DOE/ER/25052--T1

DE93 002096

ULTRACOMPUTER RESEARCH PROJECT

Progress Report

for Period January 1, 1992 - December 31, 1992

and Continuation Request

Allan Gottlieb, Principal Investigator

New York University
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, N.Y. 10012

October 1992

Prepared for

THE U.S. DEPARTMENT OF ENERGY
GRANT DE-FG02-88ER25052

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Overview

As we pass the midpoint of our three year grant, the NYU Ultracomputer project is pleased to report both significant technical progress and greater industrial acceptance of our principal contributions: the fetch-and-add processor coordination primitive and hardware combining of all memory references. During this past year we have achieved several very significant milestones. The custom VLSI switches designed under the DOE project and fabricated by MOSIS have been integrated into a fully functional 4-processor Ultra III system (the engineering and construction of Ultra III is itself an NSF project). Our Symunix highly-parallel operating system manages the resources of Ultra III and simple application programs have been run successfully.

Both the Thinking Machines CM-5 and Cray C90 include fetch-and-add. The CM-5, in its "control network" supports a synchronous form of combining and several synchronous parallel prefix operations bearing a strong resemblance to our work of the early 80's (Jacob Schwartz, "Ultracomputers", *ACM TOPLAS* **2**, #4, October 1980, pp. 484-521 and Allan Gottlieb and Clyde Kruskal, "Supersaturated Ultracomputer Algorithms", Ultracomputer Note #11, 1980). According to rumors, combining was actively considered for the C90 but the 16-processor architectural limit of the C90 was determined to be just below the threshold where combining would be needed. We have also heard from a Siemens employee working on an ESPRIT project investigating a machine design with fetch-and-add and combining.

Our development work on the Gnu C compiler (GCC version 2), work that began as a small project to obtain a high-quality compiler for the AMD 29000 in Ultra III but has grown to support other machines, continues to attract substantial industrial attention. Most recently, we have furnished DEC with a high quality port for their new Alpha microprocessor. DEC intends to donate an Alpha-based workstation to our laboratory to support further work in this area.

We are moving our VLSI fabrication target from MOSIS to a more advanced facility at NCR, a major computer manufacturer with whom we have close ties. After some effort, two major cell in our combining chip have been transferred to the NCR system and passes their design rule checker. We expect to submit a design for NCR fabrication and checking by the end of 1992.

Shared memory designs are once again becoming of major interest. The introduction earlier this year of the Kendall Square KSR1 has already generated increased interest and upcoming machines from Cray, Convex, and Fujitsu will all support shared memory accesses. Not surprisingly, the issue of scalable access to this memory, to which we have made important contributions, is getting increased attention.

More detailed descriptions of our recent contributions and future plans are given in subsequent sections.

Industrial and Laboratory Contacts

During 1992, we have continued our relationships with AMD (Advanced Micro Devices) and MCC (Microelectronics and Computer Technology Corporation). As mentioned last year, due to the ongoing relationship between NCR and Allan Gottlieb, the principal investigator, NCR is contributing substantially to our VLSI efforts. This is proving to be a bidirectional technology transfer. NCR learns about our VLSI combining switch technology and we learn the differences between MOSIS and more traditional industrial practice for design rule verification. As discussed below, we are also involved with many other commercial organizations.

The principal investigator continues to be a charter member of two NCR committees. The Parallel Processing Academic Advisory Council assists NCR in the development of their large-scale parallel processors; Gottlieb has worked specifically with the architecture and hardware development team. The Science Advisory Committee visits various NCR plants, reviewing their operation for upper management, and provides technical and other recommendations for improved operation.

As a result of these contacts, NCR is fabricating our combining switches using their 1.5-micron CMOS process with 208-pin packages. We have converted our layout to their design rules. Several cells have already passed the checker. We expect to submit the design for fabrication in a month or two.

AMD continues to support our work financially and technically. We received a grant from them for our GCC work in 1991 and they are still supplying all the microprocessors we require, most recently the 29050s. In addition, they have recently sent us another batch of in-circuit emulators. We have submitted a proposal to DARPA to have MCC construct a 256-processor Ultracomputer for which AMD will again provide the processors without charge.

