Proc. Int'l Conf. Communications in Computing (CIC-2000)
Las Vegas, Nevada, USA, 2000 June 26-29

Challenges in Interconnection Network Design In the Era of Multiprocessor and Massively Parallel Microchips

Behrooz Parhami and Ding-Ming Kwai
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA
E-mail: parhami@ece.ucsb.edu

Abstract

Custom microchips housing many simple processors have long been used in the design of massively parallel computers. Commercially available SIMD parallel systems of the late 1980s already contained tens of bit-serial processors on each chip and more recent products offer hundreds of processors per chip. Use of microchips housing multiple general-purpose processors, with large memories, has also been proposed. No matter how many processors we can put on one chip, the demand for greater performance will sustain the need for integrating multiple chips into systems that offer even higher levels of parallelism. With tens to tens of thousands of on-chip processors afforded by billion-transistor chips, small-scale parallel systems built of powerful general-purpose processors, as well as multimillion-node massively parallel systems, will become not only realizable but also quite cost-effective. Our thesis is that design challenges for single-chip multiprocessors and massively parallel systems, as well as their use in synthesizing even larger parallel systems, are not fundamentally different from those currently facing parallel computer designers, given that interconnects already constitute the limiting factor. Either way, we need to rely on multilevel (hierarchical or recursive) parallel architectures. The difference is in scale rather than substance, with the requisite theories and design strategies already in place.

Keywords: hierarchical network, interconnect, MPP, multilevel network, parallel computer architecture, recursive network, technology scaling, wire delay.

1. Introduction

Since the introduction of early MIMD-type parallel machines in the 1960s, the processing nodes in such systems have continually shrunk in size from large cabinets, to multiboard assemblies, to single boards, and now to chips or small chip sets. Multiprocessor chips are already a reality, constituting a plausible way of utilizing the higher densities that are becoming available [Oluk96], [Hamm97] and chips containing many full-blown processors are eminent [Clar00].

SIMD-type parallel processors have always enjoyed a higher integration density, given their simpler processors [Parh95]. Commercial SIMD machines of the late 1980s already contained tens of bit-serial processors on each chip and more recent products offer hundreds of such processors per chip (thousands on one printed-circuit board). Therefore, it is only a matter of time before we witness many thousands of such SIMD nodes on a single microchip.

Regardless of how many processors we can put on one chip, the demand for greater performance will sustain the need for integrating multiple chips into systems with even higher levels of parallelism. In fact, it is true that the physical size of the largest supercomputer has not changed over the years, implying that with improved integration, we simply build larger systems; this is to quench the thirst for greater performance created by novel applications or larger-scale versions of existing ones. With tens to tens of thousands of processors afforded by billion-transistor chips, small-scale parallel systems utilizing powerful general-purpose processors, as well as multimillion-processor massively parallel systems, will become not only realizable but also quite cost-effective.

Shrinkage in the size of processing nodes has brought about corresponding changes in the art of designing interconnection networks for parallel computers. The cabling between cabinets (once the sole interconnect medium) was augmented by backplane connectors and, later, by board-level links connecting chip-size nodes. The addition of another level, namely intrachip wires, to this hierarchy of interconnects (Fig. 1), may lead one to believe that the task of designing cost-effective high-performance interconnection networks will soon be draped with yet another layer of complexity. Will the grand scale and greater variety of interconnect types necessitate the development of completely new theories and/or design strategies?

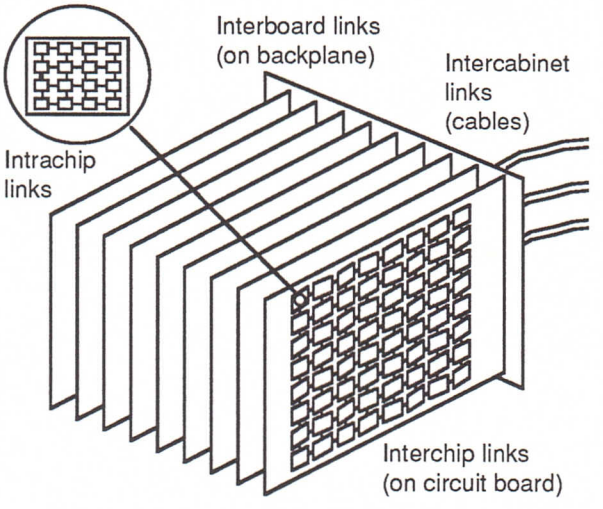

Fig. 1. Interconnect and packaging hierarchy.

