Participants in the First Workshop on CAD Engines, held June 8-9 in Tokyo, saw massively parallel computers and better interfaces and software tools in the future of simulation acceleration.

The meeting was sponsored by the International Federation of Information Processing with the cooperation of the Institute of Electronics, Information, and Communication Engineers of Japan and the Computer Society's Design Automation Technical Committee. Among the 78 attendees were 55 from Japan, 17 from the United States, three from the United Kingdom, and three from West Germany. Both industry and academia were represented.

The workshop focused on CAD engines for various phases of VLSI design, including logic simulation, fault simulation, circuit simulation, and layout. Many of the 26 technical presentations dealt with the use of parallel processing for design automation acceleration, reflecting the high interest in this topic in both universities and company labs. Hardware configurations ranging from array and vector processors to networks of multiple processors were described. The increasing use of more general-purpose multiprocessors for acceleration of CAD applications was also reflected in the presentations. The workshop concluded with a panel discussion of issues related to the integration of CAD engines into design environments and the provision of software support for them.

The workshop opened with "Man-yo: A Mixed-Level Parallel Logic Simulation Engine," presented by Nobuhiko Koike of NEC. Man-yo (a Japanese expression signifying 10,000 leaf nodes in a processor tree) is a second-generation hardware accelerator—a parallel processor array—designed to work as a global accelerator for tasks such as logic simulation, logic verification, and logic synthesis. Man-yo achieves performance in mixed-level simulation by exploiting subcircuit and algorithm concurrency and by combining hardware, firmware, and software algorithmic implementations. Each processor module handles a separate subcircuit and each is composed of a hardware engine (for gate-level simulation), a firmware engine (for functional simulation), and a general-purpose processor (for behavioral simulation). A prototype consisting of four processors and running a parallel functional-level simulator has been developed. Interprocessor synchronization is performed in software. Initial results show a factor of 2.4 speedup over the one-processor case. The maximum speedup is expected to improve to around 3.

Y. Takamine of Hitachi described Velvet, a vector-processing logic verification system implemented on the Hitachi S-810 supercomputer. It consists of a logic compiler, a test data compiler, a simulation nucleus, and an output processor. Most of the simulation nucleus and output processor was written with vector instructions. Six new vector instructions for logic simulation were added to the S-810 supercomputer. The simulation algorithm is a zero-delay, event-driven algorithm with clock suppression. The clock suppression eliminates about 60% of the events. Experiments with Velvet showed that the hardware improvements provide a 50% speedup and the software improvements a factor of 2 speedup over the performance of a comparable software simulator. Moreover, logic compilation, test data compilation, and output processing are two to three times faster. Hence, the total simulation speed of Velvet is over 100 times greater than that of the software simulator.

Fault simulation is another application that can be accelerated by vector-processing techniques. N. Ishiura of Kyoto University described an implementation of a fault simulator on a general-purpose vector processor, the Fujitsu Facom VP-200. The simulator is based on "dynamic two-dimensional parallel simulation"—a novel technique employing vectorization, fault dropping, and selective tracing. This technique follows two strategies. First, it simulates many fault cases simultaneously in pattern-parallel fashion. Second, it changes the relationship between a pattern-parallelism factor and a fault-parallelism factor, in a complementary fashion, on each pass according to the number of untested faults and the number of faults likely to be detected in the pass. For instance, it sets the pattern-parallelism factor low and the fault-parallelism factor high in early passes and the pattern-parallelism factor high and the fault-parallelism factor low in later passes. Simulation with 512K random patterns demonstrated a 10 to 15 times faster performance due to vectorization. Ishiura noted that the simulator is 130 to 400 times faster than the parallel-pattern single-fault propagation (PPSF) simulator implemented on the VAX-11/785.