Our GCC compiler development continues to attract considerable commercial attention. What started as a small-scale project to obtain a high-quality compiler for the AMD 29000 used in Ultra III, has grown to a larger effort targeting many machines and including significant machine-independent optimizations. We have had frequent technical contacts with compiler developers at Shell Oil, Data General, DEC, IBM, SRC, NeXT, Dell, NCD, and Cygnus as well as with numerous academic compiler developers world wide. To support our GCC effort, IBM has placed an RS/6000 workstation in our laboratory and Data General has sent an AViiON multiprocessor containing two Motorola MC88000s. (GCC is *the* compiler shipped with AViiON and NeXT workstations; NeXT is considering supplying a workstation for our use). Hewlett-Packard has given us a model 9000/720 "Snake" workstation, an X-terminal, and a high-speed laser printer to support our compiler and VLSI work. Finally, we are working actively with DEC on GCC for the Alpha and expect to receive a workstaton fairly soon.

LLNL has provided us with time on their BBN TC2000 multiprocessors so that we could evaluate the performance of parallel algorithms.

We have continued contact with Astronautics concerning the use of Ultra-like networks in packet routers (Asynchronous Transfer Modules or ATMs) and with Bell Laboratories on the same issue and on VLSI design.

VLSI Research

During the present reporting period, we have attained our goal of constructing a VLSI network that combines requests directed at the same memory location. Using MOSIS 132-pin packages, a 2×2 switch node is composed of four each of two types of chips: *forward path* and *return path* components. Both of these chip types have been successfully fabricated by MOSIS and are being used in the final checkout of a 4-processor Ultracomputer, which contains all the board types used in a 16-processor prototype being completed under NSF support.

To ensure that both components function correctly in the system, we adopted a sophisticated simulation methodology. Simulations were done at three levels. At the highest level, we wrote a behavioral simulation specifying how each switch component should perform. Next was a structural level simulator containing a register transfer language description in C for each cell in the design. At the lowest level we used a switch-level simulation of the circuit extracted from the VLSI layout.

We simulated a 16-PE/16-MM system with a 16×16 combining network (the size of the full Ultra III prototype). This simulation ran both the behavioral and structural models of the switch simultaneously and compared their results. A random stimulus was given to the system to verify expected operation. We simulated only a single component at the switch-level since running the switch-level simulation on a full network would be prohibitively expensive.

To assist in the layout verification process, gate-level schematics of each cell were done using DASH, a commercial schematic-capture system. A series of small programs were written that allowed extraction of transistor connections from both the schematic and layout in identical format. The resulting text files were then compared to validate each layout; a correct layout would produce identical files.

Finally, we modified the Magic technology files and CIF generation code to allow the construction of geometry that meets the design rules of NCR's 1.5μ CMOS process. We are now working with NCR, who plans to fabricate chips for us later this year.

Our major effort for our final year involves making extensive use of the simulation technology developed earlier to investigate the alternative combining structures presented in our original proposal, as well as a new, more complex structure that can process 4 input requests simultaneously. Due to the validation procedure we have developed, it will be easy to generate a correct layout for the basic cells of any promising scheme in order to estimate area and performance.

Operating Systems

Our work in this area continues to focus on the development of Symunix, an operating system designed primarily for machines like the Ultracomputer (i.e., MIMD, shared memory, hundreds or thousands of processors, Fetch-and-Add, and hardware combining of memory references). The goal of this effort is to provide efficient execution of application programs. We use two techniques to accomplish this goal. First, we avoid, whenever possible, serial bottlenecks in the operating system and language runtime system. Second, we provide the necessary hooks in the operating system that permit a user-mode runtime system to perform tasks traditionally done by expensive traps into the kernel. Portability to architectures significantly different from ours is secondary.

During the past year we improved our stable version of Symunix and ported it to our 4-processor Ultra III prototype (targeted for 16 processors with NSF funding). This effort included work on program development tools (loader, debugger, etc.), and helping to refine the programmable logic on the processor board to better support the OS.

Progress has been made on new process and memory management systems for Symunix, with the goal of significantly improving performance, flexibility, and portability, but this work is not yet complete. At a lower level, a new parallel buddy system memory allocator, causing much less fragmentation, has been devised and tested in the Symunix kernel.

We have two major goals for the next year.

(1) To finish replacing the process and memory management components of Symunix, which will validate the basic design, and allow us to begin to evaluate the effectiveness of techniques such as asynchronous system calls, asynchronous page faults, and non-preemptive scheduling policies for supporting user mode threads.