We believe not. Given that interconnects are already the limiting issue, we need to rely on multilevel (hierarchical or recursive) parallel architectures anyway; hence, our thesis that network design challenges for single-chip multiprocessors and massively parallel systems, as well as their use in building even larger parallel systems, are not fundamentally different from those currently facing the designers. The difference is in scale rather than substance. The parameters used for deriving optimal configurations under technological constraints must be determined and existing models fine tuned for the new setting. Also, appropriate modeling and verification schemes and tools need to be developed to properly handle the phenomenal increase in complexity. The key theories and design strategies, however, are already in place.

2. Scale of on-Chip Parallelism

Assuming the capability to integrate one billion transistors on one microchip, the first design issue is the number of processors built onto such a chip.

At low scale of on-chip parallelism, one might spend the one billion transistors on a few powerful processors, each with computational capabilities and on-chip memory comparable to current top-of-the-line micros. At an intermediate level, one could opt for a moderate number of processors of lesser power (e.g., single-issue, simple pipeline, no floating-point hardware) and/or memory. At the high end of the scale, one could integrate a very large number of simple processors of the type found in today's custom-chip SIMD machines which are typically bit-serial with relatively small on-chip memories.

To make our presentation more concrete, let us take four design points, with the chip containing varying numbers of processors, and give them names that facilitate further discussion (Fig. 2). An SSP, MSP, LSP, or GSP node contains the processor, its associated memory, and perhaps a router. Interconnect, clocking, and control overheads are broken down and the associated area costs allotted to each processor. Therefore, particularly with larger numbers of processors on the chip, the true transistors-per-node figure is less then the number shown in Fig. 2.

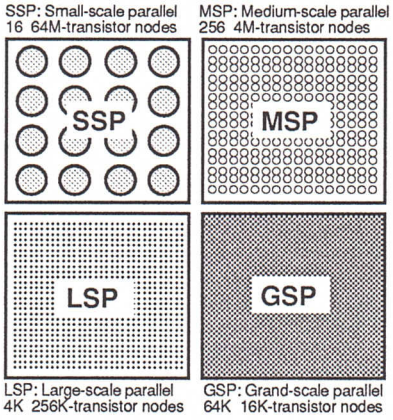

Fig. 2. Four design points for 1B-transistor chip according to the scale of on-chip parallelism.

These numbers are meant as representative points along a continuous scale. We contend that a factor of 2-3 change in the number of processors on a chip would not fundamentally alter our conclusions for each class. So the classes above can be characterized roughly as having tens, hundreds, thousands, and tens of thousands of processors on a chip. One can think of these as SSI, MSI, LSI, and VLSI in which processors have replaced cells or gates.

The cost of such chips is, of course, a matter for concern. A custom billion-transistor chip would be many times more costly to develop than today's multimillion-transistor micros. The repetitive structure of the chip, due to the multiple identical nodes placed on it, is only a minor redeeming factor. It helps, for example, with solving yield problems through the application of defect tolerance techniques developed for wafer-scale integration [Kore86]. Problems with wire delays, clock/power distribution, and power dissipation must still be solved at the chip level, as are those related to the design of glue logic for on-chip and off-chip connectivity between processors.

From past experience with parallel computers requiring custom chips, it is almost certain that custom chip development will not be economically viable for a limited application domain. Instead, off-the-shelf components will likely become available as standard building blocks for parallel systems.

