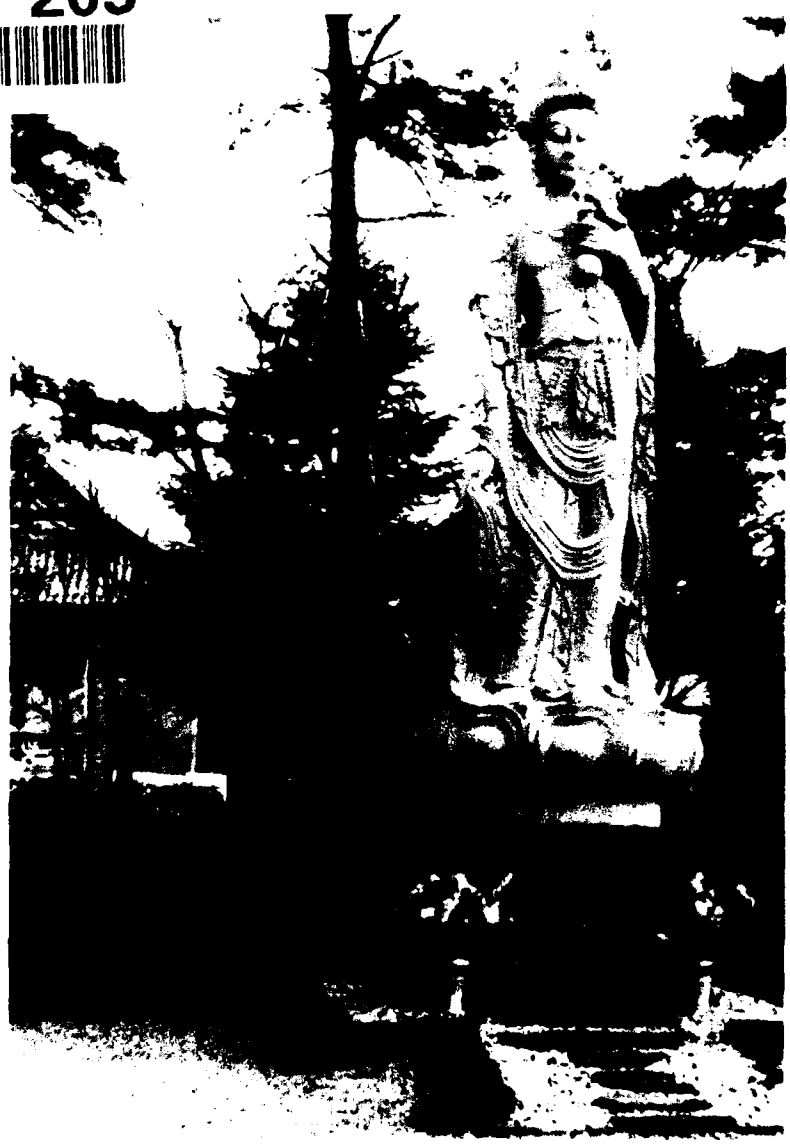
OCTOBER TO DECEMBER 1989 VOL. 14 NO. 4

SCIENTIFIC INFORMATION BULLETIN

DEPARTMENT OF THE NAVY OFFICE OF NAVAL RESEARCH FAR EAST
DEPARTMENT OF THE AIR FORCE OFFICE OF SCIENTIFIC RESEARCH FAR EAST
UNITED STATES ARMY RESEARCH OFFICE FAR EAST



NAVSO P-3580

SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
ONRFE Vol 14, No. 4	
6a. NAME OF PERFORMING ORGANIZATION: ONR/AFOSR/ARO; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code): Liaison Office, Far East, APO San Francisco 96503-0007	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)

ONR FAR EAST SCIENTIFIC INFORMATION BULLETIN

12. PERSONAL AUTHOR(S)

Arthur F. Findeis, Director; Sandy Kawano, Editor

13a. TYPE OF REPORT	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
		October-December 1989	

16. SUPPLEMENTARY NOTATION

ISSN: 0271-7077

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

Aerospace computing, Programming languages, PS-algol, Distributed PS-algol, Persistent databases, Language interfaces, Addressing mechanisms, Database interfaces, Supercomputer CPU

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This is a quarterly publication presenting articles covering recent developments in Far Eastern (particularly Japanese) scientific research. It is hoped that these reports (which do not constitute part of the scientific literature) will prove to be of value to scientists by providing items of interest well in advance of the usual scientific publications. The articles are written primarily by members of the staff of ONR Far East, the Air Force Office of Scientific Research, and the Army Research Office, with certain reports also being contributed by visiting stateside scientists. Occasionally, a regional scientist will be invited to submit an article covering his own work, considered to be of special interest. This publication is approved for official dissemination of technical and scientific information of interest to the Defense research community and the scientific community at large. It is available free of charge to approved members of the DOD scientific community. Send written request describing DOD affiliation to: Director, Office of Naval Research, Liaison Office Far East, APO San Francisco 96503-0007.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS		
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED

18. SUBJECT TERMS (continued)

Japan
Parallel inference machines
Parallel inference machine operating systems (PIMOS)
High sensitivity gas analysis
Mu-X
Natural language
Kappa
Parallel algorithm
Kernel languages
PIM/p prototype
Gigalips project
Supercomputers
University supercomputer system
University of Kyoto Data Processing Center
Ion implantation
Pohang Iron and Steel Company
Australia
Napier
POMP
Vector CPU
Peak floating point power
Paths-to-memory
Multiple pipelined architecture
Parallel computing
Cleanroom
Contamination control for IC production
Stainless steel passivation
Chemical vapor deposition
Metallization
Fifth generation computers
Sequential inference machines
Multi-PSI
Database machines
Guarded Horn Clause
Electronic Dictionary project
A'UM
DNA sequencing
PSI machine
Constraint logic programming
Load balancing
Computational fluid dynamics
University of Tokyo Computing Centre
MINOO
Korea
Persistent object systems
E
χ language
MONADS
Vector instruction set
Main memory
Vector computation
Vector start-up time
Fujii and Yoshihara benchmark
Semiconductor fabrication
Ultra clean gas processing technology
Epitaxial deposition
Thin film growth
Etching

CONTENTS

Cover: Statue of Kanon at Zojoji Temple. Kanon is the goddess of love and mercy and of humanitarian concern. The statue is a memorial to the several hundred guests who perished in a fire that destroyed the Hotel New Japan, in Akasakamitsuke, Tokyo, about 10 years ago. The hotel has not been rebuilt and compensation to the bereaved families remains pending. Courtesy of Earl Callen.

THE 1988 INTERNATIONAL CONFERENCE ON FIFTH GENERATION COMPUTER SYSTEMS

Several U.S. computer scientists attended the 1988 International Conference on Fifth Generation Computer Systems (FGCS'88). With two predecessors in 1981 and 1984, FGCS'88 was the third international conference on fifth generation computer systems. Its purpose was to report the results of the intermediate stage of the Japanese Fifth Generation project and to encourage the presentation of research papers by other researchers in related fields. The main emphasis was on logic programming, concurrency and parallelism, and artificial intelligence. The conference took place at the Tokyo Prince Hotel, from 27 November to 2 December 1988, with 1,300 participants from Japan and 300 from the rest of the world. About 355 papers were submitted with 95 selected; 39 percent of the selected papers came from Japan and 23 percent from the United States. Technical sessions were divided into four themes: theory, software, architecture, and applications. The five articles that follow give the various scientists' impressions of the conference, their comments concerning the presentations in their area of technical expertise, and highlights from site visits.

William J. Dally
Massachusetts Institute of Technology
Laboratory for Computer Science

OVERVIEW

This report describes my observations while attending the Fifth Generation Computer Systems (FGCS) Conference in Tokyo, Japan, and visiting a number of Japanese laboratories active in the field of concurrent computer architecture.

At FGCS I attended primarily the architecture sessions. The first 2 days of the conference were devoted to the FGCS project at the Institute for New Generation Computer Technology (ICOT). In the area of architecture, these sessions described work on sequential inference machines, parallel inference machines (PIM), and database machines. I found the ICOT approach to architecture to be characterized by solid engineering applied to existing ideas. The remaining 3 days of the conference were contributed papers. A substantial number of the contributed papers were also from ICOT. Most of the remaining papers were of lower quality. The architecture portion of FGCS is described in more detail below.

After the conference I gave a lecture on fine-grain concurrent computing sponsored by the Massachusetts Institute of Technology (MIT) industrial liaison program and visited the following research laboratories:

- ICOT--The research in this lab is described in the FGCS section.
- Sony Computer Science Research Laboratory--This lab, under the direction of Prof. Mario Tokoro of Keio University, is conducting research on concurrent object-oriented programming systems.

- Electrotechnical Laboratory (ETL)--At ETL I visited the laboratories of Drs. Shimada and Yamaguchi, who are developing experimental dataflow machines for numerical and symbolic applications, respectively.

- University of Tokyo--I visited Profs. Tanaka and Goto. Prof. Tanaka is building an experimental parallel inference engine (PIE). Prof. Goto is involved in the development of Josephson junction based computer technology.

- Hitachi Central Research Laboratory--I visited Nagashima's laboratory where the Hitachi S-810 and S-820 supercomputers were designed.

- NEC C&C Research Laboratories--I visited with Tadashi Watanabe to discuss the NEC SX-2 supercomputer and with numerous researchers at the C&C labs.

Overall, the Japanese have built some impressive experimental parallel computing systems. In the academic, industrial, and government research labs machines are being built to test new ideas in parallel computing and to serve as test beds for parallel software development. These machines take advantage of the latest semicustom VLSI technology and are built to industrial standards. However, with the exception of the group at ETL, they have come up with few innovative ideas to solve the outstanding problems in parallel computing such as load balancing, resource management, fast context switching, fast communication, program decomposition, and debugging. They tend to take ideas conceived elsewhere and implement them. The work at ICOT is highly specialized to logic programming and with a few exceptions does not seem applicable to other models of parallel computation.

One fact I found striking was the absence of full-custom VLSI components in any of the experimental machines I examined. The Japanese researchers have access to the latest gate-array technology. The PIM, for example, uses an 80k-gate gate array. I got the impression that much of the component and tooling costs were denoted by the Japanese companies. In contrast, most American efforts in experimental parallel computing have grown out of VLSI research groups and have a substantial custom VLSI component. This difference makes the Japanese more productive in building their machines. Far less effort is required to produce a semicustom chip than to produce a full-custom chip. However, it also limits the type of machines they can build and leaves them with an artificial measure of machine cost.

FGCS

During the first 2 days of the conference ongoing and planned ICOT research projects were described. ICOT's research is aimed at developing parallel computing systems for symbolic, *knowledge-based* tasks. Their work is strongly influenced by their choice of a logic programming language, FGHC. This artificially imposed constraint has led them to develop narrow, special-purpose solutions to many of the problems they have encountered. Their "power plane" load balancing strategy (see below) is an example.

Their work in the area of architecture has concentrated on building inference machines (for executing logic programs) and

database machines (for accessing knowledge bases). This work is characterized by a straightforward engineering approach. They have built a series of machines based on conventional ideas. They have made little attempt to define major problem areas limiting parallel computing and have made few real innovations in computer architecture.

Sequential Inference Machines

The first machines built by ICOT are sequential inference machines, the PSI and CHI. These machines are microcoded Prolog machines with an architecture similar to conventional LISP machines like the Symbolics 3600 or TI Explorer. They consist of a microcoded engine augmented by some special hardware to check and dispatch on tags. The machines are microcoded with a Prolog instruction set similar to the Warren Abstract Machine (WAM). The PSI-II and the CHI are both implemented in CMOS gate-array technology and give modest performance (300 and 500 KLIPS (logical inferences per second), respectively). They both have enormous primary memories, 64 MW and 320 MW (1W = 5 bytes), respectively.

Parallel Inference Machines

ICOT has built two versions of parallel inference machines, Multi-PSI V1 and V2, and is currently developing a third, the PIM. The Multi-PSIs consist of a number of PSIs connected by a message-passing network. The more recent machine, Multi-PSI V2. consists of 64 PSI-IIs connected by a two-dimensional grid network with 5-MB/s channels. Wormhole routing is employed. The network incorporates a load balancing mechanism that allows the mapping between a virtual two-dimensional grid (the "power plane") and the physical two-dimensional grid to be altered at run time. I was unable to get anyone at ICOT to give me figures for message startup overhead, receiving overhead, and context switch times. I was left with the impression that they were quite expensive, in the range of 50 to 500 μs. The Multi-PSI V2 is a large machine with eight PC boards and 16 MW (80 MB) of memory per PE. The entire machine contains 5 GB of memory. The major innovation in the Multi-PSI V2 is the load balancing mechanism in the network.

The PIM differs from the Multi-PSI in three respects:

1. To overcome the expensive message communication of the Multi-PSI, the PIM is constructed from clusters of eight PEs organized as bus-based shared memory multiprocessors using coherent (snooping) caches.

2. The PIM PE is more heavily integrated with one PE per board and uses a tagged RISC processor. The WAM is implemented using assembly code rather than microcode.

3. The interconnection network is a 20-MB/s hypercube.

The ultimate goal is to build a 1,024-processor (128-cluster) PIM. The machine is impressive because of its scale and the competent engineering that has gone into it.

PIMOS

The operating system for Multi-PSI and PIM is PIMOS. PIMOS is really more of a logic programming environment than it is an operating system. Its major feature is the

Shōen, a "fork" of sorts that prevents failure in a subgoal from terminating the parent. While PIMOS does apparently implement the memory, process, and I/O management tasks one expects of an operating system, there was little discussion of these components.

Database Machines

ICOT has built two database machines. Early in the project they built Delta, a relational database machine. More recently, they built Mu-X, a shared memory multiprocessor for handling knowledge-base queries. Mu-X consists of eight 68020 processors that share a common multiport, multipage memory. Several papers dealt with the memory organization. However, it is really just a convoluted way to increase the memory bandwidth of the shared memory for block transfers.

Other Papers

There were a few good architecture papers at the conference. One of the best was a paper by David Warren that described his data diffusion machine. This paper dealt with a protocol to support a tree-structured shared memory machine. Processors reside at the leaves of the tree. Cache coherence is maintained by having the directories in each node include the contents of the directories in each lower node.

SONY COMPUTER SCIENCE (CS) RESEARCH LABORATORY

The major focus of Sony's CS laboratory is object-based concurrent systems. They have developed a concurrent version of the Smalltalk-80 programming language, Concurrent Smalltalk. Another major project is the development of an object-based distributed operating system called MUSE. A few other projects dealt with Sony's new NEWS workstations. This group has successfully built some large experimental software systems but appears to have little expertise in hardware.

Sony's lab was remarkable for its working environment. While most Japanese labs (like ICOT) are high-tech sweatshops with many engineers packed into rows of desks or at best cubicles, Sony has individual offices for each researcher and a fairly comfortable open area where seminars are given. An emphasis is placed on individual accomplishment rather than the traditional Japanese team effort. They are making an effort to attract researchers from the United States and Europe to come work at their laboratory.

ELECTROTECHNICAL LABORATORY

Of the laboratories I visited, I was the most impressed with ETL. The two dataflow groups I visited at ETL combined an ability to develop creative solutions with a good experimental approach to computer architecture and a demonstrated ability to build real systems.

Shimada's group at ETL has built the Sigma-1 dataflow machine. During my visit I saw a demonstration of this machine and was able to examine the hardware. It is a pure tagged-token dataflow machine and in the present 128-processor configuration has a peak performance of 470 MFLOPS and achieves 170 MFLOPS on some real problems. The machine is programmed in a

dialect of "C." The software system assigns some resources statically at compile time and others at run time depending on load. Shimada's group is currently working on CODA, a machine that uses dataflow sequencing at a coarser level and conventional versus Neumann sequencing where possible. This approach is based on an observation from their Sigma-1 experience that pure dataflow pays an excessive time penalty to synchronize on every instruction. I was unable to obtain many details about CODA.

Yamaguchi's group at ETL is building the EM-4, a symbolic dataflow machine. The EM-4 is being designed as a single-chip dataflow processor implemented with a 50k-gate gate array. It is able to sequence instructions either by dataflow token arrivals or using a program counter (something they called *strong connection*). With the conventional sequencing, a register file is available to hold intermediate data. These innovations, similar to work done by Bob Ianucci in Arvind's group at MIT, overcome the locality penalty of dataflow. They expect this machine to be operational in March 1989.

UNIVERSITY OF TOKYO

In Professor Tanaka's laboratory at Tokyo University a group is building a parallel inference engine (PIE). This project has many objectives in common with the PIM machine at ICOT but is quite different in implementation. Two PIEs have been built to date and a third is planned. The first was a TTL version, the second was built using four 68000 microprocessors, and the third is being designed using 50k-gate gate arrays. I examined the hardware of the first two machines and the laboratory test set used to test the gate arrays (back from fab) for the third machine. I was quite impressed with their ability to build experimental machines in a university environment.

Professor Goto is involved in the development of a Josephson junction computer technology based on a circuit called the quantum-flux parametron. His goal is to build a 10-GHz machine with a power dissipation of 10 nW per gate. The machine will be constructed in three dimensions with inductive communication (no contacts) between levels. This work is in the early stages. They have simulated their devices and have built a few prototype gates in collaboration with Hitachi.

HITACHI CENTRAL RESEARCH LABORATORY

I spent an afternoon at Hitachi talking with a group of engineers responsible for designing the S-810 and S-820 supercomputers. These machines are air-cooled ECL machines with a 4-ns clock period and four parallel vector pipelines. The machine is built from ECL gate arrays with a delay of 0.2 ns/gate. The peak performance is 3 GFLOPS.

It was remarkable to me that they could achieve this level of performance in an air-cooled machine where chip crossings are quite expensive (1 ns). We discussed the design in some detail and there were no tricks, just solid engineering. The one innovation was the use of a combined memory/logic chip used for the vector registers. This chip allowed the register memory and port logic to be combined on one chip. Without the chip, it is unlikely that they could have achieved their 4-ns clock rate. Another key area of the design was the memory bank

conflict resolution logic. I was curious how they could resolve bank conflicts and reorder returning memory requests in 4 ns. The answer was that they didn't. Any bank conflicts stalled the machine and replies were always returned in order. In this logic, as in the rest of the machine, they have opted for the simplest possible solution and implemented it in very fast logic.

NEC C&C RESEARCH LABORATORY

I spent an hour at NEC with Tadashi Watanabe, the manager/principal designer of their SX-2 supercomputers. The SX-2 is an older design (1984) than the S-820 (1987) and somewhat slower with a 6-ns clock and a peak performance of 1.3 GFLOPS. Watanabe hinted many times throughout our discussions that a faster machine was in the works and would be announced within the year. However, he would not elaborate.

The two features of the SX-2 that most impressed me were its packaging technology and its scalar performance. The SX-2 is water cooled using thermal conduction modules (TCMs) that at first appear quite similar to IBM's TCMs. The SX-2 TCMs, however, use a polyimide substrate with very fine (25-micron) wires. The polyimide gives much faster signal propagation than IBM's ceramic modules. The fine pitch wires enable all the wiring to be contained on two metal layers, simplifying the manufacturing of the machine. The individual ECL gate arrays are TAB bonded to the substrate.

The SX-2 scalar processor is a RISC architecture that executes an instruction every 6 ns for a peak scalar performance of 166 MIPS. If it were marketed separately from the vector machine, it would be a viable workstation product today (5 years after its introduction).

I visited three other groups at NEC: the group working on architectures for artificial intelligence, a group making dataflow signal processing chips, and a group that builds special-purpose hardware for computer-aided design. The artificial intelligence architecture group has built the CHI-II processor (see FGCS section) and the LIME, a CHI remicrocoded to be a LISP machine. The dataflow chips and hardware accelerators were impressive experimental systems.

William J. Dally received a B.S. degree in electrical engineering from Virginia Polytechnic Institute in 1980, an M.S. degree in electrical engineering from Stanford University in 1981, and a Ph.D. degree in computer science from Caltech in 1986. From 1980 to 1982 Dr. Dally worked at Bell Telephone Laboratories where he contributed to the design of the BELLMAC-32 microprocessor. From 1982 to 1983 he worked as a consultant in the area of digital systems design. From 1983 to 1986 he was a research assistant and then a research fellow at Caltech. He is currently an associate professor of computer science at the Massachusetts Institute of Technology. Dr. Dally's research interests include concurrent computing, computer architecture, computer-aided design, and VLSI design.

Joseph A. Goguen
Professor of Computing Science
University of Oxford

TOWARDS AN ASSESSMENT OF FIFTH GENERATION RESULTS

Although the technical results reported by the Fifth Generation Project were very interesting in some areas, in certain others they were much less so. The work being done by Mukai and others in the Second Laboratory of ICOT is of great interest and promise, as is some of the work on programming languages. However, the work on architecture, which the Fifth Generation Project's publicity has led one to think of as central, seemed quite ordinary, as well as behind schedule. The following three sections discuss these three areas in greater detail.

Hardware

The centerpiece in the exhibition hall was an array of 64 PSI machines, constituting a single multi-PSI machine, running the PIMOS operating system and supporting the Guarded Horn Clause (GHC) language. Since this system had only been running for 2 weeks before the conference, no comprehensive performance figures were available, and only rather simple programs were executing. However, my impression is that the performance was not especially impressive and that the programs must have been very hard to debug. The extent to which logic programming can effectively exploit parallelism remains an open question. On the positive side, the CHI-II machine, which was designed by Konagaya from NEC and buried in one of the other demonstrations, looked as if it could develop into something quite nice. It has an enormous core memory and performs complex matching operations (on DNA molecules) with impressive efficiency.

Software

The GHC language, due to Ueda, is an elegant solution to the problem of producing a systems programming language in the logic programming tradition. Unfortunately, the semantics of GHC has little to do with logic, despite the Horn clause syntax, and it is not clear that this language will really be very convenient for programming or debugging. The latest development in this area is a preliminary design for a language called A'UM, due to Yoshida and Chikayama, which is more in the tradition of object oriented programming and which will probably be more suitable than GHC, if it is properly developed.

Natural Language Understanding

In the natural language area, a rather daring decision has been made to use the situation semantics developed by Barwise, Perry, and others at Stanford as a theoretical foundation. Furthermore, a massive Japanese/English dictionary is being built to support the project. And finally, some very interesting programming language design work has been done, including the CIL language, for writing natural language processing systems. There is also some very good work on discourse understanding using situation semantics.

OTHER RESEARCH

Among the many talks from outside the Fifth Generation Project, the one that I enjoyed the most was by Robin Milner from the University of Edinburgh. His talk was titled "Interpreting One Concurrent Calculus in Another" and presented some very interesting new ideas for relating the specification and implementation of concurrent programs in a systematic way, using some ideas similar to institutions. David H.D. Warren, now at the University of Bristol, also gave a very interesting talk on a novel multiprocessor architecture for logic programming. Futamura of Hitachi, currently visiting Harvard, gave a nice review of partial evaluation in a functional programming context. There were many papers on functional and object oriented programming as well as on logic programming and that many of these papers were by Japanese authors. Two of these papers gave a reflective semantics for object oriented programming, similar to some work done at SRI 2 years ago.

Joseph A. Goguen is Professor of Computing Science and Fellow of St. Anne's College at the University of Oxford. He also serves as co-director of the joint M.Sc. degree between the Programming Research Group and the Engineering School and is principal investigator on several grants. In addition, he is a subcontractor and frequent visitor to SRI International in Menlo Park, CA, where he was formerly a senior staff scientist and a senior member of the Center for the Study of Language and Information at Stanford University. Prof. Goguen has a bachelor's degree from Harvard and a Ph.D. from Berkeley, both in mathematics, and has previously taught computer science at Berkeley, Chicago, and UCLA, where he was a full professor. He won a Research Fellowship in the Mathematical Sciences at IBM Research in Yorktown Heights, NY, in 1972 and has held two Visiting Fellowships at the University of Edinburgh. His current research interests include software engineering; theorem proving; hardware verification; the design and implementation of programming languages based on logical systems, particularly multi-paradigm languages that combine object-oriented programming with functional and logic programming; and the design and implementation of massively parallel architectures to efficiently execute such languages. Prof. Goguen has also done research on semantics and is particularly well known for his work on abstract data types, initial model semantics, and algebraic specification. Other research interests include linguistics, logic, psychology, and computer security.

Jane W.S. Liu
Department of Computer Science
University of Illinois, Urbana

INTRODUCTION

The 1988 International Conference on Fifth Generation Computer Systems (FGCS'88) was by far the best organized conferences I have ever attended. On the average, the presentations were good. There were very few questions from the audience, however, making the presentations and panel discussions less stimulating than they could have been. In addition to attending the conference, I visited the Systems Laboratory, Oki Electric Industry Co., Ltd. and the Institute for New Generation Computer Technology (ICOT).

The first part of this article describes my activities outside of the technical sessions, including an interview with Dr. Shunichi Uchida, chief of the Fourth ICOT Laboratory, my visit to Oki Electric Industry, and discussions with several industry researchers. The second part is on FGCS'88 technical sessions.

ACTIVITIES OUTSIDE OF TECHNICAL SESSIONS

Interview with Dr. Shunichi Uchida

The sessions scheduled on the first 2 days were plenary. Besides the opening session, a panel discussion on the social impacts of information technology, and keynote speeches, the other sessions were devoted to summary reports of ICOT research and development. The present status and future plans of the FGCS project were summarized by the deputy director of the ICOT Research Center, Takashi Kurozumi. The directors of the five ICOT research laboratories reported on the work done in their laboratories. These overview presentations were followed by in-depth reports in special sessions on ICOT efforts. These special sessions were on the last 3 days, parallel with sessions of submitted papers.

As we sat in the conference rooms, holding over 1,000 people, we were highly sensitized to catch even the slightest flaws in the approaches and directions of the work presented. We viewed the presentations as if they were thesis defenses or contract reviews to which we are accustomed and duty bound to be critical. By the end of the second day, however, it became apparent that this was not the right approach for me to understand and appreciate the work done in the FGCS project. The presentations on the FGCS project were more publicity announcements than technical presentations. (The TECHNICAL SESSIONS section contains a brief discussion on special sessions on ICOT research.) It was doubtful that I could formulate an objective and unbiased opinion on the accomplishments and mistakes of the FGCS project based on these talks. I realized the impossibility of writing a credible critique without first spending some time to find out from the researchers themselves the concerns and constraints that led to their design choices; many of the choices were not what I would have made. I have had no such opportunity in the past and have no plan in the future to do so. I was pleased that ICOT's Dr. Shunichi Uchida, chief of the Fourth ICOT Laboratory, was willing to talk to me about his views on ICOT and the FGCS project. Following is a brief summary of our conversation, together with related information. It is not an objective, unbiased report.

I told Dr. Uchida that I needed to have a broader perspective; it was not enough to know that the technical objective of the FGCS project is to build a prototype, high-performance computer based on logic programming. Since the FGCS project was launched, we have become better aware of the limitations of logical programming paradigms, more capable of building high-performance machines for artificial intelligence (AI) applications, etc. By the year 1991 when the prototype system will be completed, its impact will be significantly smaller than anticipated in 1982. I wanted to know what he thought the overall impact and contributions of the FGCS project would be. His answer was that the FGCS project not only will achieve the prototype fifth generation computer system and the related research results but will also plant hundreds of seedlings in Japanese industries. The prototype system itself will not be nearly as important as the young engineers trained at ICOT.

Dr. Uchida views the FGCS project as a "great project" in computer science. A great project is a long-term (5 to 10 years) research and development effort launched to meet a specific need, focus world-wide attention, stimulate related research elsewhere, produce basic concepts and technologies as by-products, and educate future leaders. Examples of past great projects are Multics, Illiac IV, and Arpanet. The FGCS project was launched to meet the knowledge information processing needs of the 1990s; at the time of its creation, this need was not met effectively by existing computer architectures and software. It has maintained a high level of visibility and stimulated research activities in logical programming, AI, and supercomputing all over the world. Dr. Uchida hopes that like the past great projects, the FGCS project will also produce as by-products basic concepts and methods that will become foundations of future generation systems and researchers who will become future leaders in the world.

We talked about Multics and Illiac IV. In both of these cases, their critics can justifiably say that the prototype systems were "too little and too late." However, the contributions of these projects went far beyond the systems produced by them. Multics, for example, gave us memory management methods, file systems, and protection methods in use to date. These projects also gave us many influential computer scientists and engineers. Examples of these leaders include P. Denning and J. Dennis (Multics), D. Kuck and S. Chen (Illiac IV), and L. Klienrock (Arpanet). In particular, many leading Japanese computer scientists were associated with Illiac IV, including Professor Hideo Aiso, the FGCS'88 Conference Chair, Dr. Masao Kato of Nippon Telegraph and Telephone Corp. (NTT), and Dr. Akihiro Hashimoto of NTT. Dr. Hashimoto wrote his classic and much cited paper on channel routing in 1971 while he was at Illinois working on the Illiac IV project!

ICOT is one of the training places in Japan of computer scientists in the areas of logical programming, parallel processing, knowledge base systems, and natural language processing. Except for a few leaders such as Dr. Uchida, the researchers at ICOT are young and inexperienced. Most of them were sent to ICOT shortly after they joined their companies. Less competitive companies, such as Oki and Hitachi, do not have world class research laboratories of their own. Sending a young computer scientist to work at ICOT or as a subcontractor is an effective way to train him in ICOT's areas of

expertise. (The section on my visit to Oki presents a specific example.) By serving as a subcontractor of the parallel inference machine (PIM) hardware ("Research and Development of the Parallel Inference System," pages 16-32 in the *FGCS'88 Proceedings*), a company like Fujitsu will develop the expertise and capability of fabricating and building supercomputer hardware. In this way, ICOT helps these companies to become more competitive. In the past, I have heard people say that major Japanese companies such as the Nippon Electric Corp. (NEC) do not support the FGCS project enthusiastically. This lack of enthusiasm is understandable since companies with strong research and development efforts in ICOT's research areas have less to gain from the FGCS project.

Young people typically want a career path in companies that provide security and, in return, give their companies unwavering loyalty. Most Japanese students stop at the bachelor's level and seek employment in industry. Rather than expecting the universities to do most of the training job, companies expect the universities to provide only training in the fundamentals. Companies typically devote a great deal of resources and time to train their young engineers (and computer scientists). The typical length of the training period in a company for an engineer with a bachelor's degree is 2 years. Sending an engineer to ICOT for 3 years is a reasonable alternative. Dr. Uchida remarked that such an arrangement would not be acceptable to American companies since the engineers may not return after being trained.

Our conversation ended by Dr. Uchida's asking why there are no great projects in the United States any more. I would like to think that one of the answers lies in the following fact: Today there are more places doing excellent work; consequently, each project has less visibility.

Visit to Oki

On the afternoon of November 28, I was invited by Dr. Haruaki Yamazaki and Mr. Nobuyoshi Miyazaki from Systems Laboratory, Oki Electric Industry Co., Ltd. to visit their laboratory. I was asked to talk about research activities at Illinois, in general, and my own recent research activities, in particular. We spent part of the afternoon talking about their involvements in and opinions on ICOT research activities. Mr. Miyazaki is a subcontractor for ICOT. Dr. Yamazaki is Mr. Miyazaki's supervisor. Dr. Yamazaki does not interact closely with ICOT. All the research activities of ICOT are closely related to logic programming software and hardware. Dr. Yamazaki's current work is on neural nets.

Both Dr. Yamazaki and Mr. Miyazaki spent a year in the late 1970s at the University of Illinois and earned their M.S. degrees under my supervision. They are both bright, independent, and very capable. I was somewhat surprised when Mr. Miyazaki wrote upon his return to Japan in 1979 that he was very happy to be involved in the task of finishing the work on a small database machine. In my opinion, an American engineer with his talent, skills, and independent personality would not be happy to be a "finisher." Such willingness to support team work as his is one of Japan's major strengths.

Shortly after he returned, Mr. Miyazaki was sent to ICOT to work on the DBM subsystem. The project involved four researchers at ICOT together with subcontractors at Toshiba and Hitachi. Several knowledge base machine subprojects started

as successors of the DBM subproject. Research in this area is now being done in the Third Research Laboratory. Mr. Miyazaki returned to Oki in 1985 and is currently working on a distributed knowledge base control mechanism needed to support distributed cooperative problem solving. He is a coauthor of the paper "Knowledge Base System in Logic Programming Paradigm" (in the *FGCS'88 Proceedings*). This paper discusses ICOT's effort in realizing a knowledge base subsystem to support a large knowledge base shared by AI applications on the inference subsystem. They plan to integrate these subsystems in the prototype knowledge information processing system in the final stage of the FGCS project.

For Mr. Miyazaki, his short stay at ICOT seemed to be a great experience. He acquired at ICOT research expertise in knowledge base management and AI. In addition, he gained experience in working with a multidisciplinary and multiple-culture team and established contacts and a reputation outside of his company. Such positive experiences seem to be common for young researchers at ICOT as evidenced by the two-part article "Return of Former ICOT Researchers" in the *ICOT Journal*, No. 15, December 1985, and No. 16, March 1986.

I learned that unlike other Japanese national projects, ICOT does not support related research at companies, and most of these projects do not support research at universities. Companies serve as subcontractors. Thus, ICOT can better direct its outside research efforts. There are few subprojects that are done solely by researchers at ICOT. Most of ICOT's work is done jointly by ICOT and its subcontractors. For example, Mr. Miyazaki is currently working at Oki as a subcontractor on the distributed knowledge base mechanism subproject. In this case, most of the work is done at Oki. In some other subprojects, most of the research is done by ICOT, and the subcontractors implement hardware and software based on the design by ICOT. Smaller subprojects involve 2 or 3 people while the larger ones involve 20 or more people. Many subprojects are 5 to 10 persons in size.

NEC's Basic Research Institute in New Jersey

I first met Dr. Masahiro Yamamoto, assistant general manager of C&C Information Technology Research Laboratories, NEC Corporation, when he came to visit the University of Illinois in 1986. At the banquet, he told me that NEC had just established a research institute in Princeton, NJ. This institute is an American corporation and is a wholly owned subsidiary of NEC. Initially, the institute will have two divisions: Computer Science and Physical Science. In the future, there will be a Neural Science Division. Its charter is to carry out basic research. It is expected that (1) the research results of the institute will provide the technological advances leading to computers of the next century and (2) the institute will become a world class research organization with a reputation rivaling places such as Bell Laboratories.

On December 28, our department head, Dr. C.W. Gear, told us that he will leave Illinois in May 1990 to become the vice president of the Computer Science Division of this NEC research institute. His initial effort will be devoted to building his division. Knowing how capable Dr. Gear is both

as a researcher and as an administrator, I believe that the NEC Basic Research Institute in Princeton will be one of the world class research institutions in the 1990s and the next century.

NTT's Electrical Communication Laboratories

Dr. Akihiro Hashimoto of NTT was at the banquet. I first met Dr. Hashimoto in the summer of 1985 when my husband, C.L. Liu, was invited by NTT to spend 1 month at the Yokosuka Electrical Communication Laboratories. At that time, Dr. Hashimoto was the director of the Data Processing Development Division at Yokosuka. Later that year, after a reorganization of the laboratories, he became the director of the Knowledge Engineering Division of the Communications and Information Processing Laboratories. C.L. Liu found the research and development environment of the Yokosuka Laboratories stimulating. A great deal of good work is done there. For example, during his visit, C.L. Liu spent most of his time with Mr. Yukihiro Nakamura, who was working on a knowledge-based, integrated-circuit design tool at the time. This tool, called Parthenon, allows the design of an LSI circuit to be expressed in a high-level language (SFL) and produces an acceptable circuit layout as the final result. The tool has since become a commercially available tool marketed by NTT.

