

# SIMD Machines: Do They Have a Significant Future?

Report on a Panel Discussion at Frontiers '95: The Fifth Symp. on the Frontiers of Massively Parallel Computation, McLean, VA, Feb. 6-9, 1995; Sponsored by IEEE Computer Society

Behrooz Parhami, Univ. of California, Santa Barbara  
E-mail: parhami@ece.ucsb.edu

## 1. About This Report

I volunteered to write this report during the SIMD panel session held on 2/9/95 at Frontiers '95. All panelists cooperated by sending me their transparency masters. A draft report was prepared based on these transparency masters, position statements published in the Frontiers '95 proceedings on pp. 466-469, and my own notes. The draft was e-mailed on 3/17/95 to the panel organizer/moderator and the panelists for their comments. This final version of the report is based on comments and markups received through 3/31/95. I have drawn from the panelists' ideas freely, using quotation marks only when including their statements verbatim.

The panel consisted of both academic and industry experts in the field of massively parallel systems (see the table below). All but Tim Bridges, who is currently involved in a large-scale software development project for the MasPar MP-2 SIMD architecture, have built working SIMD machines. The vast practical experience of the panel was quite evident in the insightful presentations and interactions. It is indeed a privilege for me to have worked on this report.

### Introducing the panel

Name	Affiliation	Background	E-Mail Address
Ken Batcher	Kent State U.	STARAN, MPP#, Networks	batcher@mcs.kent.edu
Tim Bridges	Data Parallel Systems	DPSI Founder, Pres. & CEO	bridges@dpsi.com
Ken Iobst	SRC, Cray-3/SSS Proj.	MPP#, Cray's PIM chip+	ken@super.org
John Nickolls	MasPar Computer Corp.	MasPar Co-founder & VP	nickolls@maspar.com
Stewart Reddaway	Cambridge Par. Proc'g	Creator of ICL's DAP=	sfr@cppuk.co.uk
H.J. Siegel*	Purdue U.	Multistage nets, PASM¢	hj@ecn.purdue.edu
Charles Weems	U. of Massachusetts	IUA\$, Hetero. parallelism	weems@cs.umass.edu

\* Panel organizer and moderator.

# STARAN and Massively Parallel Processor, SIMD machines by Goodyear Aerospace.

+ Processing-In-Memory chip, described in Section 4 of this report.

= Distributed Array Processor, described in Section 4 of this report.

¢ PARTitionable Simd/Mimd machine, prototype reconfigurable parallel system.

\$ Image Understanding Architecture, designed for integrated real-time vision tasks.

## 2. What is SIMD?

The first massively parallel machines had single-instruction-stream multiple-data-stream or SIMD (Sim-Dee) designs. SIMD implies that a central unit fetches and interprets the instructions and then broadcasts appropriate control signals to a number of processors operating in lock step. This initial interest in SIMD resulted both from characteristics of early parallel applications and economic necessity. Some of the earliest applications, such as air traffic control, are what several panelists characterized as "embarrassingly parallel" (H.J. Siegel and I prefer the more positive terms "parallel-machine-friendly" and "pleasantly parallel"). Such applications tend to be much easier to program in SIMD languages and lead to more cost-effective SIMD hardware. On the economics side, full-fledged processors with reasonable speed were quite expensive in those days, thus limiting any massively parallel system (one having > 1000 processors, say) to the SIMD variety.

### **3. Why This Panel?**

Judging by what commercial vendors have introduced in the 1990s, the MIMD (Mim-Dee), or multiple-instruction-stream multiple-data-stream, paradigm has become more popular recently. The reasons frequently cited for this shift of focus are the higher flexibility of the MIMD architectures and their ability to take advantage of commodity microprocessors, thus avoiding lengthy development cycles and getting a free ride on the speed improvement curve for such microprocessors. It thus "seems like an appropriate and interesting time to assess where the industry is heading in terms of the use of SIMD versus MIMD parallel architectures, and what forces are making this the case" (Siegel).

Questions posed by the organizer to the panel upon its formation were intended to provide a comparison of SIMD and MIMD classes with respect to issues such as the following:

- a. Size/composition of both machine and user bases.
- b. Problems best/worst suited for each class.
- c. Ease of programming (program design and debugging).
- d. Suitability for general-purpose computation.
- e. Gaining cost advantage from commodity processors.
- f. Cost-effective user access through spatial sharing.

Additionally, the panel was asked to comment on ways that the advantages of these classes can be combined and the extent to which industry is influenced by scientifically unsubstantiated user perceptions, funding agency politics, and the tendency to choose economic expedience over technological soundness.