Besides serving as a building block for synthesizing large parallel machine, an SSP chip may be viewed as an alternative to a wide-issue uniprocessor that must rely on multiple independent threads to achieve its full performance potential, and even then, may have much of its computational gain nullified by the effects of interconnect delays; while the designer of the latter must struggle to come up with a modular design with short wires, an SSP chip is already structured in this way [Hamm97]. Thus multiple products may emerge at the SSP level, given the vast desktop/workstation market for a 16-processor chip, say.

In the case of MSP chips, signal processing applications will likely dominate, thus dictating the processors' microarchitecture and on-chip connectivity. Current DSP chips may move in this direction, offering MSP functionality on the same chip or as an adjunct to the main processor. The reasons given in the preceding paragraph can be repeated here as to why an MSP chip with narrow-word processors might be preferable to one based on SIMD fractional-precision and/or streaming extensions to a smaller number of conventional wide-word processors.

LSP chips will probably find a niche in computation-intensive applications, such as physical modeling, that involve both standard full-precision arithmetic and a high level of parallelism. Given the large number of processors on a chip, and the limited per-processor transistor budget, use of bit- or digit-serial arithmetic could prove essential. Alternatively, the architecture may resemble configurable logic arrays, but with advantages in flexibility (powerful word-level cells), signal delays (pipelined or systolic design), and ease of partitioning for running multiple tasks.

Given its extremely simple processors with limited memory, a GSP chip is conceptually quite close to processor-in-memory (PIM) architectures [Gokh95] that integrate the processors into the memory access logic to alleviate the memory bandwidth bottleneck. The need for "intelligent" memories [Kozy97] has been contemplated for several decades, ever since early associative memories were proposed. The billion-transistor-chip capability might be just what is needed to end the tyranny of "dumb" DRAMs. Commodity memory products with integrated processing power will be quite attractive, once we agree on the capabilities that must be built in.

Of course, many hybrid solutions are also possible. For example, a chip may contain eight powerful processors of the SSP variety, as well as 32K very simple ones of the type discussed under GSP. We will not consider such combinations any further.

3. Dominance of Wire Delays

On-chip interconnects comprise local and global wires that link circuit elements and distribute power supply and clock inputs. Downward scaling of VLSI technologies continuously improves device switching or computation times. The effect of this scaling on interconnect performance, however, can be just the opposite, given the increased current density, chip size, and noise margin, along with reduced width and spacing. Short-term solutions to the interconnect delay problem [Sylv99] will not scale indefinitely and are even now inapplicable to global wiring.

Fig. 3 depicts the ratio of wire delay to device switching time as a function of the minimum feature size, extrapolated to the point of allowing one billion transistors on a chip (dotted portion). Two scenarios are shown: Continued use of Al/SiO_2 (top curve) or changeover to less resistive copper wires and an insulator with lower dielectric constant, to reduce resistance and capacitance (bottom curve). In the latter case, downward scaling appears to improve the wire delay problem, but this may not be the case once other factors such as the transmission line effect (largely unknown at present) are taken into account.

At the physical level, the dominance of wire delay will necessitate changes in wiring material and circuit design styles [Mein96]. Architecturally, designs with local data and control flows will become increasingly more attractive. As on-chip wire delays increase, the difference between on- and off-chip transmissions, which is now a determining factor in parallel computer implementations [Basa96] will diminish. However, these changes only affect the numerical values of technology-related parameters. The basic model, based on pin and channel-capacity limitations at various packaging levels, remains valid.

The effect of rising RC delays, resulting from narrower wires and higher densities, on signal propagation constitutes only one aspect of the interconnect problem. Downward technology scaling also affects the coupling capacitance between adjacent wires due to two factors: greater proximity of wires and an increase (in relative terms) of wire height to help mitigate, in part, the effects of reduced width on wire resistance. Such increases in coupling capacitance produce noise and also constitute potential timing hazards [Sylv99], thus again making long wires undesirable.