Among the ICOT laboratories, the research activities of the Fifth Research Laboratory on Knowledge-Processing Demonstration System are concerned with technologies needed to build the next-generation tools. Specifically, their activities are in five areas: expert systems for design tasks, hypothetical reasoning, distributed cooperative problem solving systems, qualitative reasoning, and tool architectures. These technologies are considered to be critical ones for building the next-generation tools that are capable of tackling design and synthesis problems. One of the important domains of application is integrated-circuit design. The paper "Experimental Knowledge Processing System" in the *FGCS'88 Proceedings* reports that one of the experimental expert systems built in the laboratory was for VLSI logic design. It was not among the ones demonstrated, however. The December 1987 issue of *ICOT Journal*, No. 18, also contains an article on the activities of the ICOT Fifth Laboratory. In addition to the scope of research, the article also lists the five Knowledge System Shell Subworking Groups of the laboratory. These subworking groups are chaired by academicians at leading Japanese universities. This seems to be a link for interactions between researchers at ICOT and universities.

Dr. Hashimoto told me that he is now the executive manager of the Information Science Research Laboratory, which is part of the NTT Basic Research Laboratories. He said that his laboratory is currently involved in research in social impacts, human factors, psychology, etc., as well as traditional computer science and artificial intelligence areas. While the world's attention was focused on ICOT at the time, Dr. Hashimoto's presence at the banquet reminded me of the many world class research laboratories of Japan.

In my impression, both NTT and NEC support research and development at levels comparable to our AT&T and IBM. Take NTT, for example. Before its privatization, the Yokosuka facility was one of

its four research and development laboratories. The other three are located at Musashino, Ibaraki, and Atsugi. They are jointly known as the Electrical Communication Laboratories (ECL). When NTT became a private corporation in 1985, it retained these four laboratories. In 1987, they were reorganized into what is known as Research and Development Headquarters, consisting of 11 laboratories, a technical information center, and 2 development centers. The laboratories are chartered to carry out basic scientific research and advance technological development. The basic technologies developed in the laboratories are transferred to the development centers where new commercial systems and software products are developed. The information center and development centers serve as bridges between the research laboratories and the operating divisions to facilitate effective technology transfer.

During my previous visits to Japan, I briefly toured the Yokosuka facilities and had a short discussion with a couple of researchers working on language translation. I remember being impressed by the breadth and depth of their work. According to the article "NTT Electrical Communications Laboratories" in the *ICOT Journal*, No. 19, March 1988, several NTT research laboratories (ECL) are involved in research on natural language processing and knowledge processing, two of the themes of ICOT research, as well as other AI-related areas. These laboratories are the Communication and Information Processing Laboratories, the Human Interface Laboratories, the Software Laboratories, and the Basic Research Laboratories.

Examples of natural language processing technologies developed at the ECL include Japanese sentence analysis, language translation, and dialogue processing. The results of this work have already led to several commercially available products as well as basic methods in natural language processing. Examples of commercially available products include a Japanese proofreading support system called Voice-Twin for publishers and an automatic indexing system. Examples of basic methods include a multistage conversion method that aims at producing high-quality Japanese-English translations and methods for providing intelligent communication services such as automatic destination identification.

The Knowledge Systems Laboratory (in the Communications and Information Processing Laboratories), the Information Processing Laboratories (in the Basic Research Laboratories), and the Software Research Laboratory and Software Engineering Laboratories (in the Software Laboratories) are places in NTT where knowledge processing research is carried out. The integrated-circuit design tool Parthenon, developed by Mr. Yukihiro Nakamura and his colleagues, is an example of the kinds of results sought at the Knowledge Systems Laboratory. In addition to intelligent LSI-CAD, the Knowledge Systems Laboratory is also concerned with basic research in other key areas such as knowledge representation languages, knowledge-based models, knowledge acquisition mechanisms, and distributed cooperative inference mechanisms.

The Electronic Dictionary Project

Mr. Toshio Yokoi is general manager of the Japan Electronic Dictionary Research (EDR) Institute, Ltd. EDR was established in 1986. It is sponsored by the Japan Key Technology Center and private

corporations including NEC, Fujitsu, Hitachi, Toshiba, Oki, and Mitsubishi. Its budget through 1994 is over $10 million.

I learned of the Electronic Dictionary project from Mr. Yokoi. The objective of this project is to develop large electronic dictionaries needed to support the next-generation natural language processing technology and knowledge information processing. The brochure that Mr. Yokoi sent me on this project says that the dictionaries will be "of computers, by computers, and for computers. Of computers means that they can be processed and recompiled with computers into various forms. By computers means that the dictionaries are being developed by using the current computer and natural language processing technology. For computers means that the electronic dictionaries are used for computers to process and understand languages." One of the goals of the EDR project is to develop a general specification method, a development method, and support systems that are not dependent on the languages and applications. The other goal is to promote international and interindustrial cooperation.

TECHNICAL SESSIONS

Special Sessions on ICOT Research

The papers on pages 3-108 in the *FGCS'88 Proceedings* summarize the research and development activities in the five ICOT laboratories. In addition to these summaries, papers scheduled in ICOT special sessions gave overviews of ICOT research on knowledge base mechanisms, the parallel inference machine (PIM) architecture, the parallel inference machine operation system (PIMOS), knowledge base management systems, the constraint logic programming language CAL, dictionary and lexical knowledge bases, a software environment for research into discourse understanding systems, and expert system architectures for design tasks. There are also many papers on recent ICOT research results in regular sessions.

These presentations were informative. One could not help but be impressed by the thoroughness and depth of ICOT's research and development work. I was convinced that ICOT will achieve the goal of building a prototype fifth generation computer that is totally based on logic programming and capable of executing 100 million inferences or more per second in 1991. This prototype system will integrate hardware and software components developed and evaluated in the past two stages of the FGCS project and will serve as a test bed of the ideas generated in the project. This prototype system will again demonstrate the superior Japanese capability in cutting-edge, advanced development.

While the talks were informative, they were not as stimulating as I had expected. As stated before, I believe that the main reason was that the ICOT talks were intended to publicize the FGCS project. They sounded like some project reviews, designed to impress rather than to inspire and to communicate ideas; certainly, the size of the conference was not conducive to intellectual exchanges. The accomplishments of the project were described without accompanied discussions of lessons learned and mistakes made. Current results and future directions were often not compared to related work done and different approaches taken elsewhere.

Many of the results are good engineering work but did not break any new ground. It was certainly hard to do justice to these results in the time allotted for the presentations. Page limitation was likely to be another factor that prevented the authors from going into the points that I would like to have seen addressed.

Knowledge Bases and Knowledge Base Management. An example of good engineering work is the experimental knowledge base machine and the knowledge base management system described in the papers "Overview of Knowledge Base Mechanism" and "Overview of the Knowledge Base Management System." The knowledge base machine is called Mu-X, and the knowledge base management system is called Kappa.

The hardware of the knowledge base machine Mu-X is a shared-memory multiprocessor system consisting of eight processing elements, a conventional shared memory, and a multiport page memory. Each processing element contains an MC68020 microprocessor, a moving-head disk, a local memory, and a multiport-page-memory interface. Access to the multiport page memory is on a page-at-a-time basis. In a k-port memory, words in each page are stored in k memory modules. The modules are connected by a rotary switch that cyclically changes the connections between ports and memory modules. The memory allows conflict-free, concurrent read/write accesses to arbitrary pages as long as no two or more ports try to write to the same page at the same time. In the paper titled "Multiport Memory Architectures," Y. Tanaka of Hokkaido University showed how a conflict-free, multiport RAM can be built. The implementation requires each word be stored redundantly in $O(k^2)$ modules, and access time is $O(\log_2 k)$. It was shown that by using this multiport RAM as a cache and a multiport page memory as the main memory, a cost-effective, conflict-free multiport memory can outperform parallel cache architectures for k in the range from 6 to 16.

Mu-X supports an extended relational model with term relations in which attributes can have structured variables. Multitransaction support facility is being implemented. (A comparable development project is the Multiple Backend Data System (MBDS) developed at the Naval Postgraduate School. The MBDS is a database machine built on a network of Sun workstations. The hardware system is not ideally suited for this application. However, a great deal of attention was given to tune the MBDS for high performance.) The Mu-X experimental system can be developed into a commercial product that is very competitive when compared with currently available database machines.

Kappa is another knowledge base management project at ICOT. Kappa is designed to support knowledge bases in both the personal sequential inference (PSI) machine environment and the multiple-PSI and parallel inference machine environments. Its layered structure contains the database layer, the knowledge base layer, and the user interface layer. The database layer supports a nested relational model as well as a semantic network and classification hierarchy. The knowledge base layer consists of knowledge representation languages, an experimental deductive mechanism, and an object management facility. This project is also concerned with the effective integration of the deductive database approach with the object-oriented approach.

Parallel Inference Machine Architecture and Operating Systems. The projects on the parallel inference machine (PIM)

architecture and its operating system (PIMOS) are described in two of the ICOT overview papers. The breadths of these projects are very impressive. Their combined scope of work includes the design and implementation of the PIM hardware, the kernel language KL1, and the parallel operating system. Issues addressed range from the abstract instruction set, distributed resource management, job-level and goal-level scheduling, memory protection, and so on. I was told that Japanese companies usually prefer to develop what they need rather than to make use of what was developed elsewhere. These projects demonstrate this preferred approach. In any case, these projects undoubtedly provide an excellent environment in which young engineers and computer scientists can learn and practice the whole range of skills needed to build a computer.

On the other hand, one might want to ask whether it is better to concentrate one's efforts in a few critical areas rather than spreading oneself thin. Take the problem of resource management and scheduling, for example. A part of the problem is that of partitioning a computation, a *job*, into granules to be executed in parallel. Whether each granule is a goal reduction or a Fortran loop is not relevant in this problem. The granule size (or sizes) should be chosen to match the characteristics of the computation and the structure of the underlying system. Specifically, given a system configuration, one wants to achieve an appropriate degree of parallelism; that is, an ideal tradeoff between the degree of possible parallelism and the amount of communication overhead. The rest of the problem is that of assigning system resources to jobs and controlling resource usage within each job. Again, whether the jobs are logic programs or Fortran programs is not relevant in this problem. Only the characteristics (that is, precedence constraints, dynamics, communication pattern, etc.) of the jobs to be scheduled are relevant. Therefore, the best approach to the scheduling and resource management problem is to (1) understand the differences and similarities of the characteristics of jobs in logic programming and other kinds of computations; (2) survey known methods in task partitioning, synchronization, job assignment, load balancing, and scheduling; and (3) design new methods if existing ones do not work. I assume that the ICOT researchers are fully aware of the vast amount of work done in this area, although the papers do not give this impression. For example, priority management was done in PIMOS in a straightforward manner. The paper describes it fully, while the more critical issues, such as undesirable anomalies of depth-first scheduling in a parallel environment, are ignored. As another example, the paper titled "Load-Dispatching Strategy on Parallel Inference Machine" describes a simulation study of several sender-initiated load-dispatching strategies to determine their performance in the PIM environment. The results reported in the paper would be more useful if they were compared with the known results on load balancing, in particular, results on other types of strategies. (The load balancing strategy used in the prototype PIM is of the receiver-initiated type.) Similarly, the paper "Load Balancing Mechanisms for Large Scale Multiprocessor Systems and Its Implementation" did not mention any related work on load balancing methods, such as the gradient method developed at the University of Utah, that were designed to dynamically schedule granules of numerical computations on large scale multiprocessors.

Like many people, I question the decision of building a system totally based on logic programming. The reason given for this policy decision is that this allows the system designer to view all levels of the system in a logic framework. "This is an important way to solve the so-called semantic gap argument: application and implementation are closer; therefore execution is faster." This argument may be true if logic programming is indeed the only problem solving tool one uses. However, as argued by Dr. H. Simon, the keynote speaker, other problem solving methods are often more effective; we want to use other tools. (See the following section on keynote and invited speeches.) Of course, one can always extend logic programming to support other problem solving methods, for example, by introducing constraint solving techniques. These extensions are likely to introduce new semantic gaps. Moreover, how operating system functions, for example, input/output, interrupt handling, and buffering, are provided is not important to the applications as long as the interface between the applications and the underlying system is good in some sense. The interface issue is addressed in the paper that describes Aurora, an or-parallel Prolog system, developed jointly by Argonne Laboratory, Manchester University, and the Swedish Institute of Computer Science. Aurora is based on a virtual shared memory architecture. It is portable; its portability is achieved by using a macro package that is written in C. The package provides definitions of basic operations on each multiprocessor system, a uniform syntax for creation of processes, management of shared memory, and accessing locks. This is a very cost-effective approach to building a logic programming environment.

Languages. In addition to the kernel language KL1, the other FGCS languages that attracted my attention were A'UM, a stream-based, concurrent, object-oriented language, and CAL, the FGCS constraint logic programming language. One issue that was raised in the panel session on theory and practice of concurrent systems is FGCS's choice of making parallelism explicit. In contrast, the other choice is to make parallelism transparent to the programmer. Those in favor of the latter approach, including me, argue that it is better to have parallelism exploited in such a way that it impacts the application programmer as little as possible. Whether a program runs parallelly or sequentially should be of little concern to the programmer; only the performance counts. This choice is a subjective one. It is said in one of the ICOT papers that the FGCS languages let "the application programmers have explicit access to parallelism *if they want*." If this means that parallelism transparency is also supported somehow, it is ideal, of course.

A'UM is an object-oriented, stream-based language. I was interested in this language because its goals are similar in many ways to those of Mentat, an object-oriented, data flow language developed at the University of Illinois. ("Mentat, an Object-Oriented, Macro Data Flow System," by Andrew Grimshaw, Ph.D. thesis, 1987.) Mentat is a concurrent, object-oriented programming (COOP) language that adds a parallel mechanism on sequential control. With this mechanism, objects can be run in parallel, but the computation in each object is sequential. (Finer granularity within objects can be realized by using a parallelizing compiler such as the one developed by Kuck Associates to produce parallel code for the objects.) By combining the data flow and

object-oriented paradigms. Mentat provides an easy-to-use, transparent mechanism to exploit parallelism. This mechanism automatically detects data flow at run time and constructs dynamic program graphs for programs written in an extended C++ programming language. (Extensions to C++ are to support persistent objects and facilitate data flow detection.) Mentat also allows computation to be relational like A'UM. It does not have the disadvantages of other COOPs pointed out by the authors. Data on performance of several benchmark applications running on a Encore Multimax are encouraging.

A problem that has received a great deal of attention recently is how to make use of mathematical tools, such as the simplex method, as well as consistency checking and constraint propagation methods available in logic programming. The solution to this problem is to extend a pure logic programming language by introducing three computation domains: finite domain restricted terms, boolean terms, and linear rational terms. The extended language can then be used to solve many practical constrained-search and optimization problems (such as scheduling and circuit layout) that cannot be solved using logic programming. It is obvious that successful development and use of constraint logic programming languages is important to advocates of logic programming. The paper titled "The Constraint Logic Programming Language CHIP" gives an excellent overview of advances in the last 3 years. In addition to CHIP, which was developed by the European Computer-Industry Research Center, it briefly describes Prolog III, developed at the University of Marseille, and CLP, developed jointly by the IBM Watson Research Laboratory and the University of Monash in Australia. The constraint logic programming language CAL being developed at ICOT is based on the language CLP.

Other Highlights

Parallel Algorithms. Satoru Miyano of Kyushu University presented a paper titled "Parallel Complexity and P-Complete," which is a short tutorial on parallel complexity theory, a field of study that is concerned with determining what problems allow efficient parallel algorithms. A parallel algorithm for a problem with input size n is said to be efficient if it runs in time $O(\log^k n)$ for some constant $k \geq 0$ on a polynomial number of processors that work synchronously and communicate via a shared random access memory. A problem is said to be in the class NC if there are efficient parallel algorithms to solve it. The class of problems that are in NC is a subclass of the problems that are in P (and therefore are known to have polynomial time solutions.) It is believed that NC is a proper subset of P; a problem is said to be P-complete if it is in P but no efficient parallel algorithms exist for the problem. Hence, one can say that a P-complete problem is inherently sequential and cannot achieve drastic speedup. Of course, linear speedup (or even superlinear) speedup is still possible for a P-complete problem. Many important problems, including linear programming, the maximum flow problem, and the unifiability problem, are known to be P-complete. The paper then presents three general P-completeness theorems. Based on these theorems, a new series of P-complete problems that can be solved by simple greedy algorithms is identified.

Ernst Mayr of Stanford University, in "Parallel Approximation Algorithms," surveys problems for which efficient parallel approximate algorithms are known. These

problems include bin packing, list scheduling, and high density subgraph problem. The paper also shows that efficient parallel approximate algorithms exist for some NP-complete problems. For example, there are efficient parallel algorithms for first-fit-decreasing bin packing.

These two papers pointed out the need to carry out further research in parallel complexity, especially the complexity of parallel approximate algorithms. An example of the kind of results that are of practical interest is the one reported in "Parallel Complexity Results About Greedy Breadth and Depth First Search," by R. Greenlaw (Technical Report no. 88-07-05, University of Washington). Greenlaw studied the greedy breadth first search problem to determine how quickly lexicographic breadth first search numbers can be assigned in parallel to vertices of a graph. He showed that the problem is in NC by finding an efficient parallel algorithm. This result is interesting because the problem of greedy lexicographic depth first search is known to be P-complete. As said by Greenlaw, "the breadth first search problem is easier than the depth first problem." Another fruitful direction of research is to develop efficient parallel approximate algorithms that can be implemented in message-passing architectures such as hypercubes.

We also need to better understand the behaviors of approximate search algorithms and heuristics in parallel implementations. It is known that anomalous behaviors can lead to detrimental speedup (that is, parallel implementations run slower than sequential implementations of the algorithm.) This problem is discussed in the paper "Parallel Processing of Combinatorial Search Problems," by B. Wah, *Computer*, July 1987.

DNA Sequencing. One of the demonstrations at the conference was on the use of a knowledge base system for DNA sequencing. The problem is, given a DNA sequence, typically encoded in a string of letters, we want to search a database and retrieve all sequences or parts of sequences that significantly match the given sequence. This pattern matching problem is complicated by the fact that there are gaps, duplications, and large-scale segment rearrangements in the sequences. Searching through a commercially available database is a long and tedious process.

This demo points to an application area of fruitful research in knowledge bases (e.g., homology search and reasoning with incomplete information), computational geometry (e.g., in design and simulation of experiments to determine DNA structures), and pattern recognition. To carry out research in this area, we need computer scientists who understand problems in life science and life scientists who are comfortable with computer science theories and methods.

Keynote and Invited Speeches

The keynote speech was given by Dr. Herb A. Simon. His paper is titled "Prospects for Cognitive Science." Dr. Simon not only made strong position statements but also outlined several research frontiers in his speech. His paper deserves to be strongly recommended.

Dr. Simon challenged the preference and emphasis of the FGCS project by pointing out the following:

(1) "Among the central principles is the idea that problem solving is heuristic search." Logic is not the universal law governing human reasoning. Logical

reasoning proceeds in small steps; large numbers of steps are needed for even the simplest proofs. It is not an effective method for solving most practical problems. Solving practical problems often requires heuristic search, to discover rather than to verify, since often the completeness and guaranteed correctness of a search are computationally infeasible.

(2) Problems that require an exponentially explosive search remain infeasible on parallel hardware, an indisputable fact. Consequently, hardware development has not been the bottleneck that limited the rate of progress in cognitive science. Lack of good heuristics is.

(3) Special Lisp machines and Prolog machines are now available. They allow important primitives to be executed faster, but these machines are not necessarily cost effective compared with powerful general-purpose hardware. They definitely do not represent a "breakthrough" in cognitive science.

(4) Many problems in reasoning and control are inherently serial; the degree of parallelism is limited by precedence constraints between subtasks. Only in applications, such as visual and auditory pattern recognition, where there is little connection among tasks, can a high degree of parallelism be achieved.

Among research frontiers mentioned by Dr. Simon are large scale experimentation with databases, applications of connectionism to sensory stimuli processing, and nonverbal representation of knowledge in AI. One problem in robotics mentioned by him is concerned with the integration of the low-level control system and a high-level planning system. Specifically, the problem is how to use a planning system that works with an inexact model of the real world to guide a robot that must have exact information in order to operate and survive. This problem in intelligent control was also discussed at the Workshop on Embedded AI Languages, Ann Arbor, MI, 16-17 November 1988. Several subproblems, including stability of intelligent control systems, the interface between symbolic and numerical computations, and planning with time constraints, were identified at the workshop.

Among the invited talks, I found one particularly interesting. It is "Multiple Reasoning Styles in Logical Programming" by H. Gallaire. This paper complements H. Simon's and defends logic programming as an effective, universal tool. It gives an insightful overview of the recent advances in logic programming and its strengths and limitations as a problem solving tool. Specifically, the paper discusses several reasoning styles from the point of view of logic programming. These reasoning styles include reasoning from first principles, taxonomic reasoning, hypothesis reasoning and truth maintenance systems, goal-directed reasoning, mixed-mode reasoning, and incremental reasoning. The paper shows that logic programming can often be extended to support these styles. It then addresses the question on how far logic programming should be extended. The paper also contains a brief survey on extensions of logic programming and argues in favor of tight integration of extensions to support multiple reasoning styles. The issue of symbolic modeling versus numerical modeling is also discussed here.

Microelectronics and Computer Technology Corp. (MCC) Presentations

Eugene Lowenthal, director of the Advanced Computer Architecture (ACA) program at MCC, was at the conference to talk about the activities of MCC. His paper is titled "A Review of MCC's Accomplishments and Strategic Outlook for Knowledge-Based Systems." ACA is the largest and the oldest of five programs at MCC. Lowenthal said that the name of this program reflects the original charter of MCC, which was to compete head-on with the ICOT-sponsored effort in the fifth generation computer systems. However, over the years, it has broadened its focus, and its emphasis has shifted from computer architecture to "innovations in software." Currently the program is divided into four laboratories: AI, Human Interface, System Technology, and Experiment Systems.

One could not help comparing ICOT and MCC. The latter was established in 1983, modeling itself after ICOT. However, there are more differences than similarities between ICOT and MCC. The most important difference between them is their research staffs. The researchers at ICOT are temporary, young, and inexperienced. Its effort is led by a few senior members. The ICOT environment is good for its focused effort in the FGCS project. The project's objective, to integrate "good ideas" together in a prototype system in a relatively short time, is easier to meet without distractions from "updated perspective" and "better ideas." MCC, on the other hand, has managed to assemble a talented and experienced staff with diverse interests and backgrounds. That "MCC's research efforts lack focus," as it is often said of MCC, is an unavoidable consequence; talented and experienced people tend to choose their own directions rather than focus on a direction chosen for them several years ago. As time goes on, they will want to move on to face newer challenges. Lowenthal said: "Most of the research undertaken by ACA has been motivated by a mission and goals established at the time of MCC's inception. Even as we continue to work towards fulfillment of these goals, it is clear that new research must be motivated by an updated perspective on future competitive pressures. Thus we have found it appropriate to define new long range beacons predicated upon a collective vision of how people and institutions will use computers at the turn of the century." He then went on to say that they are in the process of identifying new targets to motivate their research and focus their work.

Among the work reported by Lowenthal, I have been following the Orion database project. It was initiated in 1985. Its objective was to find ways of supporting persistent and shared objects in object-oriented programming environments and applications systems. In particular, the project studied the impact of the object-oriented concept on database management strategies and the requirements of the underlying database system imposed by object-oriented applications. I was impressed by the recently released (that is, transferred to the shareholders) Orion database system when I read its description in *ACM Transactions on Data Base Systems*. The system supports version control, change notification, and long-duration transactions, making it ideally suited for CAD/CAE applications. I was disappointed when told that it cannot be made available to the University of Illinois for experimental use.

D.B. Lenat of the AI Laboratory reported on their CYC project in the paper titled "When Will Machines Learn." CYC

is a 10-year project on knowledge acquisition that began in 1984. Lenat and Lowenthal said that in their opinion, the CYC project will have tremendous impact on automatic knowledge acquisition, that is, machine learning. The paper's abstract says that "if we succeed, knowledge acquisition in the post-CYC era will be not unlike the human teacher-pupil paradigm." Specifically, CYC is based on the common belief that the more we know about something, the easier and faster we will be able to learn more about it. In other words, "learning occurs at the fringe of what we already know." Hence we should be able to achieve effective machine learning if we provide as a starting point an immense knowledge base containing a large fraction of all the interrelated facts, heuristics, representations, etc. CYC is a full-scale effort to encode this very large knowledge base containing millions of pieces of common sense knowledge that make up what is called "late 20th century reality." It was reported that the CYC knowledge base now has half a million entries in it and will have 2 million entries in 1989. In the meantime, it is expected that CYC will become an increasingly more active intelligent agent that will help in its own construction. By 1994, there will be enough knowledge base so that the dominant knowledge entry mode can be natural language understanding. This work was cited by Dr. H. Simon as an example of good experimental work. My limited background in machine learning prevents me from fully appreciating this project. I can see the value of building and experimenting with an immense and ever-growing knowledge base that contains millions of facts and algorithms to keep track of the interrelationship between the facts. At least, it will allow experimental evaluation of some of the knowledge base management methods and will provide the large semantic database for applications such as natural language processing.

Jane W.S. Liu received her B.S. degree in electrical engineering in 1959 from Cleveland State University, OH, and her M.S.E.E. and D.Sc. degrees in 1966 and 1968, respectively, from the Massachusetts Institute of Technology (MIT). Before joining the University of Illinois, she worked as an electronics engineer for the U.S. Department of Transportation, Transportation Systems Center, Cambridge, MA; as a postdoctoral fellow in the Department of Electrical Engineering, MIT; as a member of the technical staff of the Mitre Corp., Bedford, MA; and as an engineer for the Radio Corp. of America, Needham, MA. Dr. Liu joined the Department of Computer Science at the University of Illinois in 1973 and is currently a professor of the Department of Computer Science and the Department of Computer and Electrical Engineering. Her current research activities are in the areas of real-time systems, scheduling and load balancing, data bases, data communications, and distributed operating systems. She is the co-director of the Illinois Computing Laboratory for Aerospace *Systems and Software, a National Aeronautics and Space Administration center of excellence. Dr. Liu served as the chairman of the IEEE Computer Society Technical Committee on Data Base Engineering in 1981 and 1982 and is the chairman of the IEEE Computer Society Technical Committee on Distributed Processing. She is an associate editor of the* IEEE Transactions on Computers and Data and Knowledge Engineering. *She is a member of the Association of Computing Machinery and the IEEE Computer Society.*

John M. Mellor-Crummey
Computer Science Department
University of Rochester

INTRODUCTION

In 1981, the Japanese Ministry of International Trade and Industry (MITI) announced the Fifth Generation Computing Project, a national project formulated as a 10-year plan dedicated to research and development (R&D) of symbolic inference machines for knowledge information processing. This project was undertaken as a joint venture between the Musashino Laboratory of Nippon Telegraph and Telephone (NTT), MITI's Electrotechnical Laboratory (ETL), and eight companies: Fujitsu, Hitachi, Matsushita, Mitsubishi, Nippon Electric Corporation (NEC), Oki, Sharp, and Toshiba (Ref 1). In 1982, the Institute for New Generation Computer Technology (ICOT) was founded with a core group of scientists from the various participating laboratories and companies to coordinate the fifth generation project.

Kazuhiro Fuchi noted in his keynote address at the conference that artificial intelligence (AI) is not the direct aim of the fifth generation computing project; it is the means rather than the end. The major objective of the project is the construction of parallel inference machines to enable high speed knowledge information processing. Based on previous AI research, it is estimated that fifth generation computers will require inference speed that is 1000 times greater than conventional computers (Ref 2, p. 3).

The initial 3-year phase of the project (1982-1984) focused on research and development of basic computer technology to efficiently support machine inference and knowledge-based processing. During this phase, researchers experimented with various techniques for machine inference, including dataflow and reduction, as well as evaluating the feasibility of using a relational database scheme as the basis for constructing a parallel knowledge base (Ref 2, p. 5). This research led to the construction of several prototypes including PSI, a Personal Sequential Inference machine developed to support logic programming (Ref 3), and Delta, a parallel relational database machine (Ref 4). Research in software for fifth generation computing led to the development of Guarded Horn Clauses (GHC) for parallel logic programming and two sequential inference languages: Kernel Language version 0 (KL0), a descendant of Prolog, and Extended Self-Contained Prolog (ESP), an object-oriented language implemented on top of KL0 that provides hierarchical inheritance and a macro expansion facility. Construction of the Sequential Inference Machine Programming and Operating System (SIMPOS) (Ref 5) using ESP provided experience developing system software within a logic programming paradigm. Experience developing applications using logic programming languages was obtained by building prototype systems for applications including knowledge-based information retrieval, natural language understanding, and expert systems (Ref 6).

The intermediate phase of the project (1985-1988) was based on a 4-year plan for research and development of prototypes that will serve as the basis for fifth generation computers. The focus of research in this phase was on investigating how parallelism can be incorporated into models, algorithms, architectures for logical inference, and knowledge-based processing. The four major research themes of this phase, as stated by Kurozumi, were:

1. Basic software (including development of a logic-based kernel language, problem solving and inference software, knowledge base management, interfaces, and intelligent programming support)

2. An inference subsystem

3. A knowledge base subsystem

4. A development support system

During this phase, researchers developed the Multi-PSI (version 2) machine, a parallel inference machine constructed as a mesh of up to 64 PSI machines. The Multi-PSI machine serves as a testbed for parallel algorithms and provides an environment for obtaining practical experience in parallel software development. Researchers are developing the Parallel Inference Machine Operating System (PIMOS) for the Multi-PSI; parallel logic languages are being used as the basis for PIMOS kernel development. Also, much of the design of a pilot Parallel Inference Machine (PIM) was completed during the intermediate phase. This machine is intended to efficiently support KL1, a parallel logic language, and provide roughly four to five times the inference performance of a 64-processor Multi-PSI machine.

The research and development plan for the final phase of the fifth generation project (1989-1991) calls for integration of the results from hardware and software R&D in the earlier phases into a prototype fifth generation computer system. This machine is intended to be a high performance parallel architecture, consisting of approximately 1000 processing elements. It will support software for high speed inference and knowledge retrieval as well as a programming environment for developing knowledge information processing applications using parallel logic languages. Although the goal of the final stage is a prototype parallel inference machine, Fuchi sees 1989-1991 as a beginning rather than the conclusion of the study of parallel inference.

The first part of this report summarizes and evaluates the status of the fifth generation computing project based on information gathered at the Fifth Generation Computer Systems (FGCS) 1988 Conference, held at the end of the project's intermediate phase (after 7 years of the 10-year plan). The rest of the report describes: (1) some of the goals and infrastructure of the fifth generation project, (2) the hardware and software prototypes developed in the project's intermediate phase, (3) an excellent invited talk by Herbert Simon on the prospects for cognitive science and AI research, and (4) some interesting research projects presented at FGCS that are not part of the fifth generation project. The report concludes with a discussion of some technical issues that are critical for the success of fifth generation computers.

FIFTH GENERATION RESEARCH INFRASTRUCTURE

A researcher from ICOT indicated that one of the important goals of the fifth generation project was to foster cooperative applied research involving both corporations and the government. The fifth generation project sought to emulate the successful cooperation in the United States on projects such as MULTICS. It is hoped that the fifth generation project will provide a seed for future technology.

This researcher questioned why no large cooperative projects such as MULTICS currently exist in the United States. In his opinion, the technology that grew out of MULTICS, such as Unix, gave the United States a great edge in the field. In discussing this point with Dr. Thomas LeBlanc (University of Rochester) after returning from the conference, LeBlanc suggested that the Mach project initiated at CMU has fostered similar cooperation, producing several interesting advances in OS design.

Funding

Funds allocated in the budget were ¥8.3 billion for the initial 3-year phase, and about ¥21.5 billion for the 4-year intermediate phase (¥4.7 billion for 1984, ¥5.5 billion for 1985, ¥5.6 billion for 1986, and ¥5.7 billion for 1987) (Ref 2, p. 6).

ICOT Organization

The essential parts of the ICOT organization are a research planning department and five technical laboratories. The five laboratories are respectively devoted to the following five areas:

1. Kernel language software development
2. Interface software
3. The knowledge base subsystem
4. The inference subsystem, including PIMOS
5. Knowledge base management, including acquisition and use of knowledge

FIFTH GENERATION PROTOTYPES

The hardware and software of the fifth generation project were developed in tandem using a stepwise refinement development strategy. This strategy worked more effectively than had been anticipated (Ref 7, p. 17).

Software

In his opening address, Hideo Aiso, FGCS'88 conference chairman and professor at Keio University, indicated that at the current point in the project, software is the major emphasis. He stated that there have been large advances in hardware and that software systems must grow to match. In my opinion, developing the systems software to intelligently manage resource allocation on fifth generation computers will be one of the greatest challenges of the project.

Parallel Logic Programming. Guarded Horn Clauses (GHC) was proposed as the basis for parallel logic programming in the fifth generation project. GHC is a committed choice language: goal reduction commits to the first matching clause whose guard evaluates to true. The semantics of GHC enable goals to be reduced in parallel.

The software being developed for the parallel inference machine is based on a family of languages designated Kernel Language version 1 (KL1). The development of KL1 has its roots in GHC and experiences with KL0, the sequential logic programming language developed as the machine language for the PSI. Measurements of interprocessor communication in an implementation of flat GHC executing on the Multi-PSI (version 1) significantly shaped the

parallel processing facilities provided by KL1. The layers of languages in the KL1 family are as follows:

1. At the bottom is KL1-b (base), the base language that provides a virtual machine model similar to WAM (David Warren's Abstract Machine for Prolog).

2. The second layer is KL1-c (core), essentially flat GHC (flat means that only built-in predicates may be used in guards) with the Shōen meta-call facility). Also part of the second layer is KL1-p, a pragma annotation language for specifying how to divide and distribute jobs so they can be processed in parallel.

3. The top layer is KL1-u (user), user-defined languages for concurrent logic programming such as A'UM (Ref 8).

Although various types of parallelism have been exploited in logic programs (i.e., and-parallelism, or-parallelism, and stream-parallelism), logic programming languages are unable to express data parallelism. This makes it impossible to use logic programming languages to construct high performance parallel programs for tasks such as low-level image processing, even though these tasks are highly suited for parallel implementation. In my opinion, the inability to write efficient parallel programs for problems that are best suited to parallel implementation is a good reason to be skeptical of the goal to base all fifth generation parallel software on logic programming.

System Software. PIMOS, the operating system for the parallel inference machine, is currently being developed in KL1-c. PIMOS provides facilities for execution control, resource management, and database management. The PIMOS implementation under development on the Multi-PSI currently relies on a front end processor to provide all I/O. A more efficient model of I/O is needed for PIMOS (Ref 9). Although implicit synchronization in KL1 made the task of developing PIMOS easier, my impression is that logic languages are not particularly suited to operating system implementation. Being unfamiliar with the particulars of the KL1 language, I was confused by the description of some of the operating system facilities (Ref 9); however, my general impression is that kernel calls will be costly and low-level device handling is awkward using logic languages.

In KL1, goal reduction can be controlled using the Shōen facility for meta-programming. A Shōen call to control resource management for a goal reduction accepts parameters including minimum and maximum priorities for reduction of subgoals, the number of subgoal reductions that can be performed, streams for control and error reporting, and a mask of exceptions that Shōen will handle. Shōens can be hierarchically nested forming a tree with KL1 goals at the leaves. Since child and parent Shōens communicate frequently, the Shōen facility provides a "foster-parent" mechanism that enables reduction of inter-cluster communication when child and parent reside in different clusters.

In the design of the PIM prototype, KL1 goals are distributed within clusters on a demand basis. Idle processors request goals from busy processors. Currently, load distribution across clusters occurs only as a result of pragma annotations of the form *goal@node (C)*. When goal reduction

commits a clause containing a goal with such a pragma, the goal is sent to processor *node* in cluster *C*. Future plans call for implementation of intercluster dynamic load balancing.