"Is SIMD dead?" was how Weems paraphrased the main question facing the panel. His next three questions provide a convenient framework for discussing the panelists' views.

### **4. Why is SIMD Still Alive?**

The main reasons can be found in questions a, b, and c above. SIMD is alive because it provides more performance per dollar for a vast collection of pleasantly parallel problems that are of considerable interest to the scientific computation, embedded control, and database communities. This is because SIMD uses SIMPLE, more readily scalable, hardware to implement data parallelism, which is a SIMPLE programming model. SIMD also offers advantages in hardware testing, reliability, and speed/precision trade-offs. Parallel SIMD and vector SIMD (Cray, Fujitsu, ...) machines have collectively dominated the high end of the supercomputer market thus far and will continue to do so for the foreseeable future. SIMD is not just alive but thriving.

The relatively large installed base and long-term customers of parallel SIMD machines are attributable to this cost-effectiveness. For example, ASPRO, a little-publicized parallel SIMD machine with 1792 processing elements, was first built in the late 1970s. With over 170 systems delivered for use in aircraft early warning radar surveillance and command and control processing, ASPRO is still being built by Loral Defense Systems in Akron. The Distributed Array Processor (DAP), introduced by ICL in the mid 1970s, is now in its fourth generation. Currently marketed by Cambridge Parallel Processing, DAP has enjoyed similar success in its niche markets, such as signal and image processing, with well over 100 installations.

Commercial development and sales of SIMD machines for somewhat more general applications is also continuing. MasPar has been fairly successful in its target market of signal/image processing, decision support, and bioinformatics with its MP-1 and MP-2 systems, with over 240 systems shipped since 1990. A major software development effort by Data Parallel Systems, Inc., supported by NASA, MasPar, DEC, NSF, and IBM, is in progress with the aim of making MasPar systems even more attractive for decision support applications. Cray Computer Corp. will provide 512K SIMD processors in its forthcoming Cray 3/SSS Super Scalable System. These are single-bit

processors associated with each column of an otherwise standard RAM within custom processing-in-memory (PIM) chips that form a portion of the system's memory address space. This appears to be a promising approach for scalability beyond one million processors.

## 5. Why is SIMD Ailing?

The main reasons can be found in questions d, e, and f above. SIMD suffers, as do all other parallel processing paradigms, from the recent decrease in demand for high-end systems and the ever-present difficulty of finding and exploiting application parallelism. It is no secret that parallel programming languages and software development tools have not kept pace with the phenomenal growth in performance and the decline in cost of hardware. "We desperately need better high-level languages ... simple extensions to C or FORTRAN do not do it" (Ken Iobst).

Moreover, parallel systems find themselves in competition with powerful microprocessors that are conveniently accessible in personal workstations compared to the time-shared availability of parallel systems through a network with unpredictable delays and waits. As Ken Batcher put it: "Queuing theory is important." Virtually all SIMD developments to date have been based on custom chips, with their attendant design and testing costs borne by a relatively small user base (compared to main line microprocessors). While an SPMD system (MIMD with all processors executing copies of the same program) can effectively emulate SIMD computing, it is ultimately not cost-effective for most pleasantly parallel applications for which SIMD has been found attractive. Spatial sharing can be achieved by providing multiple SIMD (M-SIMD), but the added overhead and complexity may not prove worthwhile.

The SIMD paradigm is perceived as being inefficient for applications that require high-precision arithmetic, conditional computations (especially multiway branching), and indirect references, as well as for applications with limited parallelism. Thus, the perception that SIMD machines have very narrow application areas. SIMD is also ailing because it is approaching scalability limits (in terms of interfacing, clock distribution, and synchronous communications) and implies large incremental expansion steps in its current implementations.

## 6. How Do We Save SIMD?

Clearly we need to be more diligent in addressing the few real technical problems (as opposed to a much larger number of imaginary or perceived problems) outlined in the preceding section. Suggested approaches in dealing with real technical challenges include improvements in processor density and speed to remain competitive, mixing synchronous (intrachip) and asynchronous (interchip) communications, cooperating rather than competing with microprocessors by allowing direct interaction channels, developing and using commodity components similar to the PIM chip, and improving downward (perhaps down to a single chip) as well as upward scalability.