(2) To begin measuring the behavior of highly parallel algorithms and data structures on Ultra III and with the simulator described later in this proposal. For the first time one will be able to evaluate real code running on real hardware with and without combining.

The GNU C Compiler

Our efforts on the GNU C compiler over the past year continue to draw the attention of a number of companies, several of whom have provided equipment for our use in this effort. Due to this interest, our work has expanded from providing the basic support for GCC that is required for our Ultra III prototype to producing high-quality compilers for the IBM RS6000, the DEC Alpha, and the IBM Romp used in an older IBM workstation as well as for the 29000 (our original target). We continue to be the principal developers and maintainers of the back end of GCC (i.e. the code generator and optimizer).

During the last year, we produced a GCC compiler for the DEC Alpha microprocessor. We understand this compiler is used at a number of DEC facilties and has been distributed to Cray, who is using the Alpha processors in a parallel system they are developing. DEC plans to provide us with an Alpha system in exchange for this work. Most recently, we have been asked by a group of consultants to assist in their port of GCC to the WE1610 DSP chip. This is both the first port of GCC to a DSP chip and the first C compiler for that chip. We expect to receive a grant from this group as a recognition of our help in their effort.

We continue to work with Cygnus Support, a commercial organization founded to support GNU software, in improving an instruction scheduler that tries to remove data dependencies between instructions that cause pipeline stalls. This scheduler separates memory loads from uses of the register loaded, which increases the latency tolerance of the system. Tolerating memory latency is particularly important for shared-memory multiprocessors. Over the past year, this scheduler has been significantly improved for the superscalar processors (such as the Motorola 88110 and the Alpha) that have become common. Much of this work was done by Data General, with our assistance.

Our plan for next year is similar: we will continue improving the optimization abilities of GCC in addition to supporting the compiler for both local and external users. For use with our prototypes, we plan to include support for a *shared* keyword and to allow selected optimizations on *volatile* variables. It is likely that coordination variables will be declared volatile.

GNU Fortran (g77) is now in alpha test. This uses the same backend as GCC and therefore will generate code for all the processors supported by GCC. This FORTRAN compiler will be a freely available, high quality FORTRAN compiler that we will provide to users of our prototypes.

The high quality of GCC has attracted the attention of the Ada language community. The NYU Ada group has recently received funding to produce an Ada 9X compiler using GCC. We have assisted this effort in the past and they are now supporting 1/2 of Richard Kenner, who is doing our GCC work.

Simulation

Since the project's inception, we have continually built and used simulators to study multiprocessor performance. Simulations driven by synthetic reference streams have been used extensively by our group and by others for studying the performance of multiprocessor networks. In contrast to these simulators, we also built an address trace driven simulator to study the performance of multiprocessor TLBs.

We are currently building a flexible, complete multiprocessor simulation environment that will allow us to perform either execution- or trace-driven simulations. The network and memory simulators are complete as is a very primitive "processor" simulator. The network simulator operates as a behavioral simulator of each of the individual switches composing the network. Unlike the simulators described in the VLSI section, this one is very highly parameterizable, capable of supporting networks different from any we are considering building and is well integrated into the simulation environment. The processor simulator presently under construction is much more specific, targeting the AMD 29000 family and several of the features present on the Ultra III processor board.

When complete, the simulation environment will allow us to simulate a multiprocessor in detail or to simulate specific subsystems in detail, while others merely are modeled. For example, the network can be bypassed, effectively providing an idealized, single-cycle memory. The processor simulator will support both kernel- and user-mode operation.

Such a simulation environment will allow us to study the performance of components and subsystems of multiprocessors, in addition to continuing our study of network and TLB performance. In particular, we would like to study the performance of different size Ultracomputers and the effect of combining, to characterize parallel programs, to compare different parallel algorithms and different methods of ensuring cache consistency, and investigate the feasibility of demand paging.

Coordination Algorithms

We continue our longstanding study of coordination algorithms, emphasizing bottleneck-free solutions. During the past year, we have built upon our experience in process coordination by improving well-known algorithms and by exploring several new paradigms for synchronization. This work has touched several major classes of coordination algorithms including synchronization, queueing, and memory allocation. These new algorithms have attracted the interest of others working in parallel processing and have been incorporated into a networking and streams package from Spider Software (Ian Heavens, "Experience in Fine Grain Parallelisation of Streams-Based Communications Drivers", *OpenForum '92*, Nov. 1992).