In my opinion, reliance on explicit pragmas for distributing load will make writing substantial high performance parallel applications for knowledge information processing extremely difficult. Knowledge information processing applications have much less regular structure than typical numerical computations. For numerical programs, it is often easy to predict the computational needs of each process before execution, and a good load distribution strategy can be determined statically. However, the characteristics of computational load for knowledge information processing applications are highly variable depending on intermediate results; thus, static strategies will be ineffective. In particular, static pragma annotations do not take into account dynamic size of arguments during execution, so they are of limited utility. Also, many goal reductions can proceed in parallel using the same clause; therefore, annotations assigning specific subgoals in a clause to particular processors could result in some processors (and clusters) being swamped during execution by subgoals arising from multiple reductions initiated in parallel using the same clause. Dynamic load balancing is an utmost necessity for efficient utilization of highly parallel inference hardware.

A dynamic load balancing strategy is being investigated for the Multi-PSI-v2 (Ref 10). In their work, they have identified two important factors that will determine the success of dynamic load balancing strategies: communication locality and prediction of the amount of computation that each subproblem will require. In their technique, the processing power of the Multi-PSI machine is represented as a plane. The initial goal of a computation is assigned to the entire processing plane. Rectangular regions of the plane are dynamically apportioned out to subgoals as specified by static pragmas. The pragmas specify whether to split the plane along its length or width and what fraction of the area to assign to each division. Initially, the processing power plane is divided as a grid with each processor responsible for a square region (reflecting the mesh topology of the machine). During computation, each goal is identified with the center point of the region in the plane to which it is assigned. Divisions of the processing power plane between processors are dynamically adjusted using local communication between neighboring processors; adjustments are based on the number of active goals on each processor. Each time computation of a subgoal is initiated, the subgoal is sent to the processor responsible for its identified point in the plane. A forwarding mechanism in the mesh routing hardware supports this algorithm. A good feature of this technique is that it requires no centralized communication: all load balancing decisions are made between adjacent processors. However, while assignment of subgoals to neighboring regions attempts to recognize communication locality, the strategy is too naive and it is easy to end up with tightly coupled goals separated by several hops through the communication network. I am not aware of any empirical studies evaluating the effectiveness of this load balancing strategy. All demonstrations on the Multi-PSI shown at the conference exclusively used static assignment of goals to processors using pragmas. It would be worth tracking experimental work in this area since the success of the fifth generation computers ultimately depends on dynamic load balancing strategies.

The lack of a uniform global address space on the PIM and the use of compaction algorithms require clusters to maintain import and export tables to manage external references that arise when handling arguments during assignment of a goal to a remote cluster. Each time compaction occurs, export tables must be updated to keep the retain correspondence between internal addresses and external identifiers. In order to garbage collect import and export table entries that refer to multireferenced data, an incremental intercluster garbage collection scheme using weighted export counts was developed.

Management of external references with all of this machinery will be very costly. To minimize the overhead of maintaining intercluster references, partitioning a collection of goal reductions across multiple clusters must be done with great care. Any schemes for dynamic load balancing that are developed for the PIM will have to pay close attention to the cost of creating and maintaining intercluster references when moving goals between clusters.

Applications. Fuchi stated in his keynote address that the development of applications software currently underway as part of the fifth generation project is primarily a vehicle to guide further research.

Research and development of knowledge-based software in the intermediate phase focused on knowledge representation, utilization, and acquisition. The traditional focus of expert systems has been using a knowledge base and heuristic rules to drive an inference engine. The new focus in the fifth generation project is on problem solving using frameworks that match application domains. These include assumption-based reasoning, abductive reasoning, and qualitative reasoning. Additional research is in progress on distributed cooperative reasoning.

The focus of intelligent interface software R&D during the intermediate phase was on natural language understanding. Researchers studied both grammar, morphology, syntax, and phrase structure. Further goals in this area include the study of semantic and context analyses. The primary software development in this area during the intermediate phase focused on building a general-purpose language tool box (LTB) for understanding Japanese. LTB forms the basis for the DUALS-II and DUALS-III experimental discourse understanding systems. DUALS-III, which is under development, will be able to understand the meaning of 100 sentences (2,000 words) (Ref 2). DUALS-III will apply a constraint-based approach that includes contextual analysis for anaphora and ellipsis and a conceptual dictionary and thesaurus. DUALS-III will use planning to perform sentence generation.

A third area of research in this area was on intelligent programming software. In particular, there are investigations of computer-aided proof systems. These investigations aim to build a system for program transformation and verification composed of a proof checker, a term rewriting system, and a theorem prover.

Fuchi indicated that one of the ICOT objectives is to integrate constraint programming with logic programming. Three systems support various types of constraint logic programming: algebraic CAl, boolean CAL, and typed CAL.

Hardware. During the initial phase of the fifth generation project, the PSI machine was developed as a workstation for

software development during the intermediate and final stages. A novel feature of this machine is that it was the first machine to use a logic language as its machine language. It is interesting to note that PSI version 1 did not support virtual memory; instead, the designers supplied the machine with 80 MB of real memory, which they felt would be sufficient for most applications (Ref 11, p. 61). A benefit claimed for this approach was that it simplified garbage collection. The same paper seems to indicate that PSI version 2 does not possess virtual memory support either. The implication of this design choice is that the memory system capacity needs to be large enough to directly support the largest intended application. In talking with a researcher during our visit to ICOT, he indicated that they had no idea how much of the memory was actually in use on a PSI at runtime. A system with virtual memory could have a more modest amount of real memory and likely achieve nearly the same performance with a reduction in hardware cost. In addition, virtual memory would provide much greater flexibility for handling large problems. My impression is that the garbage compaction algorithms they developed could work similarly in a system with virtual memory by compacting in virtual space. Although it seems that the decision not to include virtual memory was an expedient choice during development, it appears an expensive choice since the intent is to provide PSI machines to a large number of programmers for development of logic programming software for later fifth generation prototypes.

The Multi-PSI-v1 consists of six PSI machines with a two-dimensional (2D) mesh interconnect. This machine was the basis for experiments with GHC: each node runs a flat GHC interpreter written in ESP capable of 1K logical inferences per second (LIPS). The Multi-PSI-v2 is a 64-processor machine with a 2D mesh interconnect. It has a KL1-b interpreter in firmware capable of 100K LIPS per processing element (PE) plus garbage collection. The KL1-b interpreter supports about 160 instructions, heap-based data allocation, and incremental garbage collection. The mesh network in Multi-PSI-v2 is capable of 5 MB/s/channel.

The PIM machine is structured as multiple clusters of processor elements. The current prototype calls for interconnection of these clusters using a pair of four-dimensional hypercubes. The PIM architecture does not support global shared memory: intracluster and intercluster addressing are different. Each cluster contains eight processing elements organized as a shared-bus shared memory multiprocessor; each PE is equipped with a write-back data cache. Interestingly, the data cache contains support for interprocessor messages and a novel lock operation that facilitates exclusive locking of data words, in some cases without using any bus cycles (Ref 12, p. 223).

Each processing element in the PIM is a tagged architecture built around a RISC style processor. The processor has a 50-ns cycle time and a four-stage pipeline fed by separate 64-KB address and data caches. The processor supports about 170 primitive instructions. In addition to main memory for the PE, there is a special instruction memory that contains "macros" for high level language support. The use of macro instructions kept the processor design simple, and use of the special instruction memory requires only a one-cycle delay for a call to one of the macro routines rather than a full

pipeline flush required for regular subroutine calls. Although the macro instruction memory is novel, its application seems limited since it will only provide a significant performance benefit in the presence of very frequent calls to extremely short procedures.

The prototype fifth generation parallel inference machine to be constructed in the final phase of the project is expected to have about 1,000 processors organized in clusters of about 10 processors. Clusters will be interconnected with some form of hierarchical interconnection network.

PROSPECTS FOR COGNITIVE SCIENCE

At FGCS'88, Herbert Simon, professor at Carnegie-Mellon University, gave an excellent invited lecture on the prospects for cognitive science in which he outlined a set of predictions for the next 10 or so years. Simon believes that acquiring a deeper understanding of our own intelligence is the best route for advancing AI.

Although advances in hardware have been essential for AI, hardware has not been the bottleneck since ideas can be developed independently of hardware. In his view, Lisp and Prolog machines do not represent a breakthrough in AI, rather only a speedup, and he questions whether they will remain cost effective. Simon indicated that he is skeptical of the application of parallel hardware as the answer for the exponential explosion of search.

Simon argued that speed and brute force heuristics do not go far toward achieving AI; intelligent behavior requires using vast amounts of domain specific knowledge (for example, semantic knowledge is essential for natural language processing; syntax is not enough). He remarked that the response of human experts is largely intuitive or judgmental based on a vast knowledge (>50,000 things about a target domain) as well as solution search techniques such as means-end analysis or hill-climbing.

Simon's views on parallel programming and logic programming differ with those of the FGCS project goals. Simon's view for the future:

- In robotics we need to focus on sensors and feedback from a robot to a planning system.

- The best use of connectionism in the immediate or near future is in the sensory domain. Most higher level functions are serial and inappropriate for connectionist modeling; however, low level processes make good use of parallelism.

- Experimentation with large-scale knowledge bases is important. We must learn how to bring large amounts of knowledge to bear to solve problems effectively. Memory and retrieval capabilities seem more important than parallel processing.

- In learning, the most promising approaches seem to be adaptive production systems in semantic domains and connectionist systems for sensory input.

- Simon's stand in the learning versus programming controversy is that our application of learning for computers probably should be more selective than it currently is.

- In knowledge representation, Simon believes that nonpropositional representations such as pictures and diagrams are important parts of human thought. A

challenge is to determine how we can adapt and use such representations in AI. The solution of many problems is a matter of finding the right representation.

Simon identifies four key issues in hardware and software systems for AI. First, Simon commented on the question of serial versus parallel systems for AI. For most parts of the brain, psychological research shows no evidence of massively parallel computation. Although sensory processing (vision and hearing) is demonstrably parallel, higher level functions are mediated by attention and appear sequential. Simon believes that taking advantage of massively parallel hardware is difficult in AI, except for special-purpose tasks. There is no reason to believe that general-purpose parallelism will be easy: dense, rigid connections between tasks and precedence relations make this difficult. Second, Simon indicated that we should heed the evolutionary lesson and recognize the importance of hierarchy in large systems. An important question in connectionist systems is how should they be organized hierarchically. Third, Simon criticized the use of logic programming for AI. The foundation for the logic programming movement is that reasoning should be logical and that programming languages should make logic accessible. Simon indicated that the lack of progress in computer theorem proving is evidence that adherence to the principles of logic is crippling; he stressed a need for higher level rules such as equality, transitivity, and commutativity. Domain-specific knowledge is very important for problem solving; however, this is more readily exploited by production systems than logic programs. Completeness and correctness are hard to come by; it is better to get a likely answer soon. Simon sees heuristic search as a theme for intelligent behavior: it is important to use best-first search rather than depth first as directly supported by logic programming. Fourth, he sees the need for nonverbal representations, both procedural and declarative (including diagrammatic) knowledge. Diagrammatic knowledge has been neglected.

OTHER RESEARCH PRESENTED AT FGCS

David Warren reported on the Aurora Or-Parallel Prolog system (Ref 13), developed by the Gigalips Project. This system seems to be a good step towards parallel execution environments for logic programs. Its use of implicit or-parallelism keeps parallelism transparent to the programmer. In contrast, the PEPsys system developed at the European Community Research Center (ECRC) utilizes independent and-parallelism identified by explicit program annotations (Ref 14); if a programmer incorrectly annotates a pair of goals as independent when they are in fact dependent, the program's correctness is violated. The Aurora system implements full Prolog with an additional *commit* operation that can be used in nondeterministic parallel evaluation (it prunes branches in the search tree to the left and right and is not guaranteed to prevent side effects in the pruned branches). Aurora provides both synchronous and asynchronous database predicates (synchronous predicates can be executed only when in the leftmost branch of the search tree; asynchronous predicates can be executed on demand). The Aurora system seems to be a solid, well-thought-out implementation based on solid principles that performs well on shared-bus multiprocessors (Encore and Sequent).

The data diffusion machine, another project presented by David Warren, also looked interesting (Ref 15). The data diffusion machine is a design for an architecture to support a shared global address space by using a hierarchical shared-bus organization. The design of the data diffusion machine incorporates a novel scheme for data migration at the word level (as opposed to page level). It is intended that data words in the machine migrate independently to where they are needed. The word level granularity of this scheme is intended to reduce unnecessary data movement and thrashing. Although not all of the details of this scheme are clear to me, the paper claims that the word level management of virtual memory needs only double the memory of more traditional page-based schemes. Two fundamental questions remain in my mind about the machine: (1) whether the hierarchical bus design can provide sufficient throughput for the data diffusion scheme, and (2) how will protection be handled in the machine (e.g., could the machine protect multiple users or programs from each other).

Some of the most exciting AI work that I saw at the conference was presented by David Waltz from Thinking Machines (Ref 16). Enormous speedups relative to sequential processors are possible using the Connection Machine if one can manage to design an algorithm to solve a problem in a way amenable to SIMD implementation. In his talk, Waltz described clever parallel algorithms for assumption-based truth maintenance, memory-based reasoning, computer vision, and chess endgames. The astounding performance of this variety of applications demonstrates a versatility of the SIMD approach that I had not previously recognized.

CONCLUSIONS

In the panel discussion on the social impact of information technology and international collaboration, Fred Weingarten, U.S. Office of Technology Assessment, noted that the availability of fifth generation computing systems would change the structure of business and education. The ability to use "humanized interfaces" based on these machines to communicate and access data through global networks would offer a vast array of possibilities. While I concur with Weingarten that the impact of fifth generation computers on society would be profound, I believe that significant technical challenges remain for the project in the construction of the systems software for the fifth generation parallel inference machine prototype.

In my opinion, dramatic advances in dynamic load balancing will be necessary if applications executing on fifth generation machines are to utilize a fraction of the cycles available from the highly parallel hardware. Naively hiding the nonuniform structure of the parallel machine from the programmer (the disparity in cost between intracluster and intercluster operations) will prove extremely difficult without incurring a heavy cost in program performance. Experiences in developing software for the BBN Butterfly (Ref 17) show that failure to recognize the nonuniform nature of the machine leads to tremendous performance bottlenecks that result from communication with remote nodes and resource contention. Furthermore, constructing efficient programs for a nonuniform machine is a very difficult task that requires intimate knowledge of all levels of the software in the system.

The most successful method of constructing high performance parallel applications is one of iterative refinement. The cycle of development, measurement, and redesign is critical to success in this area. However, there is an appalling lack of tools and techniques for performance analysis of parallel programs (this is a general criticism that applies to most, if not all, parallel computer systems, not just the fifth generation prototypes). In a conversation with Dr. Chikayama (ICOT), he indicated that there are no current plans to provide hooks in PIMOS for monitoring program execution performance and isolating bottlenecks. Lacking such tools, it will be extremely difficult to construct high performance applications without intimate knowledge of the machine structure. Even with such tools, high performance applications will require explicit partitioning of applications among clusters and individual processors using pragmas. Furthermore, to effectively annotate a program with pragmas, programmers must consider the procedural interpretation of the logic program, subverting the declarative nature of logic programs.

Successful strategies for dynamic load balancing will depend on several factors. First, it must be possible to accurately measure the degree of coupling between elements in a network of communicating entities. This will be necessary to approximate least cost partitions of the network. Second, methods for accurately predicting the resources required for each schedulable entity will be necessary. These will serve two purposes: to avoid migrating entities for which the cost of migration is a significant fraction of the computation they require, and to try to accurately estimate work available on each processor to avoid unnecessarily shuffling work back and forth due to inaccurate load estimations. At FGCS, Evan Tick presented a technique for estimating computation granularity of logic programs using compile-time analysis (Ref 18). Although Tick observed only limited improvement using his technique to guide scheduling on a shared-bus shared-memory multiprocessor, his success was likely tempered by the low cost of spawning a task on a remote processor in his environment. I predict that for nonuniform architectures such as the PIM in which the cost of intercluster operations is high, techniques such as Tick's will prove invaluable. Third, heuristic methods must be developed for assigning processes to clusters to balance load while minimizing communication. Finally, strategies for load balancing should avoid centralized communication. Centralized communication would limit the scalability of algorithms.

In conclusion, the successful creation of fifth generation computers hinges on being able to effectively exploit parallelism for knowledge-based information processing. Dynamic load balancing strategies will be essential to attain that goal. Since many of the problems that need to be solved for dynamic load balancing are intractable (i.e., least cost multiway network partitioning), success in this area will largely depend on the construction of effective heuristics.

REFERENCES

1. P. McCorduck, "Introduction to the fifth generation," *Communications of the ACM*, 629-630 (Sep 1983).

2. T. Kurozumi, "Present status and plans for future research and development," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 1, 3-15, Tokyo, Japan (Nov 1988).

3. K. Taki, M. Yokota, A. Yamamoto, H. Nishikawa, S. Uchida, H. Nakashima, and A. Mitsuishi, "Hardware design and implementation of the personal sequential inference machine (PSI)," in *Proc. of the International Conference on Fifth Generation Computer Systems 1984*, 398-409, Tokyo, Japan (Nov 1984).

4. H. Sakai, K. Iwata, S. Kamiya, M. Abe, A. Tanaka, S. Shibayama, and K. Murakami, "Design and implementation of the relational database engine," in *Proc. of the International Conference on Fifth Generation Computer Systems 1984*, 419-426, Tokyo, Japan (Nov 1984).

5. T. Yokoi, S. Uchida, and ICOT Third Laboratory, "Sequential Inference Machine: SIM its programming and operating system," in *Proc. of the International Conference on Fifth Generation Computer Systems 1984*, 70-81, Tokyo, Japan (Nov 1984).

6. K. Furukawa and T. Yokoi, "Basic software system," in *Proc. of the International Conference on Fifth Generation Computer Systems 1984*, 37-57, Tokyo, Japan (Nov 1984).

7. S. Uchida, K. Taki, K. Nakajima, A. Goto, and T. Chikayama, "Research and development of the parallel inference machine in the intermediate stage of the FGCS project," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 1, 16-36, Tokyo, Japan (Nov 1988).

8. K. Yoshida and T. Chikayama, "A'UM - A stream-based concurrent object-oriented language," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 2, 638-649, Tokyo, Japan (Nov 1988).

9. T. Chikayama, H. Sato, and T. Miyazaki, "Overview of the parallel inference machine operating system (PIMOS)," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 1, 230-251, Tokyo, Japan (Nov 1988).

10. Y. Takeda et al., "A load balancing mechanism for large scale multiprocessor systems and its implementation," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 3, 978-986, Tokyo, Japan (Nov 1988).

11. S. Uchida and T. Yokoi, "Sequential Inference Machine: SIM progress report," in *Proc. of the International Conference on Fifth Generation Computer Systems 1984*, 58-69, Tokyo, Japan (Nov 1984).

12. A. Goto et al., "Overview of the parallel inference machine architecture (PIM)," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 1, 208-229, Tokyo, Japan (Nov 1988).

13. E. Lusk et al., "The Aurora or-parallel Prolog system," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 3, 819-830, Tokyo, Japan (Nov 1988).

14. U. Baron et al., "The parallel ECRC Prolog system PEPsys: An overview and evaluation results," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 3, 841-850, Tokyo, Japan (Nov 1988).

15. D. Warren and S. Haridi, "Data diffusion machine - A scalable shared virtual memory multiprocessor," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 3, 943-952, Tokyo, Japan (Nov 1988).

16. D. Waltz and C. Stanfill, "Artificial intelligence related research on the Connection Machine," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 3, 1010-1024, Tokyo, Japan (Nov 1988).

17. J.M. Mellor-Crummey, "Experiences with the BBN Butterfly," in *Proc. of 1988 COMPCON*, 101-104, San Francisco, CA, IEEE (Feb 1988).

18. E. Tick, "Compile-time granularity analysis for parallel logic programming languages," in *Proc. of the International Conference on Fifth Generation Computer Systems 1988*, volume 3, 994-1000, Tokyo, Japan (Nov 1988).

John M. Mellor-Crummey received a B.S.E. degree in electrical engineering and computer science from Princeton University, Princeton, NJ, in 1984 and an M.S. degree in computer science from the University of Rochester, Rochester, NY, in 1986. From 1984 to 1986 he was a Sprouli Fellow in the Department of Computer Science, University of Rochester, where he is currently completing a Ph.D. degree. The title of his dissertation is "Debugging and Analysis of Large-Scale Parallel Programs." In the fall of 1989 he will join the Center for Research on Parallel Computation at Rice University, Houston, TX, as a research associate. Mr. Mellor-Crummey's current research interests include programming environments for parallel processing, parallel algorithms, concurrent data structures, multiprocessor operating systems, and parallel computer architectures. He is a member of Tau Beta Pi and Phi Beta Kappa and a student member of the Association for Computing Machinery and the IEEE.

Hervé Touati
Computer Science Division
University of California, Berkeley

INTRODUCTION

Technical sessions at the 1988 International Conference on Fifth Generation Computer Systems (FGCS'88) were divided into four themes: theory, software, architecture, and applications. This article covers the area of computer systems, including the Institute for New Generation Computer Technology's (ICOT) main research project: the *parallel inference machine* (PIM), a large scale multiprocessor whose operating system and application software are based on concurrent logic programming.

This article has two main sections. The first section contains an overview of the Fifth Generation project and a description of its current state of advancement. The second section contains a commented summary of a few other technical communications given at the conference. The Appendix contains a *glossary* to help the reader with the terminology used in published materials related to the Fifth Generation project.

THE FGCS PROJECT

Overview

The Fifth Generation Computer Systems (FGCS) project was informally initiated by the Japanese Ministry of International Trade and Industry (MITI) in 1979. MITI set up a committee to study the implications of FGCS technology and decided to start in April 1982 a 10-year research project involving all the major computer and telecommunications companies of Japan under the control of a national research center: ICOT.

The project was divided into three stages of, respectively, 3, 4, and 3 years. So far, MITI has invested ¥8.3 billion in the first stage and ¥21.5 billion in the second stage of the project (at current exchange rates $68 million and $177 million, respectively).

The ICOT research staff is now composed of 90 to 100 researchers, up from 50 in 1984. Its research staff comes from industry (Fujitsu, Hitachi, Toshiba, NEC, Oki, Mitsubishi, Sony, NTT, KDD) and the national Electrotechnical Laboratory (ETL). Most researchers are sent by their mother companies and stay 3 to 4 years in ICOT; a few are selected directly by ICOT.

Main Purpose

The main purpose of the project is to develop a prototype of a parallel computer system based on logic programming and targeted at artificial intelligence applications.

The project also actively contributes to the education and training of young Japanese researchers in the important research areas it covers. Its high visibility also provides them with early opportunities for direct exposure to the international research community.

The PSI Machine

In the first stage of the project, ICOT designed the PSI machine (a sequential Prolog workstation) and developed its operating system entirely in an object-oriented extension of Prolog. The machine itself was

manufactured by Mitsubishi and Oki and demonstrated at FGCS'84. A faster version based on the Warren Abstract Machine, called the PSI-II, was designed and developed in the second stage of the project. Three hundred PSI workstations are now in operation, mostly in Japan, either in ICOT or in industrial and academic laboratories.

The Multi-PSI Machine

The Multi-PSI machine was the machine demonstrated at FGCS'88. It was developed as a platform for parallel software research. The version demonstrated at FGCS'88 was composed of 64 PSI-II processors connected together by a two-dimensional mesh network. There are plans for an improved version of this machine, under the code name of PIM/m, but it is unlikely to become as important as the other PIM prototypes (its processors are tuned to the execution of Prolog, not a parallel logic programming language).

The PIM

The PIM (parallel inference machine) is the final hardware prototype to be designed by ICOT. A first version with 128 processors, called PIM/p, will be completed by April 1989. The final prototype, which is not expected before the end of the project, should contain on the order of 1,024 processors. Four companies are working on the development of the PIM: Fujitsu, Hitachi, Oki, and Mitsubishi.

The PIM/p Prototype

The PIM/p is composed of 16 clusters of 8 processors, connected by a four-dimensional hypercube network. Each cluster is a tightly coupled, shared-memory multiprocessor. Five 8Ok gate LSI chips implement one processor, including floating point and communication hardware. With a cycle time of 50 ns, one processor will have an average performance of 200 to 500 KLIPS (see the Appendix); the PIM/p itself will have a total aggregate performance of 10 to 20 MLIPS.

The processors are RISC-like, with separate data and instruction caches. The main difference with a RISC architecture is the addition of a writable control store. The processors can execute either simple one-cycle instructions directly or more complex macro instructions from the writable control store. Macro instructions were introduced to reduce memory traffic.

The Final PIM Prototype

Plans for the final PIM prototype are still evolving. One design under investigation is the PIM/c, which adopts a cross-bar network instead of the hypercube network of the PIM/p. Though all the details are not yet completely decided, our impression at this point is that ICOT should be able to attain its final goal in peak hardware performance within 3 years without encountering major difficulties and will concentrate its efforts on software issues during the last stage of the project.

The PIMOS Operating System

PIMOS, the operating system of the PIM, will be entirely written in KL1, a stream-AND-parallel committed-choice language (see the Appendix). Parts of PIMOS are already operational and were demonstrated with the Multi-PSI machine during the conference.

KL1 provides a simple and elegant framework to express concurrency and communication. The implicit dataflow synchronization feature of the language can be used to solve most of the synchronization problems within an operating system. However, the elegance and generality of this framework have their limitations. Low level I/Os are not straightforward to implement efficiently, as KL1 supports directly only fine-grained communication protocols. In addition, KL1 natural communication channels between user processes and the operating system have to be protected by a filtering process to guarantee the integrity of the system.

PIMOS is not yet a complete system. Among the problems left to be solved, load balancing is the most crucial for high performance. In the context of PIMOS, the task of the load balancer is not limited to allocating new jobs to processors: it is also to exploit parallelism within user programs. It is further complicated by the fact that the target machine is composed of a large number of nonequidistant processors.

Distributed memory management, in particular distributed garbage collection, is another important problem that remains to be solved.

Other Research Activities

Other research activities in ICOT include: constraint logic programming and its integration to concurrent logic programming; meta-programming, program transformations, and partial evaluation; natural language understanding, based on Barwise's situation semantics theory; knowledge acquisition, induction, inference, and learning, with currently an emphasis on hypothetical and nonmonotonic reasoning.

REPORT ON OTHER TECHNICAL SESSIONS

This section is an incomplete overview of the technical sessions of the conference. The main emphasis will be on software and computer architecture. We will start with Herbert Simon's skeptical remarks on parallelism and logic programming, followed by the views of Hervé Gallaire and Mehmet Dincbas on constraint logic programming. We will then present a summary of the current research activities within the Gigalips project, followed by a few comments on three proposals for load balancing and parallel scheduling in the context of KL1. Finally, the last section regroups some remarks on other unrelated but interesting talks.

A Skeptical View on Parallelism and Logic

Herbert Simon, from CMU, a world-renowned expert in psychology, economics (Nobel Prize), and computer science, discussed the prospects of cognitive science in his invited talk. He included some criticisms on parallelism and logic that are summarized below.

Parallelism is not a miracle solution to efficiency problems in cognitive science for two reasons: (1) it provides only a very limited answer to combinational explosion and (2) there is no evidence that a genuinely general-purpose massively parallel computer can be built. The brain itself, though clearly parallel at the sensory or motor system level, is a relatively slow, sequential system at the level of conscious activity. Moreover, it is quite possible that large parts of the brain are made of mainly passive memory-like devices. It is more likely that parallelism will

help to simulate low level activities like pattern recognition than high level activities like problem solving.

The idea behind logic programming is that reasoning should be logical, from axioms and inference rules. Ideally, axioms and inference rules are independent from the subject matter and results are valid in all generality. When logic is applied to a particular domain, separate domain-specific axioms are added, and inference rules are kept severely restricted to make rigorous reasoning and verification as clear and simple as possible. But rigor is a heavy price to pay. In contrast, human reasoning uses many different inference rules, not all logical but also domain-specific; it often proceeds by long jumps. The lack of rigor is not a virtue but a necessity to cope with the complexity of the problem at hand. Problem solving is heuristic search; logic for problem solving is a misconception of the basic principles that underlie intelligence.

Herbert Simon's criticism should be taken as a warning that logic alone is not likely to be a fruitful approach to problem solving. In fact, many people in the logic programming community agree with this view. Not only Prolog is often used to program heuristic searches, but also important research activities are being focused on integrating other reasoning paradigms to logic programming, as explained in more detail in the next section.

Constraint Logic Programming

Several constraint logic programming systems have been proposed in the past 2 or 3 years: CHIP, CLP, Prolog-III, Trilogy, to name the most influential ones. It is currently one of the very active research areas within the logic programming community.

Hervé Gallaire [director of the European Community Research Center (ECRC)] stressed the importance of expanding the range of applicability of logic programming. The method he recommends is to incorporate into logic programming multiple reasoning styles; he mentioned constraint logic programming as a successful effort in this direction. He also stressed the importance of tight integration for efficiency and ease of use.

Mehmet Dincbas gave a presentation of CHIP, a constraint logic programming language developed at ECRC. CHIP extends Prolog in three domains: terms restricted to finite domains, boolean terms, and linear rational terms. The main idea is to add to the logic programming framework new computation domains by extending unification to give a semantic interpretation to symbols and to use constraints actively to reduce the search space. CHIP was reported to have been used to solve nontrivial planning, scheduling, or circuit design problems with reasonable efficiency.

Current Activity Within the Gigalips Project

The Gigalips project is an informal research collaboration between the teams of David Warren at Manchester (now at Bristol), Seif Haridi at SICS (Sweden), and Ewing Lusk and Ross Overbeek at Argonne National Laboratories.

David Warren reported the results of their experiments with the Aurora OR-parallel Prolog system. Aurora was implemented as an extension of a fast Prolog implementation and supports OR-parallelism with a sequential overhead of only 25 percent. Aurora Prolog demonstrated speedups ranging from 5.8 to 14 on a

16-processor system. The results are encouraging but not yet competitive with the best sequential implementations, mainly for economical reasons.

David Warren also presented the Data Diffusion Machine, a new design for a fast parallel machine to support the execution of a restricted AND-parallel extension of the Aurora model, called Andorra. The suggested machine architecture consists of clusters of tightly coupled processors connected through a hierarchy of buses in a treelike fashion. Its main characteristic is to have no directly addressable memory but only set associative caches. Virtual addresses are not mapped to a specific memory location; rather, data items *diffuse* from cache to cache on demand.

Load Balancing

Takeda et al. from Mitsubishi and ICOT proposed a semi-automatic load balancing strategy that exploits load information provided by the program. The programmer (or possibly a smart compiler) tries to spread the computation on a virtual square. Initially, this square is subdivided into smaller squares of equal size, which are allocated to processors. During the execution of a program, imbalances are corrected dynamically by shrinking or enlarging the squares (which may become arbitrary quadrilaterals). To do this, one corner of a quadrilateral is moved in a way that compensates the imbalances between the four processors adjacent to that corner. This method has the advantage of being local and is semi-automatic in the sense that it hides the hardware configuration from the programmer. Unfortunately, it requires the intervention of a programmer or a smart static scheduler.

Sugie et al. from Hitachi proposed a fully automatic load balancing strategy for the PIM machine that only attempts to balance the load among PIM clusters (within clusters load balancing is straightforward: one common job queue implemented in shared memory is all that is needed). They compared several load balancing strategies and recommend the use of *random*, which allocates tasks from a loaded cluster to a less loaded cluster chosen at random. Their simulation shows a utilization rate of 70 percent for an overhead in dispatching and communication of 30 percent with 16 clusters. Random and similar strategies have been proposed before in the context of distributed operating systems. Sugie et al. extended their scope of application to the parallel scheduling of KL1 programs on clusters of processors, but their simulations are still too small in scale to be convincing.

Evan Tick looked at the problem of improving parallel scheduling of FGHC programs on shared memory multiprocessors. His main idea is to use a priority scheduler, which schedules first the task of largest estimated granularity. He estimates granularity with a simple, one-pass static analysis of the program. Compared to a simple depth-first scheduler, this technique provided less than 10 percent improvement with an eight-processor machine. The main reasons for this limited improvement are that priority scheduling increases the scheduling overhead and the number of process suspensions for synchronization, while task spawning was cheap on the multiprocessor used in the experiment. Larger benchmarks, better static estimators of granularity, and a faster implementation of the priority scheduler and the suspension mechanism can still improve the performance of this approach.

Nevertheless, these results already indicate that it is not straightforward to improve over simple schedulers.

Miscellaneous Presentations

Dr. Ebcioglu et al. from IBM presented their work on a VLIW (very large instruction word) architecture to execute a new logic programming language called BSL (backtracking specification language). BSL supports backtracking and is a single assignment language like Prolog, but it does not support unification. Preliminary results give a 20-fold speed advantage to hand-compiled BSL over a fast Prolog interpreter (VM/Prolog). Moreover, simulation results promise an additional speedup of 3 from the use of a VLIW architecture over a conventional IBM mainframe.

Doug Lenat, from MCC, presented his 10-year project on machine learning called CYC, which started in the fall of 1984. This project mainly consists of entering manually a large amount of data (on the order of tens of millions of facts, heuristics, representations) into a computer, to test Lenat's working hypothesis that learning can take off rapidly once a machine has accumulated enough knowledge.

John Lloyd centered his presentation around the semantics of meta-programming, currently a popular topic in the logic programming community (meta-programming concerns itself with techniques related to the manipulation of programs by other programs). His main point was that in current meta- programming applications there is confusion between the meta level and the object level logic variables. He recommended instead the use of a *ground* representation of object level variables at the meta level. He acknowledged that efficient implementation of the ground representation and other facilities for meta-level programming would require some effort from implementors, the price to pay for cleaner semantics.

Micha Meier, from ECRC, analyzed *statically* a large number of nontrivial Prolog programs for the purpose of better understanding the compilation of clause indexing for Prolog. His main conclusions are as follows: only 50 percent of procedures are indexable; most of those that are not indexable are either single clause procedures or procedures consisting of a single variable block. Indexing is worth doing only for the first two procedure arguments. In nonunit list blocks, indexing on the first element of the list can be used in 20 percent of the procedures to restrict further the number of matching clauses.

CONCLUSION

The main objective of the Fifth Generation Computer Systems project is to develop a *parallel inference machine*, that is, a multiprocessor specialized in the execution of concurrent logic programs. The final form of ICOT's prototype is still under study, but some features are already emerging: the use of powerful sequential processors, currently in the performance range of 200 to 500 KLIPS; the use of a simplified, RISC-like architecture to implement them; the use of clusters of tightly coupled processors as building blocks. It is likely that ICOT will reach its peak performance goal in hardware by the end of the project with a 1,024-processor prototype.

Most of the remaining pitfalls lie in software, in particular the issue of load balancing and efficient parallel execution of programs, i.e., the issue of efficient utilization of the hardware resources of the multiprocessor. ICOT is betting on concurrent logic programming to help in this process. It will be quite interesting to see whether the use of a large multiprocessor and an efficient parallelization scheme for a relatively slow, high-level programming language can lead to a competitive approach to symbolic computing. The next FGCS conference should provide at least parts of the answer.

Hervé Touati received a master's degree in mathematics from Ecole Normale Supérieure, Paris, in 1981. He is currently working towards a Ph.D. degree in computer science at the University of California, Berkeley. He was the first researcher sent by the French National Research Institute of Automation and Computer Science to ICOT, where he spent 4 months in 1985. His main research interests are logic programming, performance analysis of software systems, and IC logic synthesis.

