the maximum link constraint. We define a *link vector* as a bit vector with one entry for each possible interconnection between two processors. For a system with N processors, there are $N(N-1)$ entries in the link vector.

We evaluated our GA-based interconnection synthesis algorithm on a neural network classification benchmark called RBFNN. This neural network consists of 8 input layers, 2 hidden layers, and 4 output layers. This

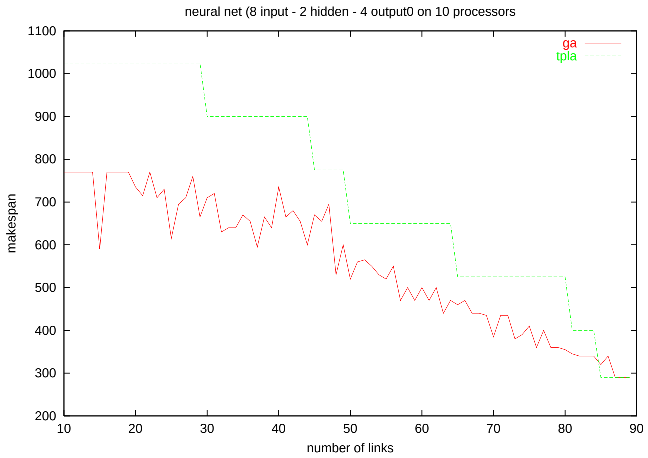

Figure 4. Comparison of TPLA and genetic algorithm for neural network application.

benchmark was chosen in part since it exhibits a large amount of inter-processor communication.

We compared the GA-based algorithm to a greedy, heuristic algorithm called *TPLA* described in [1]. The TPLA algorithm starts with a fully connected network, and operates in down and up phases. Each step of the down phase removes one link, while each step of the up phase adds one link. At each step, TPLA chooses the topology which maximizes the throughput. The genetic algorithm has several advantages over the TPLA algorithm. The first advantage is that it is able to incorporate fanout constraints, which the TPLA algorithm does not. Cost and area considerations often dictate fanout constraints. In a free-space optical system, as already mentioned, fanout is dictated by the number of VCSELs and photoreceivers that can be placed adjacent to a processor. In a WDM system, cost constraints dictate the number of wavelengths used. The second advantage is that, in order to synthesize a network for a given link constraint, the TPLA must evaluate many intermediate topologies that don't meet the link constraint during its construction phases. This makes it much less efficient, especially for systems with a large number of processors. Neither of these algorithms take into account isomorphically unique link topologies. Doing so could significantly pare the search space and is an area for future work. Figure 4 shows the best latency achieved for each level of connectivity between zero connectivity and fully connected for both algorithms. This gives a Pareto curve of the trade-off between number of links and latency for the application. In order to properly compare the different algorithms, the GA run time was limited to the run time required by TPLA. The results show that the algorithm based on the GA performs 21% better (producing lower makespan schedules), when averaged over the different link configurations, for this benchmark.

8. Conclusions

Optical interconnect technology holds the potential to relieve interconnect bottlenecks on SoC. It is particularly well suited for embedded systems since the interconnection patterns can flexibly be optimized and reconfigured to match the target applications. We have presented a model for determining optimal partitioning size and an improved interconnect topology synthesis algorithm for these systems.

9. Acknowledgement

This research was supported in part by the DARPA funded Optoelectronics Center for Innovative Photonic Chipscale Technologies (contract number MDA972-00-1-0023, through Brown University).

References

[1] N. K. Bambha and S. S. Bhattacharyya. System synthesis for optically connected, multiprocessors on-chip. *Proceedings of the International Workshop for System on Chip*, pages x–x, July 2002.

[2] P. Guilfoyle. Digital optical computing architectures for compute intensive applications. *Proceedings 1994 International Conference on Optical Computing*, 1994.

[3] M. Haney, M. Christensen, and P. Milojkovic. Description and evaluation of the fast-net smart pixel-based optical interconnection prototype. *Proceedings of the IEEE*, 88(6), June 2000.

[4] R. Kostuk, J. Goodman, and L. Hesselink. Optical imaging applied to microelectronic chip-to-chip interconnections. *Applied Optics*, pages 2851–2858, 1985.

[5] T. Leighton. *Introduction to Parallel Algorithms and Architectures; Arrays, Trees, and Hypercubes*. Morgan Kaufmann, San Mateo, CA, 1992.

[6] N. McArdle and M. Ishikawa. Experimental realization of a smart-pixel optoelectronic computing system. *Proceedings Massively Parallel Processing with Optical Interconnects '97*, pages 190–195, 1997.

[7] N. McArdle, M. Naruse, A. Okuto, and M. Ishikawa. Implementation of a pipelined optoelectronic processor: Ocular 2. *Technical Digest of Optics in Computing*, 1999.

[8] C. Seo and A. Chatterjee. A cad tool for system-on-chip placement and routing with free-space optical interconnect. *Proceedings of IEEE International Conference on Computer Design (ICCD'02)*, 1992.

[9] S. Sriram and S. S. Bhattacharyya. *Embedded Multiprocessors: Scheduling and Synchronization*. Marcel Dekker Inc., 2000.

[10] http://www.darpa.mil/mto/vlsi.