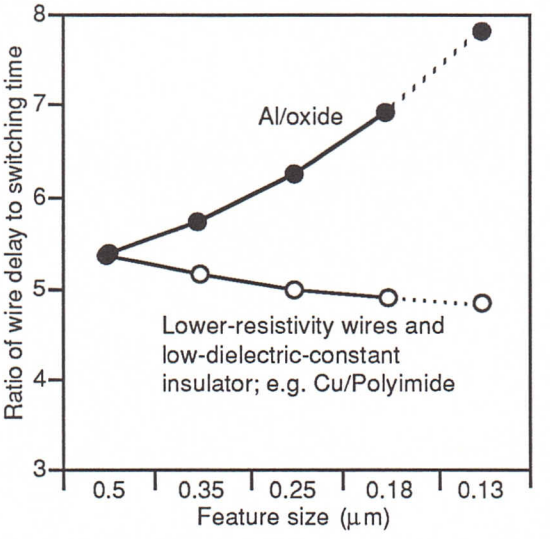

Fig. 3. Changes in the ratio of wire delay to device switching time as the feature size is scaled down. It is assumed that the length (1 cm) of the wire is reduced with feature size and that coupling between wires contributes 20% increase to capacitive loading.

4. Intrachip Connectivity

Multiprocessor and massively parallel microchips require a pattern of on-chip connectivity and must also encompass a way of connecting several chips together for building larger systems, preferably with no "glue" component required. Interchip connectivity schemes will be discussed in the next section. Here, we focus primarily on intrachip connectivity issues.

The nodes of an SSP chip can be richly interconnected with on- and off-chip points without serious area overhead or pin-out problems. For example, several high-speed buses can be provided on the chip for the ultimate in flexibility and performance. Of course, it is also possible to interconnect the 16 nodes as a 4×4 mesh or torus (4-cube), thus achieving higher throughput at a slight increase in latency.

The 256 nodes of an MSP chip can be interconnected, e.g., as a 16×16 mesh/torus or an 8-cube. The factor of two difference in diameters of the torus versus hypercube connectivity of an MSP chip is relatively insignificant once other variables such as channel capacities, router complexities, and signal propagation delays on long wires are taken into account [Parh99]. Any 2D layout of an N-node hypercube leads to wires of length $O((N \log N)^{1/2})$. Extrapolating from current feature sizes and delay models, Fig. 4 shows the expected signal propagation delays on a 1B-transistor chip as a function of wire length. The mesh, torus, and cube architectures are identified on Fig. 4 based on the estimated lengths of their wires in 2D layouts that are optimized with regard to wire length.

The significant propagation delay penalty associated with the 8-cube makes the torus (or even mesh) connectivity quite competitive, especially when the relative ease of connecting I/O ports to its 60 boundary processors, or a subset thereof, is taken into account. Intermediate architectures between the preceding two (e.g., a 4-ary 4-cube) will likely not be attractive in view of added algorithmic complexity or cost over mesh/torus, without a noteworthy decrease in the maximum wire length over the hypercube. Similarly, richer connection schemes, with node degrees greater than eight, are unlikely to be cost effective.

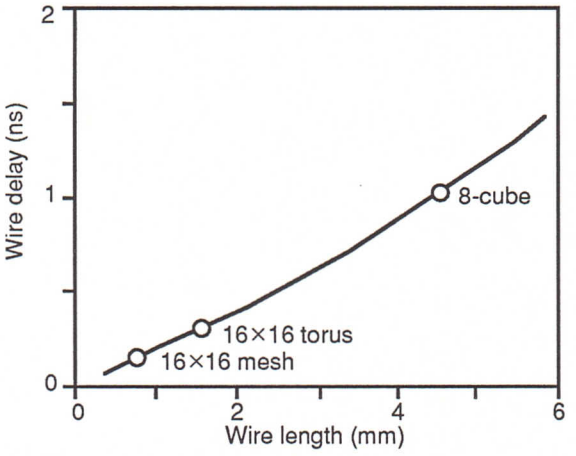

Fig. 4. Intrachip wire delay as a function of wire length. The use of copper wires and node layout area of 0.15 × 0.15 cm are assumed. For mesh topology, intraprocessor wire delays dominate. For torus, intra- and interprocessor wire delays are comparable (folded layout). For hypercube, interprocessor delay is clearly dominant.