Much of our recent work has been on incremental improvements to well understood algorithms. In our continuing efforts to "minimize the constants" that are often swept under the asymptotic rug, we have developed variations of known algorithms for counting semaphores and readers/writers locks that require the theoretical minimum of a single shared access in the absence of contention. Our highly-parallel, buddy-system memory allocator (James Wilson, "Operating System Data Structures for Shared-Memory MIMD machines with Fetch-and-Add", NYU Doctoral Dissertation, 1988) has undergone an incremental improvement during this past year resulting in reduced fragmentation of memory. A less synchronous, and hence more efficient version, is planned for next year.

We are continuing the program begun in (Eric Freudenthal and Allan Gottlieb, "Process Coordination with Fetch-and-Increment", *Proc. ASPLOS IV*, 1991, pp. 260-268) to examine the algorithmic implications of architectural alternatives to fetch-and-add. This year, we have observed that many coordination algorithms utilizing fetch-and-add do not require the exact numeric result of the operation, but only an indication if the value is zero or negative. This observation enables these algorithms to be implemented on processors like the Intel 80386 that return only the sign of an atomic add. In particular, the algorithms can be implemented on early Sequent Symmetries (later Symmetries use the Intel 80486, which supports full fetch-and-add semantics).

The Ultracomputer's combining network is expected to minimize the effects of hot-spot contention (such as the counter variable of a barrier). However, significant network traffic may be caused by unrelated processors spin-waiting on distinct variables. The recent work of Mellor-Crummey and Scott ("Synchronization Without Contention", *Proc. ASPLOS IV*, April 1991, pp. 269-278) has demonstrated the effectiveness of local shared memory that can be accessed by the one co-located processor without generating network traffic.

We have observed that substantial amounts of such memory are not required. Indeed, a single bit of shared memory local to each processor is sufficient. Processors spin-wait on their local bit; when another processor sets it, the local processor checks ordinary shared memory for more information. These single bit "triggers" may be significantly simpler to implement than full-size local, shared memory with consistent caches. In particular, the *reflect* operation on Ultra III supports such triggers.

LLNL is supporting this research by providing access to their 128 processor BBN TC-2000 on which we have implemented a trigger library and several basic coordination algorithms. We will port the trigger library to Ultra III next year.

In order to gain access to some resource, busy-waiting synchronization algorithms traditionally utilize either a reservation system (e.g. Lamport's bakery algorithm) or repeatedly attempt to sieze the resource. Reservation-based algorithms achieve fairness at the cost of issuing more shared accesses. Furthermore, many reservation systems serialize access, which is inappropriate for readers/writers synchronization and for counting semaphores.

Recently, we have discovered a family of fair coordination algorithms with costs comparable to the fastest known (starvation-susceptible) algorithms. These new algorithms follow a two-phase paradigm: All requests first check for contention (usually requring only a single shared fetch-and-add access). If no contention is detected (a frequent occurrence in practice), the lock is granted immediately. Otherwise, arbitration is managed by a starvation-free second phase.

These two-phase algorithms require only fetch-and-increment and fetch-and-decrement but often the number of shared accesses is halved if full fetch-and-add is available. We anticipate using Ultra III over the coming year to measure and further refine these algorithms.

Fetch-and-add based algorithms for bottleneck-free queues and multiqueues (queues in which a single "multi-item" could be inserted representing multiple instances of an item) were among the original contributions of our group. We had two classes of multiqueues one with better worst-case performance (logarithmic-time) and one with better "normal"-case performance (constant-time). New multiqueues have been descovered that combine the best qualities of their forebears: constant normal-case complexity and graceful degradaton to logarithmic worst-case complexity in degenerate cases. An implementation on the LLNL TC-2000 is underway; measurements on Ultra III will be undertaken next year.

Recent Publications

(1) George Almasi and Allan Gottlieb, *Highly Parallel Computing*, second edition, Benjamin-Cummings, approx. 600 pages, to appear in 1993.

(2) Guoying Chen, "A Wiring Tool for Ultracomputer", Ultracomputer Note #176, January, 1992.

(3) Guoying Chen, "Primary-node Scheme for Cache Coherences in Large Scale Shared-Memory Multiprocessors", Ultracomputer Note #181, January, 1992.