Appendix

GLOSSARY OF FGCS TERMS

- FGCS: Fifth Generation Computer System project.
- FGHC: Flat GHC. A simplified version of GHC that keeps most of its expressive power.
- GHC: Guarded Horn Clauses. A stream-AND-parallel committed-choice concurrent logic programming language developed by ICOT.
- ICOT: Institute for New Generation Computer Technology. It was created by the Japanese Government in April 1982 to lead the FGCS effort.
- KL1: Kernel Language 1. KL1 is the generic code name for low level system or application languages designed for ICOT's multiprocessors. Among others, KL1-b is an abstract machine instruction set; KL1-c is an extended version of FGHC that supports operating system primitives.
- LIPS: Performance unit, corresponding to the number of logical inference per second. There is no full agreement on the definition. One general (but imprecise) definition is the number of procedure calls executed per second. A workstation of the Sun 3/160 class with a good Prolog compiler can achieve 80 KLIPS (80,000 LIPS).
- Multi-PSI: The first multiprocessor developed by ICOT. The current version is composed of 64 PSI-II processors connected by a two-dimensional mesh network. It was demonstrated at the FGCS'88 conference.
- PIM: The parallel inference machine. A prototype with 128 processors should be running by April 1989. The final hardware goal of the project is a prototype with an order of magnitude as many processors.
- PIMOS: The operating system for the PIM machine, written in KL1-c.
- PSI: Personal sequential machine. It was developed by ICOT and its industrial partners during the first stage of the project. It achieved a performance of 30 KLIPS.
- PSI-II: An improved version of the PSI developed during the second stage of the project. It achieved an order of magnitude improvement in speed over the PSI.

SUPERCOMPUTER USER ENVIRONMENT IN JAPAN

H. Yoshihara

The supercomputer environment for users in Japanese universities, government laboratories, and aerospace companies is reviewed. Compared to the United States, the number of users per supercomputer and computer costs in Japan are lower by at least an order of magnitude. Such a conducive environment is a key ingredient in the development of computational fluid dynamics in Japan.

INTRODUCTION

Progress in computational fluid dynamics (CFD) in Japan is proceeding at a rapid pace, and the availability of supercomputers has been an important factor. In the following, accessibility of supercomputers for researchers in representative universities, government laboratories, and aerospace companies is described.

SUPERCOMPUTERS IN UNIVERSITIES

Similar to the U.S. National Science Foundation (NSF) Supercomputer Network, the Ministry of Education, Science, and Culture (Monbusho) has established a network of supercomputers at the seven former Imperial Universities. The universities and supercomputers are the Universities of Tokyo (Hitachi S-820/80), Hokkaido/Sapporo (Hitachi S-820/80), Tohoku/Sendai (NEC SX-2), Nagoya (Fujitsu VP-200), Kyoto (Fujitsu VP-400E & VP-200), Osaka (NEC SX-2), and Kyushu/Fukuoka (Fujitsu VP-200) (see Figure 1). The speed and memory of the supercomputers are as follows:

Computer	Speed (GFLOPS)	Memory (MB)
Fujitsu VP-200	0.533	64
VP-400	1.14	256
VP-400E	1.7	1,024
Hitachi S-820/80	3.0	512
NEC SX-2	1.3	256

All of the computers installed in the NSF computing centers are single-CPU computers in contrast to the multi-CPU computers, as the four-CPU Cray X-MP and Cray 2. In a centralized computing center, absence of a multiple-CPU system will reduce system flexibility, leading to reduced throughput.

Supercomputers are leased by universities usually at a substantial discount, typically 80 percent. Leasing, in contrast to buying, permits, in principle, easier updating of the equipment as improved models are offered. This has not occurred in a timely fashion in the Japanese university network. In the U.S. National Aerodynamic Simulator (NAS) at the NASA-Ames Research Center, supercomputers are updated almost routinely as new generations are offered. Thus, for example, the Cray X-MP and ETA 205 were replaced in turn by the Cray 2 and Cray Y-MP. From the user's point of view, these changes have not been without disruptions with programming re-tuning required in the Cray X-MP/Cray 2 and Cray 2/Cray Y-MP transitions.

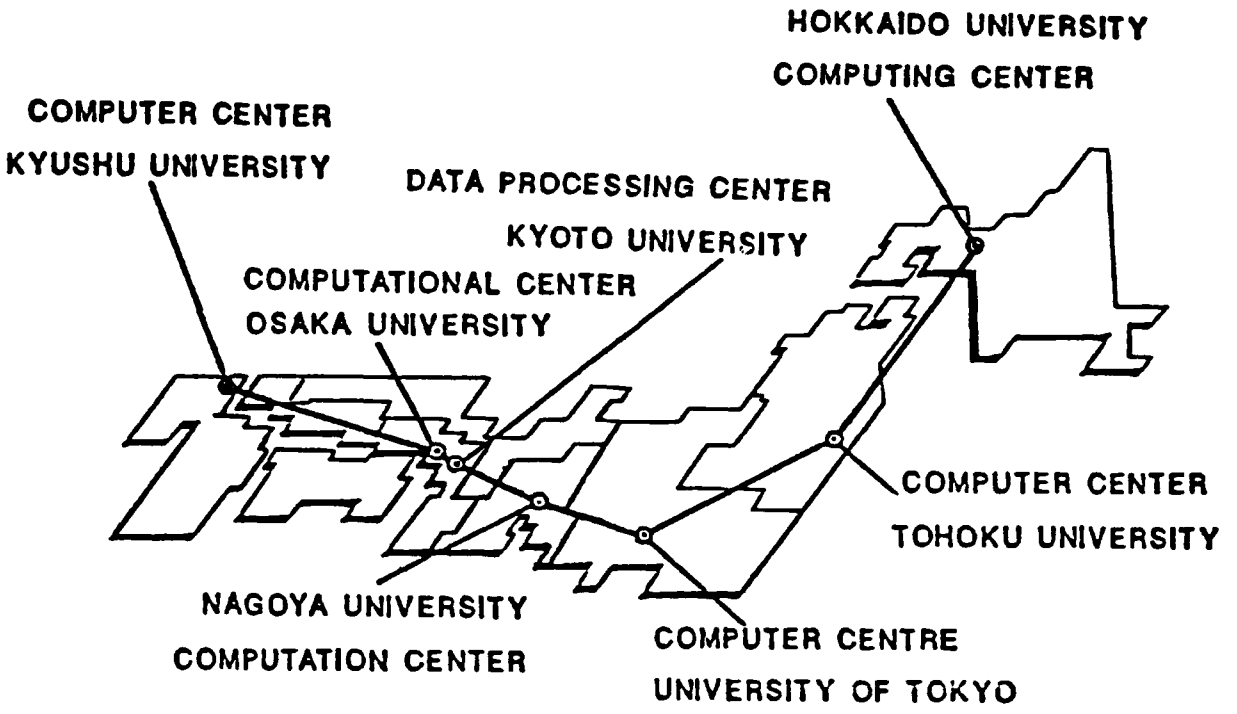


Figure 1. University supercomputer system (N1 network).

The above university computing centers are connected via the N1 (interuniversity) network, which additionally connects 42 other mainframe computers at other universities. The ETA 10 computer presently in the checkout phase at the Tokyo Institute of Technology is a notable example. With the demise of ETA Inc. the future of this computer is uncertain. A dedicated digital network (DDX) operated by the National Telephone & Telegraph Company (NTT) connects these computers with a transmission rate of 48 Kbps (bps = bits per second). To date, computations have, however, been largely carried out locally with little remote processing using the network. This, however, is rapidly changing as local centers are inevitably becoming saturated.

Universities assigned to a given computing center are connected to the center supercomputer through a local area network (LAN), an ethernet (coaxial cables) with a transmission rate of 10 Mbps. Thus, for example, 49 universities are connected to the Computing Centre of the University of Tokyo, while 12 universities are serviced by the Data Processing Center of Kyoto University.

Use of the university computers is restricted to basic research with users confined to graduate students and faculty. (Undergraduates use smaller computers.) Researchers in government laboratories and aerospace companies are essentially precluded from using the university supercomputers, though there are informal arrangements between industry (nonmilitary) and university researchers in which university computers are used.

The charge per hour occupancy at all centers is ¥10,000/h (about $80), a trivial amount relative to U.S. costs. For researchers carrying out independent research, however, this charge is prohibitive since there is little, if any, computing budget within an academic department. Supercomputer usage is primarily confined to researchers with contracts. Several senior professors at the University of Tokyo expressed frustration over this situation. In reflection, the situation in the United States is perhaps not dissimilar. Many researchers, including the author, have had to seek "free" computing time at various Department of Defense and NASA computing centers to carry out large independent computations.

Computing at the University of Tokyo Computing Centre

To obtain a perspective of the operation of the computing centers, some statistics are given, first for the University of Tokyo Computing Centre in this section and then for the University of Kyoto Data Processing Center in the next section.

The computer system within the University of Tokyo Computing Centre is shown in Figure 2).* There are 311 direct-line terminals, 2,025 phone terminals, and 36 remote job entry (RJE) stations. There are 10 work stations that include the Sun 3/260C (8 MB memory; 280 MB disk), Sony NEWS (4 MB memory; 80 MB disk), MicroVAX Station II (2 MB memory; 31 MB disk), and MicroVAX AI (4 MB memory; 71 MB disk), all UNIX-based. (Contradicting numbers in Figure 2 are only for the Centre itself.)

In 1986 the number of users totaled 6,500 with the following breakdown:

- By organization: University of Tokyo 55%; others 45%
- By position: faculty 64%; graduate students 33%
- By department: science 40%; engineering 52%; others 8%

The center processed between 4,000 and 7,300 jobs per day with a total of 1.2 million jobs in the 1986 academic year; 70 percent were time-sharing (TSS) jobs with batch jobs accounting for about 80 percent of the CPU time. The hours for TSS job entries are 0930 to 2300 during the week and 0930 to 1800 on Saturday. The University of Tokyo Computing Centre is staffed by about 50 persons, 29 being technical and 21 administrative.

The University of Kyoto Data Processing Center

The supercomputers in this center are the Fujitsu VP-200 and VP-400E. There are about 1,000 remote terminals connecting into these supercomputers that are located within the University of Kyoto or in surrounding universities such as the Kyoto Institute of Technology, where extensive CFD calculations are undertaken by Professor N. Satofuka and his staff. Interactive terminals are connected with either 300- or 1,200-bps phone lines or 4,800- and 9,600-bps digital PBX lines.

---

*"Computing Centre, University of Tokyo," brochure of the Computing Centre (February 1988).

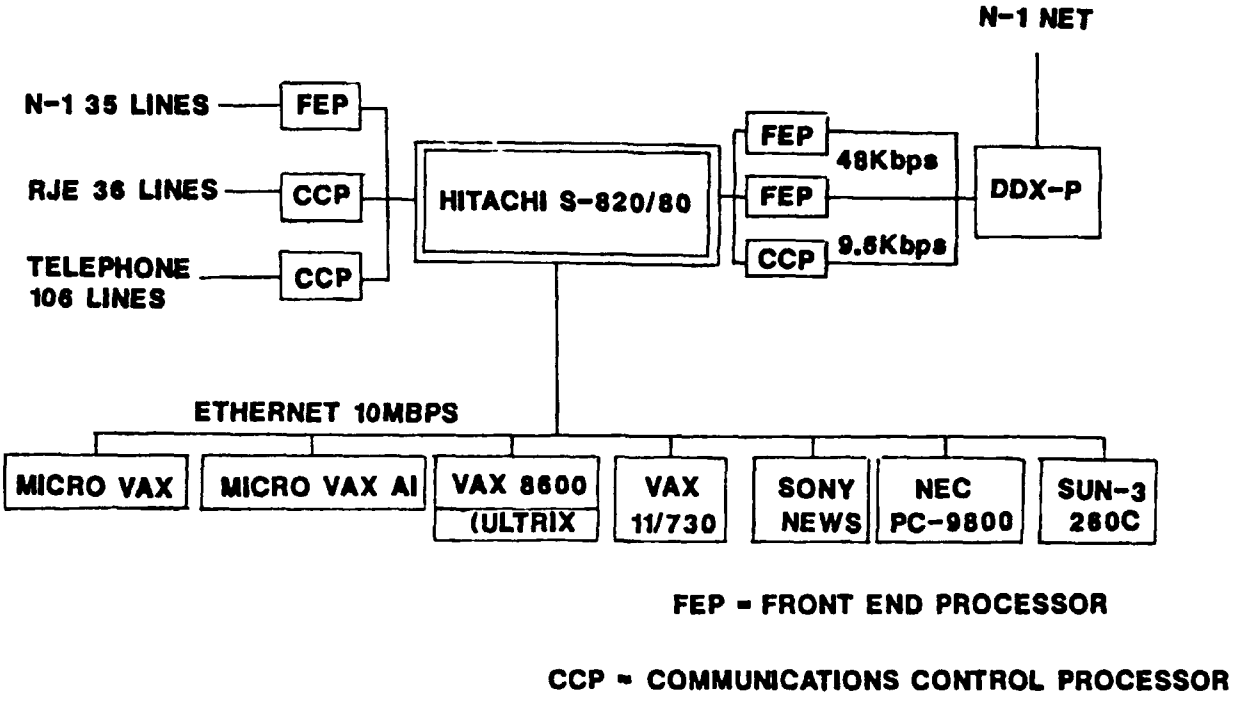


Figure 2. Computer network at the University of Tokyo.

In 1987 there were about 3,600 persons using the computers at the Kyoto Center with an excess of one million jobs for the year. The computing center is typically open from 0920 to 1830 for onsite users and from 0920 to 2200 for offsite users. There was apparently some user dissatisfaction over these hours. The Kyoto Data Processing Center is staffed by 19 computer engineers and 23 administrative personnel.

The just-announced Fujitsu VP-2600 computer was to be installed at the University of Kyoto Center in January 1990, but this has been delayed 1 year to January 1991. The VP-2600 is a single-CPU computer with a speed of 4 GFLOPS and a main memory of 2.048 GB. With the delay the "soon-to-be-announced" four-CPU version could be substituted.

Institute for Space and Astronautical Sciences (ISAS)

ISAS is a Monbusho laboratory located in Sagamihara, an hour's train ride east of Tokyo. Its main supercomputer is a leased Fujitsu VP-200, which has a memory that is inadequate for CFD calculations. Acquisition of an updated computer is presently in progress. Recommendations for the new computer have been made to the Executive Committee of ISAS, which then selects the computer to be leased. Cost of the VP-200 is a token ¥1,800/h (about $15) for the first 4 hours and free thereafter up to a maximum of 10 hours. On several visits to ISAS, the computer appeared to be readily available during prime time. There probably is unlimited "free" use of this computer.

Professor K. Fujii of the High Speed Aerodynamics Section recently installed a Stellar (U.S.) work station that has a parallel CPU with a speed of 40 MFLOPS and a memory of 128 MB. Its response time for complex CFD graphics and for inspection of complex meshes is typically one-quarter to one-third of that of the IRIS work station installed, for example, at NASA-Ames and the Boeing Company.

COMPUTING CENTER OF THE NATIONAL AEROSPACE LABORATORY (NAL)

NAL is in the Science and Technology Agency and is located in Chofu City in the eastern suburb of Tokyo. It functions much like NASA in the United States, though on a much smaller scale. It contains the largest CFD group in Japan, which may number 15 to 20 senior researchers, which is relatively small in comparison to U.S. organizations, for example, that of the Boeing Company, which has a CFD group of 50 to 60 persons.

The NAL Numerical Simulator System* is centered about the Fujitsu VP-200 and VP-400E. The latter has a very large memory of 1.024 Gbytes, adequate for the computation of complete aerospace configurations. NAL does not have "quick response" work stations for graphics. Use of supercomputers for inhouse researchers is formally allocated by division, but in fact each researcher has had unlimited use of the computers. During 1987 the average CPU hours per month on the VP-200 was about 300 hours while for the VP-400E it was about 400 hours per month.

COMPUTING IN THE AEROSPACE INDUSTRY

At present there are no supercomputers at the two largest aerospace companies, Mitsubishi Heavy Industries (MHI) and Kawasaki Heavy Industries (KHI), both located in the Kinki region surrounding Nagoya. However, it was rumored that MHI was in the process of purchasing one of the new supercomputers, perhaps in connection with the recently awarded major contract to develop the FSX support fighter with General Dynamics. MHI engineers use the NAL VP-400E, either without charge in a cooperative project with NAL or by rental at ¥17,000/h (about $140). There is a single dedicated fiber-optic digital line between NAL and MHI with a transmission rate of 60 Kbps which, according to CFD users, is completely inadequate.

KHI has a Fujitsu VP-50 (a 100-MFLOPS class computer) and recently acquired a Titan (Ardent) work station (64 MFLOPS CPU; 32 MB memory). KHI engineers access the NAL supercomputers via a leased 9.6-Kbps line. In addition KHI researchers frequently travel to NAL to use the computers interactively, for example, to generate the mesh for complex configurations as the ASKA STOL transport and the space shuttle HOPE for Navier/Stokes calculations.

CONCLUSIONS

Supercomputers in Japan are readily accessible to most users in universities, government laboratories, and aerospace companies even in prime time. This is in

*Miyoshi, H., and M. Fukuda, "On the NAL numerical simulator system," Report SP-8 (National Aerospace Laboratory, November 1987).

contrast to the situation in the United States, where the number of users per supercomputer and computer occupancy charges are larger by at least an order of magnitude. As a result, research, for example, to reduce the computing time for the extremely slowly converging Navier/Stokes codes has near-zero priority in Japan, quite in contrast to the United States. Though the number of computational fluid dynamicists is on the increase in Japan, growth of supercomputer power is such that the present enviable position of the Japanese supercomputer user should persist for years to come.

Hideo Yoshihara arrived in Tokyo in April 1988 for a 2-year assignment as a liaison scientist for the Office of Naval Research. His assignment is to follow the progress of advanced supercomputers and to review and assess the viscous flow simulation research in the Far East. Dr. Yoshihara formerly was with the Boeing Company, where he was Engineering Manager for Applied Computational Aerodynamics. He was also an affiliate professor in the Department of Aeronautics and Astronautics of the University of Washington, an AIAA Fellow, and a former member of the Fluid Dynamics Panel of AGARD/NATO.

THE POHANG IRON AND STEEL COMPANY: ITS RESEARCH INSTITUTE AND TECHNICAL UNIVERSITY IN SOUTH KOREA

Fred Pettit

Pohang Iron and Steel Co. Ltd. (POSCO) has established over the past 3 years the Research Institute for Industrial Science and Technology (RIST) and the Pohang Institute of Science and Technology (POSTECH). RIST is to develop advanced iron and steel technologies and new technologies that will help POSCO diversify. POSTECH is a private coeducational research university whose goal is to become the premier technical university in Korea. The progress that has been made in establishing these two institutions is described.

INTRODUCTION

Ground was broken for Pohang Iron and Steel Co. Ltd. (POSCO) in 1968 in the southeastern port city of Pohang, South Korea. In May 1983 the annual production capacity at this location was 9.1 million tons. The facilities include four blast furnaces, one foundry blast furnace, five coke plants, and five sinter plants as ironmaking facilities; two steelmaking plants and three continuous casting plants as steelmaking facilities; and two hot strip mills, two cold rolling mills, two plate mills, two wire rod mills, and one silicon steel mill as rolling facilities. In March 1985 work commenced on construction of the Kwangyang steelworks on the coast in South Cholla Province about 200 kilometers southwest from Pohang. The Kwangyang steelworks now has an annual capacity of 5.4 million tons, which gives POSCO an annual capacity of about 15 million tons. Moreover, the Kwangyang steelworks has the most up-to-date, state-of-the-art steelmaking facilities in the world, including continuous casting facilities.

With the successful operation of the Pohang and Kwangyang steelworks, POSCO began to become more concerned with research and the supply of properly educated engineers and scientists for the steelworks and especially the research laboratories. In December 1986 the Pohang Institute of Science and Technology (POSTECH) was founded by POSCO. It is a private coeducational university in science and engineering with a heavy emphasis on research. Furthermore, in March 1987 POSCO formed the Research Institute of Industrial Science and Technology (RIST) to develop advanced iron and steel technologies, to diversify POSCO, and to develop new, valuable technologies.

This article will describe RIST and POSTECH, the relationship between them, and their functions in regards to POSCO.

POSCO

The construction of the steelworks at Pohang was planned and directed by Japanese steel experts from Nippon Steel. By 1973 these steelworks were operated totally by Koreans. Furthermore, the subsequent expansions at Pohang and Kwangyang were carried out completely by Koreans. The products of POSCO are iron and steel.

Business and profits have been good and POSCO has been looking for ways to improve its products and to diversify. Furthermore, Pohang is about 350 kilometers southeast of Seoul and it was difficult to attract researchers and scientists to work there because of the lack of cultural and social amenities. Consequently it was decided to establish a research institute and a high quality university in Pohang. The university was to provide excellent teaching, perform high quality research, and develop scientists and engineers of the highest caliber for POSCO's and Korea's future development. The research institute was to interface between POSCO and POSTECH, in particular, to introduce the technical problems of POSCO to the POSTECH faculty and to transfer to POSCO and other Korean industries the new technology innovated by POSTECH.

Since Pohang is in a rural area, POSCO has tried to provide comfortable housing and cultural facilities. Complexes including privately owned homes, apartments, and dormitories now exist, forming a small city. Various facilities are also available for cultural events and for other forms of recreation, including a concert hall, gymnasium, and sports stadium. Medical facilities as well as outstanding educational facilities from kindergarten to college have been established.

RIST

The organization of RIST is described in Figure 1. It consists of four technical divisions. RIST currently has about 780 people, with 465 researchers, 210 technicians, and 105 administrative staff. The research staff includes 239 with Ph.D. degrees, of which 115 are adjunct researchers from the POSTECH. By 1995 the number of personnel is expected to reach 1,050, with 650 researchers; of these researchers, 425 will have Ph.D. degrees, including 200 adjunct researchers from POSTECH.

The POSTECH faculty as well as graduate students work cooperatively with RIST researchers. RIST and POSTECH are located in the same complex of buildings. Thus it is very convenient for students at POSTECH to do research at RIST. Nevertheless, the buildings for RIST and POSTECH are separate and both organizations have clearly defined geographical boundaries.

RIST has been in operation less than 3 years, and all of the divisions are still in the process of acquiring new researchers. Therefore, while research topics to be emphasized have been defined, results presented via publications are few. The Iron and Steel Division is the most advanced since this division was in existence prior to the formation of RIST. It was part of the POSCO Technical Research Laboratories and became a division of RIST upon RIST's inception.

Iron and Steel Division

The Iron and Steel Division focuses on developing new technologies in iron and steelmaking as well as advancing conventional technologies in manufacturing processes, steel products, energy, and factory automation. In the case of steelmaking, research is being performed on new casting techniques such as strip casting, horizontal continuous casting, and rheocasting. Research is also being performed on processing of raw materials, quality control, and analysis of gas-solid-liquid reaction systems. Special emphasis is placed upon the development of the smelting-reduction ironmaking process as a means to replace the blast furnace.

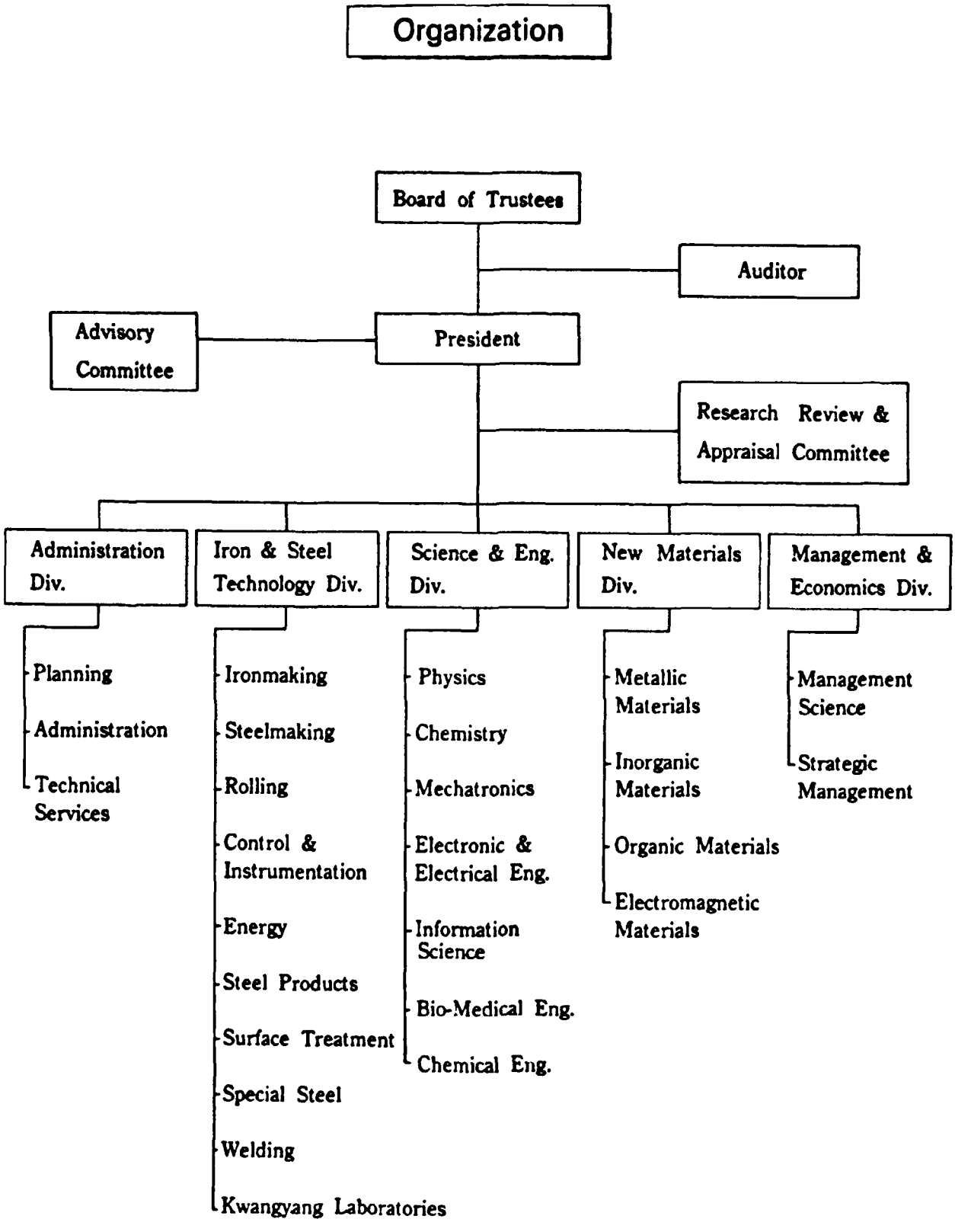


Figure 1. Organization of the Research Institute of Industrial Science and Technology (RIST).

A significant amount of research is being performed on the rolling of steels. Kwon and coworkers (Ref 1) have studied the effects of composition and hot rolling conditions on the mechanical properties of low carbon bainitic steels. As carbon concentration was increased up to 0.05 wt. %, strength was increased but elongation and toughness were decreased. Increases in carbon beyond 0.05 wt. % did not significantly affect the mechanical properties. The addition of 0.3 wt. % Mo produced an increase in strength without any decrease of low temperature toughness. In contrast, the addition of 0.5 wt. % Cu had little influence on strength but significantly improved impact properties. The combined addition of Mo with Cu or Ni resulted in an improvement in both strength and toughness. Reheating temperature and finish rolling temperature had little influence on strength; however, toughness was slightly improved by using lower temperatures. A decrease in coiling temperature did not affect strength, but a significant improvement in low temperature toughness occurred as coiling temperature was lowered. These effects were explained in view of the observed microstructural refinement and the formation of ultra-fine polygonal ferrites.

Kim and Kwon (Ref 2) have studied the formation of abnormally coarse grain structure in hot-rolled steel strips. For steels deformed in the ferrite-austenite two-phase region, abnormal grain growth occurred by the growth of strain-free, transformed ferrite into the surrounding deformed matrix. For steels deformed in the ferrite region, however, the coarse grain structure was proposed to develop by the preferential growth of certain grains following extensive recovery.

Control and instrumentation, cokemaking, fuel combustion, waste heat recovery, surface treatments, high strength alloys, specialty steels, and weldability of steels are also topics being investigated in this division. Paek et al. (Ref 3) have developed an automatic hot slab surface inspection system using a laser scanner with a photo multiplier tube and a microcomputer. Longitudinal cracks of 5 mm width on test slabs were detected with good reliability. Lee et al. (Ref 4) have developed a mathematical model to estimate the temperature profiles in the slab mold under various operating conditions. Of the variables examined, water velocity, mold thickness, and scale deposition had strong effects on the mold temperature distribution, but the water inlet temperature and casting speed had negligible effects.

Science and Engineering Division

Research and development in the Science and Engineering Division is directed at achieving technical innovations in physics, chemistry, mechatronics, information science, biomedical engineering, and chemical engineering. The activities in physics are focused currently upon optics, lasers, high T_c superconductors, and ultra high vacuum. Research in chemistry involves process development for chemicals and pharmaceutical intermediates along with drug development. In the mechatronics section research is involved with robotics, computer-aided design/computer-aided manufacturing (CAD/CAM), factory automation, fluid flow, and heat transfer. The information science research is concerned with the principles of computers and their applications. Special emphasis is placed upon the development

of a parallel computer and the implementation of expert systems. In the biomedical engineering area research is currently emphasizing the design and manufacture of artificial joints using CAD/CAM. The technical areas being studied in the Chemical Engineering Section include the development of new technologies in fine chemicals, advanced catalytic materials, and processes related to polymeric materials.

Jeong et al. (Ref 4) are studying a partitionable, parallel processing system being designed to support 64 or more transputers. A reconfigurable interconnection switch controlled by software provides great flexibility in selecting any interconnection topology dynamically in the program.

New Materials Division

The New Materials Division is concerned with developing improved engineering materials by using advanced processing techniques. This division has sections investigating metallic, inorganic, organic, and electromagnetic materials.

The Metallic Materials Section can fabricate metallic alloys using a variety of techniques including rapid solidification, alloy powder fabrication and consolidation, squeeze casting, and superplastic forming and shaping. Special consideration is being given to materials for aerospace applications.

Kim and Suh (Ref 3) have developed a mathematical model to describe time dependent pressure, relative density, and temperature relations of metal powders during hot compaction.

The Inorganic Materials Section has projects on high temperature structural ceramics for the steel industries, synthesis of high purity fine powders, wear and heat resistant ceramics, composites, as well as electronic and superconducting ceramic materials. Kim et al. (Ref 4) have fabricated alumina-10 vol % SiC whisker composites by pressureless sintering at 1,750 °C. Ultrasonic dispersion and ball milling were beneficial to sintering. The addition of a liquid sintering aid increased density. To obtain densities greater than 90 percent, submicron alumina was essential. Jeong et al. (Ref 3) have studied the bond geometry of an oxygen-silicon complex in an oxide film on a Si (100) surface by using high resolution electron energy loss spectroscopy. The observed values of vibrational energies of the four normal modes were in good agreement with the calculated values. A bond length of 3.00 Å and a bond angle of 103° were obtained by using a continuous random network model. These values indicate that the bond geometry in the oxide layer is quite similar to that of chemisorbed oxygen at high coverage.

The Organic Materials Section is involved with research on carbon fibers, polymeric materials, and composites. Special consideration is being given to composite materials for the aerospace and automobile industries. Park and coworkers (Ref 5-7) are investigating the fabrication of carbon fibers from pitch to utilize byproducts from POSCO's coking operations. Carbon fibers are also being made from polyacrylonitrile (PAN). This section is attempting to obtain fibers with improved properties as well as fabricating components from densified carbon. Kim (Ref 8, 9) is studying the thermal behavior and morphology of polymer blends by using differential scanning calorimetry and scanning electron microscopy (SEM).

The Electromagnetic Materials Section is attempting to develop magnet materials such as ferrites and Nd-based permanent magnets. Emphasis is placed on chemical vapor deposition processing and magnetic recording materials.

Management and Economics Division

The Management and Economics Division attempts to assist managers in decisionmaking. The topics being investigated in the area of management science are productivity management, material handling analysis, and quality control. The strategic management studies include human resource management and industrial labor relations, marketing, finance and accounting, and management information systems. The economics research covers demand forecasting, economic trend analysis, and economic feasibility studies.

Research Support

The physical plant and the experimental facilities at RIST are outstanding. The various divisions at RIST are housed in three interconnected buildings that are new and spacious. The environment is pleasant. A great variety of the very latest equipment is available as shown in Table 1. An excellent library is available with copies of virtually all of the important technical journals and periodicals.

POSTECH

The Pohang Institute of Science and Technology is a research-oriented university. This institute was described in a previous *Scientific Bulletin* article about 2-1/2 years ago (Ref 10). Its goal is to be the premier technical university of Korea. Currently undergraduate and graduate programs offering B.S., M.S., and Ph.D. degrees are available in the following 10 departments:

- Chemistry
- Life Sciences (graduate program to be initiated in March 1990)
- Mathematics
- Physics
- Chemical Engineering
- Computer Science
- Electronic and Electrical Engineering
- Industrial Engineering
- Materials Science and Engineering
- Mechanical Engineering

An Economics Department is to be established in about 1992.

The first undergraduate class of 249 freshmen matriculated on March 5, 1987, with 80 faculty members present. The average college board examination score of those admitted was 300.6 out of a possible 340, which was the highest overall average of all Korean colleges and universities. It is POSTECH's plan to accept only students in the top 2 percent in Korea. The graduate program was inaugurated in March 1988. As of October 1988 the numbers of undergraduate and graduate students were 490 and 110, respectively, with 140 faculty members. Currently there are about 250 graduate students with 50 students in the Ph.D. programs. It is planned to have an enrollment of 1,200 undergraduate and 1,000 graduate students with a faculty of 300 by 1995.

Table 1. Some of the Research Equipment Available at RIST and POSTECH

FT-NMR (Bruker 300 MHz)
Mass Spectrometer (Kratos 25-RFA)
FT-IR (Bomem DA 3.26)
IR and UV/Visible Spectrophotometers
Laser-Raman Spectrometer (Spex Ramalog-101)
Tunable YAG-Dye Laser (Molectron)
X-ray Diffractometer
Low Energy Ion Scattering Spectrometer
DNA Synthesizer and Automatical DNA Sequences
Peptide Synthesizer and Peptide Sequencing System
Diode Array Spectrophotometer
High Performance Liquid Chromatography System
Liquid Scintillation Counter and Gamma Counter
Image Analyzer with Laser-Aided Confocal Microscope
Computer Vision Laboratory
Helium Liquefier (KPS-1410)
Low Energy Electron Diffraction
ESCA, SIMS, TEM, STEM, SEM, SAM EELS/LEED, EPMA
Mossbauer Spectroscopy
Electron Spin Resonance Spectroscopy
Particle Size Analyzer
Atomic Absorption Apparatus
Chemisorption Apparatus
Low Shear Rheometer
Raman Spectroscopy
Plasma Etcher
Universal Testing Machines
Optical Microscopes
Capillary Rheometer
Hot Presses
Injection Molding Machine
Vax 880 (YMS)
Vax 810 (UL TRIX)
IBM 4381 (VM/VSE)
SUN/APOLLO/HP workstation
IBM PCs
Clean Room
Acoustic Emission Equipment
MTS
Fatigue Tester
Creep Machines
Various Engineering Pilot Plants
Wind Tunnel (under construction)
Three Component Laser Doppler Velocimeter
2 GeV Synchrotron Radiation Source (under construction)

The facilities at POSTECH are excellent. The buildings are new and well maintained. Ample space is available for classroom instruction including laboratory courses. The library is computerized for cataloguing, search, and circulation. A computer center and audio-visual facilities are available. POSTECH is one of the better equipped universities in the Pacific Basin region (see Table 1).

All undergraduates receive financial assistance unless their grade point average is below 2.0 out of a 4.3 maximum. About one-third of the undergraduate students are free from all tuition and fees except meals. The other students pay only one-third of tuition and fees. The rooms for POSTECH students are free. Most graduate students are paid stipends as teaching or research assistants that cover tuition, other fees, and living expenses. Excellent housing facilities are available for single and married students as well as faculty. POSTECH has agreements with six universities in the United States, United Kingdom, Germany, and France for student and faculty exchanges.