As we shift our focus to an LSP chip, the importance of network diameter starts to build up. A 64×64 torus (mesh) has a diameter of 64 (126) which can lead to significant delays. A 12-cube has a much smaller diameter of 12, but besides needing more complex nodes of degree 12, its layout implies significant area overhead and signal propagation delays. Thus, we should consider a hierarchical or recursive architecture on the chip to cope with diameter and long-wire problems in much the same way as we now do for multichip/board MPPs [Yeh98].

One way to combine low diameter with simple layout, and thus shorter wires, is to apply systematic pruning to a dense network [Kwai99], [Kwai00], [Parh99a]. Fig. 5 [Kwai00] exemplifies the layout area savings that can be obtained with pruning. For k-ary n-cube networks, pruning strategies are known that yield node-symmetric networks with substantially the same diameters as the unpruned networks. The negative effect of pruning on the bisection bandwidth is, at least in part, mitigated by the use of wider channels made possible through node degree reduction.

The often made assertion that network diameter is no longer important, because we tend to use wormhole routing in modern parallel machines, is worth a mention here. Even in the limited context of current practice (no more than a few thousands of processors, wormhole switching, and relatively long messages to mitigate the send/receive overheads), this assertion is debatable [Parh00]. As we begin to seriously entertain the notion of million-processor architectures [Clar00], network topology once again assumes a crucial role.

Finally, in the case of GSP, our options for on-chip connectivity are quite limited. Given many thousands of nodes on a chip, any increase in interconnection complexity over a simple 256×256 mesh or torus is likely to consume so much of the chip area as to leave little real estate for the nodes' computing and storage functions. This can be self-defeating in that such an underpowered node can do little before requiring communication with other nodes.

Use of configurable interconnects may offer some advantages for GSP. With due care during the design, such configurable interconnects can serve the dual purpose of yield enhancement and run-time setup of efficient communication paths with relatively low overhead. By its nature, a GSP chip is more rigid than, say, an SSP chip, making its application domain more limited. Thus, configurability can also be viewed as desirable for mitigating the economic disadvantage that comes with narrow applicability.

It thus appears that on-chip connectivity problems for SSP and MSP are currently within our grasp; but LSP and GSP present certain challenges that must be confronted with intensified research on interconnection schemes and their VLSI layouts.

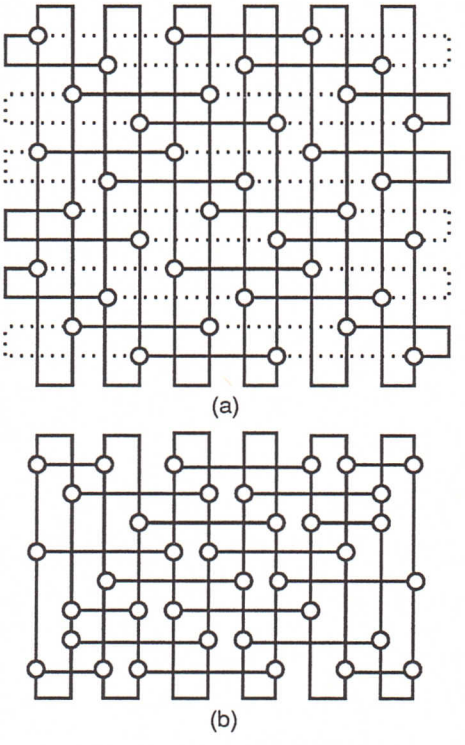

Fig. 5. The effect of pruning on network layout area complexity. (a) Folded layout of a 2D torus, with links to be pruned depicted as dotted lines. (b) More compact layout for the pruned torus.