(4) Susan Dickey and Richard Kenner, "Using Qualified Clocks in the NORA Clocking Methodology to Implement a Systolic Queue Design", *Proc. of the Brown/MIT Conference on Advanced Research in VLSI*, March 1992.

(5) Susan Dickey and Richard Kenner, "Hardware Combining and Scalability", *Proc. of 4th Annual ACM Symposium on Parallel Algorithms and Architectures*, San Diego, June 29 - July 1, 1992, pp. 296-305.

(6) Susan Dickey and Richard Kenner, "Combining Switches for the NYU Ultracomputer", to appear in *Proc. of the Fourth Symposium on the Frontiers of Massively Parallel Computation*, McLean, Virginia, Oct 19-21, 1992.

(7) Susan Dickey and Ora E. Percus, "Performance Difference Among Combining Switch Architectures", *Proc. of 21st International Conference on Parallel Processing,* August, 1992.

(8) Susan Dickey and Ora E. Percus, "Performance Analysis of Clock-Regulated Queues with Output Multiplexing in Three Different Types of 2 by 2 Crossbar Switch Architectures" *Journal of Parallel and Distributed Computing,* **16**, 1992, pp. 27-40.

(9) Eric Freudenthal, "Evaluation of Interprocessor Triggers as a Coordination Primitive for Shared-Memory MIMD computers", in Brooks et al. (Eds) *The 1992 MPCI Annual Report: Harnessing the Killer Micros,* UCRL-ID-107022092, Lawrence Livermore National Laboratory, pp. 241-242.

(10) Allan Gottlieb, "Architectures for Parallel Supercomputing", *Proc. PACTA '92 (Parallel Computing and Transputer Applications),* Barcelona, Spain, September, 1992, pp. 39-47.

(11) Allan Gottlieb, "Puzzle Corner", in *Technology Review,* MIT Press, 1966-present.

(12) Torbjörn Granlund and Richard Kenner, "Eliminating Branches using a Superoptimizer and the GNU C Compiler", *Proc. SIGPLAN '92 Conference on Programming Language Design and Implementation* San Francisco, June 17-19, 1992, pp. 341-352.

(13) Ora E. Percus, "A Multistage Clocked Queueing Network", Ultracomputer Note #183, January, 1992.

CURRICULUM VITAE
ALLAN GOTTLIEB

EDUCATION

BS Mathematics (elected to ΣX)	M.I.T	June, 1967
MA Mathematics	Brandeis Univ.	June, 1968
PhD Mathematics	Brandeis Univ.	January, 1973

ACADEMIC EXPERIENCE

1990-	Professor of Computer Science	Courant Institute of Mathematical Sciences, NYU
1989-	Director	Ultracomputer Research Laboratory Courant Institute
1986-1989	Associate Director	Ultracomputer Research Laboratory
1985-90	Associate Professor of Computer Science	Courant Institute
1981-85	Associate Research Professor	Courant Institute
1979-81	Visiting Member, on leave from York College (City Univ. of NY)	Courant Institute
1979-81	Associate Professor of Mathematics	York College
1976-81	Coordinator of Computer Mathematics	York College
1973-79	Assistant Professor of Mathematics	York College
1972-73	Instructor of Mathematics	State College of MA (No. Adams)
1971-72	Acting Instructor of Mathematics	Univ. of CA (Santa Cruz)

PROFESSIONAL SOCIETIES

American Mathematical Society
Association for Computing Machinery
IEEE Computer Society
N.Y. Academy of Sciences

SCIENTIFIC ACTIVITIES

Science Advisory Committee, NCR, 1987-present

Parallel Processing Advisory Council, NCR, 1989-present

Program Committee chairman for the *18th Annual Symposium on Computer Architecture*, 1992.

Editor, Hardware and Software Systems, *Journal of Parallel and Distributed Computing*, Academic Press, beginning May 1992 (Distributed Operating Systems Editor 1986-1992).

Editorial Board member of the *Cambridge International Series on Parallel Computation* 1988-present.

PUBLICATIONS

Highly Parallel Computing, Benjamin-Cummings, 1989, 509 pages (with George Almasi). Second edition to appear 1993.

"Puzzle Corner", in *Technology Review*, MIT Press, 1966-present.