POSTECH has not been in existence long enough for publications to be available resulting from various research programs. Eighty-five percent of the faculty were educated in the United States, with 9 percent from Korean universities and 6 percent educated in other countries. The faculty is relatively young, with 80 percent having received Ph.D. degrees in 1980 or later. The major areas of study and research in the 10 departments of POSTECH are as follows:

Chemistry Department

- Bio-organic and medical chemistry related to new drugs
- Development of new synthetic organic and organometallic technologies
- Experimental physical chemistry concerned with chemical kinetics, laser-induced reactions, surface and catalytic science, and polymers
- Theoretical and computational chemistry investigating kinetics, structures of condensed matter, reaction mechanisms, and molecular design
- Electrochemistry applied to conducting polymers, metal and semiconductor corrosion, and semiconductor/electrolyte interfaces
- Chemical instrumentation for sensor development and computer-automated instrumentation using artificial intelligence
- Practical applications of vibrational spectroscopy techniques

Life Sciences Department

- Cellular and molecular biology
- Biochemistry and protein engineering
- Plant molecular genetics and biochemistry
- Virology
- Neurobiology
- Human genetics
- Microbiology

Mathematics Department

- Pure mathematics
  - Analysis--Harmonic analysis, singular integral operators for partial differential equations, infinite holomorphy in functional analysis and several complex variables

- Algebra--Evaluation of zeta functions, elliptic curves and modular forms in algebraic number theory, ideal structure of domains such as valuation, Pruefer, and Krull rings in commutative algebra, and studies on the family of algebraic curves in algebraic geometry

- Geometry/topology--Dynamics on Lorentz spaces, parallelism of manifolds, and immersions and imbeddings of differentiable manifolds

- Applied mathematics

  - Partial differential equations and mathematical physics

  - Fluid dynamics--Measure valued solutions of Euler equations, zero viscosity limit of the statistical solutions of the Navier-Stokes equations

- Computational mathematics

  - Numerical analysis--Numerical solutions for ordinary and partial differential equations, large-scale scientific computing, modeling in mathematical biology, numerical models of fluid motion in a blast furnace, mathematical programming, and related branches of analysis

  - Mathematics for computer vision--Application of differential geometry, topology, and catastrophe theory to 3D object recognition; industrial visual inspection; neural networks modelled after the human brain; integral transform for image reconstruction; and computational geometry

Physics Department

- Accelerator and plasma physics research directed at various types of accelerators as well as beam dynamics and instabilities; plasma diagnostics

- Condensed matter experiments--Amorphous materials, low temperature physics, high T_c superconductors, and surface physics

- Computational physics--Monte Carlo simulation of statistical systems and development of numerical algorithms for parallel computers

- Theoretical physics--Many body theories and their application to condensed matter physics; research on properties of solids by using electronic band structure theory, many body theory, and computer simulations; and phase transitions and transport processes occurring in condensed matter

- High energy physics--Cosmology, astrophysics, lattice quantum chromodynamics and superstring theory

Chemical Engineering

- Catalysis and reaction engineering--Chemistry (utilization of carbon monoxide and hydrogen, especially derived from steelmaking processes to generate products with higher carbon numbers); environmental protection by catalytic abatement of carbon monoxide, hydrocarbons, SO_x, and NO_x; selective oxidation to produce specialty chemicals; polymerization catalysis; development of novel catalytic

materials, electronic materials, and ceramics; physicochemical studies of catalysts, their interaction with reactant molecules, and the nature of elementary steps occurring on the catalyst surfaces

- Polymers--Rubber toughening of glassy polymers, fiber-reinforced plastic, polymer application to semiconductors, development of submicron resists, synthesis of polyimides, injection molding of optical disks, study of birefringence patterns by rheo-optics, development of toughened engineering plastics using interfacial agents, nylon-based polymer alloys, polymer-polymer adhesion, structures and properties of liquid crystalline polymers, and phase separation kinetics of polymer blends

- Advanced materials--Chemical vapor deposition for silicon-integrated circuit metallization, metal organic chemical vapor deposition for compound semiconductors, glow discharge plasma chemical processes for deposition and dry etching, plasma diagnostics and reactor design, gas phase synthesis of ceramic materials, thermal plasma processes for fine powders and inorganic materials processing, and thermal plasma diagnostics

- Biotechnology--Recombination of DNA, hybridization of animal and plant cells, design and control of bioreactors, transport phenomena inside living organisms, new separation techniques of bioproducts, and biomass conversion techniques

- Control and optimization--Advanced process control, optimal control, computer-aided process control, computer-aided process design, and process software development

- Energy and environment--Chemical, physical, and biological changes in the environment through contamination or modification

- Chemical engineering fundamentals--Dynamic behavior of free surfaces; transport phenomena in multiphase flow; separation processes such as affinity chromatography, membrane separation, supercritical fluid extraction, etc.; deterministic and/or stochastic simulations; and statistical (molecular) thermodynamics

Computer Engineering Department

- Computer systems--The POPA (POSTECH Parallel) machine project; application-specific, parallel computer-like database machine to perform image processing and scientific calculations; parallel algorithms; parallel language; operating systems; topology; protocol engineering; computer networks; and distributed systems

- Artificial intelligence (AI)--Expert systems, shells and AI machine for symbolic processing, image analysis neural networks, and parallel processing of pattern recognition

- Computational theory--Computational geometry and parallel algorithms

Electronic and Electrical Engineering Department

- Communications and signal processing--Information, communication, and signal processing; information and coding theory; communication and queuing networks and optical communications; signal detection, processing, and estimation; and instrumentation

- Control and power electronics--Control theory; control in robotics; power electronics, power systems, and factory automation (hardware and software); linear and nonlinear systems; sensing and vision; and electric machines

- Solid state and quantum electronics--Solid state materials and device physics, silicon devices, III-V compound semiconductors and quantum-well devices, and high-speed and optical devices

- Electromagnetics (EM) and microwave engineering--Interaction of EM waves with materials (radiation, propagation, scattering, reception); EM measurement analysis, modeling, and computation; antennas and radar systems; remote sensing; microwave and millimeter waves; and electromagnetic compatibility

- Computer engineering--VLSI design and CAD, computer architecture and cognitive architecture, artificial intelligence and man-machine interface, fault tolerance, computer vision, pattern recognition, computer graphics, and microprocessor design and application

Industrial Engineering Department

- Manufacturing engineering--Knowledge-based engineering systems, applications of industrial robots and computer vision, factory automation, production planning, and process control

- Human factors engineering--Human performance in engineering systems, man-machine systems, biomechanics, work physiology, human-computer interface, work measurement and method analysis, and industrial safety management

- Information systems and computer applications--Database design and management, management information systems, artificial intelligence including expert systems, manufacturing information systems, simulation, and other computer applications

- Operations research and applied statistics--Mathematical programming and optimization, stochastic processes, quality control and reliability, decision analysis, and applied statistical and probabilistic models

Materials Science and Engineering Department

- Processing of metallic materials--Development of a new inelastic deformation theory for general mechanical behavior of crystalline materials including formation and propagation of microcracks and local concentration of plastic deformation, theoretical development of fracture

mechanics, in-situ transmission electron microscopy analysis on micromechanics, microstructure-property relationships of high temperature materials, and solidification processing including rapid solidification and near net shape continuous casting

- Process metallurgy--Fine ceramic materials for the electronic, aerospace, information, and medical industries including fine powder synthesis of ultra purity, colloidal processing, sinter-forging, hot isostatic pressing, and thin film processing

- Corrosion and surface treatment--Research on understanding corrosion phenomena and proper surface treatment to extend the service life of engineering components operating under harsh environments and on high temperature corrosion to develop protective coatings for superalloys

- Polymer materials research--Improving mechanical and thermal properties of polymer materials by means of new polymer synthesis, polymer blends, and composites

Mechanical Engineering Department

- System and design--CAD/CAM (design of machine elements and systems by computer coupled with computer-aided manufacturing); material forming technology (simulation/optimization of metal forming and continuous casting processes, die/preform design in forging, die/mold design and optimization in the processing of polymers, ceramics, and composites); robotics (control and servo mechanisms, vision, artificial intelligence, design of advanced industrial robots and autonomous mobile robots)

- Thermal and fluid engineering--Energy conversion and conservation (ocean thermal and wind energy conversion systems, wave-body interactions, wind engineering, waste heat recovery systems); environmental engineering (aerosol science, air pollution control, clean room technology); power plant thermal-hydraulics (operating transient analysis, boiling and condensation research, analysis of severe accidents in nuclear power plants, plant safety evaluation); special thermo-fluid topics (heat/mass transfer in manufacturing processes, microelectronics cooling, heat exchanger design); turbomachinery (fluid flow and heat transfer around blades, stress and material problems, nozzle design, vibration, blade cooling, thermodynamic cycle analysis); vacuum technology (heat/mass transfer in the rarefied gas region, vacuum pump and system design)

- Applied mechanics--Biomechanics (dynamic characteristics of skeletal elements, biomaterials, design and manufacturing techniques of artificial joints); composite materials (mechanics of composite materials, optimal design, fatigue and fracture, fabrication techniques); mechanics of porous media (constitutive modeling of powder compaction and deformation of porous materials and ceramics, densification mechanisms, thermo viscoelastic-plastic and creep behavior, forming technology)

CONCLUDING REMARKS

RIST and POSTECH are two very impressive institutions. Both have physical plants with equipment comparable to the best at equivalent institutions in Japan or in the United States. The relationship between RIST and POSTECH appears to be ideal with regard to obtaining interaction between the more applied researchers at RIST and the more fundamentally research-oriented POSTECH faculty. It also appears that POSTECH has been successful in attracting very high quality students. It is now necessary for the researchers at RIST and the faculty at POSTECH to begin to publish in journals so their research can be examined by their peers. It is also necessary for POSTECH to develop students that can go to various institutions and organizations in Korea to start careers that will also contribute to POSTECH's reputation. Finally, it is necessary for POSTECH and especially RIST to contribute in a meaningful way to POSCO's needs.

The progress of these two institutions should be closely watched over the next 5 to 10 years. An excellent start has been made. The potential exists for the establishment of an excellent research institute and an outstanding university.

REFERENCES

1. O. Kwon, K.S. Ro, R.W. Chang, and W.S. Lee, "Effects of composition and hot rolling conditions on the mechanical properties of low carbon bainitic steels," World Materials Congress 1988, Symposium on Microalloyed HSLA Steels.

2. G. Kim and O. Kwon, "Formation of abnormally coarse grain structure in hot-rolled strips," International Conference on Physical Metallurgy of Thermomechanical Processing of Steels and Other Metals, ISIJ (1988).

3. K.N. Paek, K.J. Oh, T.Y. Kim, Y.W. Kwak, and C.I. Son, "The development of an automatic hot slab surface inspection system using a laser," *RIST Technical Research Report* 3(2), 49-59 (1989) (in Korean); K.T. Kim and J. Suh, "Densification mechanisms of metal powders for hot compaction," 123-130; J.I. Jeong, B.O. Kim, and J.W. Chung, "Structure of the oxide film on the reconstructed Si (100) surface," 137-145.

4. S.M. Lee, Y.S. Koo, S.W. Lee, and I.R. Lee, "Studies on the optimal primary cooling for the slab continuous casting through heat transfer analysis," *RIST Technical Research Report* 3(1), 33-41 (1989) (in Korean); C.S. Jeong, J.Y. Lee, and S.Y. Bang, "Parallel computer POPA (I)," 143-149; D.H. Kim, K.H. Lee, B.H. Park, and J.H. Jang, "Pressureless-sintering of alumina/SiC-whisker composite," 151-156.

5. Y.D. Park, Y. Korai, and I. Mochida, "Selective preparation of anisotropic spheres from commercial pitches by carbonization under vacuum," *High Temperature-High Pressure* **16**, 689-694 (1984).

6. Y.D. Park and I. Mochida, "Extractive stabilization of mesophase pitch fiber," *Carbon* **26**, 375-380 (1988).

7. Y.D. Park, Y. Korai, and I. Mochida, "Preparation of anisotropic mesophase pitch by carbonization under vacuum," *J. Mat. Sci.* **21**, 424-428 (1986).

8. C.M. Burns and W.N. Kim, "Solution blending of polystyrene and poly(methyl methacrylate)," *Polymer Engineering and Science* **28**, 1362-1372 (1988).

9. W.N. Kim and C.M. Burns, "Thermal behavior, morphology and the determination of the polymer interaction parameter of polycarbonate-poly(butylene terephthalate) blends," *Makromal. Chem.* **190**, pp 661-676 (1989).

10. J.H. McCarthy, "POSTECH: Korea's new research-oriented university," *Scientific Bulletin* **12**(1), 81-85 (1987).

F.S. Pettit was a liaison scientist with ONR Far East from June 1988 through August 1989 while on sabbatical from the Materials Science and Engineering Department at the University of Pittsburgh. Dr. Pettit's professional interests are in high-temperature materials and surface stability and the use of coatings for protection.

WORKSHOP ON PERSISTENT OBJECT SYSTEMS: THEIR DESIGN, IMPLEMENTATION, AND USE

Edward F. Gehringer

In a persistent object system (POS), a process should be able to create objects that outlive its execution and these objects should be held in on-line storage in the same format used by the process itself. This article surveys the state-of-the-art in POSs as presented at the Persistent Object Systems Workshop. A description of POSs is followed by a discussion of papers presented at the workshop and suggestions of areas for further work.

PERSISTENT OBJECT SYSTEMS

The field of persistent object systems is closely related to programming languages, database systems, and computer architecture. The thesis behind a persistent object system (POS) is that a process should be able to create objects that outlive its execution, and that these objects should be held in on-line storage in the same format used by the process itself. Conventional programming languages, by contrast, read from files and write to files. A file is the only object type that may persist from one execution of a program to the next; files, whether sequential or random access, are incapable of representing all the relationships between objects that are present at run time in the form of pointers. The consequence is that virtually all programs must contain code for reading and writing files and spend time executing this code. Atkinson (Ref 1) quotes a study that concluded that typically 30 percent of the code in programs is concerned with transferring data to and from files or a database management system.

Persistent object systems have much in common with object-oriented languages and database systems. Like object-oriented languages, they have facilities for the creation and use of objects. Object-oriented languages, however, are not usually able to save objects from one execution to the next. A standard Smalltalk-80 system (Ref 2), for example, can create no more than 2^{16} objects and cannot save them in a form that can be used by another program. (It is capable of *suspending* a single user session and resuming where it left off; however, this is not the same as saving objects in a form that can be used by other users or programs.) Also, according to Wegner (Ref 3), *inheritance* of attributes by subtypes or subclasses is an essential feature of object-oriented languages, but many persistent programming languages do not support inheritance.

Persistent object systems can be contrasted with database systems in that they support general-purpose programming languages that are much more powerful than query languages (Ref 4). The object-based philosophy of POSs implies a stronger notion of object identity, which assures that the code for manipulating a single object type is centralized in a single location and prevents unauthorized programs from corrupting or misinterpreting the representation of an object (Ref 5).

The relationship of computer architecture to POSs derives from the fact that POSs need to allow a potentially large number of processes to address the same objects. Consequently, they require very large address spaces–orders of magnitude larger than those found in ordinary virtual memories (Ref 6). This problem can be attacked by clever software implementations, but for greatest efficiency, hardware enhancements are frequently suggested.

THE POS COMMUNITY AND THE POS WORKSHOP

On 10-13 January 1989, a workshop on persistent object systems was held in Newcastle, NSW, Australia. This workshop was the third in a series; its two predecessors were held in Appin, Scotland, in 1985 and 1987. The next is scheduled for somewhere in the Northeast United States, perhaps Massachusetts, in September 1990. The largest persistent object research project seems to be the persistent programming project at the Universities of Glasgow and St. Andrews. Though not represented at this workshop, the Mushroom project at the University of Manchester also has made significant contributions. In Australia, the pioneering effort was the MONADS project at Monash University; key personnel moved to Newcastle and continued the project, and work also has been done at the Australian National University. In the United States, important work has been done by the object-oriented database vendors Ontologic and Servio-Logic, as well as by Texas Instruments. Academic contributions have come from Brown, the Massachusetts Institute of Technology (MIT), the Oregon Graduate Center, North Carolina State, the University of Southern California (USC), and the Universities of Massachusetts, Pennsylvania, and Wisconsin. There also have been scattered contributions from Europe and Canada (but not Japan). Attendance at the recent workshop was limited to 45, with the vast majority coming from Australia, the United Kingdom, and the United States.

TECHNICAL CONTENT OF THE WORKSHOP

As noted above, POS research concerns programming languages, database systems, and computer architecture. The work reported at the conference can be divided roughly along these lines into three categories: (a) programming languages and programming environments, (b) issues in object-based databases, and (c) hardware/software implementations and architectures. Nine papers fell into category (a), 6 into (b), and 11 into (c). Two papers on performance evaluation concern both (a) and (c).

Programming Languages and Environments

Programming Languages. The most widely used persistent programming language is PS-algol (Ref 7,8), developed originally at Edinburgh and enhanced later at Glasgow and St. Andrews. Two papers at this workshop relate to languages based on PS-algol. *Distributed PS-algol* (Ref 9) is designed to allow concurrency. It permits objects to be referenced independent of location (a shared-memory model), but in order to permit efficient execution on distributed-memory machines, it allows relative object locations to be specified and permits data to be explicitly copied from one locality to another. Synchronization is

controlled by semaphores, but higher level linguistic constructs are provided to promote the correct use of semaphores. An efficient remote procedure call is also provided. *MINOO* (Ref 10) is an implementation of a "minimal" object-oriented language in PS-algol. It encompasses the facilities ordinarily found in object-*oriented* languages, such as inheritance, message passing, and encapsulation. However, it omits "such orthogonal concepts as program structuring constructs and expressions." The project--including implementation--was completed in about 2 weeks, demonstrating, according to the author, that persistent programming languages provide prototyping facilities similar in power to those of object-oriented languages.

Two other papers relate to extensions of existing languages. *E* (Ref 11) is a database programming language designed as an extension to C++. It is an endeavor of the EXODUS project at the University of Wisconsin, a project that is taking a toolkit approach to extending database systems. Its goal is to provide a convenient and high-level means for clients to express their interactions with the database. Linguistically, its main extension is to add a *db* (database) attribute to C++ constructors. Another paper (Ref 12) explores how a persistent Prolog might be implemented. Prolog's dynamic clause base normally persists only during the execution of a program. The main issue in extending the language is how an efficient implementation of a persistent clause base might be derived. The author suggests a bitmap index so that the index to the table of clause records can be held in primary memory although the table itself is much too large to fit in main memory. Partial match searches can then be performed in primary memory.

Napier (Ref 13), developed at Glasgow and St. Andrews, is a descendant language of PS-algol, aimed at five problems that must be solved to make persistent languages widely useful. One of these is protection of data; Napier provides a combination of static and dynamic type checking to assure complete type safety with modest overhead. Another is orthogonal persistence; Napier allows any data type to be persistent regardless of its other attributes. For concurrency control, Napier uses a model based on that of CSP and Ada. In the view of this observer, Napier is currently the most advanced persistent programming language.

The final two papers in the language arena relate more to techniques than to languages per se. A paper on the representation of null values (Ref 14) describes aspects of the persistent language Galileo. The paper, however (which was not actually presented at the workshop), relates more to type theory than to persistence, which is only incidental to the discussion. Another paper (Ref 15) deals with the issue of making a persistent object store accessible to different languages and proposes a grammar for representing the way data is structured in a language-independent way. Table 1 summarizes the status of the various languages described at the workshop.

Programming Environments. The workshop sessions included a description of two programming environments for persistent programming languages. The first paper (Ref 16) describes an object browser for PS-algol that provides functionality similar to that of the Smalltalk-80 browser, except that it permits the user to navigate through persistent data structures instead of code. It contains an adaptive knowledge base to display the structure of, and relationships

between, objects in a way that enables a user easily to comprehend how the database fits together. The second programming environment (Ref 17) is oriented toward teaching programming to novices. It provides facilities for compiling, running, and keeping track of the status of programs, which are objects in the persistent object system. It is an interesting exercise in using persistent object systems, but the interface it provides is similar to that provided by other programming environments geared toward teaching.

Table 1. Persistent Languages Described at the Workshop

Language	Orientation	Extends	Designed?	Implemented?
Distributed PS-algol Univ. of Glasgow	Concurrency and efficient execution	PS-algol	Yes	Partially
MINOO Univ. of Glasgow	Showing that persistent programming languages are good prototyping environments	--	Yes	Yes
E Univ. of Wisconsin	Interfacing client processes to a database	C++	Yes	Yes; optimization in progress
Persistent Prolog CSIRO, Sydney	Adding a persistent set and clause base	Prolog	Partially	No
Napier Univ. of St. Andrews	Controlling complexity, protection of data, concurrency	Descendant of PS-algol	Yes	?
Galileo Univ. da Pisa	Hierarchical type system, inheritance, dealing with null values	--	Yes	?
χ[a] Monash University	Exploiting POS support in a capability architecture	--	Yes	Yes, interpreter

[a]See discussion in text section Implementations and Architectures.

Databases

Persistent Databases. The workshop included two papers on particular persistent databases, Worlds at the USC Information Sciences Institute and Iris at Hewlett-Packard Laboratories. Both are large projects with complete, but evolving, implementations. The Worlds paper (Ref 18) describes the concepts underlying the system. A "world" is similar to a blueprint that characterizes some aspect of a building. Similarly, a world characterizes some aspect of a complex object. The Iris paper (Ref 19) concentrates on the architecture of Iris, an object-oriented database system. The architecture includes a generalized function evaluator that allows new operations to be prototyped by writing procedural database functions. To obtain better performance, these functions can be replaced by programs in a language such as C that make calls to the function evaluator. The contents of the dictionary can be modified by function updates, just like ordinary user data.

Problems in Persistent Databases. Four papers relate to specific problems in databases of persistent objects. The first (Ref 20) shows how a persistent object store can be used to integrate a database with the virtual-memory system, such as by providing locking at the page level. A layered transaction mechanism for general operations on objects can be built on top of the virtual memory. Although the paper describes an unimplemented design, it is a very nice integration of databases, operating systems, and proposals for special hardware support.

The next two papers describe methods for concurrency control. The first of these (Ref 21) introduces an algorithm for concurrency control that uses semantic information about an object (commutativity of operations) to allow more concurrency than the algorithm due to Moss currently uses in distributed databases like Argus and Camelot. The algorithm is provably correct, and a detailed outline appears in the paper. The second (Ref 22) describes the concurrency control mechanisms of the ObServer object-oriented database under development at Brown University. It shows how concurrency control operations can be extended to allow different "design groups" working on a project to cooperate in using shared objects. For example, it allows one transaction to be notified if another transaction needs the object. It also allows reading while another transaction writes, with provision for notifying the first transaction when the object is modified.

The last paper (Ref 23) explores how "foreign" objects created by other applications might be integrated into an object-oriented database. It outlines how "surrogates" might be used to assign unique identifiers to objects in the database without modifying the object so that it would be unusable to the application that created it. The paper describes work in progress; no implementation has been undertaken.

Implementations and Architectures

Implementations of persistent object systems span the software-hardware continuum. Most of the papers about implementations were oriented toward the implementation of a particular persistent language; the rest were more concerned with interfaces to databases. Some needs of persistent object systems can arguably best be met with hardware support. Chief among these

is the need to provide a large address space, a topic of two workshop papers. Another paper presents an add-on to a computer architecture, and the last two papers describe complete architectures.

Language Interfaces. The four papers in this category explain how persistent languages can be implemented on existing computers. The first paper (Ref 24) begins with a short description of the χ language designed at Monash University. χ is designed for a Monash-built capability-based computer (although the paper discusses how it could be implemented on standard architectures), with persistent objects being accessed through pointers of type "capability." Its parallel constructs are similar to, but more powerful than, those of Ada. Like Napier, it is a strongly typed language, with type compatibility of persistent objects checked when they are first referenced. The majority of the paper describes the implementation of χ. Objects are accessible through "windows" in the address space, similar to the implementation on the Cm* multiprocessor (Ref 25). The rules for copying transient and persistent objects are derived from the properties of the capability implementation.

The second paper (Ref 26) describes an abstract machine for running the Napier language. The machine is part of a layered architecture and is heap-based in order to support retention of blocks such as activation records after a procedure has returned. In this respect, it is very similar to the Smalltalk-80 virtual machine (Ref 2). Its type system contains just enough information to allow machine instructions to behave differently on operands of different types. It can efficiently implement polymorphic procedures, abstract data types, and bounded universal quantification. The third paper (Ref 27) describes the "object storage service" (OSS) of the SOS operating system. The OSS is generic and requires little compiler support. Currently, it is capable of storing only C++ objects. A new data type is defined to point to permanent objects, since standard C++ pointers are indistinguishable from ordinary data. The last paper (Ref 28) describes the SSE data system, designed to support a persistent object system that interfaces with general-purpose languages on general-purpose hardware. It is quite difficult to assess the impact of this work, since details of the implementation are sketchy and very few references are provided.

Database Interfaces. There is little difference between POSs that interface to languages and databases, except that the latter have built-in transaction support. Portlandia (Ref 29) is a distributed object server that uses a transaction/commit model for concurrency control. Checkpointing of objects is provided, except that immutable objects such as code need not be written to disk. When main memory fills with objects, they are paged to disk using a generation-scavenging algorithm, as in the Smalltalk system SOAR (Ref 30). Multiple reading of an object at different sites is allowed; if an object needs to be written, the copies must be "merged" back into one. A reasonably detailed design exists at a conceptual level, but the system has not yet been implemented. The other paper on database implementation (Ref 31) is simply an extended discussion of the relevant issues, concluding that the most suitable architecture is one that resembles the EXODUS project at the University of Wisconsin.

Addressing Mechanisms. Two papers describe mechanisms for implementing the very large address spaces required by persistent object systems. The first (Ref 32) describes a system that provides multiple address spaces per process. Into these address spaces are mapped *memories*, which encompass objects in both on-line and off-line secondary storage. This scheme is capable of being implemented on modern microprocessors like the Intel 386. The second scheme (Ref 33) employs *names* rather than virtual addresses to make all intermodule references and thereby create an effectively infinite address space, since a process can keep making up new names as it creates new objects. Names are mapped to physical addresses by a multilevel translation scheme similar to that of the Intel 386 with an additional level of caching. Both of the schemes presented at the conference have much in common with the Multics virtual memory as a means of controlling sharing in a large object space. Neither scheme has been implemented yet, but the performance of the second has been simulated.

Hardware Support. POMP (Ref 34) is a persistent management coprocessor, designed to speed up address translation for accessing persistent objects. It interfaces to a Motorola 68020 just like other coprocessors like the Weitek floating-point accelerators. POMP can translate addresses to persistent objects almost as fast as the processor can reference local memory, instead of the approximately 10-instruction overhead required to follow an addressing path using standard 68020 instructions. The initial design of POMP has been completed, and funding is being sought for its construction.

MONADS, by contrast, is a complete architecture, described in two papers from the conference. The first (Ref 35) describes the MONADS-PC architecture, which is capability based, and provides a 60-bit address space, large enough to hold a persistent object store. It attains efficiency comparable to conventional architectures by confining capabilities to standard locations within activation records so that no special protection hardware is necessary and by avoiding the use of a central object table with its lookup and indirection overhead. Current work includes the development of a local area network of MONADS machines with shared virtual memory. The second paper (Ref 36) describes the MONADS-MM architecture, a "massive-memory supercomputer" with a 128-bit address space and a main memory of at least 4 GB. This paper focuses on address-translation issues and has few specifics pertinent to persistent objects.

Performance Measurement

Two papers report on the measurement of PS-algol programs. The first (Ref 37) is an excellent survey of techniques for monitoring the run-time behavior of programs, useful to studies of almost any language, not just PS-algol. The most important technique is to record the execution of basic blocks in the code and keep a list of actions that each basic block performs. In that way it is possible to tell how often the program performs each action, without the need to record and interpret a large file of trace information. Other techniques, such as Lempl-Ziv compression, can be used to reduce the size of the output further. A

combination of these techniques can produce savings of three orders of magnitude over the raw trace data.

Bailey (Ref 38) instrumented the PS-algol abstract machine to measure characteristics of program execution. The data are very preliminary, consisting of results on just two programs. They indicate that loading of persistent objects into main memory is clustered at working-set transitions, such as when a program begins execution and when it writes its results. This leads to a very skewed distribution for times between persistent identifier dereferences--the median is 18, but the mean is 1879, reflecting the existence of long periods of time during which the program references no new objects. To minimize accesses to the database, the author is investigating object preloading--automatically loading objects referenced by other objects that are being loaded.

SUMMARY AND CONCLUSIONS

Most recent progress in persistent object systems seems to be concentrated in programming languages. Hardware and software support for the efficient execution of these languages is not keeping pace with language development. One partial exception is database interfaces, but even here, much more has been designed than has been implemented. The MONADS series of machines are practically the only architectures that provide hardware support for persistent objects.

In several areas, progress is conspicuously absent. No work seems to focus on supporting persistent objects at the operating system level. Such support would have obvious advantages. Unlike language- or database-level support, it would be available to all languages that run on a machine. (Several papers noted that minimal change to a language--the simple addition of a *persistent* attribute for data--enables it to take advantage of some of the facilities of a persistent object store.) Unlike hardware support, it does not require a new computer architecture. At first glance, one would assume hardware support would be more efficient. But that does not necessarily imply that operating system support would be *perceptibly* less efficient. The Mach operating system is an example of an operating system that provides efficient software facilities for techniques that were once thought to require special hardware.

One critical issue in storing persistent objects is how they should be grouped. It is clearly far too inefficient to swap them in and out of main memory individually. Previous studies focused on static (Ref 39) and dynamic (Ref 40) strategies for grouping objects onto pages in a paged virtual memory. At the Workshop on Object-Oriented Database Implementation held at OOPSLA-87 (Ref 41), object grouping was a topic of major interest. Yet no new strategies or results were reported at all at this workshop.

Also conspicuous by their absence at this workshop were the commercial vendors of object-oriented databases, such as Graphael, Ontologic, and Servio-Logic. It is difficult to gauge the state of development of these commercial systems, since few details of their implementation have been published. Widespread skepticism has greeted some of the performance measurements (e.g., Ref 42) that have been presented.

This survey has been intended to describe the state-of-the-art as presented at the Persistent Object Systems Workshop. The author hopes that it will suggest fruitful avenues for future investigation.

REFERENCES

1. M.P. Atkinson, P.J. Bailey, K.J. Chisholm, P.W. Cockshott, and R. Morrison, "An approach to persistent programming," *Computer Journal* **26**(4) (1983).

2. A. Goldberg and D. Robson, *Smalltalk-80: The Language and its Implementation* (Addison-Wesley, 1983).

3. P. Wegner, "Dimensions of object-based language design," Proceedings of OOPSLA '87: Object-Oriented Programming Systems, Languages, and Applications, Orlando, October 1987, pp 168-182. Printed as *ACM SIGPLAN Notices* **21**(12) (December 1987).

4. R.L. Cooper, M.P. Atkinson, A. Dearle, and D. Abderrahmane, "Constructing database systems in a persistent environment," Persistent Programming Research Report 34 (Department of Computer Science, University of Glasgow, and Department of Computational Science, University of St. Andrews, 1987).

5. D. Maier and J. Stein, "Development of an object-oriented database," in *Research Directions in Object-Oriented Programming*, edited by B. Shriver and P. Wegner (MIT Press, 1987, pp 355-392).

6. W.P. Cockshott, "Building a microcomputer with associative virtual memory," Persistent Programming Research Report 20 (Department of Computing Science, University of Glasgow, and Department of Computational Science, University of St. Andrews, 1985).

7. M.P. Atkinson, K.J. Chisholm, and W.P. Cockshott, "PS-algol: An algol with a persistent heap," *ACM SIGPLAN Notices* **17**(7), 24-31 (July 1982).

8. W.P. Cockshott, M.P. Atkinson, K.J. Chisholm, P.J. Bailey, and R. Morrison, "Persistent object management system," *Software--Practice and Experience* **14**, 49-71 (1983).

9. "Distributed PS-algol," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 343-357 (January 1989).

10. R. Cooper, "The implementation of an object-oriented language in PS-algol," addendum to the *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle* (January 1989), 16 pp.

11. J. Richardson and M. Carey, "Implementing persistence in E," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 302-319 (January 1989).

12. R.M. Colomb, "Issues in the implementation of a persistent Prolog," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 65-79 (January 1989).

13. R. Morrison, A. Brown, R. Carrick, R. Connor, A. Dearle, and M.P. Atkinson, "The Napier type system," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 253-270 (January 1989).

14. G. Ghelli, "Null values: How nicely they fit a hierarchical type system," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 158-172 (January 1989).

15. G. Michaelson, "Grammars and implementation independent structure representation," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 242-252 (January 1989).

16. A. Dearle, Q. Cutts, and G. Kirby, "Browsing, grazing and nibbling persistent data structures," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 96-112 (January 1989).

17. M.S. Powell, "A program development environment based on persistence and abstract data types," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 286-301 (January 1989).

18. D.G. Allard and D.S. Wile, "Aggregation, persistence, and identity in Worlds," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 1-18 (January 1989).

19. P. Lyngback and K. Wilkinson, "The architecture of a persistent object system," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 229-241 (January 1989).

20. P. Brössler and B. Freisleben, "Transactions on persistent objects," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 19-35 (January 1989).

21. A. Fekete, N. Lynch, M. Merritt, and W. Weihl, "Commutativity-based locking for nested transactions," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 113-127 (January 1989).

22. M.F. Fernandez and S.B. Zdonik, "Transaction groups: A model for controlling cooperative transactions," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 128-138 (January 1989).

23. S. Heiler and B. Blaustein, "Generating and manipulating surrogates for heterogeneous distributed objects," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 173-185 (January 1989).

24. A.J. Hurst and A.M.S. Sajeev, "A capability based language for persistent programming: Implementation issues," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 186-201 (January 1989).

25. E.F. Gehringer, D.P. Siewiorek, and Z.Z. Segall, *Parallel Processing: The Cm* Experience* (Digital Press, 1987).

26. R. Connor, A. Brown, A. Carrick, A. Dearle, and R. Morrison, "The persistent abstract machine," *Proceedings of the*

Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle, pp 80-95 (January 1989).

27. M. Shapiro and L. Mosseri, "A simple object storage system," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 320-327 (January 1989).

28. S.L. Wright, "The evolution of the SSE data storage system into a persistent object system," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 358-372 (January 1989).

29. H.H. Porter, III, "Persistence in a distributed object server," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 272-285 (January 1989).

30. D. Ungar, "The design and evaluation of a high-performance Smalltalk system," Ph.D. thesis, University of California, Berkeley (MIT Press, 1986).

31. D. Stemple, "Exploiting the potential of persistent object stores," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 328-342 (January 1989).

32. P.A. Buhr and C.R. Zarnke, "Addressing in a persistent environment," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 36-50 (January 1989).

33. E.F. Gehringer, "Name-based mapping: Addressing support for persistent objects," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 139-157 (January 1989).

34. W.P. Cockshott, "Design of POMP--Persistent object management coprocessor," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 51-64 (January 1989).

35. J.L. Keedy and J. Rosenberg, "Support for objects in the Monads architecture," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 202-213 (January 1989).

36. J. Rosenberg, D.M. Koch, and J.L. Keedy, "A massive memory supercomputer," addendum to the *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 387-394 (January 1989). Also in *Proc. 22nd Hawaii International Conf. on System Sciences*, Vol. I: Architecture, pp 338-345 (January 1989).

37. C.Z. Loboz, "Monitoring execution of PS-algol programs," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 214-228 (January 1989).

38. P.J. Bailey, "Performance evaluation of a persistent object system," *Proceedings of the Workshop on Persistent Object Systems: Their Design, Implementation, and Use, Newcastle*, pp 373-385 (January 1989).

39. J.W. Stamos, "Static grouping of small objects to enhance performance of a paged virtual memory," *ACM Transactions on Computer Systems* 2(2), 155-180 (May 1984).

40. I.W. Williams, M.I. Wolczko, and T.P. Hopkins, "Realisation of a dynamically grouped object-oriented virtual memory hierarchy," *Proceedings of the Second Appin Workshop on Persistent Object Systems* (to be published by Springer-Verlag, 1987).