5. Interchip Architecture

Given a particular on-chip connectivity, two issues must be considered for building larger parallel systems. The first of these, the provision of off-chip links, is really within the chip design realm. However, one must look at the potential overall architectures in order to decide on suitable off-chip connectivity. Perhaps the most general and flexible option is to provide one (or a handful of) off-chip port(s) per processor. This is feasible for SSP chips and workable, with serial or time-shared ports, for MSP. However, such ports will suffer from the double penalty of off-chip propagation delays and long on-chip wires.

Again, hierarchical interconnection networks provide a solution. A variety of hierarchical architectures can be built when every processor on the chip is directly accessible from outside [Yeh98]. An example is shown in Fig. 5, where a chip (cluster) is connected to other chips via intercluster links. In such a scheme, all routers will be identical, thus leading to manufacturing simplicity (e.g. fault tolerance for yield enhancement) and algorithmic uniformity.

In most known hierarchical architectures, performance advantage is gained via the replacement of off-module communications with (a larger number of) on-module transfers. Thus, the communication performance of the low-level modules (chips) is a determining factor in the overall performance. This points to the importance of research on hierarchical architectures, based on large building blocks, whose performance is less sensitive to the low-level connectivity.

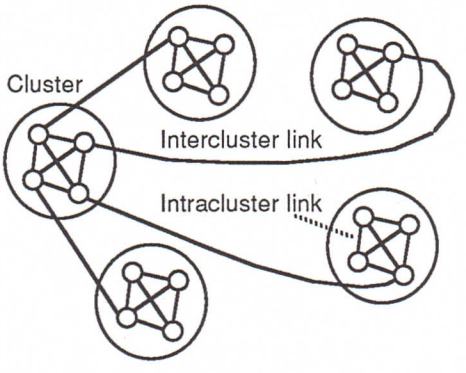

Fig. 5. Two-level hierarchical architecture (partially shown), with clusters, intracluster links, and possible intercluster connectivity using a single off-cluster link per processor.

For the sake of concreteness, let us focus on two-level hierarchical networks to demonstrate some of the design issues and challenges. Figure 6 depicts example two-level 64-node networks built of 8-node clusters. The two networks, which are different in the densities of their intracluster and intercluster connectivities, can be compared in a variety of ways. With respect to ease of packaging, the network of Fig. 6a is preferable. The bisection bandwidth, which is often more sensitive to intercluster connectivity than the intracluster one, is larger for the network in Fig. 6b, making it more likely to do better in random routing.

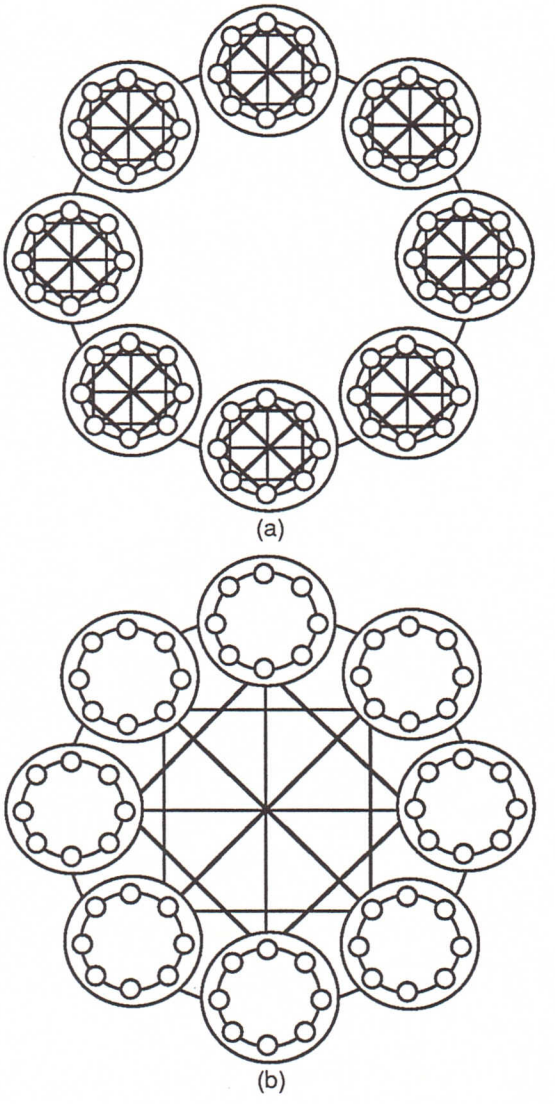

Fig. 6. Examples in the design space for two-level hierarchical interconnection networks: (a) Sparse connection of dense clusters, and (b) Dense connection of sparse clusters.