Automatic test-pattern generation is one of the most computationally intensive CAD tasks. It requires the same process to be repeated applied to the same data. Hence, it is a good candidate for acceleration through the use of a multiprocessor system. The acceleration of automatic test-pattern generation in a parallel CAD environment was the subject of a presentation by Anthony Ambler of the UK’s Brunel University. He noted that performance analyses have shown that the transferal of existing automatic test-pattern generation algorithms onto powerful, highly parallel multiprocessor accelerators results in only modest speed improvements. He suggested an alternative approach to automatic test-pattern generation, one using multilevel descriptions of complex sequential circuits that allows efficient implementation of a test-generation algorithm in a highly parallel...
CAD accelerator. Ambler noted that work is in progress on porting this algorithm—and one for logic simulation, circuit simulation, and placement and routing—to the Intel iPSC hypercube and to an Inmos Transputer “supernode” developed by Esprit, a European research consortium.

Richard Newton and Alberto Sangiovanni-Vincentelli discussed research in multiprocessing at the University of California, Berkeley. They reviewed past and present work in applying multiprocessors to electrical simulation, simulated annealing, logic verification, and 3-D device simulation. Many of these applications have been developed on “conventional” multiprocessors with up to 1000 processors, and there is growing interest in utilizing massively parallel computing systems with more than 10,000 processors—for example, the Connection Machine. The speakers strongly expressed the view that massively parallel computing offers interesting opportunities for research. They also asserted that

- breakthroughs will come from new algorithms, not from remapping old ones
- the type of hardware implementation (for example, whether shared memory or message passing should be chosen) should not be a concern to the user.

- much work in software development environments remains to be done
- general-purpose multiprocessors with special-purpose coprocessors will meet most needs in price/performance and will capture most of the market

Rob Rutenbar of Carnegie Mellon University described his experience with parallel annealing in shared-memory and message-passing environments. He reviewed three cases of parallel annealing, each on a different parallel machine: standard cell placement on a four-processor, shared-memory VAX-11/784; standard cell placement on a 16-processor, shared-memory Encore Multimax; and floor planning on a 16-processor, message-passing Intel iPSC hypercube.

Rutenbar reported that the consensus is that annealing is very adaptable to parallel implementation on many architectures. Reasonable speedups (2x to 3x) and good results were achieved on the VAX-11/784. Speedups of 7x to 8x were achieved on the iPSC. Progress on the Encore has been temporarily halted by the instability of the Multiprocessor C compiler and the Mach operating system (both under development at CMU). Rutenbar also noted that the architecture and software environment have a big influence on partitioning and implementation. The important question is whether this influence should exist, he said. An ideal virtual machine—one in which all hardware details are hidden—makes development easier but the performance obtained is less than optimal. The pragmatic view would be to have some hardware details visible to get better performance.

In the panel discussion concluding the workshop, the critical importance of the software environment and of software tools was emphasized. According to Leslie Smith of DEC, a fast accelerator that cannot communicate is worse than a slow accelerator that can. Her statement was supported by Pratima Agrawal of AT&T, who stressed the importance of the integration of CAD engines into the existing CAD environment: They should provide easy access and a familiar interface. Nobuhiko Koke of NEC pointed out the necessity of using the same interface and obtaining the same results in an accelerated environment as in a nonaccelerated one. It is also necessary to provide a debugging and user interface that supports mixed-mode acceleration, he said. The bottom line, the panelists concluded, is that CAD engines should evolve with design styles and methodology.

---

First Annual International Conference on Neural Networks

Mike Parks, MCC

More than 2000 people—four times the expected turnout—attended the First Annual International Conference on Neural Networks June 21 to 24 in San Diego. Questions raised by the attempt to understand these networks often look like the questions raised by the attempt to design complex VLSI systems—hence once source of the high level of interest in this conference.

The meeting was cosponsored by the IEEE San Diego Section, the IEEE Systems, Man, and Cybernetics Society, and the IEEE Control Systems Society. Almost 200 papers were presented.

Jim Reggia of the University of Maryland reported results in implementing a text-to-speech system using indirect or allocational competition. A winner-take-all network, in which every node can directly inhibit every other node, would have required 1.2 million lateral inhibitory connections. The allocational approach required only a few thousand connections.

W.B. Press, in “Frequency-Coded Artificial Neural Networks,” reported work implementing neural networks with nodes consisting of variable-frequency oscillators. This approach draws on the fact that real neurons emit pulse trains, although it is not clear whether the information is transferred by the frequency of the pulses or by the
recognition of more complex patterns in the pulses.