41. S.M. Thatte, "Report on the object-oriented database workshop: Implementation aspects," *ACM SIGPLAN Notices* 23(5), 73-87 (May 1988).

42. J. Duhl and C. Damon, "A performance comparison of object and relational databases using the Sun benchmark," *Proc. OOPSLA '88*, pp 153-163. *ACM SIGPLAN Notices* 23(11) (November 1988).

Edward F. Gehringer received a B.S. degree from the University of Detroit and a B.A. degree from Wayne State University, Detroit, both in 1972, and M.S. and Ph.D. degrees from Purdue University in 1974 and 1979, respectively. Beginning in 1979, he was a research associate and lecturer, Carnegie-Mellon University, associated with the Cm and StarOS projects. In 1981 he held a Fulbright Postdoctoral Research Fellowship at Monash University, Melbourne, Australia, where he assisted the Monads project. Since 1984 he has been an assistant professor of electrical and computer engineering and computer science at North Carolina State University, Raleigh. His research interests include object-oriented systems, persistent object stores, very large address spaces, and parallel architectures. Dr. Gehringer is a member of the Association for Computing Machinery and the IEEE.*

SUPERCOMPUTERS: THE NEXT GENERATION

Kenneth W. Neves

An update of supercomputing technology is presented utilizing information on the latest supercomputing technology from Japan and the United States. Notably the Cray-2, Cray Y-MP, Fujitsu VP-2600, Hitachi S-820, and NEC SX-3 are discussed and contrasted with earlier designs. While complete and uniform information on all of the above machines is not available, enough information is presented to establish likely near-term trends and assess performance characteristics.

INTRODUCTION

The purpose of this paper is to offer an early technological update of what can be loosely called the third generation of supercomputing. While debates rage among vendors, users, and even procurement organizations as to what is, and what is not, a supercomputer, a simple definition would be:

> *The class of general purpose computers that is both faster than commercial competitors AND has sufficient central memory to compute problem sets of general scientific interest.*

This definition could be used to define a class of hardware that has evolved from the very first computers. Generally, the term "supercomputer" was coined in the early 1970s and first applied to machines such as the CDC CYBER 200 series and the Cray-1. Certainly today, the most popular "supercomputer," by virtue of machine placements, is the Cray X-MP. In 1983-84 the Japanese entered the supercomputer market, with impressive first entries in the Fujitsu VP-200, Hitachi S series, and the NEC SX series. CDC had spun off a subsidiary in ETA Systems, which developed the ETA-10 series, delivering several systems before the latter's recent demise in 1989. Cray Research, Inc. (CRI), subsequent to the X-MP, developed two new systems, the Cray-2 and more recently (1988) the Y-MP. The Cray-2 (like the Cray-1 and Cray-3) is a Seymour Cray designed machine, while the Y-MP is patterned after the X-MP designed by Steve Chen. Both Cray and Chen have left CRI. Steve Chen started a new supercomputer company with support from IBM, called Supercomputer Systems, Inc. (SSI). Recently CRI announced the departure of Seymour Cray to start a new company, the Cray Computer Corp. These events have shocked the U.S. computer industry. Seymour Cray's new company will be making and marketing the Cray-3 and 4; Steve Chen's SSI machine is expected in 1992, and CRI is planning a 1992 release of the C-90. These computers will not be addressed in any detail in this report as public information is not available.

In 1989, Fujitsu and NEC announced machines with single central processing unit (CPU) peak performance of 4 and 5.5 GFLOPS, respectively. (1 GFLOP is a billion floating point operations per second.) The NEC SX-3 packages four CPUs for a combined peak performance potential of 22 GFLOPS, the fastest peak performance rating of any computer announced as of this writing. Both computers are expected to be available in early 1990. Both companies are utilizing the same chip technology found in their more traditional mainframe products.

This represents reliable technology with little technological risks in maintenance and manufacture.

Thus, the latest generation of supercomputers (in the 1990 time frame) boasts both U.S. and Japanese computers competing at unprecedented computational rates (measured in terms of peak floating point operations per second). The growth of memory capacity in these systems is keeping pace with the growth of computational power.

These recent entries in the market from CRI and its Japanese competitors offer an opportunity to assess likely trends in their future products. Several caveats are in order, however. First, predictive capability in the supercomputer industry is fraught with error. Some of the forces in the market place are worthy of mention before we embark on a technical discussion focused on the products of established vendors. First, the supercomputer industry has undergone, and will continue to undergo, enormous change from market pressures. Up until the mid-1980s, supercomputers were manufactured by relatively new or smaller companies. Certainly, Cray Research and CDC could not be considered fully integrated computer companies like IBM and its Japanese competitors. Texas Instruments made a valiant attempt at this specialized market with the ASC Vector Processor, and Burroughs has flirted with the market on several occasions. Despite the demand for supercomputers worldwide, the supercomputer industry remains fragile. In 1983, the Japanese were among the first fully integrated companies to seriously enter the market. Despite lack of success in the United States, Japanese supercomputer sales are growing and their products are improving at a rapid pace. Unfortunately, CDC recently has announced its withdrawal from the market. A short time later Cray Research, Inc. announced the inability to continue to fund the two development projects discussed earlier. These events and the announcements of the Fujitsu VP-2600 and the NEC SX-3 have made 1989 a very dynamic year in this industry.

A second caveat is that this report will concentrate on the products of the companies mentioned above, but it is critical to point out that parallelism is coming of age. Hypercube architectures and massively parallel systems are beginning to demonstrate that they can solve "real" problems at impressive rates. Any prediction of supercomputing over the next decade would be remiss not to mention the possibility of completely new approaches in architecture for supercomputers. Ten years from now, it is likely that one or more "supercomputer" designs will be highly (if not massively) parallel. The theoretical characteristics of hardware dictate that distributed memory parallelism will dominate high-end computing. The unknown is when this parallelism will come of age. The decade of the 90s will be a pivotal time for computer architecture evolution. This coupled with the challenges these designs pose for software promises to make the next decade one of the most challenging decades in scientific computing, for both users and manufacturers.

In this report we will concentrate on vector CPU supercomputers available today or within 2 years. While this work is intended to be self-contained, it is primarily an update to Chapters 2 and 3 of Reference 1. The reader is referred to this work for a more complete introduction to high performance computing.

THE SUPERCOMPUTER CENTRAL PROCESSING UNIT

A framework for discussing the similarities, differences, and relative advantages of various supercomputers requires a fundamental description of the architecture of modern supercomputer CPUs. In Figure 1 a simple generic diagram of a supercomputer central processing unit is displayed. The figure itself is more symbolic than an actual replica of a manufacturer's hardware diagram. While several manufacturers have introduced multiple CPU architectures (which we will discuss later), it is worth noting that the fundamental architecture of their individual CPUs has remained a vector pipelined architecture that fits the general description given in the section titled "Features of a Vector CPU."

Historical Perspective

In and of itself, no feature above is a sufficiently important parameter for assessing computer performance. The interplay or "balance" between these elements, however, gives the computational character of a given computer. The most distinctive feature of modern supercomputers is their orientation toward processing "vectors" or arrays of elements as operands. For years the computational bottleneck in scientific computing was the processing of floating point computations. The CDC 6600, for example, tried to improve this bottleneck by using two floating point multiply units. Later the CDC 7600 exploited the pipelined concept in the functional units. Interestingly enough, the floating point units on the 7600 were, all too often, left idle. The bottleneck to computation was the rate of instruction processing associated with the overhead of fetching and storing each pair of operands and/or result. The first modern supercomputers (the CDC Star-100 and the Cray-1) circumvented this "instruction-issue" bottleneck by extending the instruction set to include vector operations. This coupled with a high bandwidth connection between the vector units and memory, through an interface (buffer or registers), characterizes modern supercomputer CPUs. In these, and subsequent supercomputer designs, one single instruction could launch a process that operated on not one but many operand pairs. The production rate of floating point operations became a much more meaningful measure of performance than did machine instruction rates. With the advent of the "vector" computer, the performance rating of millions of machine instructions per second (MIPS) gave way to the MFLOP (millions of floating point operations per second) as a first order indicator of performance.

The MFLOP is now being replaced by the GFLOP in recognition of the increased speeds of these machines. With the increased use of parallel CPUs and advances in electronics, we anticipate "tera" (trillion) FLOP machines in the mid to late 1990s. It is perhaps worthy to note that even though the computation rate seems to be the paramount measure of performance, the ability to achieve this performance on any given machine is dictated by the peak sustained flow of data from memory to the vector units. This will become more apparent in the discussions below. These issues figure greatly in the use of multiple (parallel) CPU systems, another observed trend in computing at all levels.

Before we discuss the trends, it is necessary to delve further into the single CPU architecture.

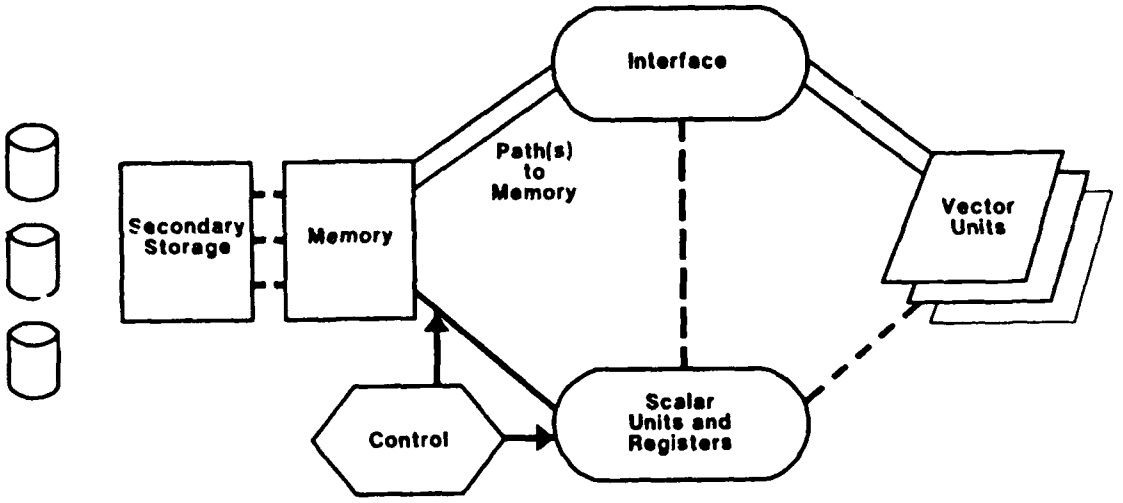


Figure 1. A supercomputer CPU.

Features of a Vector CPU

As described above, the breakthrough in performance with modern supercomputers came with the combination of pipelined arithmetic units and enhanced instruction processing allowing vector data to flow more freely between memory and the vector computational units. The critical features in this data flow process are discussed below. A more detailed and indepth account is found in Reference 1.

The Number of Paths-to-Memory. On vector computers the operands are "vectors," large arrays of data stored in memory. In order to "operate" (e.g., add, multiply) on vectors one must fetch the vector operands from memory to the interface and eventually store the results. A simple dyad operation requires three memory references. One must fetch two vector operands and store the result. Several computer designs, notably the Japanese machines, have multiple arithmetic pipelines. However, in these machines the multiple pipelines are logically treated as one, and the vector operands are fetched (or stored) in such a manner as to sustain full computation rates among the multiple pipeline units according to a fixed hardware scheme. The number of paths-to-memory refer to the number of simultaneous "vectors" that can be passed to and from the vector units at full computation rates. There are differences in the number of these "logical" paths among supercomputers and these differences can materially affect performance.

The Interface Between Memory and the Vector Units. The interface between memory and vector units is usually a vector register set. Notable exceptions were the CDC/ETA Systems computers, which used a buffer, and the IBM 3090/VF, which used an associative cache. Registers are supported by the vector instruction set, and vector data are passed modulo the length of the vector registers. For example, on the CRI machines this length is 64 words. Buffers are not supported in the instruction set but are managed by the hardware, perhaps through microcode. The notable difference is that compilers or application software

cannot utilize buffers as temporary resting places for accumulation of data or storage of interim results. Other types of interfaces include caches, local memories, and scalar registers. With the recent demise of ETA Systems, all the high-end supercomputers are vector register oriented.

The Allowable Vector Data Structure in Memory. To a mathematician or scientist, a vector is simply an ordered array of numbers. For a computer, this array doesn't exist until it is designated by the memory location of its ordered components. Various supercomputers have limitations as to the flexibility allowed for the specification of vectors. There are three common complexities found in manipulating vectors stored in supercomputer memories:

1. Contiguously stored vectors (stride 1)

2. Regularly stored vectors (allowing an arbitrary stride between array components as stored in memory)

3. Randomly stored (allowing successive vector components to be designated according to a list or index vector--a form of indirect addressing)

The random storage of vectors, when supported by a sufficiently rich vector instruction set, provides very useful capabilities across a broad range of linear algebra and other mathematical algorithm classes. While all supercomputers generally support type 1 above, not all support types 2 and 3. The CDC/ETA machines, for example, support only type 1 in all of their vector operations. Today most other machines support all three types, yet some of them degrade in full performance on type 3 operations. The Cray-2, for example, performs at one-quarter of the peak vector computation rate on random stride vectors.

The Vector Instruction Set. As pointed out above, vector instruction processing is the distinguishing feature of supercomputers from scalar computers. The performance potential of modern supercomputers can only be achieved in "vector mode," i.e., utilizing vector instructions. A full complement of vector instructions includes dyad and triad operations (the latter being linked add/multiply operations). These instructions are to be supported on all allowable vector types, contiguously, regularly, and randomly stored. While all the latest machines support these constructs on basic operations, some degrade with noncontiguous storage. For example, Fujitsu's VP-200 and VP-2600 degrade from peak performance when operating on noncontiguously stored vectors. Some machines have more vector instruction types than others, but usually the extra instructions do not impact mainstream scientific computations. (There are significant exceptions, but the details are beyond the scope of this article.)

The Characteristics of the Vector Floating Point and Fetch/Store Units. This area has become most critical in rating performance of supercomputers. With the advent of parallel CPUs, deficiencies in this area can be even more pronounced. Pipelined units work like "mini" assembly lines. The number of pipe segments is analogous to "work stations" in the assembly line. The more segments, the longer it takes the assembly line to fill up and the longer the start up time, or time to produce the first result from a dead start. This "wait" time is the overhead of vector computing. Memory fetch/store units and arithmetic units all have pipe segment latency, which must be considered in performance evaluations. This will be discussed in much more detail later.

A SUMMARY OF SUPERCOMPUTER HARDWARE CHARACTERISTICS

The above features, along with computer memory, comprise the basic tools that compilers and application programs use to achieve high performance. In this section we will catalog, via tables and brief discussions, the characteristics of various supercomputers. In later sections we will look at the coupled effects of using multiple CPUs and vector architectures together and their combined impact on software.

Information on the newest Japanese machines is somewhat incomplete owing to the fact that little has been published with first deliveries being some time away, and some information is proprietary. Yet, a fairly complete picture can be drawn from available information on the new hardware, particularly when compared to its predecessors. The design characteristics are quite similar with improvements in packaging, cycle times, and a few new "tricks" to improve performance.

Peak Floating Point Power

Taken alone, peak performance ratings are very misleading indicators of performance. Peak performance is an indication of what computational rate a given processor is guaranteed not to exceed. Taken in this light these rates (often dubbed "macho FLOPs" by supercomputer hackers) provide an indication of underlying asymptotic performance potential for very ideal operations of vectors of enormous lengths. In subsequent discussions, inhibitors to attaining peak performance will be explored. Table 1 gives the single CPU performance of various machines and, where applicable, includes multiple CPU ratings. All the data are for 64-bit precision, the current standard word length for high-end computation. The italicized entries indicate machines that are not yet delivered, with deliveries expected in early 1990. The Cray-1 and CYBER 205 are included for historical perspective, and the Cray X-MP is included because it is the most used supercomputer model worldwide at this time.

Memory

Main memory is an important characteristic to memory-constrained applications. Many important computations in science and industry are constrained by memory and storage bandwidth. Computational fluid dynamics (CFD), large structures problems, electromagnetics, seismic analysis, and medical imaging are all examples of applications whose model sizes are constrained by memory size. The Cray-2 at time of introduction set a new standard for memory size. Several applications ran on the Cray-2 at slower computation rates than the Cray-X-MP, yet the Cray-2 was the only option for the bigger problems. The Y-MP is much faster than the Cray-2, yet its early models are memory deficient by comparison. Table 2 lists memory sizes at time of introduction. As chip technology improved later models offered larger memories. The bank (interleave) characteristics give an indication of potential conflicts in random or strided memory references. (See Reference 1 for details.)

Table 1. Peak Performance Rates

Computer	Cycle Time (ns)	Single CPU Peak MFLOP Rating	Multiple CPU Peak MFLOP Rating
Cray-1	12.5	160	
Cray X-MP	8.5	233	932, 4 CPUs
Cray-2	4.1	488	1,952, 4 CPUs
Cray Y-MP	6.0	333	2,666, 8 CPUs
Cray-3	2	1,000	16,000, 16 CPUs
CYBER 205	20	200	
ETA-10/G (1988)	7	625	5,000, 8 CPUs
Fujitsu VP-400E	7	1,700	
Fujitsu VP-2600	4	4,000	
Hitachi S-810/20	14	630	
Hitachi S-820/80	4	2,000/3,000	
NEC SX-2	6	1,300	
NEC SX-3	2.9	5,500	22,000, 4 CPUs

Table 2. Main Memory Sizes (at Introduction) (64-bit words)

Computer	Size (MW)	No. of Banks	Secondary Memory Size
Cray X-MP/4	8	64	512
Cray-2	256	128	none
Cray Y-MP/8	32	256	512
Fujitsu VP-400E	128	256	none
Fujitsu VP-2600	256	256	1,024
Hitachi S-820	64	128	1,500
NEC SX-2	32	512	256
NEC SX-3/44	256	1,024	2,048

Memory sizes from all vendors are growing. The Japanese machines seem to have more latency in memory fetch/store operations (as does the Cray-2) due to their decisions on packaging and cooling. As the vendors move more toward parallel CPUs, memory will have to be more distributed. The Cray-3 at 16 CPUs, sharing one memory, should prove to be the limit of pure shared memory designs. Beyond 16 CPUs, the need for local memories and/or large caches to alleviate memory conflicts seems inevitable. This will introduce new complexities to large application program designs.

In recent years, the growth in main memory sizes has not been matched by the increase in disk technology access speed or capacity. As a result almost all manufacturers offer some type of secondary memory and disk striping (allowing parallel input/output (I/O) of a single file to several disks).

Paths-to-Memory

A quick listing of this characteristic among vendors (see Table 3) reveals that this feature continues to be critical. Anything less than three paths-to-memory puts a burden on efficiency since so many algorithms in use today have inner loops that require at least two fetches and one store. Often these algorithms can be recast to provide less demanding inner loop memory traffic, but all too often this is not done.

Table 3. Paths-to-Memory

Computer	No. of Logical Paths-to-Memory Per CPU	Latency (cycles)
Cray-1	1	11
Cray X-MP	3	14
Cray-2	1	35-50
Cray Y-MP	3	17
CYBER 205	3	50
ETA-10	3	[a]
Fujitsu VP-200	1 or 2	31-33
Fujitsu VP-400	1	31-33
Fujitsu VP-2600	1 or 2	
Hitachi S-820	2	[a]
IBM 3090/VF[b]		
NEC SX-2	4	[a]
NEC SX-3	3	60-70

[a]Not available.
[b]Associative cache.

One path-to-memory is simply too restrictive. The Cray-2 has only one path as did its predecessor, the Cray-1. It was the upgrade of this feature in the Cray X-MP that made a big difference in its performance. In fact, as reported in Reference 1, an optimally coded triad operation on the X-MP is four times faster than an optimally coded triad on the Cray-1, despite the X-MP boasting only a 25-percent faster cycle time. This improved performance is solely due to the extra paths-to-memory. Later CRI designs, the Cray-3 and Y-MP, both have three paths-to-memory. Many basic computational loops in FORTRAN algorithms are severely handicapped by only one path-to-memory. As indicated in Table 3 the Fujitsu VP-200 has two paths-to-memory on contiguously stored vector operands, but for strided or random vectors it has only one path. The VP-2600 is similar to the VP-200 in paths-to-memory. The VP-400 added more pipes (four sets), which would necessarily double the number of words per cycle needed to be fetched from memory to support peak speeds. Thus, in the case of the VP-400, the designer was forced to provide a single logical path for all vector operations. While the number of paths-to-memory has not been consistent among vendors, within a single vendor's line, or even (in the case of the Fujitsu VP-200 and 2600) within a single machine, it is generally accepted that a minimum of three paths-to-memory (two load and one store) is desirable.

PARALLELISM, VECTOR COMPUTATION, AND LATENCY IN DESIGN

Perhaps the most striking realization in reviewing supercomputer architectures is that all manufacturers confront the same barriers of time and space. Scalar architecture is limited in its ability to produce floating point operations. The only alternative is parallelism. In fact, all manufacturers are employing about the same amount of parallelism in their designs, but in different forms. For example, the Cray X-MP four-CPU machine has two floating point vector units per CPU (an add and a multiply). Thus, the whole system has eight floating point units. It is termed a parallel CPU machine because it has four CPUs. On the other hand, the NEC SX-2 is a single CPU machine. Yet, within the CPU it has four add pipes and four multiply pipes. In terms of floating point operations, the two machines have the same degree of parallelism. To appreciate the differences between the two machines one must look first at the bottlenecks, latencies, and flow of vector computation and the

likely use of parallel CPUs. This information, when coupled with an understanding of the intended application, can be useful in predicting performance.

Vector Computation

In this section the dynamics of vector computations will be examined among the various manufacturers. The tradeoffs of vector computation when used in parallel CPU designs versus single CPU with multiple pipelines are examined.

Multiple Pipelined Architectures. The latest generation of machines from the Japanese manufacturers have followed their earlier trend of using multiple pipelines within a single processor to achieve greater peak performance. In order to do this, they must provide greater bandwidth between the CPU and main memory. For example, in order to support four multiply pipes producing one result per cycle each, a stream of eight operands must be available each cycle after start-up to support two logical paths (i.e., a fetch of two vectors). The Hitachi S-810, the NEC SX-2, and the Fujitsu VP-200 and VP-400 all have multiple pipelines as indicated in Table 4. The CRI series of machines has consistently provided only one add and one multiply pipe. In order to achieve higher performance rates, more CPUs were added.

Table 4. Floating Point Pipes Per CPU (Earlier Machines)

Computer	No. of Pipes/CPU
Cray Y-MP	2
Fujitsu VP-400	8
Hitachi S-810	8
NEC SX-2	8

The second generation of Japanese supercomputers promises to offer more variety in the approach to offering multiple pipelines. It is not really known how many floating point pipes make a well-balanced vector/parallel architecture. Many believe one add and one multiply per CPU is not enough, and others argue that 16 is excessive. The answer, of course, is application dependent. Table 5 indicates what the Japanese have done in their more recently announced machines.

Table 5. Floating Point Pipes Per CPU (New Machines)

Computer	No. of Pipes
Fujitsu VP-2600[a]	8
Hitachi S-820[a]	8
NEC SX-3	16

[a]These designs use a combination add/multiply pipe that is chained or linked, allowing an add result to be input to a multiply. This achieves two floating point results per cycle per pipeline for chained operations. This is in contrast to NEC, which provides fully independent add and multiply pipelines.

The Fujitsu VP-2000 series has taken the following approach to improving the multiple pipeline computational output. In earlier Fujitsu designs, as already observed, providing the proper bandwidth to memory (as evidenced by reduced paths-to-memory) has been a design problem (i.e., a bottleneck in data flow). Adding pipes exacerbates this

design problem for the architect. In the VP-400E, an approach was taken to create a linked pipe. This was an add/multiply pipe that could perform a vector multiply followed by a vector add without going back to registers with the intermediate result. This could be considered hardware "chaining" or "linking" of the multiply and addition operations. If only a multiply is desired, the add function is not performed, but the pipe length remains as long as the chained operation. On the VP-2600 this concept was extended as follows.

The VP-2000 series has the following floating point units--two add/multiply units and one divide unit. This could be called a pipe set. On the VP-2600, there are four such sets. Each arithmetic unit is composed of four pipelines, which are run simultaneously. The add/multiply pipes are hardware "chained" to produce linked add multiply results. Therefore, the VP-2600, in theory, can perform eight vector adds chained with eight vector multiplies per cycle. This results in 16 floating point operations every 4 ns. The divide cannot execute while all the add/multiply pipes are going. Thus, one arrives at the 4-GFLOP peak performance figure. An optional feature of the VP-2600 is the addition of a second scalar CPU, the model VP-2600/20. The rationale for this feature is that most applications cannot keep a vector CPU busy. A second scalar CPU will improve job stream improvement and allow sharing of the vector CPU. This may find usefulness in mixed business scientific environments.

NEC seems to have pushed the multiple pipeline concept to the extreme in the SX-3. This machine has 8 multiply and 8 addition pipelines, for a total of 16 floating point units per CPU. Unlike Fujitsu's VP-2600, these are independent units, not requiring chaining to attain peak performance. This is double that of the SX-2. With this many pipelines to fill, a much higher bandwidth has been established between registers and main memory. To support this feature the pipelines are grouped into two logical sets of four add/multiply pairs each. Each logical set is operated much like the SX-2. A vector operation is automatically spread among the four pairs of pipelines as appropriate. To achieve simultaneous functioning of both logical sets, the instruction process supports issuing overlapped instructions. That is, a vector instruction takes two clock periods to execute, but a second vector instruction can be initiated one cycle after the first. Subsequently, two vector operations will proceed simultaneously, each using a separate logical set of pipes. The net result is 16 floating point operations per cycle. Both instructions are fully supported by two vector fetch and one vector store pipeline. This means that three logical paths-to-memory are fully supported. With a 2.9-ns clock, the NEC SX-3 (called the SX-X in the United States) is a 5.5-GFLOP/CPU design--in its four-CPU configuration, it is a 22-GFLOP peak performance machine.

A Study of Vector Start-Up Time. The pipelined concept in vector operations and memory references is a great equalizer. For example, a manufacturer with slower memory technology (longer latency to retrieve data) can pipeline the memory fetch operations with slightly more segments (stages) and achieve a high effective bandwidth for large amounts of data. In reality for long vector operations the resulting rates are as good as another manufacturer who has faster memory. The problem with this approach occurs when vectors are not "large." This concept has been a known performance issue for years and was well catalogued by R. Hockney and C. Jesshope (Ref 2). Reference 2 defined a term called $N_{1/2}$. The term

will be redefined here, and a latency result in Reference 1 for single pipelines will be extended to multiple pipeline operations.

> *Given an operation (or even an algorithm), $N_{1/2}$ is defined to be the length of a vector to achieve one-half the asymptotic peak performance for the given operation (or algorithm).*

Perhaps the best way to describe $N_{1/2}$ is graphically. Figure 2 displays the performance in MFLOPS for the vector triad, A*X+Y (a Scalar "A" times a vector "X" Plus a vector "Y," called SAXPY), as a function of vector length. The curve is obtained by modeling the time of a vector operation as follows:

```
T = S + K*N
```

where S is the start-up time to fill the pipeline and K is a constant (related to the pipe cycle time). Solving this equation for computation per unit of time yields,

```
N/T = 1/(S/N+K)
```

As N approaches infinity, N/T approaches the asymptotic rate, R, for the vector process, as indicated in Figure 2.

The value of the $N_{1/2}$ parameter is largely heuristic. While peak performance gives one an ideal of potential performance on long vectors, $N_{1/2}$ reveals performance in less than ideal situations. If performance were being measured on vector operations of length equal to $N_{1/2}$, then performance would be, by definition, 50 percent of peak. $N_{1/2}$ gives us a "feel" for what is a short vector and what is a long vector. A common heuristic to apply is:

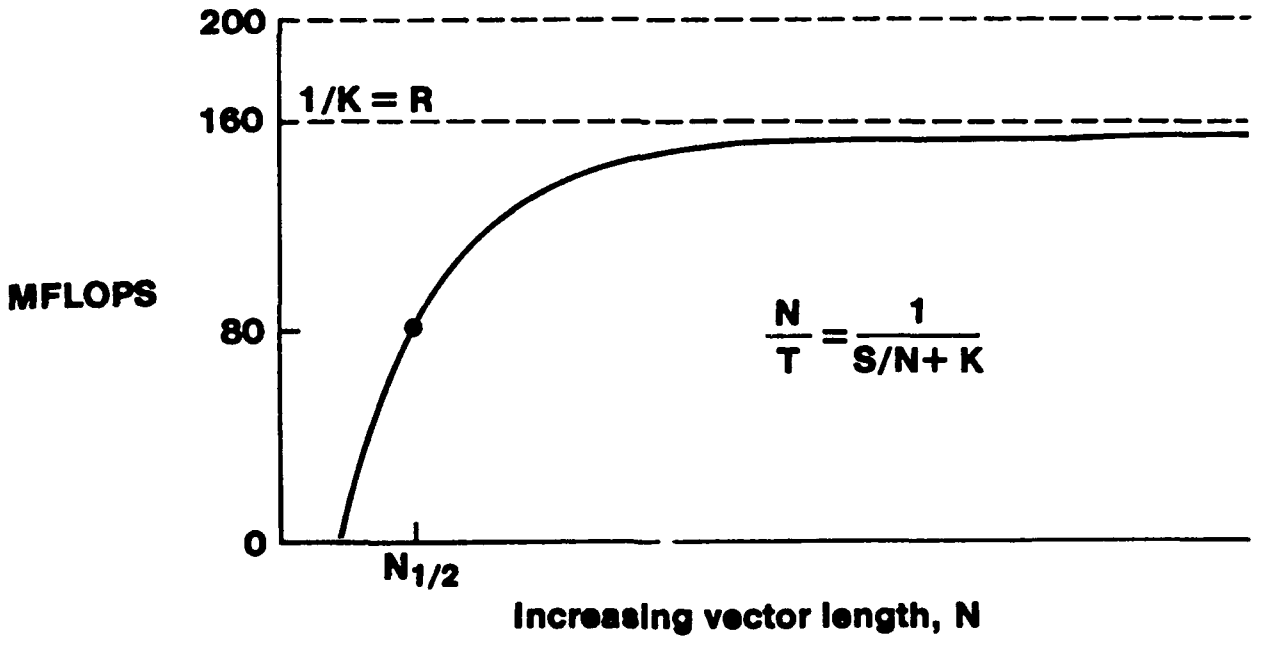


Figure 2. $N_{1/2}$ for a vector operation.

If vectors are very much shorter than $N_{1/2}$ performance is very poor. Vectors of length twice or three times that of $N_{1/2}$ usually operate significantly close to the peak performance.

One consequence of this heuristic parameter is that what is a "long" vector on one machine might be a "short" vector on another, for $N_{1/2}$ is a function of peak speeds and start-up time for vector operations. It is therefore worthy to study $N_{1/2}$ in more depth. In Reference 1 it was pointed out that multiple CPUs generally double the asymptotic peak rates as well as $N_{1/2}$. The following result was also established in that reference:

If an operation is the result of an M-segmented pipelined process with one result per cycle, then $N_{1/2}$ is simply equal to the number of pipe segments M.

The number of segments in the pipe has been called the "depth of parallelism" (defined in Reference 3) of a vector process, for at any instant in time a full pipeline is computing (at some stage) M operations simultaneously. This result, and the definition of $N_{1/2}$, makes it a very accessible parameter. Given an operation or algorithm, one can experimentally compute $N_{1/2}$ by plotting speed at various vector lengths to obtain a figure like Figure 2 and simply read an approximation to $N_{1/2}$ off the graph or table. For machine instructions such as simple vector adds or multiplies, one can use the above result and add the number of memory fetch/store segments to the number of floating point unit segments. The resulting sum is $N_{1/2}$. In multiple pipelined machines, or in multiple CPU machines, which produce more than one result per cycle, the above result is extended as follows.

Assume that a vector operation achieves a rate of P floating point results per machine cycle. This could be due to having P parallel pipelines sharing the chores by chaining or parallel processing the vector stream. Also assume that the vector operation is composed of a memory reference pipe chained with the floating point unit pipe so that the resulting combined segmentation is M segments long. Let C_T be the cycle time, then the time to perform a vector operation of N floating point results is given by

$$T = C_T\ (M + N/P)$$

The time to compute N operations per operation is given by

$$N/T = N/[C_T\ (M + N/P)]$$

or

$$N/T = \frac{1}{C_T(M/N + 1/P)}$$

Allowing N to approach infinity results in the expected asymptotic rate of P/C_T. Half of this performance rate is $P/(2*C_T)$. This is achieved when $N = P*M$. Thus, $N_{1/2} = P*M$. Simply stated:

For multiple pipelined operations (either chained pipes or duplicated pipes) that achieve P results per machine cycle, $N_{1/2}$ is equal to the total number of segments in the combined pipelined operations considered times P.

We now have a full set of tools to analyze vector latency in supercomputers. Experimentation for those machines available can be used to compute $N_{1/2}$, or for more

simple operations one can estimate the $N_{1/2}$ value from the number of segments in memory pipelines, segments in the floating point pipelines, and the number of results per cycle. One must take care in computing the result rate depending on paths-to-memory, stored register data allowed in the operation, and conflicts in data flow (such as bank conflicts or other memory delays such as cache misses, paging. etc.).

As an example of computed results, Table 6 gives $N_{1/2}$ values for several machines from Reference 4. The FORTRAN SAXPY operation, which is scalar times a vector with the result added to another vector, is used. This operation can be chained on machines that allow chaining.

Table 6. Values for $N_{1/2}$ (FORTRAN SAXPY)

Computer	$N_{1/2}$	FORTRAN Peak Performance
Cray-1	20	45
Cray X-MP	37	101
Cray-2	30	55
CYBER 205	238	170
Fujitsu VP-200	120	190
IBM 3090/VF	34	53
NEC SX-1	30	240
NEC SX-2	80	575

One can compute the results in Table 6 from hardware design information. For example, using the Cray X-MP, where there is a 14-cycle wait time for read, a 13-segment chained add/multiply, with 2 results per cycle, we can expect an $N_{1/2}$ of 54. One should also expect a peak rate of close to the maximum of 210 to 230, depending on the model. The FORTRAN results in Table 6 show a smaller $N_{1/2}$ and a smaller peak rate than theoretical. However, careful assembly coding can achieve the higher peak. If start-up time remains constant, the higher the peak rate, the larger the value of $N_{1/2}$. Reference 1 reports an experimentally computed $N_{1/2}$ value of close to 80 and peak of over 200 from Cray assembly coded loops. The theoretical model above does not consider the fact that all vector operations are modulo the length of vector registers, introducing some additional overhead and increases to theoretical $N_{1/2}$ values.

Another example illustrates how computing theoretical $N_{1/2}$ values can depend on often obscure information. The Fujitsu VP-200 has a 31-segment read pipe (33 for noncontiguous data). The two add and two multiply pipes have six and seven segments each. Assume the scalar value for the multiply is in a register and assume that the vector to be added is in the register as well. [This is not unreasonable since in Gaussian elimination, where this operation is often used, one can accumulate the inner loop results in a register via column-ordered elimination (Ref 1).] After 31 cycles the first pair of fetched elements reaches the registers. After 14 more cycles the first quartet of results is computed. The result rate is 4 per cycle (2 adds and 2 multiplies) after a total 45-cycle start-up. According to the theoretical result

$$N_{1/2} = 45 * 4 = 180$$

The experimental result was 120 and the theoretical result is 180. Disparities of this kind in $N_{1/2}$ values computed experimentally occur between assembly coded loops and FORTRAN. FORTRAN typically achieves lower peak rates and, consequently, smaller $N_{1/2}$ values. (A host of examples like this can be found in Reference 1.)