Note that the diameters of the two networks in Fig. 6, or their performance in executing various parallel algorithms, cannot be compared without additional information on hardware details, and in particular how the intercluster links are attached to the nodes within clusters. We, thus, begin to appreciate the intricacies of network design for multilevel implementation.

The fact that direct off-chip access to all processors cannot be provided for LSP and GSP chips reinforces our earlier conclusion that a hierarchical on-chip connectivity, or else a PIM-type design philosophy, is required. Design issues now become more complicated as the particular hierarchical scheme chosen for on-chip connectivity may not match that used for the higher-level architecture, potentially leading to inefficiencies and algorithmic complexity. The resulting design problems can be handled, e.g., by taking advantage of

techniques for parallel system implementation using both hierarchical and recursive compositions [Yeh98].

Even though different connectivities can be combined in an ad hoc manner to build a multilevel network, it would be better from the viewpoint of algorithmic simplicity, as well as ease of design and analysis, to apply a uniform design scheme that determines the node connectivity rules in such a way that modularity is automatically ensured. This would lead to simpler routing and computational algorithms that do not need to keep track of the level of hierarchy in order to adjust their behaviors accordingly.

The foregoing desirable property can be ascertained through the use of index-permutation graphs [Yeh00]. As a unified model of hierarchical networks, index-permutation graphs allow us to modify network structure and capabilities via suitable adjustments in network components (nuclei), connectivity rules (generators or supergenerators), and other parameters.

To summarize, known design methods for hierarchical interconnection networks must be extended and refined to allow systematic cost-performance tradeoffs during the design process of large systems employing parallel microchips. Whereas off-chip connectivity decisions for SSP and MSP can be handled with ad hoc methods, the use of a common design methodology is beneficial even in these latter cases.

6. Conclusion

As these words are being written, the state of available computational power can be summarized as follows: GFLOPS on desktop, TFLOPS in supercomputer center, PFLOPS on drawing board. The foregoing prefixes used to be M, G, and T a little over a decade ago, and we almost routinely expect them to become T, P, and H (?) early in the next decade, if not sooner. However, it is almost certain that the latter transition will not occur solely by improving the architectural methods that fueled the growth of computational power in the 1980s and 1990s.

Emergence of multiprocessor and massively parallel microchips are expected to help in this regard. We do not see any insurmountable problem in building and utilizing multiprocessor or massively parallel chips containing of the order of one billion transistors. Multiprocessor chips, already a reality, will enhance the capabilities of future workstations as well as provide building blocks for higher performance systems. With massively parallel microchips, multimillion-processor MPPs no longer constitute an unrealizable or unaffordable dream.

Emerging research results on the synthesis of parallel architectures under propagation delay, channel capacity, and packaging constraints, when suitably modified to take the new physical and technological parameters into consideration, can lead to the solution of anticipated design problems. Optimal operating points with regard to system partitioning and interconnection structure may change, but the models and know-how for dealing with the most fundamental design problems are already in place.

References

[Abra91] Abraham, S. and K. Padmanabhan, "Performance of Multicomputer Networks under Pin-out Constraints", *J. Parallel and Distributed Computing*, Vol. 12, pp. 237-248, July 1991.

[Bald96] Baldi, L., "Industry Roadmaps: The Challenge of Complexity", *Microelectronic Engineering*, Vol. 34, No. 1, pp. 9-26, Dec. 1996.

[Basa96] Basak, D. and D.K. Panda, "Designing Clustered Multiprocessor Systems under Packaging and Technological Advancements", *IEEE Trans. Parallel and Distributed Systems*, Vol. 7, pp. 962-978, Sep. 1996.