"Architectures for Parallel Supercomputing", *Proc. PACTA '92 (Parallel Computing and Transputer Applications)*, Barcelona, Spain, September, 1992, pp. 39-47.

"Locating Multiprocessor TLBs at Memory", Ultracomputer Note, Courant Institute, NYU, 1991 (with Patricia J. Teller).

"TLB Performance in Multiprocessors", Ultracomputer Note #173, Courant Institute, NYU, 1991 (with Patricia J. Teller).

"Process Coordination with Fetch-and-Increment", *Proc. ASPLOS IV*, Santa Clara CA, April, 1991, pp. 260-268 (with Eric Freudenthal).

"An Outsider's View of Dataflow", in *Advanced Topics in Data-flow Computing*, Lubomir Bic and Jean-Luc Gaudiot, ed., Prentice Hall, 1991, pp 573-581.

"Scalability, Combining, and the NYU Ultracomputer", *Proc. 1990 Parallel Computing Workshop*, Ohio State University, March, 1990.

"An Overview of the NYU Ultracomputer Project", in *Experimental Parallel Computing Architectures*, Jack J. Dongarra, ed., North Holland, 1987, pp. 25-95.

"Designing VLSI Network Nodes to Reduce Memory Traffic in a Shared Memory Parallel Computer", *Circuits, Systems, and Signal Processing*, **6** 1987, pp. 217-238 (with Susan Dickey, Richard Kenner, and Yue-Sheng Liu).

"The NYU Ultracomputer–Designing an MIMD Shared Memory Parallel Computer", in *Computer Architecture* Daniel D. Gajski, V.M. Milutinovic, Howard J. Siegel, and B.P. Furht, ed., The Computer Society of the IEEE, 1987 pp. 471-485 (with Ralph Grishman, Clyde P. Kruskal, Kevin P. McAuliffe, Larry Rudolph, and Marc Snir).

"Using VLSI to Reduce Serialization and Memory Traffic in Shared Memory Parallel Computers", *Adv. Res. in VLSI: Proc. 4th MIT Conf.*, Ed. Charles Leiserson, MIT Press, 1986, pp. 299-316 (with Susan Dickey and Richard Kenner).

"Considerations for Massively Parallel Unix Systems on the NYU ULtracomputer and IBM RP3", *Proc. USENIX Assoc. Winter Conf.*, 1986 (with Jan Edler and Jim Lipkis).

"Adapting UNIX for Large Shared-Memory MIMD Systems", *Proc Meet/IX Conf.*, July 1985, pp. 4-20 (with Jan Edler and Jim Lipkis).

"A Brief Update on the NYU Ultracomputer", *IEEE Software*, page 71, July, 1985.

"Issues Related to MIMD, Shared-memory Computers: The NYU Ultracomputer Approach" *Proc. 11th Annual Symposium on Computer Architecture*, 1985 (with Jan Edler, Ralph Grishman, Clyde P. Kruskal, Kevin McAuliffe, Larry Rudolph, Marc Snir, and James Wilson).

"Operating System Considerations for Large-Scale MIMD Machines", *Computers in Mech. Eng.*; also in *Proc. 1985 ASME International Computers in Engineering Conference* (with Jan Edler and Jim Lipkis).

''A Remark on 'A Variable Length Shift Register''' Ultracomputer Note #77, Courant Institute, NYU, 1985.

''The NYU Ultracomputer: An MIMD Shared Memory Parallel Computer'', in *Supercomputers: Design and Applications*, Kai Hwang and Bob Kuhn ed., Computer Society Press, 1984 (with Ralph Grishman, Clyde P. Kruskal, Kevin McAuliffe, Larry Rudolph, and Marc Snir).

''Simulating Shared-Memory Parallel Computers'', *Proc. Fifteenth Annual Pittsburgh Modeling and Simulation Conference*, Apr. 1984, pp. 1417-1424 (with Seth Abraham and Clyde P. Kruskal).

''Complexity Results for Permuting Data on Parallel Processors'', *JACM* 31 #2 (April 1984), pp. 193-209 (with Clyde P. Kruskal).

''Avoiding Serial Bottlenecks in Ultraparallel MIMD Computers'', *Proc. Compcon*, San Francisco CA, Feb. 1984, pp. 354-359.

''Comparing the NYU Ultracomputer with Other Large-Scale Parallel Processors'', *Proc. Conf. on Frontiers of Supercomputing*, Los Alamos NM, Aug. 1983.