Another factor, not considered in the theoretical calculation, is the use of overlapping operations. It is possible to issue the next vector instruction in a sequence, before the first is completely done. This greatly reduces the start-up time of a vector operation since the pipeline filling process can be shared with the "emptying" of the previous vector operation. The Fujitsu VP-2600 can overlap as many as four vector instructions. Most of the time the programs used to time such operations issue nested do-loops of vector operations. Thus, in a series of vector operations, the segmented read pipe need not completely empty itself before a successive read can take place. This judicious scheduling could have the effect of reducing the wait time of the successive vector read operation. NEC machines do this as well.

For this reason, another parameter can be defined:

> *Sequenced $N_{1/2}$: The vector length to achieve half of peak performance for a vector operation within a sequence of vector operations.*

As one can observe, the calculation of $N_{1/2}$ is mathematically simple, but in practice, it is quite difficult owing to the lack of detailed hardware information, particularly as machines become more complex.

Since it is a primary purpose to update Reference 1, it is important to focus on recently designed supercomputers and examine their characteristics in vector computation. The focus of the discussion will be the following machines: the Cray Y-MP, NEC SX-3, Hitachi S-820/80, and Fujitsu VP-2600. These machines are likely to be competitors in 1990. The Cray Y-MP and Hitachi machines are available now, although Hitachi is not yet marketing the S-820 outside of Japan. While we have observed that the theoretical estimates of $N_{1/2}$ values can differ from FORTRAN, they do serve as a comparator of latency in vector operations when benchmarks are not available. The first ingredient in this computation is the memory latency measured in cycles of delay until first word availability. The second is the number of results produced per cycle once all the pipes are full. Finally, one must pick an operation or computation to analyze. We shall select the contiguously stored SAXPY operation discussed previously. However, we shall make the following simplifying assumptions:

- The operation is performed once.
- One vector operand is stored in the registers.
- The final result will not be stored (simulating an accumulation for a next iteration).

Table 7 lists the latency of the fetch, add, and multiply pipes measured in cycles (pipe segments). This is information to be used in the calculation of the theoretical $N_{1/2}$ listed in Table 8. Unfortunately, the values F, H, and N are considered proprietary as of the time of this writing. In the past the Japanese manufacturers have had large memory latencies when measured in cycles of delay for initial components of a vector load to register. One can estimate the values of F, H, and N in Table 7 based on other machine statistics (for example, see the discussion below).

After the pipeline is full, a number (equal to the length of the vector register) of operations are completed (one result per cycle per pipe) until the vector register length is exhausted. The next set of elements is then processed until the entire vector length is exhausted. The subsequent set is usually

fetched in overlapped fashion so that the pipe start-up time is not required (i.e., a prefetch has occurred). There may be a several cycle gap between sets. For the Cray designs the register length is 64 words. For the Japanese machines these vector "sets" are generally larger, but so is the overlapped latency. This effect will be ignored for all machines. Table 8 lists the number of results per cycle achieved and the estimated theoretical $N_{1/2}$ (for a single CPU).

Table 7. Vector Operation Latency

Computer	Memory Latency (cycles)	Floating Point Latency (add + multiply)
Cray Y-MP	20	6+7
Fujitsu VP-2600	F	11
Hitachi S-820	H	6+8
NEX SX-3	N	8+8

Table 8. Theoretical $N_{1/2}$ (Linked Triad, With One Vector Fetch)

Computer	Results/ Cycle	$N_{1/2}$
Cray Y-MP	2	66
Fujitsu VP-2600	16	16*F
Hitachi S-820	8	8*H
NEC SX-3	8+8	8*N

In the past, the primary reason the Japanese entries have such large $N_{1/2}$ values is the large memory latency. While this can be shortened considerably in a sequence of overlapped instructions in the newer designs, it still is of concern (enough for all three manufacturers to hold latency figures proprietary). An estimate (lower bound) of memory latency in cycles can be made by multiplying the ratio (memory speed)/(cycle time) times the number of full bandwidth elements delivered to registers per cycle per path. For the NEC SX-3 this lower bound estimate would be

```
(20 ns/2.9 ns) * 8 = 55.2
```

Clearly, there would be a number of potential reasons why this number might be larger owing to resolution of conflicts and internal microcode delays, etc. For the sake of discussion, assume that F, H, and N were on the order of 70. With this value of latency, the $N_{1/2}$ values for the Fujitsu VP-2600 and NEC SX-3 would be on the order of 1120 and 560, respectively. Using these theoretical estimates, and the heuristic rule of thumb, one can observe that, to be efficient in vector operations, one must have vector lengths on the order of 2000 to 3000 for the VP-2600 and 1000 to 1500 for the SX-3, while only on the order of 150 to 200 for the Cray machines. One must understand that greater peak performance generally will require larger latencies. Unless problem sizes and, consequently, vector lengths grow with CPU power, less efficient use of the hardware will result.

REMARK: *One should note the very exacting assumptions used in Tables 7 and 8. For memory-to-memory operations (i.e., requiring the final vector store), machines with path deficiencies would suffer lower peak performance and require even longer memory delays than listed in these tables. For example, a two-path machine would require an N-cycle wait plus start-up time for the final store operation, effectively reducing peak performance by 2. On the other hand, three-path machines would be able to almost completely overlap the start-up time of the store operation to achieve full performance, but an increased value*

of $N_{1/2}$ due to store pipe start-up time would result. Similarly, if two vectors had to be fetched (instead of assuming that one was already in the register), the one path-to-memory architecture would perform much worse.

Parallel Computation

All of the manufacturers represented in Table 8 have indicated intentions to offer multiple CPU machines. In fact, except for the Hitachi and Fujitsu machines listed, they all are multiple CPU machines as indicated in Table 1. Reference 5 details how compiler technology today, and for the near future, can exploit parallelism. There are a number of broad issues in parallelizing application programs. At the loop level a form of parallelization is easily implemented by the compiler. There are two possibilities. First, a nested loop can be divided among CPUs at the outer loop level giving each CPU a sequence of inner loop vector operations. This is the most advantageous way to deploy the multiple CPUs at the loop level. Vector lengths remain the same, and the number of vector operations remains the same. The overhead of parallelization is the only penalty. Unfortunately, not all loops are nested, and not all that are nested can be recognized as being independent and allow for this technique.

A second possibility is that a single loop is segmented and spread among available CPUs. This technique results in the length of each loop being divided by M, the number of processors used. This has two effects. First, what was once a single vector operation becomes M vector operations. Second, each vector operation is using a shorter vector. In this situation what was considered a long vector for a single CPU now must be M times longer. That is, what was an adequately long vector length for efficiency on single CPU vector operations may not be long enough to be efficient on multiple CPU machines. Consider an eight-CPU computer. If a single CPU vector operation had an $N_{1/2} = 150$, then 450 might be considered an adequate length to achieve efficiency on a single CPU, while a vector length of 3600 is necessary for a single unnested loop implemented on eight CPUs. This doesn't scale. That is, it is unlikely that an eight-CPU machine can handle problems uniformly scaled to be eight times bigger since, so far, multiple CPU supercomputers are not providing eight times the real memory of their single CPU versions.

It is quite possible that as parallel CPUs are exploited, the efficiencies enjoyed by vector CPUs will decrease for lack of algorithmic approaches that avoid the degradation of the second scenario above. In other words, the current trend to automatic parallelization must be used with an awareness that parallelizing vector instructions can produce less efficient vector operation in the individual CPUs. Parallelization from the top duplicates entire processes and offers the opportunity to gain the advantages of parallelism without the needless sacrifice of vector efficiency. To ameliorate this degradation users will have to increase the opportunity for outer loop or top-down parallelization.

While $N_{1/2}$ values are increasing due to parallelism within the CPU, as well as among CPUs, so are the memory sizes. The NEC SX-3 requires longer vectors to be efficient than other designs, yet it provides large enough memory and secondary storage to support more complex computations with, perhaps, longer average vector lengths. In the meantime, multiple pipeline machines with very high computation rates will probably be competitive on smaller problems. Both Fujitsu and NEC have attempted to ameliorate their shortcomings in start-up

times with techniques for overlapped instructions, appropriately placed caches, and large register sets.

Comments on the Fujii and Yoshihara Benchmark (Ref 6)

In Table 8, the supercomputers likely to compete in the 1990-91 period are listed. With the exception of the Cray Y-MP and Hitachi S-820, benchmarking these machines is not yet possible. Although the recent announcements from CRI indicate the Cray-3 is planned to be available in 1990, all indications are that it will not be available in large numbers until 1992. Fujii and Yoshihara (Ref 6) have recently benchmarked one particular program, an unsteady Reynolds-averaged Navier/Stokes code, on available Japanese and U.S. machines. Their work gives some insight to the issues discussed in previous sections and is worthy of discussion here. We can also use the data to conjecture on the likely performance of the newer machines on this application, as well as on other applications.

No single application program provides a "level playing field" to evaluate computer hardware products. In fact, many benchmarks provide misleading information about the specific application they are related to due to anomalies in the code structure, algorithm formulation, or coding techniques. Nevertheless, benchmarking provides, over time, an accumulated expectation of performance of various computers and hardware designs. Over time, codes adapt to the architectures that become popular. Certainly, many of today's application programs in many industries contain a much higher percentage of sound vectorizable code when compared to those of a decade ago.

The Fujii/Obayashi code used in Reference 6 has several shortcomings as an initial benchmark, which they point out. No inner loop vector exceeds 10,000. This is a sizeable inner loop, yet no machine benchmarked achieves more than 53 percent of peak performance, which would indicate some inhibitors to efficient vectorization. Inhibitors to vectorization fall into a number of categories:

1. Compiler barriers
    - The dependence analysis of compiler optimizers sometimes cannot vectorize loops without the user's knowledge (directive) that it is safe to do so.
    - Some compilers are inefficient by not recognizing "vectorizable" loops.
    - Sometimes compilers optimize loops with conditionals (i.e., IF tests) inefficiently.
2. Hardware barriers
    - Large start-up times for vector processes relative to vector lengths.
    - Path deficiencies.
    - Lower bandwidth from memory to the vector units than peak performance potential in floating point operations.

In the case of the Fujii/Obayashi benchmark code, the indications are that the compiler efficiency was very good in the sense of item 1. This would lead one to the conclusion that there were problems in the hardware data throughput (item 2).

Since the Y-MP is the only multiple CPU machine in the benchmark, elapsed time is provided as a means of comparison. The Y-MP, through the use of autotasking directives, was able to take advantage of simultaneous computation of the right-hand sides per iteration and other parallel tasks. The single CPU machines were, of course, required to do this sequentially, albeit at vector rates. Table 9 shows that the Y-MP was able to achieve an impressive reduction in elapsed time through parallel computation (a factor of over 7).

Table 9. Reynolds-Averaged Navier/Stokes (from Ref 6)

Computer	CPU (min)	Elapsed Time (min)
Cray Y-MP/832 (8 CPUs)	550	78
Fujitsu VP-400E	255	258
Hitachi S-820	162	164
NEC SX-2A	200	201

The Japanese machines, as detailed in earlier sections, are moving toward multiple CPUs, employing overlapped instruction streams to reduce memory latency and providing the necessary parallelization tools. There is little known about the overhead of the equivalent of macro- and microtasking on the NEC machine, but this no doubt will be important. If the SX-2 were a four-CPU machine (in the style of the SX-3), the opportunity to achieve the parallelization speedup would exist, as it did for the Y-MP.

To put some perspective on the benchmark as it might perform on future machines, examine Table 10 (from Reference 6), which lists the peak performances achieved on the single CPU runs.

As a (perhaps poor) predictor, we could assume that the SX-3 design could achieve the same percentage efficiency of 41 percent as the SX-2. This would put the sustained single CPU performance at 2.26 GFLOPS. The Y-MP was able to achieve a factor of 7 (out of a possible 8) increase using eight CPUs, suggesting a reasonable amount of exploitable parallelism in the problem. Should the SX-3 achieve only a factor of 3 (out of its possible 4), this would achieve a 6.7-GFLOP sustained performance on this benchmark, without undue optimization effort. While these figures enjoy a large degree of conjecture, it would seem that a reasonable goal for sustained performance on CFD and other scientific applications could be 10 GFLOPS in the early to mid-1990s. This is a significant increase in what has been thought of as "obtainable" computing power in the near term. It was only a few years ago that 10 GFLOPS peak performance was thought an incredible figure, and we are quickly closing on this being a realizable sustained performance goal on an actual application.

Table 10. Single CPU Performance

Computer	Peak GFLOPS	Actual	Ratio
Cray Y-MP	0.334	0.175	0.524
Fujitsu VP-400E	1.700	0.395	0.232
Hitachi S-820/80	3.000	0.602	0.201
NEC SX-2A	1.300	0.414	0.319

SUMMARY

It is clear that the "balance" of the Y-MP's single CPU architecture and the tools to efficiently utilize its multiple CPUs allow it to compete with hardware with much higher peak performance rates. It is equally clear that the next generation of Japanese machines promises to be much more competitive than their current offering. While their latency to memory will continue to drive their respective $N_{1/2}$ values higher, multiple CPUs, overlapped instructions,

larger memories, and aggressive compiler tools will provide them a technological base to compete on very large problems.

Over the longer term, all the high-end manufacturers will receive serious challenge from the more massively parallel machines. It is not clear that machines with 500 to 65,000 CPUs will initially have to compete as general purpose machines. For specialized problems that are amenable to highly parallel computing designs, the price/performance picture for highly parallel machines may be worthwhile. In single application environments, or as nodes on an integrated system, massively parallel machines may gain market acceptance. The decade of the 1990s will, no doubt, be challenging for users and manufacturers alike.

REFERENCES

1. W. Gentzsch and K.W. Neves, *Computational Fluid Dynamics: Algorithms and Supercomputers*, edited by H. Yoshihara, NATO AGARDograph No. 311 (March 1988).

2. R. Hockney and C. Jesshope, *Parallel Computers 2* (Adam Hilger Ltd., Bristol, 1988).

3. K. Neves, "Mathematical libraries for vector computers," *Computer Physics Communications* **26**, 303-310 (1982).

4. J. Dongarra, Lecture Notes, Supercomputer Seminar, San Diego State University AMCEE Education Seminar, KMPS-TV (4 March 1987).

5. J. Kowalik and K. Neves, "Supercomputing: Issues and challenges," Keynote Paper, *Second NATO Workshop on High Speed Computing*, edited by Kowalik, (Springer Verlag, Norway, June 1989) (to appear).

6. K. Fujii and H. Yoshihara, "A Navier/Stokes benchmark for Japanese and U.S. supercomputers," *Scientific Information Bulletin* **14**(2), 69-74 (1989).

Kenneth W. Neves received a B.A. degree in mathematics from California State University at San Jose and M.A. and Ph.D. degrees in mathematics (specializing in numerical analysis) from Arizona State University while holding a National Science Foundation Research Fellowship. Dr. Neves is currently Manager of Research and Development Programs in the Boeing Computer Services Company (BCS) in Seattle, WA. His primary responsibilities include the definition and management of research and development programs encompassing most aspects of engineering/scientific computing, notably hardware, software, and algorithms. A major activity under Dr. Neves' direction is the High Speed Computing Program, which currently has three parallel processors and advanced work station equipment. Previously he was Manager of the Computational Mathematics Group responsible for maintenance, development, and certification of mathematical and statistical software libraries resident on the BCS computer centers nationwide. Before joining BCS in 1975, Dr. Neves was Senior Mathematician for the Nuclear Power Division of Babcock and Wilcox Company, Lynchburg, VA. He is currently vice-chair and co-founder of the SIAM Special Interest Group on Supercomputing, the chair and founder of the Special Interest Committee on Applications and Algorithms of the Cray User Group, and serves on the IEEF Subcommittee on Supercomputing.

A REVIEW OF ADVANCED SEMICONDUCTOR PROCESSING AT TOHOKU UNIVERSITY'S LABORATORY FOR MICROELECTRONICS

Henry Berger and Jeffrey M. Davidson

At Tohoku University, Prof. Tadahiro Ohmi has helped set up a cleanroom facility to develop the design and processing technology for next-generation electronic chip manufacture. A review is given of the various programs underway at this facility.

INTRODUCTION

The Laboratory for Microelectronics, also known as the Super Cleanroom (SCR), at Tohoku University's Research Institute of Electrical Communication*, was constructed in March 1986 at a cost of about $13.3 million [funded in part by Japan's Ministry of Culture and Education (Monbusho program)]. Within the Division of Microfabrication (directed by Prof. Nobuo Mikoshiba), Prof. Tadahiro Ohmi's research group consists of 40 to 50 staff and students.

Prof. Ohmi and his group have developed what is termed an "Ultra Clean Technology" (UCT) to set the guidelines (Ref 1,2) for successfully producing future generation, or ultra large scale integrated (ULSI), electronic circuits. His specific device design goals include 100-Mb LSI solid-state imaging sensors (Ref 3-5) using ~0.1 μm CMOS structures combined with new bipolar designs (so-called Bi-CMOS devices). UCT guidelines are applied, prototype fashion, to the semiconductor fabrication equipment at the SCR.

Currently, most Super Cleanroom projects are in research and development (R&D) environments. The majority of the UCT semiconductor fabrication schemes, though highly innovative, are not yet ready for direct transfer to industrial production line environments. Presently, UCT experimental processing steps require a high degree of operator attention and prohibitively long run times (for chamber bake-out, wafer load, moisture purge, etc.); use small diameter wafers (32-mm wafers rather than industrial sized, >100-mm wafers); and allow only low wafer throughputs (wafers processed/time). Thus, these programs might be viewed as experiments that set the boundary conditions for fabrication of future generation integrated circuit (IC) chips.

On the other hand, actively supported by the semiconductor industry, Tohoku University has developed products that serve to increase the cleanliness of semiconductor processing. Research work has been aimed at showing how these improvements in process purity lead to more efficient device fabrication. The significant amount of interest by the industry in the work of Prof. Ohmi is evident by the high level of industrial collaboration taking place at Tohoku University (Table 1). The English language technical publications and presentations from Tohoku University have been specifically cited by the Ministry of International Trade and Industry (MITI) (Ref 6) as examples of Japanese technology sharing and transfer in the area of semiconductor processing.

* 1-1, Katahira 2-chome, Sendai-shi, Miyagi-ken 980, Japan; tel: 0222-27-6200; headed by Prof. Shunichi Iwasaki.

Table 1. Industry Researchers at Tohoku University's Super Cleanroom Laboratory

NTT
Canon Inc.
Hitachi Plant Co., Ltd.
Oki Electric Industry Co., Ltd.
Alps Electric Co., Ltd.
Tokyo Electronics OK
Seiko Instruments and Electronics Co., Ltd.
Kokusai Electronics Co., Ltd.
Takasago Thermal Engineering Co., Ltd.
Hashimoto Chemical Industrial Co., Ltd.
Tokuyama Soda Co., Ltd.
Nihon Chemical Industrial Co., Ltd.
Osaka Sanso Kogyo Ltd.
Nihon Sanso Co., Ltd.
Daido Sanso K.K.
Mitsubishi Gas & Chemical Co. Inc.
Siemens AG[a]
The BOC Group Inc.
Applied Materials (U.S.A. and Japan Inc.)
SAES Getters U.S.A. Inc.
IBM

[a]Under the direction of Professor Mikoshiba, Tohoku University.

In October 1988, Prof. Ohmi helped establish the Institute of Basic Semiconductor Technology Development, also known as the Ultra Clean Society. This Tokyo-based organization acts as a clearing house for semiconductor-related technical information. It is intended that this information will be circulated through newsletters and symposia to member companies and universities (numbering about 155) representing all segments of the Japanese electronics industry.

Over the past few years there have been a significant number of publications describing the work of Prof. Ohmi and his coworkers at the SCR. Indeed, during 1988 alone, Prof. Ohmi's contribution to the electronics industry included 20 patents, coauthorship of over 100 technical papers, and numerous guest and keynote addresses. He has also been invited to a number of leading semiconductor companies in the United States for consultation.

The purpose of this report is to bring together and review the various programs underway at Tohoku University's Super Cleanroom. Assessments on the production line feasibility of UCT or discussions of the marketability of Prof. Ohmi's work will not be considered in this report.

ULTRA CLEAN TECHNOLOGY ITEMS

Work at Tohoku University's Super Cleanroom is aimed at developing higher levels of contamination control for advanced IC device production. The primary thrust of this effort is focused on minimizing the physical and chemical contaminants in the semiconductor fabrication process and in the raw materials used. It is widely acknowledged (Ref 7) that such purity improvements will be necessary in order to economically produce future generations of electronic chips. Integrated into this effort is the development work for new and innovative device designs, along with the concatenate processing techniques.

As reviewed below, a broad range of processing issues is considered--the cleanroom facility itself has been made "cleaner" and more efficient in operation; the handling and control of the chemical (Ref 8-11) and gas (see below) raw materials have been developed for higher levels of purity; and in key fabrication steps, processing tools and run procedures have been redesigned to allow lower temperature processing and higher levels of cleanliness. In this report gas technology items are emphasized.

SUPER CLEANROOM FACILITY

With a process floor space of 600 m^2 (~6,500 ft^2), Tohoku University's Super Cleanroom itself has been the starting point for a significant amount of the R&D work. This has led to facility improvements that provide an overall contamination-free setting for electronic chip processing. In terms of UCT, contamination control is used here in a broad sense to include not only chemical and particle impurities but also detrimental sources of thermal, vibrational, and electromagnetic field contaminations.

Fabrication work at the SCR is predominantly silicon based and a concerted effort has been aimed at improving contamination control in the cleanroom so that Si wafer surfaces remain clean. Cleanroom air-handling systems have been designed (Ref 12,13) to efficiently control the room air flow, temperature, and humidity. Also, the sodium content in room air is minimized. In this way, condensation of chemical impurities on wafer surfaces can be reduced and the number of particles can be minimized. Cleanroom personnel garments have also been studied (Ref 14) to further eliminate sources of particle contamination.

Other areas of development in the cleanroom include: noncontaminating wafer transport schemes (Ref 15); cleanroom vibration control, voltage charging, and magnetic field control (Ref 16,17); and deionized (DI) water handling (Ref 16).

CONTAMINATION CONTROL OF WAFER SURFACES

Control of particle contamination on wafer surfaces has been related to the effects of applied electrostatic charging of wafers and wafer handling procedures (Ref 17). To actively remove particles from wafers, drying equipment has been developed that uses an ultra-pure isopropanol wash (Ref 18). Also, an Ar sputter-cleaning process for silicon surfaces has been reported (Ref 19) using low kinetic energy particle bombardment. This in-situ cleaning serves as a necessary pretreatment for the low-temperature sputter deposition process of epitaxial silicon or Al metallization.

ULTRA CLEAN GAS PROCESSING TECHNOLOGY

Increasingly, during semiconductor IC manufacturing, the wafer is in almost constant contact with process gases. Much attention has thus been given to the precise control of process gas parameters. For example, liquid nitrogen and liquid argon supplied to the SCR typically contain impurity levels in the 1 to 10 parts per billion (ppb) range. These purity specifications are more stringent than presently required by even the most advanced semiconductor facilities. Such high purity levels for both atmospheric and specialty gases have resulted from the collaboration (Ref 20) between Prof. Ohmi and Japanese gas supply and gas-handling equipment companies. For example, in the transfer process from tank lorry to the SCR's cryogenic storage tanks, liquefied gases are loaded via a differential pressure filling technique. Transfer pumps, conventionally used, have thus been eliminated because of their associated potential to contaminate the gases with carbon from pump seals.

The high purity levels of gases delivered to a facility must be maintained to the point-of-use, where the wafer is processed. Much of the Tohoku University resources have been directed towards developing a gas delivery system (Ref 21-23) that will not add contaminants to the gas. The SCR tubing design features minimization of external leak rates and dead-space volumes while maximizing gas purging capability. Also, as a result of Prof. Ohmi's widespread use of bakable materials in tubing components (i.e., all metal), a reduction of outgassed moisture and organic contamination has been measured (Ref 24). As a specific example, consider Figure 1, which compares the gas displacement time, a measure of residual gas from dead-space, between newly developed metal diaphragm valves and conventional bellows valves.

HIGH SENSITIVITY GAS ANALYSIS

The concurrent development of high sensitivity gas analytical capabilities to measure the diminishing impurity levels (ppb, ppt) of the ultra clean gases has been necessary. From Prof. Ohmi's group, atmospheric pressure ionization mass spectrometry (APIMS) measurements have been reported (Ref 25,26) that represent next-generation detection limits for impurities in gases.

APIMS uses a selective, two-stage ionization process that increases impurity-to-host gas ratios for mass spectrometry sampling (Ref 27). In addition to the gas purity monitoring at Tohoku University, APIMS contributes to research in a variety of areas. These include the measurement of outgassing levels of contamination from process equipment as well as from new semiconductor materials. These tests are accomplished by measuring the impurities in carrier (purified) gas after it has flowed past test pieces subjected to temperature cycling.

STAINLESS STEEL PASSIVATION

As the purity specifications for wafer processing increase, use of electropolished stainless steel is gaining wide acceptance within the semiconductor industry for reactor chambers (chemical vapor deposition (CVD), sputter deposition, and etch) and gas tubing systems. It is thought that these specially prepared steels can be used to decrease the level of particle and chemical contamination from walls to process environments. However, very long periods of time may be required for a piping system to

yield ultra high gas purity levels, even using electropolished stainless steel tubing. This is illustrated in Table 2. An integral part of the Tohoku University contamination control program involves decreasing the "clean-up" time for the overall gas delivery system (thus decreasing the "start-up" time for a semiconductor fabrication line). Research work is directed to further improve the inertness of stainless steel surfaces by forming thin oxide passivation layers (Ref 28). Over conventional, electropolished stainless steel, the passivated surface allows faster clean-up times after being exposed to moisture contamination (Figure 2). In addition, passivated steel surfaces show enhanced resistance to corrosive gases (Ref 28).

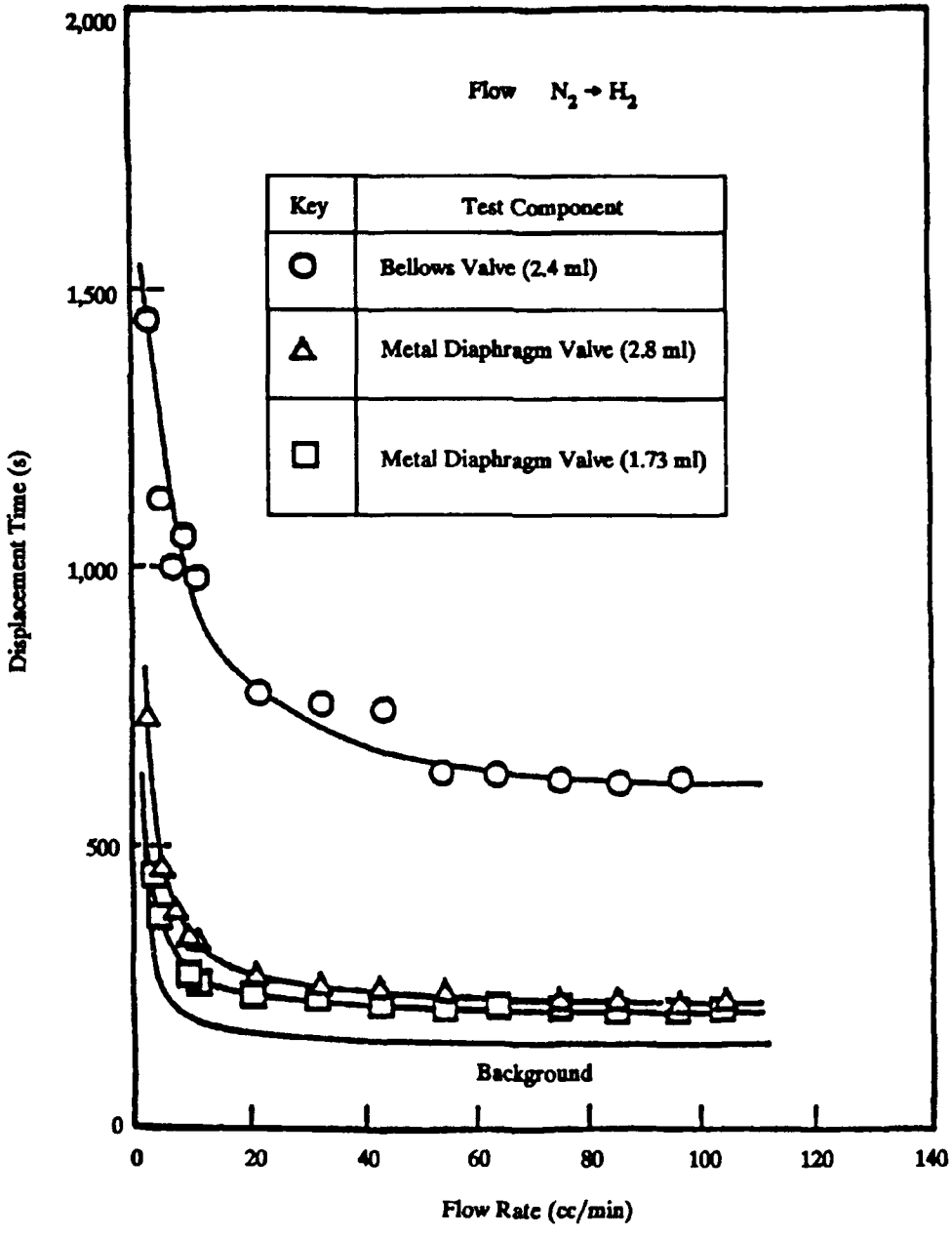


Figure 1. Gas displacement characteristics for various commercially available valves.

Table 2. Impurity Contents (ppb) in Nitrogen Gas at Point-of-Use as a Function of System Operation Time

Operation Time (h)	Condition	H_2O	O_2	NO	CO_2	T.H.C.
100	bulk	170	9.0	4.2	2.7	6.5
	purified	85	5.0	1.3	1.0	3.3
500	bulk	72	6.8	7.2	9.4	0.2
	purified	15.5	1.7	1.5	0.5	<0.1
4,300	bulk	4.6	0.9	<0.1	2.0	0.1
	purified	4.0	0.5	<0.1	<0.1	<0.1
11,000	bulk	2.0	0.9	<0.1	0.2	<0.1
	purified	1.8	<0.1	<0.1	<0.1	<0.1

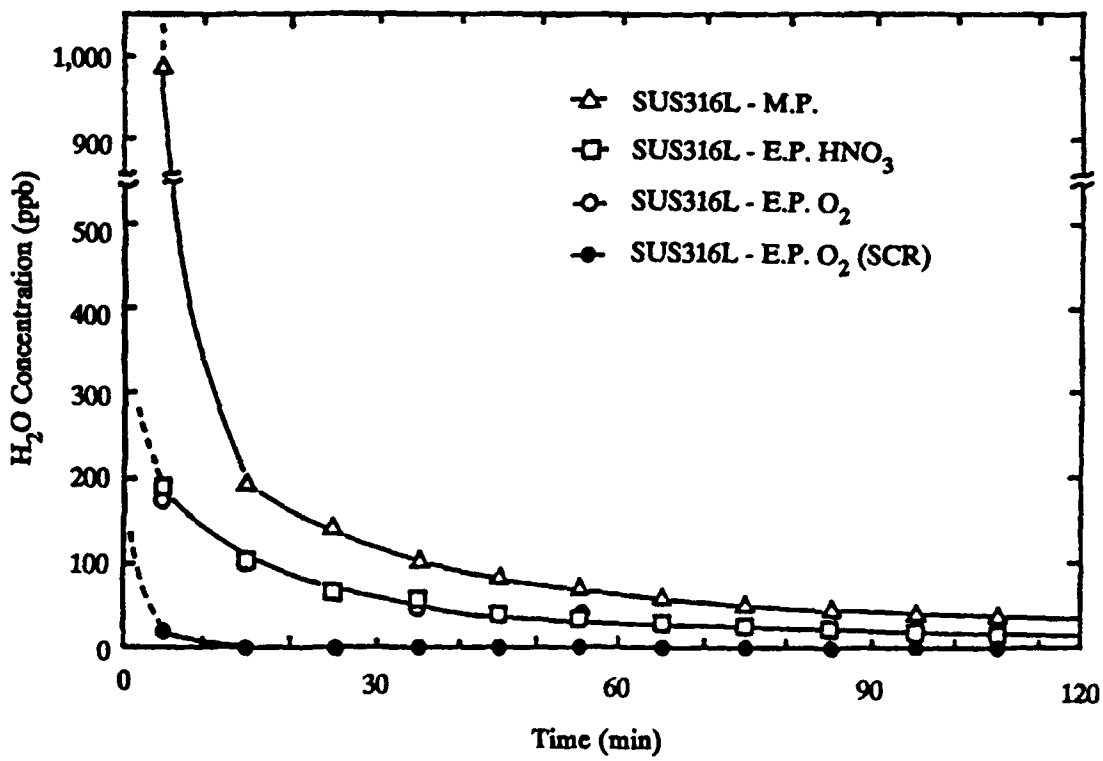


Figure 2. Time dependence of water concentration in argon gas passed through passivated stainless steel tubes at 25 °C.

UCT SEMICONDUCTOR PROCESSING TOOLS

Shrinking ULSI device geometries are increasingly placing more stringent demands on the fabrication process of IC materials. For example, processing techniques that feature lower temperatures are required in order to preserve doping distributions in shallow junctions and to allow the formation of multilevel metallizations. The quality of epitaxial layers is becoming increasingly more important for new silicon-on-insulator (SOI) types of isolation patterns. More reliable dielectric thin film materials are necessary for the thinner gates of advanced device designs.

Prof. Ohmi has set out to address these issues by systematically improving the purity of the ambient in the reactor chamber. This has led to the design of new processing tools and techniques. Outlined below are the specifications for some of these prototype tools that are under development at the Super Cleanroom. Generally, UCT equipment for semiconductor processing uses quartz or stainless steel reactor chambers that are high vacuum compatible (for initial leak checking and outgassing), are bakable (for moisture outgassing), and feature some sort of wafer load-locking system that uses nitrogen or argon gas purging. In addition, SCR equipment is connected to the high purity gas sources and delivery manifolds, discussed above.

Epitaxial Depositions With Simultaneous Doping

- In collaboration with Prof. Mikoshiba, improvements in selective low-pressure CVD (LPCVD) of Si (Ref 29,30) or Ge epitaxy (Ref 31) are being pursued. Tool #1 is a hot-walled, resistance-heated (600 to 800 °C) quartz reactor with batch loading through an "N_2 purge box." This equipment is designed for selective Si epitaxy (at 650 °C, with no HCl) or Ge epitaxy (350 °C) for contact hole filling. Reactor #2, with selective components made from electropolished stainless steel, is rf heated and has full load-locking capability (using N_2). Preliminary results include depositions with good Si thickness uniformity over an 8-inch susceptor (within 3 percent), without polymerization on the chamber walls. These two reactors are under the direction of Prof. N. Mikoshiba.

- The CVD of epitaxial Si (Ref 32) in a quartz, rf-heated (cold-walled) reactor has full load-locking capability, with high vacuum compatible, stainless-to-quartz flanges. This tool features a new carbon susceptor designed to limit outgassing contamination to the deposition process. Reported results include defect-free epitaxial Si films, deposited at 100 nm/min, for temperatures as low as 900 °C. Also, p^+n diodes are reported with low reverse current ($< 10^{-10}$ A/cm^2).

- A CVD process for epitaxial Si features a gas supply design in the reactor chamber (*gas-jet*) such that the predeposition reaction is confined near the region of the substrate surface. This leads to decreased amounts of deposition on the reactor chamber walls which, in turn, reduces equipment contamination and maintenance. The reactor chamber is fully load-locked and lamp heated (xenon) and uses high purity disilane for depositions at 10^{-3} Torr. Reported results include high rates of epitaxial Si deposition, at ~540 to 700 °C, for via hole filling with good step coverage (Ref 33-35).

- Epitaxial silicon is also deposited via rf-dc coupled bias sputtering using a load-locked, passivated, stainless steel reactor chamber. With precise dc-biased control of the argon plasma energy distribution, low kinetic energy depositions are possible (thereby minimizing radiation damage to the substrate). Simultaneous impurity doping for epitaxial Si films is reported for very low temperatures (~370 °C) (Ref 36-38).