In “Programming Parallel Computers Using Energy Minimization Algorithms,” S.D. Simmes presented evidence that energy minimization algorithms (for example, neural network algorithms) are very well suited to decomposition for solution on multiprocessors. Current simulations on uniprocessors exhibit very poor performance, suggesting that “real” problems will yield to neural techniques only when hardware accelerators are available. A multiprocessor approach may mitigate the need for special-purpose hardware.

Bart Kosko, in “Adaptive Inference in Fuzzy Knowledge Networks,” reported on the combination of neural-network expert systems into a single large system without the performance degradation attendant on a conventional match-select-act architecture.

In “Representing Conceptual Structures in a Neural Network,” Dave Touretzky of CMU reported on a two-level associative retrieval system that corrects partially incorrect key specifications. It uses a connectionist-based frame-like knowledge representation approach. Inheritance is not a part of the frame system. Instead, microfeatures are explicitly encoded to handle exceptions to captured regularities in the knowledge.

Judith Styles, in “Detection of Favorited Patterns in the Temporal Structure of Nerve Cell Connections,” presented evidence that favored patterns do exist and can be correlated to sensory input in animals. A favored pattern is a temporal sequence within a neuron’s spike train that occurs more often than would be expected if the neuron’s output were random. Patterns were found in the cortex of a monkey expecting food that were not present while the animal was eating (or not expecting food).

Deborah Walters of the State University of New York at Buffalo reported a project that found that biological networks tend to use representations that fall in the middle of the variable representation space. In other words, many neurons are neither fully “on” nor fully “off.” This is in contrast to the representation used in most artificial neural network models. Walters suggested the devising of a theoretical framework for variable representation in artificial neural network models that would be more closely aligned with biology.

In “Self-Organization of Stable Category Recognition Codes for Analog Input Patterns,” Gail Carpenter and Stephen Grossberg of Boston University presented the results of a government-sponsored project in which automatic classification of 20 different shapes was done under conditions of scaling, rotation, and noise.

**Plenary sessions**

Several luminaries in the field presented longer invited papers in the evening sessions. One of Caltech professor Carver Mead’s messages was “Bad news: we must learn how to design analog VLSI systems.” He was referring to his belief that the most efficient way to implement neural networks in hardware is to make the neurons out of analog circuits rather than digital ones. He also pointed out a fact important to the long-term future of neural network research: The cortex is basically a two-dimensional structure. It is roughly one meter × one meter × one millimeter, with most of the thickness composed of white matter, i.e., of wiring. Knowing this simplifies somewhat the technology needed to wire up something similar to the cortex, although it will still be many years before as many processing elements and as many connections as the cortex can be put into an artificial neural network.

Bernard Widrow put the research discussed at the conference into a historical perspective by presenting some work done in the mid-1960’s, including some work on a connectionist approach to speech recognition. The results he showed were almost too good and the reports he referenced seemed almost too contemporary, raising the question of how much progress has really been made in the last 20 years.

Representatives of several government agencies—including DARPA, NSF, the Office of Naval Research, the Air Force Office of Scientific Research, and the Air Force Wright Aeronautics Laboratory—described work they are interested in supporting and told the plenary session attendees how to apply for funds.

**Vendor exhibits**

About 20 vendors were represented at the conference, in booths ranging from the large and elaborate to the small and simple. Hecht-Nielsen Neurocomputer Corporation exemplified the large exhibitor—they demonstrated several applications of their IBM PC/AT coprocessor board and software designed to support neural network paradigms. At the other end of the scale, a university was giving away copies of a simulator for the Apple Macintosh.

**Attention readers!**

**Save the reputation of your colleagues!**

Rumor has it that design and test engineers, technical staff, and even management have no opinions, no sense of humor, and nothing to complain about. We at *D&T* just couldn’t believe that, so we started a department called “D&T Forum” to give those silent types a chance to change their reputation. We even agreed to withhold the names of contributors to prevent undue embarrassment.

So far, only two brave souls have ventured forth. Could it be that the rumor was true?

There’s still time to renew our faith in you. Send humor—cartoons, jokes, parodies, etc.—opinions, or beefs to *D&T* Forum, IEEE Design & Test, 10662 Los Vaqueros Cir., Los Alamitos, CA 90720.