''Basic Techniques for the Efficient Coordination of Very Large Numbers of Cooperating Sequential Processors'', *ACM TOPLAS* 5 #2 (April 1983), pp. 164-189 (with B.D. Lubachevsky and Larry Rudolph); also in *Proc. Intern. Conf. on Par. Proc.*, 1981 (winner of most original paper award).

''The NYU Ultracomputer – Designing an MIMD Shared Memory Parallel Computer'', *IEEE Trans.* C-32 (Feb. 1983), pp. 175-189; also in *Proc. 8th Annual Symp. on Computer Architecture*, April 1982 (with Ralph Grishman, Clyde P. Kruskal, Kevin McAuliffe, Larry Rudolph, and Marc Snir).

''The NYU Ultracomputer – A Paradigm for Future Large-Scale Parallel Processors'', *NCC 82*, June 1982 (omitted from printed proceedings due to NCC error) (with Ralph Grishman, Clyde P. Kruskal, Kevin McAuliffe, Larry Rudolph, and Marc Snir).

''Networks and Algorithms for Very Large Scale Parallel Computation'', *Computer* 15 (January 1982), pp. 27-36 (with J.T. Schwartz).

''The NYU Ultracomputer – A General Purpose Parallel Processor'', *Proc SPIE*, 298 *Signal Processing IV* Aug. 1981 (with Ralph Grishman, Clyde P. Kruskal, Kevin McAuliffe, Larry Rudolph, and Marc Snir).

''Coordinating Large Numbers of Processors'', *Proc. 1981 Intern. Conf. on Parallel Processing*, Bellaire MI (with B.D. Lubachevsky and Larry Rudolph). Designated ''Most Original Paper''.

''Coordinating Parallel Processors: A Partial Unification'', *Computer Architecture News* 9 (October 81), pp. 16-24 (with Clyde P. Kruskal).

''A Historical Guide to the Ultracomputer Literature'', Ultracomputer Note #36, Courant Institute, NYU, 1981.

''Washcloth 81'', Ultracomputer Note #21, Courant Institute, NYU, 1981.

''Comments on 'Concurrent and Insertion in AVL Trees''', *IEEE Trans* C-30 (October 81), p. 812.

''MULT - A Multitasking Ultracomputer Language With Timing'', Ultracomputer Note #15, Courant Institute, NYU, 1980 (with Clyde P. Kruskal).

''PLUS - A PL/I Based Ultracomputer Simulator, II'', Ultracomputer Note #14, Courant Institute, NYU, 1980.

"MOP - A (Minimal) Multiprocessor Operating System Extending WASHCLOTH", Ultracomputer Note #13, Courant Institute, NYU, 1980.

"WASHCLOTH - The Logical Successor to SOAPSUDS", Ultracomputer Note #12, Courant Institute, NYU, 1980.

"Supersaturated Ultracomputer Algorithms", Ultracomputer Note #11, Courant Institute, NYU, 1980 (with Clyde P. Kruskal).

"PLUS - A PL/I Based Ultracomputer Simulator, I", Ultracomputer Note #10, Courant Institute, NYU, 1980.

"Another Remark on the Planarity of the Shuffle-Exchange Network of sizes 16 and 32", Ultracomputer Note #9, Courant Institute, NYU, 1980.

"A Data Motion Algorithm", Ultracomputer Note #7, Courant Institute, NYU, 1980.

"A Note on Sorting Integers from a Bounded Range", *ACM SIGACT News* 12 #3 (1980) pp. 66-67 (with Clyde P. Kruskal).

"Time Handicapping Will Lower Game Quality", *Intern. Comp. Chess Assoc. Newsletter* 2 #1 (Feb. 1979) pp. 4-6.

"Traveling Salesman Problems, Generating Functions and Markov Chains", *Proc. ORSA Regional Meeting*, April 1978 (abstract) (with Samuel Kohn and Meryle Kohn).

Introduction to Scientific Programming, York College Press, 1978, 221 pp.

"A Generating Function Approach to the Traveling Salesman Problem", *Proc. ACM 77*, pp. 294-300 (with Samuel Kohn and Meryle Kohn).

"A Flowcharting Proposal", *SIGPLAN Notices* 11 #12 (1976) pp. 35-37.