[Burg96] Burger, D., J.R. Goodman, and A. Kagi, "Quantifying Memory Bandwidth Limitations of Current and Future Microprocessors", *Proc. Int'l Symp. Computer Architecture*, May 1996, pp. 78-89.

[Clar00] Clark, D., "Blue Gene and the Race Toward Petaflops Capacity", *IEEE Concurrency*, Vol. 8, pp. 5-9, Jan.-Mar. 2000.

[Gokh95] Gokhale, M., B. Holmes, and K. Iobst, "Processing in Memory: The Terasys Massively Parallel PIM Array", *IEEE Computer*, Vol. 28, pp. 23-31, Apr. 1995.

[Hamm97] Hammond, L., B.A. Nayfeh, and K. Olukotun, "A Single-Chip Multiprocessor", *IEEE Computer*, Vol. 30, pp. 79-85, Sep. 1997.

[Kore86] Koren, I. and D.K. Pradhan, "Yield and Performance Enhancement Through Redundancy in VLSI and WSI Multiprocessor Systems", *Proceedings of the IEEE*, Vol. 74, pp. 699-711, May 1986.

[Kozy97] Kozyrakis, C.E. et al, "Scalable Processors in the Billion-Transistor Era: IRAM", *IEEE Computer*, Vol. 30, pp. 79-85, Sep. 1997.

[Kwai99] Kwai, D.M. and B. Parhami, "Comparing Torus, Pruned Torus, and Manhattan Street Networks as Interconnection Architectures for Highly Parallel Computers", *Proc. Int'l Conf. Parallel and Distributed Computing and Systems*, Nov. 1999, pp. 19-22.

[Kwai00] Kwai, D.-M. and B. Parhami, "A Unified Formulation of Honeycomb and Diamond Networks", *IEEE Trans. Parallel and Distributed Systems*, to appear.

[Mein96] Meindl, J.D., "Gigascale Integration: Is the Sky the Limit?", *IEEE Circuits and Devices*, Vol. 12, No. 6, pp. 19-23 & 32, Nov. 1996.

[Oluk96] Olukotun, K., B.A. Nayfeh, L. Hammond, K. Wilson, and K. Chang, "The Case for a Single-Chip Multiprocessor", *Proc. Int'l Symp. Architectural Support for Programming Languages and Operating Systems*, Oct. 1996, pp. 2-11.

[Parh95] Parhami, B., "Panel Assesses SIMD's Future", *IEEE Computer*, Vol. 28, pp. 89-91, June 1995. Extended version: www.ece.ucsb.edu/faculty/parhami

[Parh99] Parhami, B., *Introduction to Parallel Processing: Algorithms and Architectures*, Plenum Press, 1999.

[Parh99a] Parhami, B. and D.-M. Kwai, "Periodically Regular Chordal Rings", *IEEE Trans. Parallel and Distributed Systems*, Vol. 10, pp. 658-672, June 1999.

[Parh00] Parhami, B. and C.-H. Yeh, "Why Network Diameter is Still Important", these Proceedings.

[Sylv99] Sylvester, D. and K. Keutzer, "Rethinking Deep-Submicron Circuit Design", *IEEE Computer*, Vol. 32, pp. 25-33, Nov. 1999.

[Wain97] Waingold, E. et al, "Baring It All to Software: Raw Machines", *IEEE Computer*, Vol. 30, pp. 86-93, Sep. 1997.

[Yeh98] Yeh, C.-H. and B. Parhami, "Synthesizing High-Performance Parallel Architectures under Inter-Module Bandwidth Constraints", *Proc. Int'l Conf. Parallel and Distributed Computing and Systems*, Oct. 1998, pp. 414-416.

[Yeh00] Yeh, C.-H. and B. Parhami, "A Unified Model for Hierarchical Networks Based on an Extension of Cayley Graphs", *IEEE Trans. Parallel and Distributed Systems*, to appear.