However, we also need to embark on an educational campaign to dispel some of the myths about the inefficiency of the SIMD approach. Insisting that all hardware be busy most of the time is no more valid for SIMD processing elements than for computer memory cells. "To say that the inactive processors are doing nothing is like saying that the memory you are not currently accessing in your uniprocessor is doing nothing" (Chip Weems). Tim Bridges and his colleagues have demonstrated that redundant silicon is inevitable even in MIMD machines. Ken Batcher cited the work of Hank Dietz at Purdue, suggesting that MIMD code can be automatically converted to SIMD with reasonable efficiency. Because the silicon area of one MIMD processor can be packed with at least eight SIMD processors, even a conversion efficiency of 12.5% is a win for SIMD. As for the difficulty of spatial sharing of SIMD systems, Stewart Reddaway pointed out that an equivalent benefit is provided by low-overhead time-sharing in DAP when the memory can hold two or more programs.

Furthermore, system functionalities and their associated costs are the primary criteria for selecting a particular hardware platform to develop large-scale software. "We do not pose the question 'SIMD or

MIMD?' until long after we have identified the characteristics of the application that our software will support" (Tim Bridges). Similarly, customers would not reject orders-of-magnitude improvement in performance/cost ratio on the basis of a dislike for the underlying architecture, when provided with an overall packaged solution to their problem(s). We must thus emphasize and exploit the strengths of the SIMD approach in developing SIMD-friendly applications that provide compelling value to customers. Such applications abound in signal/image processing, text retrieval, large databases (e.g., fingerprints), and data fusion for command and control, among others. We will no doubt witness the emergence of such compelling values in the near future as a result of ongoing development efforts.

The SIMD approach is highly effective in dealing with associative data dependence, variable- or low-precision arithmetic, and high-bandwidth I/O. As an example of the strengths of the fine-grain massively parallel SIMD approach, Stewart Reddaway pointed out the performance advantage of building square-root and exponential functions, among others, from more fundamental building blocks than word-length additions and multiplications for which microprocessors are optimized. An interesting case in point is the ability of DAP to exceed its "peak" floating-point performance rating for some library FFTs in which special code executes both an addition and a subtraction in well under twice the normal add time.

## 7. Conclusions

So, does SIMD in fact have a significant future? Ken Batcher summarized it best when he said: "SIMD has a significant future IF AND ONLY IF massive parallelism has a significant future AND supercomputing has a significant future." One does not have to look beyond the worldwide interest in grand-challenge application problems and the steep sustained rise in performance/cost ratio for integrated circuits, allowing multiple-processor chips and in some cases highly parallel chips, to conclude that the answer to the latter two questions is a definite YES.

Almost no one argues against the usefulness of massively parallel SIMD machines as special-purpose servers within heterogeneous computing environments. But, based on cautious extrapolation, even a parallel SIMD coprocessor embedded in a single-user workstation may not be such a far-fetched idea. SIMD machines were first in achieving giga-bit-ops (operations per second) and tera-bit-ops capabilities. It seems that SIMD will continue to lead the way in performance improvements for many real applications, perhaps reaching the peta-bit-ops milestone around the turn of the Century.

## 8. Bibliography

Almasi, G.S. and A. Gottlieb, *Highly Parallel Computing*, Benjamin/Cummings, 2nd Ed., 1994. [Good general reference on high-performance and parallel architectures. See, in particular, the SIMD/MIMD overview on pp. 377-379 and comparison on pp. 442-443.]

Bridges, T., S.W. Kitchel, and R.M. Wehrmesiter, "A CPU Utilization Limit for Massively Parallel MIMD Computers," *Proc. of the 4th Symp. on the Frontiers of Massively Parallel Computation*, McLean, VA, pp. 83-92, Oct. 1992. [See Section 6 in this report.]

Cypher, R. and J.L.C. Sanz, *The SIMD Model of Parallel Computation*, Springer-Verlag, 1994. [Surveys a variety of SIMD architectures and provides an extensive bibliography on SIMD parallel computing, emphasizing algorithm design and analysis.]

Hord, R.M., *Parallel Supercomputing in SIMD Architectures*, CRC Press, 1990. [Describes several SIMD machines, old and new, such as the ILLIAC IV, MPP, DAP, GAPP, and CM-2. A companion volume, published in 1993, deals with MIMD architectures.]

Siegel, H.J., J.B. Armstrong, and D.W. Watson, "Mapping Computer-Vision-Related Tasks onto Reconfigurable Parallel Processing Systems", *Computer*, Vol. 25, No. 2, pp. 54-62, Feb. 1992. [Besides the parallel computer vision overview, provides sidebars on SIMD/MIMD comparison.]