"A Computer Generated Bibliography", *ACM SIGSOC Bulletin* 7 (1976) pp 19-23 (with Joan Bodoff).

"Integration is Differentiable", *Proc. AMS* 53 (1975), pp. 167-171.

"Converses to the Ω-stability and Invariant Lamination Theorems", *Trans. AMS* 202 (1975), pp. 269-283.

"Absolutely and Differentiably L-stable Diffeomorphisms", *University Microfilms*, 1973, 92 pp.

"Lectures on Groups of Homotopy Spheres", (lectures by Jerome Levine), Tech. Rept., Brandeis Univ., 1970 (with Clint McCrory).

INVITED ADDRESSES

''Hardware Combining and the Ultra III Prototype'', Third Workshop on Scalable Shared Memory Multiprocessors, San Diego, May, 1993.

''Architectures for Parallel Supercomputing'', PACTA '92 (Parallel Computing and Transputer Applications), Barcelona, Spain, September, 1992.

''The Relevance of University Research in Computer Architecture'', Intern. Symp. on Comp. Arch., May, 1991.

''Processor to Memory Interconnection Networks'', Distinguished Lecture Series, Univ. of Wisc., October, 1990.

''Parallel Computer Architectures (Panel)'', Intern. Conf. on Par. Proc., August, 1990.

''Scalability, Combining, and the NYU Ultracomputer'', *Parallel Computing Workshop*, Ohio State Univ., March 1990.

''Interconnection Networks for MIMD Computers'', Amherst College, January 1990.

''Simulating and Packaging Shared-Memory Interconnection Networks'' Cornell Theory Center Lecture, November 1989.

''The Symunix II Operating System and Packaging Ultracomputers'' Cray Research, Mendota Heights MN, October 1989.

''Network Research at NYU'' Univ. Ill., June 1989.

''The NYU Ultracomputer Architecture Project'', Buscon 88 East, New York NY, October, 1988.

''An Introduction to the NYU Ultracomputer Architecture and Systems Software'', Third International Conference on Supercomputing, Boston MA, May, 1988.

''The NYU Ultracomputer: A Shared Memory MIMD Parallel Computer'', New York Academy of Sciences, December, 1985.

''The Ultracomputer – An Experimental Shared Memory MIMD Computer'', SIAM Conference on Parallel Processing for Scientific Computiation, Norfolk, VA, November, 1985.

''The NYU Ultracomputer'', Concurrent Processing Symposium, Washington DC, July 1985.

''The NYU Ultracomputer'', International Colloquium on High-Performance Computer Architectures for Large-Scale Numerical Computations, Cologne, October, 1984.

''The Ultracomputer Operating System'', Operating Systems and Environments for Parallel Computers, Los Alamos National Laboratory, Aug. 1984.

''A Taxonomy of Parallel Computers'', AAAS Annual Meeting, New York, May 84.

''Avoiding Serial Bottlenecks in Ultraparallel MIMD Computers'', Compcon, San Francisco, Feb. 84.

''The NYU Ultracomputer - The Ultimate Commercial Machine'', (title created by conference organizer) Data Processing Management Association (DPMA) Conference, Washington DC, Oct. 83.

''A Status Report on the NYU Ultracomputer'', IEEE Electronics and Aerospace Conference (EASCON), Washington DC, Sept. 83.

''Comparing the NYU Ultracomputer with Other Large Scale Parallel Processors'', Frontiers of Supercomputing, Los Alamos National Laboratory, Aug. 83.

(Co-organizer and session chairman) ''ICASE Workshop on Array Architectures for Computing in the 80's and 90's'', NASA Langly Research Center, Hampton VA, April 80.

''Ultracomputer Simulation and Supersatured Algorithms'', Nov. 79, NASA Langly Research Center, Hampton VA.

''Solving Combinatorial Problems with Generating Functions'', Bell Labs, Holmdel NJ, Jan. 78 (with S. Kohn & M. Kohn).

''Artificial Intelligence - Can Computers Think?'', NSF Summer Science Program, Bard College, Annandale-on-Hudson, July 77.

''Operations Research'', Bard College, Annandale-on-Hudson NY, May 78.

''Computer Chess and the North American Championship'', Bard College, Annandale-on-Hudson NY, Dec. 76.

Budget removed. do

END

DATE FILMED

1 / 13 / 93