Thin Film SiO_2 Growth

- Lamp-heated (cold-walled) thermal oxidation (Ref 39) of Si wafers is carried out in a full load-locked quartz tube with stainless-to-quartz flanges (for high vacuum capability) using a floating magnetic loading arm for particle-free loading. For the dry oxidation process steps (no HCl added), special low nitrogen containing oxygen gas*, made from the electrolysis of pure DI water, is used. Ultra clean oxidation for gates is reported to give ideal SiO_2/Si interface properties.

- The ultra clean gate oxidation effort is coupled with a project aimed at modeling and controlling native oxide (room temperature) growth (Ref 40). For thin SiO_2 gates (<5 nm), uncontrolled native oxide growth (~0.5 nm) can lead to serious problems in processing uniformity. Prof. Ohmi reports limiting the growth of native oxides on bare Si wafer surfaces by using air ambients with very low moisture concentrations or pure DI water containing low levels of dissolved oxygen. Also, an anhydrous HF process has been proposed that selectively etches native oxide species from thermal oxide films (see Etching below).

Metallization

- Bias sputtering of Al (Ref 41) and Cu (Ref 42) is carried out in stainless steel reactors using high purity argon gas and metal targets. These depositions take place using the same type of precise rf-dc coupled plasma control used for the sputtered epitaxial Si depositions (see above). Both metallizations are low-temperature operations with Al films reportedly showing increased resistance to hillock formation during subsequent post-metal annealing.

Etching

- Prof. Ohmi has contributed (Ref 43) to the development of a dry etching process that uses anhydrous (high purity) HF gas for the selective removal of native oxides from Si surfaces. Native oxides are etched off of bare silicon without damaging regions of the wafer covered by thermal oxides. In conjunction with this, ways to then remove unwanted fluorine residues are proposed.

- Sputtering processes are also applied to plasma etching for high accuracy pattern formation. Tohoku University etching programs include work aimed at improving plasma energy distribution control, development of corrosion-resistant reactor chamber surfaces, and utilization of electron cyclotron resonance (Ref 44) for reactive ion beam etching (RIE).

*For example, for high purity gases: Osaka Sanso Kogyo Ltd., 1-14, Miyahara 4-chome, Yodogawa-ku, Osaka 532, Japan, tel 06-396-3168; Airco Electronic Gases, Research Triangle Park, NC 27709, tel 201-464-8100.

Ion Implantation

- The SCR ion implantation system (Ref 45) features an ultra clean gas supply system and a high purity implant chamber capable of ultra-high vacuum. Arsenic implanted pn junctions with superior IV characteristics have been reported. Also, improvements in reverse-bias current are associated with lower levels of defect generation during post-implant anneals (for n^+p diodes).

Lithography

- Tohoku University lithography programs for submicron patterning include work with an e-beam stepper system and development of new photoresist technologies. Results from these programs are, as yet, unpublished.

SUMMARY

Prof. Ohmi's Ultra Clean Technology can be considered as a criterion for cleanliness in any one of the many aspects of semiconductor processing. As such this technology is increasingly being used by the semiconductor industry as a value of merit or certification for: (1) a facility for the preparation of semiconductor raw materials (Ref 40), (2) specific semiconductor process steps (the Tohoku University Super Cleanroom projects), or (3) a complete chip fabrication line (not yet built). Also, the specifications of ultra clean technology change with new development work. An upgraded, new cleanroom is presently being constructed in order to incorporate some of the most recent Tohoku University advances.

To date, there are scant published data that systematically relate processing techniques and raw materials to the resulting device properties and yields. Prof. Ohmi has been one of just a handful of researchers contributing in this area. In addition, he has managed to establish a collaborative environment in the semiconductor industry among countries and companies that are normally intensely competitive.

ACKNOWLEDGMENTS

The authors express their thanks for the persistent support from the staff of the Tohoku University Super Cleanroom, specifically Prof. Tadahiro Ohmi, Prof. Tadashi Shibata, Prof. Mizubo Morita, and Mr. Kazuhiko Sugiyama. Also, the help of Osaka Sanso Kogyo Ltd. technical personnel in Sendai, Mr. Masakazu Nakamura, Mr. Yasumitsu Mizuguichi, and Mr. Fumio Nakahara, was indispensable. The support of Mr. Masatoshi Goto, Mr. Yoshiyuki Nakahara, Mr. Satoshi Mizogami, and Mr. Mike Solomon, Osaka Sanso, is also gratefully acknowledged.

REFERENCES

1. T. Ohmi, "Ultraclean technology: ULSI processing's crucial factor," *Microcontamination* 6(10), 49 (October 1988).

2. T. Ohmi, "What's the contamination control target in ULSI manufacturing?" *Proc. Microcontamination Conf. & Expo. '88* (November 1988), p. 5.

3. N. Tanaka, S. Hashimoto, M. Shinohara, S. Sugawa, M. Morishita, S. Matsumoto, Y. Nakamura, and T. Ohmi, "A 310k pixel bipolar imager (BASIS)," *Digest of Tech. Papers - 1989 IEEE International Solid-State Circuits Conf.* (February 1989), p. 96.

4. N. Tanaka, T. Ohmi, and Y. Nakamura, "A novel bipolar imaging device with self-noise-reduction capability," *IEEE Transactions on Electron Devices* 36(1), 31 (1989).

5. N. Tanaka, T. Ohmi, Y. Nakamura, and S. Matsumoto, "A low-noise Bi-CMOS linear image sensor with auto-focusing function," *IEEE Transactions on Electron Devices* 36(1), 39 (1989).

6. Y. Honda, "Industry news," *Microcontamination* 6(9), 8 (September 1988).

7. C.M. Osburn, H.R. Berger, R.P. Donovan, and G.W. Jones, "The effects of contamination on semiconductor manufacturing yield," *Journal of Environmental Sciences* 31(2), 45 (1988).

8. S. Seitaro, I. Sugawara, F. Tanaka, and T. Wakabayashi, "Stabilizer-free high purity hydrogen peroxide for ULSI fabrication," *Proc. Ninth Int. Symp. Contamination Control (High Purity Chemicals I)* (September 1988), p. 41.

9. H. Kikuyama and N. Miki, "Property-controlled high purity buffered oxide etchants for ULSI processing," *Proc. Ninth Int. Symp. Contamination Control (High Purity Chemicals I)* (September 1988), p. 45.

10. H. Kikuyama, N. Miki, J. Takano, and T. Ohmi, "Developing property-controlled, high purity buffered hydrogen fluorides for ULSI processing," *Microcontamination* 7(4), 25 (April 1989).

11. S. Hashimoto, M. Kaya, and T. Ohmi, "Improving and maintaining electronics-grade chemical quality requires technological advances," *Microcontamination* 7(6), 25 (June 1989).

12. T. Ohmi, N. Mikoshiba, and K. Tubouchi, "Super clean room system - Ultra clean technology for submicron LSI fabrication," *Proc. of Electrochem. Soc. ULSI Science and Technology Symp.*, PV87-11 (Abst. No. 212) (1987), p. 761.

13. T. Ohmi and T. Shibata, "Personnel training for cleanrooms - Cleanliness management for an ultra-high-grade super cleanroom," *Proc. Ninth Int. Symp. on Contamination Control* (September 1988), p. 267.

14. N. Shiromaru, K. Shimoda, S. Shibuya, T. Saiki, M. Morisaki, T. Yoneda, T. Takenami, and T. Ohmi, "Measurement of the number of particles on cleanroom garment," *Proc. Ninth Int. Symp. on Contamination Control* (September 1988), p. 77.

15. T. Ohmi, M. Onodera, G. Sato, T. Shibata, and M. Morita, "Ultra-high-vacuum compatible wafer transport and holding system using electro-static chucks," *Proc. Electrochem. Soc. 174th Mtg., Extended Abstracts* (Abst. No. 407) (October 1988), p. 596.

16. K. Yabe, Y. Motomura, H. Ishikawa, T. Mizuniwa, and T. Ohmi, "Responding to the future quality demands of ultrapure water," *Microcontamination* 7(2), 37 (February 1989).

17. H. Inabu, T. Takenami, and T. Ohmi, "Evaluation of air velocity, air flow distribution and scattering of dust-charged particle adhesion to electrostatic charged wafer," *Proc. of Eighth Symp. on ULSI Ultra Clean Technol.*, Tokyo (January 1989), p. 247.

18. H. Mishima, T. Mizuniwa, M. Abe, T. Ohmi, and T. Yasui, "High purity isopropanol and its application to particle-free

wafer drying," *Proc. Ninth Int. Symp. on Contamination Control* (September 1988), p. 446.

19. T. Ohmi, T. Ichikawa, T. Shibata, K. Matsudo, and H. Iwabuchi, "In situ substrate-surface cleaning for very low temperature silicon epitaxy for low-kinetic-energy particle bombardment," *Appl. Phys. Lett.* 53(1), 45 (1988).

20. S. Mizogami, Y. Kunimoto, and T. Ohmi, "Ultra clean gas transport from manufacture to users by newly developed tank lorries and gas storage tanks," *Proc. Ninth Int. Symp. on Contamination Control* (September 1988), p. 352.

21. K. Sugiyama and T. Ohmi, "Part I: ULSI fab must begin with ultraclean nitrogen system," *Microcontamination* 6(11), 49 (November 1988).

22. Y. Kanno and T. Ohmi, "Part II: Components key to developing contamination free gas supply," *Microcontamination* 6(12), 23 (December 1988).

23. K. Sugiyama, T. Ohmi, T. Okumura, and F. Nakahara, "Electropolished, moisture-free piping surface essential for ultrapure gas system," *Microcontamination* 7(1), 37 (January 1989).

24. Y. Kanno and T. Ohmi, "Development of contamination-free gas components and ultra clean gas supply system for ULSI manufacturing," *Proc. Ninth Int. Symp. on Contamination Control* (September 1988), p. 345.

25. K. Sugiyama, F. Nakahara, T. Okumura, T. Ohmi, and J. Murota, "Detection of sub ppb impurities in gases using atmospheric pressure ionization mass spectrometry," *Proc. Ninth Int. Symp. on Contamination Control* (September 1988), p. 332.

26. F. Nakahara, T. Ohmi, K. Sugiyama, Y. Mizuguichi, H. Berger, M. Nakamura, H. Mihara, and K. Sato, "Ultra clean gas dilution system and its evaluation by APIMS," *Proc. Eighth Symp. on ULSI Ultra Clean Technol.*, Tokyo (January (1989), pp 49-75; *Proc. Microcontamination Conf. & Expo. '89* (at press).

27. H. Kambara and I. Kanomata, "Determination of impurities in gases by atmospheric pressure ionization mass spectrometry," *Analytical Chemistry* 49(2), 270 (1977).

28. T. Ohmi, T. Okumura, K. Sugiyama, F. Nakahara, and J. Murota, "Outgas-free corrosion-resistant surface passivation of stainless steel for advanced ULSI processing equipment," *Proc. Electrochem. Soc. 174th Mtg., Extended Abstracts* (Abst. No. 396) (October 1988), p. 596.

29. J. Murota, N. Nakamura, M. Kato, N. Mikoshiba, and T. Ohmi, "Ultraclean low-pressure CVD technology with high selectivity," *Electrochem. Soc. Proc. Advanced Materials for ULSI*, PV-88-19 (Abst. No. 192) (1988), p. 299.

30. J. Murota, N. Nakamura, M. Kato, N. Mikoshiba, and T. Ohmi, "Low-temperature silicon selective deposition and epitaxy on silicon using the thermal decomposition of silane under ultraclean environment," *Appl. Phys. Lett.* 54(11), 1007 (1989).

31. S. Kobayashi, M. Cheng, A. Kohlhase, J. Morita, and N. Mikoshiba, "Selective germanium epitaxial growth on silicon using

CVD technology with ultra-pure gases," to be published in *Jpn. Soc. of Appl. Phys. Int. Conf. Solid State Devices and Materials* (August 1990).

32. T. Ohmi, S. Yoshitake, J. Murota, T. Okumura, and H. Aikawa, "High quality epitaxial silicon layers formed by ultra clean technology," *Electrochem. Soc. Proc. Advanced Materials for ULSI*, PV-88-19 (Abst. No. 189) (1988), p. 80.

33. T. Ohmi, M. Morita, T. Kochi, M. Kosugi, H. Kumagai, and M. Itoh, "High-rate growth at low temperature by free jet molecular flow: Surface reaction film formation technology," *Appl. Phys. Lett.* 52(12), 1173 (1988).

34. T. Ohmi, H. Kumagai, M. Morita, M. Itoh, T. Kochi, M. Kosugi, and G. Tei, "Surface reaction film formation utilizing free jet molecular flow," *Electrochem. Soc. Proc. Advanced Materials for ULSI*, PV-88-19 (Abst. No. 185) (1988), p. 36.

35. T. Ohmi, M. Kosugi, M. Morita, G.S. Jong, and H. Kumagai, "A step coverage and a hole filling of Si film by surface reaction film formation technology," *Proc. Electrochem. Soc. 175th Mtg., Extended Abstracts*, vol 89-1 (Abst. No. 190) (May 1989), p. 276.

36. T. Ohmi, K. Matsudo, T. Shibata, T. Ichikawa, and H. Iwabuchi, "Very-low temperature epitaxial silicon growth by low-kinetic-energy particle bombardment," *Jpn. J. Appl. Phys.* 27(11), L2146 (1988).

37. T. Ohmi, H. Iwabuchi, T. Shibata, and T. Ichikawa, "Electrical characteristics of epitaxial silicon films formed by low kinetic energy particle bombardment," *Appl. Phys. Lett.* 54(3), 253 (1989).

38. T. Ohmi, T. Ichikawa, and H. Iwabuchi, "Crystal structure analysis of epitaxial silicon films formed by low kinetic energy particle process," *Appl. Phys. Lett.* 54(6), 523 (1989).

39. T. Ohmi, M. Morita, and T. Hattori, "Defects and impurities in SiO_2 for oxides prepared using superclean methods," *Proc. Electrochem. Soc. 173rd Mtg., Extended Abstracts* (Abst. No. 256) (May 1988), p. 387.

40. T. Ohmi, M. Morita, E. Hasegawa, M. Kawakami, and K. Suma, "Control of native silicon oxide growth in air or in water," *Proc. Electrochem. Soc. 175th Mtg., Extended Abstracts*, vol 89-1 (Abst. No. 160) (May 1989), p. 227.

41. T. Ohmi, H. Kuwabara, T. Shibata, N. Kowata, and K. Sugiyama, "Low kinetic-energy particle process for hillock-free aluminum metallization," *Proc. Fifth Int. IEEE VLSI Metallization & Interconnection Conf.* (June 1988), p. 446.

42. T. Ohmi, T. Saito, T. Shibata, and T. Nitta, "Room temperature copper metallization for ultra large scale integrated circuits by a low kinetic-energy particle process," *Appl. Phys. Lett.* 52(26), 2236 (1988).

43. N. Miki, H. Kikuyama, M. Maeno, J. Murota, and T. Ohmi, "Selective etching of native oxide by dry processing using ultra clean anhydrous hydrogen fluoride," *Tech. Digest, 1988 Int. Electron Device Mtg.* (December 1988), p. 730.

44. S. Matsuo and Y. Adachi, "Reactive ion beam etching using a broad beam ECR ion source," *Jpn. J. Appl. Phys.* 21(1), L4 (1982).

45. T. Ohmi, K. Masuda, T. Hashimoto, T. Shibata, M. Kato, and Y. Ishihara, "Formation of arsenic-implanted pn junctions using high vacuum ion implanter," *19th Conf. Solid State Devices & Materials, Extended Abstracts*, Tokyo (August 1987), p. 299.

Henry Berger, Manager, Microelectronics Process Technology for The BOC Group Technical Center at the Microelectronics Center of North Carolina, joined the research staff of Dr. Tadahiro Ohmi at Tohoku University in October 1988. A device physicist, Dr. Berger studies the relationship between cleanliness of raw materials and processing (with a focus on gases) and new semiconductor materials for integrated circuit devices. Dr. Berger received his Ph.D. in physics from the University of North Carolina. Prior to joining The BOC Group, he was a development scientist with Energy Conversion Devices, Troy, MI. There he was responsible for research and development of amorphous silicon, thin film, photovoltaic solar cells. He also worked at Exxon Research, Linden, NJ, on solar energy-related technology. Dr. Berger, a member of the Electrochemical Society, has authored numerous articles on his work. Along with Dr. Davidson, he is organizing an Electrochemical Society symposium, scheduled for May 1990, on gas and chemical purity for semiconductor processing.

Jeffrey M. Davidson is Manager, Materials Selection and Performance, at The BOC Group Technical Center in Murray Hill, NJ. Since joining the Center in 1984, he has been engaged in research on methods of characterizing and minimizing contamination of high purity gases used in semiconductor fabrication. Most recently, Dr. Davidson has completed a 1-year assignment in Japan as technical advisor to Osaka Sanso Kogyo, a member of The BOC Group, with primary emphasis on coordination of BOC Group R&D on industrial gases applications related to semiconductors and other advanced technologies. Dr. Davidson received his B.S. in materials science and M.S. and Dr. Eng. Sc. in metallurgy from Columbia University. Prior to joining BOC, he was a senior metallurgist with the International Nickel Research and Development Center, where his research focused on advanced materials development. Dr. Davidson is a member of the Metallurgical Society of AIME, the American Society for Metals, and the Institute of Environmental Sciences.

INTERNATIONAL MEETINGS IN THE FAR EAST 1989-1995

Compiled by Yuko Ushino

The Japan Convention Bureau, the Science Council of Japan, and journals of professional societies are the primary sources for this list. Readers are asked to notify us of any upcoming international meetings and exhibitions in the Far East which have not yet been included in this report.

1989

Date	Title/Attendance*	Site	Contact for Information
November 20-December 1	The 1st International Symposium and Exhibition of SAMPE JAPAN CHAPTER	Makuhari, Japan	SAMPE P.O. Box 2459 Covina, CA 91722
November 28-December 1	1st Japan International SAMPE Symposium & Exhibition: New Materials and Processes for the Future	Chiba, Japan	1st Japan International SAMPE Symposium & Exhibition c/o The Nikkan Kogyo Shinbun, Ltd. 1-8-10 Kudan Kita Chiyoda-ku, Tokyo 102
December 4-6	The 1st International Conference on Deductive and Object-Oriented Databases (DOOD89)	Kyoto, Japan	Professor Kiyoshi Agusa ASTEM RI, 9F Asahi Building Oike Yanaginobanba Nakagyo, Kyoto 604
December 4-8	The 4th International Conference on Fusion Reactor Materials	Kyoto, Japan	Professor S. Ishino General Chairman, ICFRM-4 Department of Nuclear Engineering University of Tokyo Bunkyo-ku, Tokyo 113
December 5-7	Symposium on the Application of Mechatronics	Hong Kong	Mr. T.P. Leung Secretariat for Symposium on the Application of Mechatronics c/o Dept. of Mechanical & Marine Engineering Hong Kong Polytechnic Hung Hom, Kowloon, Hong Kong
December 6-9	Asian Pacific Education Network Regional Workshop II Computer Software Development for Physics Institution	Chiang Mai, Thailand	Dr. Samran Lacharojana, Head Department of Physics Faculty of Science Chiang Mai University Chiang Mai 50002, Thailand
December 11-13	The 3rd International Workshop on Petri Nets and Performance Models (PNPM 89)	Kyoto, Japan	Dr. Shojiro Nishio Department of Applied Mathematics and Physics Faculty of Engineering Kyoto University Kyoto 606

*Note: Data format was taken from the Japan International Congress Calendar published by the Japan Convention Bureau.

No. of participating countries
F: No. of overseas participants
J: No. of Japanese participants

1989

Date	Title/Attendance	Site	Contact for Information
December 11-15	The 10th Australasian Fluid Mechanics Conference	Melbourne, Australia	10AFMC c/o Professor A.E. Perry Department of Mechanical Engineering The University of Melbourne Parkville, Victoria 3052
December 11-21	The 5th International Symposium on World Trends in Science and Technology Education	Manila, Philippines	Dr. Adracion D. Ambrosio IOSTE Symposium Chairman Philippine Science High School Diliman, Quezon City 1104

1990

Date	Title/Attendance	Site	Contact for Information
January 9-11	Symposium on High Magnetic Field Generation and Its Application to Materials and Biological Systems (ISEF-KANAZAWA) 10-F20-J80	Kanazawa, Japan	ISEF-KANAZAWA Secretariat c/o Faculty of Technology Kanazawa University 2-40-20 Kodatsuno, Kanazawa 920
January 22-26	International Conference on Recrystallization in Metallic Meterials	Wollongong, Australia	Metallurgical Society of AIME Conference Department 420 Commonwealth Drive Warrendale, PA 15086
January 24-26	The 2nd International Symposium on Advanced Nuclear Energy Research - Evolution by Accelerators	Mito, Japan	Secretariat c/o Atomic Reactor Engineering Japan Atomic Energy Research Institute Tokai-mura, Naka-gun, Ibaraki 319-11
February 4-8	The 18th Australian Polymer Symposium	Bendigo, Australia	Dr. E. Rizzardo CSIRO, Division of Chemicals & Polymers Private Bag 10 Clayton, VIC 3168
February 4-9	The 17th International Symposium on the Chemistry of Natural Products (IUPAC)	New Delhi, India	Professor Sukh Dev Multi-Chem. Research Centre Nandesari, Baroda-39340
February 5-9	International Workshop on Polarized Ion Source 10-F40-J20	Tsukuba, Japan	National Laboratory for High Energy Physics 1-1 Oho Tsukuba, Ibaraki-ken 305
February 13-17	International Workshop on Polarized Ion Sources and Polarized Gas Jet 10-F40-J20	Tsukuba, Japan	National Laboratory for High Energy Physics 1-1 Oho Tsukuba, Ibaraki 305
March 1	Workshop on Advanced Motion Control	Yokohama, Japan	Professor Kohei Ohnishi Department of Electric Engineering Keio University 3-14-1 Hiyoshi Kohoku, Yokohama 223
March 12-14	International Forum on Fine Ceramics '90 10-F100-J900	Nagoya, Japan	Japan Fine Ceramics Center 2-4-1 Mutsuno Atsuta-ku, Nagoya 456

1990

Date	Title/Attendance	Site	Contact for Information
March 12-16	International Conference on Supercomputing in Nuclear Applications	Mito, Japan	Kiyoshi Asai Conference Secretariat Computing Center, JAERI Tokai-mura, Naka-gun Ibaraki 319-11
March 15-17	International Bio Symposium 90 Nagoya "BIOTECHNOLOGY/ Today & Tomorrow" 10-F50-J350	Nagoya, Japan	International Bio Symposium 90 Organizing Committee c/o Chubu Bioindustry Promotion Council 2-17-22 Sakae Naka-ku, Nagoya 460
March 22-24	Kyoto Bioscience Symposia VI "Role and Regulation of Heart Shock Response" N.A.-F12-J50	Kyoto, Japan	Institute for Virus Research Kyoto University 53 Shogoin-Kawahara-cho Sakyo-ku, Kyoto 606
March 29-31	IEEE International Workshop on Advanced Motion Control 10-F30-J70	Yokohama, Japan	Dr. K. Ohnishi Department of Electrical Engineering Faculty of Science and Technology Keio University 3-14-1 Hiyoshi, Kohoku-ku Yokohama-shi, Kanagawa 223
April 1-6	The 1990 National Engineering Conference of the Institution of Engineers Australia	Canberra, Australia	The Conference Manager 1990 National Engineers Conference The Institution of Engineers 11 National Circuit Barton, ACT 2600
April 4-6	The 2nd International Symposium on Power Semiconductor Devices & JCs (ISPSD '90)	Tokyo, Japan	Yoshiyuki Uchida Fuji Electric Co., Ltd. Matsumoto Factory 2666 Tsukama, Matsumoto Nagano 390
April 8-12	1990 International Topical Meeting on Optical Computing 10-F100-J300	Kobe, Japan	OC'90 Secretariat Business Center for Academic Societies Japan (BCASJ) 3-23-1 Hongo Bunkyo-ku, Tokyo 113
April 12-14	1990 International Topical Meeting on Photonic Switching	Kobe, Japan	PS'90 Secretariat Business Center for Academic Societies Japan (BCASJ) 3-23-1 Hongo Bunkyo-ku, Tokyo 113
April 13-16	The 25th Yamada Conference on Magnetic Phase Transition (MPT '90) 10-F100-J200	Osaka, Japan	MPT '90 Secretariat Professor Y. Miyako Faculty of Science Hokkaido University Nishi 8-chome, Kita 10-jo Kita-ku, Sapporo 060
April 17-19	The 5th International Symposium on "Advanced Technology in Welding and Materials Processing and Evaluation"	Tokyo, Japan	Japan Welding Society 1-11 Kanda Sakuma-cho Chiyoda-ku, Tokyo 101
April 23-25	The 3rd Japan-China Joint Conference on Fluid Machinery 8-F60-J100	Osaka, Japan	Professor Yutaka Miyake Department of Mechanical Engineering Faculty of Engineering Osaka University 2-1 Yamada-Oka Suita, Osaka 565

1990

Date	Title/Attendance	Site	Contact for Information
April 23-27	Nankai Conference: International Conference on Physics Education Through Experiments	Tianjin, People's Republic of China	Professor Zhao Jing-yuan Department of Physics Nankai University Jianjin
May (tentative)	Recent Developments and Applications of Hot Cold Rolled and Coated Products	Kaohsiung, Taiwan	South East Asia Iron and Steel Institute P.O. Box 7759 Airmail Distribution Center NAIA, Pasay City 1300, Philippines
May 2-4	1st World Congress on Biosensors	Hong Kong	Penny Moon, Conference Manager Elsevier Seminars Mayfield House 256 Banbury Rd. Oxford OX2 7DH, U.K.
May 14-18	The 14th World Mining Congress and Exhibition	Beijing, People's Republic of China	14th World Mining Congress 54 Sanlihe Road Beijing
May 19-26	The 27th International Navigation Congress 62-F500-J500	Osaka, Japan	Japan Organizing Committee for 27th International Navigation Congress of PIANC c/o Port and Harbor Bureau City of Osaka 2-8-24 Chikko Minato-ku, Osaka 552
May 20-25	The 9th International Symposium on Carotenoids	Kyoto, Japan	Professor Masayoshi Ito Kobe Women's College of Pharmacy 4-19-1 Motoyamakita-Machi Higashinada-ku, Kobe 658
May 20-25	The 17th International Symposium on Space Technology and Science	Tokyo, Japan	Ms. Hiroko Sakurai 17th ISTS Secretariat c/o Institute of Space and Astronautical Science 3-1-1 Yoshinodai Sagamihara, Kanagawa 229
May 20-26	The 27th Congress of Permanent International Association of Navigation Congress (PIANC) 70-F500-J500	Osaka, Japan	Secretariat Japan Organizing Committee for 27th Congress of PIANC c/o Port & Harbor Bureau, City of Osaka 2-8-24 Chikko Minato-ku, Osaka 552
May 21-22	Conference and Exhibition: Foundry Asia '90	Hong Kong	FMJ International Publications
May 21-23	4th Symposium on Our Environment	Singapore	Wong Ming Keong Dept. of Chemistry National University of Singapore Singapore 0511
May 29-June 1	The International Conference on Manufacturing Systems and Environment - Looking Forward to the 21st Century	Tokyo, Japan	T. Nakajima The Japan Society of Mechanical Engineers Sanshin Hokusei Building 2-4-9 Yoyogi Shibuya-ku, Tokyo 151
June (tentative)	The 10th International Conference on Vacuum Metallurgy	Beijing, People's Republic of China	The Chinese Society of Metals 46 Dongsixi Dajie, Beijing 100711

1990

Date	Title/Attendance	Site	Contact for Information
June 4-7	Joint International Conference on Marine Simulation and Ship Maneuverability (MARSIM & ICSM 90) N.A.-F130-J120	Tokyo, Japan	Secretariat: MARSIM & ICSM 90 c/o ISS International, Inc. 5F, Shinkawa Building 2-2-21 Shiba-koen Minato-ku, Tokyo 105
June 5-8	International Symposium on Reliability and Maintainability 20-F200-J400	Tokyo, Japan	Union of Japanese Scientists and Engineers (JUSE) 5-10-11 Sendagaya Shibuya-ku, Tokyo 151
June 11-15	1990 International Conference: Metallurgical Coatings	Beijing, People's Republic of China	Chinese Society of Metals 46 Dongsixi Dajie, Beijing 100711
June 11-15	1990 International Conference: Special Melting	Beijing, People's Republic of China	Chinese Society of Metals 46 Dongsixi Dajie, Beijing 100711
June 15-20	The 2nd International Conference: Aluminum Alloys - Physical and Mechanical Properties	Beijing, People's Republic of China	Beijing University of Aeronautics and Astronautics
June 19-21	The 1990 Coal Handling and Utilization Conference	Sydney, Australia	The Conference Manager Coal Handling and Utilisation Conference 1990 The Institution of Engineers, Australia 11 National Circuit Barton, ACT 2600
June 22-26	International Conference on Dynamics, Vibration, and Control	Beijing, People's Republic of China	Professor Wei Jinduo Chinese Society of Theoretical and Applied Mechanics No. 15 Zhong Guancun Street Beijing
June 26-30	International Symposium on High Temperature Corrosion and Protection	Shenyang, People's Republic of China	Professor Man Yongfa Institute of Metal Research Academia Sinica 2-6 Wenhua Road Shenyang, Liaoning Province China
July 1-5	The 1st Tokyo Conference on Advanced Catalytic Science and Technology (TOCAT 1) 20-F100-J200	Tokyo, Japan	Secretariat: TOCAT 1 c/o Department of Synthetic Chemistry Faculty of Engineering Tokyo University 7-3-1 Hongo Bunkyo-ku, Tokyo 113
July 1-6	The 3rd International Conference on Technology of Plasticity (3rd ICTP) 10-F300-J700	Kyoto, Japan	The Organizing Committee 3rd ICTP c/o The Japan Society for Technology of Plasticity Torikatsu Building 5-2-5 Roppongi Minato-ku, Tokyo 106
July 6-7	The 1st KSME-JSME Fracture and Strength Conference (Fracture and Strength '90)	Seoul, Korea	Professor Hideaki Takahashi Research Institute for Strength and Fracture of Materials Tohoku University Aoba Tsurumaki Aza Sendai 980

1990

Date	Title/Attendance	Site	Contact for Information
July 9-11	Japan-U.S.A. Symposium on Flexible Automation - A Pacific Rim Conference	Kyoto, Japan	Professor Toshihiro Tsumura c/o Institute of Systems, Control at Engineers 14 Yoshida-Kawahara-cho Sakyo-ku, Kyoto 606
July 11-13	The 5th International Conference on Manufacturing Engineering	Wollongong, Australia	The Conference Manager The Institution of Engineers, Australia 11 National Circuit Barton, ACT 2600
July 11-13	The 3rd Optoelectronics Conference (OEC '90) 8-F20-J350	Tokyo, Japan	Katsuyoshi Ito OEC '90 Publicity & Registration Subcommittee Chair c/o Business Center for Academic Societies Japan Conference Department, Crocevia Crocevia Hongo 2F 3-23-1 Hongo Bunkyo-ku, Tokyo 113
July 15-21	The 10th International Congress of Nephrology 10-F1,000-J4,000	Tokyo, Japan	Japanese Society of Nephrology c/o 2nd Department of Internal Medicine School of Medicine, Nippon University 30-1 Oyaguchi-kamicho Itabashi-ku, Tokyo 173
July 16-20	Pacific Congress on Marine Science and Technology (PACON 90)	Tokyo, Japan	PACON 90 College of Science and Technology Nihon University 1-8-14 Surugadai, Kanda Chiyoda-ku, Tokyo 101
July 16-21	ISEC '90 International Solvent Extraction Conference	Kyoto, Japan	Conference Secretariat ISEC '90 Department of Chemistry Science University of Tokyo Kagurazaka, Shinjuku-ku, Tokyo 162
July 18-20	Advanced Research on Computers in Education	Tokyo, Japan	Professor Setsuko Otsuki Faculty of Computer Science and Systems Engineering Kyushu Institute of Technology 1-1 Sensui-cho, Tobata-ku Kitakyushu-shi, Fukuoka 804
July 30-August 2	The 15th International Conference on International Association on Water Pollution Research and Control	Kyoto, Japan	Japan Society on Water Pollution Research and Control Yotsuya New Mansion 12 Honshiocho Shinjuku-ku, Tokyo 173
August 2-8	The 25th International Conference on High Energy Physics 1990	Singapore	Professor K.K. Phua South East Asia Theoretical Physics Association c/o Dept. of Physics National University of Singapore Kent Ridge, Singapore 0511
August 7-11	International Symposium on Analytical Chemistry	Changchun, People's Republic of China	Professor Qinhan Jin Dept. of Chemistry Changchun, China
August 12-17	The 15th International Carbohydrate Symposium	Yokohama, Japan	Dr. Ishido, General Secretary Faculty of Science Tokyo Institute of Technology Ookayama, Meguro-ku, Tokyo 152

1990

Date	Title/Attendance	Site	Contact for Information
August 13-17	The 4th Asia Pacific Physics Conference	Seoul, Korea	Program Committee, AAPC Department of Physics Yonsei University Seoul 120-749, Republic of Korea
August 18-20	General Assembly, International Mathematical Union 52-F124-J6	Kobe, Japan	ICM 90 Secretariat c/o Research Institute for Mathematical Sciences Kyoto University Oiwake-cho, Kitashirakawa Sakyo-ku, Kyoto 606
August 20-24	1990 International Symposium on Symbolic and Algebraic Computation 15-F60-J140	Tokyo, Japan	ISSAC '90 Conference Office c/o Scientist, Inc. Yamazaki Building 3-2 Kanda-Surugadai Chiyoda-ku, Tokyo 101
August 21-29	International Congress of Mathematicians 1990 84-F1,500-J1,500	Kyoto, Japan	ICM 90 Secretariat c/o International Relations Office Research Institute for Mathematical Sciences Kyoto University Kitashirakawa Oiwake-cho Sakyo-ku, Kyoto 606
August 22-24	1990 International Conference on Solid State Devices and Materials N.A.-F100-J900	Sendai, Japan	c/o Business Center for Academic Societies Japan Crocevia Building 2F 3-23-1 Hongo Bunkyo-ku, Tokyo 113
August 23-30	V International Congress of Ecology 62-F900-J1,000	Yokohama, Japan	Secretary General's Office for INTECOL 1990 c/o Institute of Environmental Science and Technology *Yokohama National* University 156 Tokiwadai Hodogaya-ku, Yokohama 240
August 28-31	International Conference & Exhibition on Computer Applications to Materials Science and Engineering (CAMSE 90)	Tokyo, Japan	Professor M. Doyama CAMSE '90 c/o The Nikkan Kogyo Shimbun, Ltd. Business Bureau 1-8-10 Kudan Kita Chiyoda-ku, Tokyo 102
August 29-September 4	The 11th International Symposium on Biotelemetry N.A.-F120-J250	Yokohama, Japan	Professor A. Uchiyama Department of Electronics & Communication School of Science and Engineering Waseda University 3-4-1 Okubo Shinjuku-ku, Tokyo 169
August 30-September 4	International Conference on Potential Theory 24-F50-J200	Nagoya, Japan	Secretariat International Conference on Potential Theory c/o Department of Mathematics College of General Education Nagoya University Furo-cho, Chikusa-ku, Nagoya 464-01
August 30-September 4	International Symposium on Computational Mathematics 10-F30-J50	Matsuyama, Japan	Professor T. Yamamoto Department of Mathematics Ehime University 2-5 Bunkyo-machi, Matsuyama 790
September 3-5	International Symposium on Diagnostics and Modeling of Combustion in Internal Combustion Engines	Kyoto, Japan	Professor Makoto Ikegami Dept. of Mechanical Engineering Kyoto University Sakyo-ku, Kyoto 606

1990

Date	Title/Attendance	Site	Contact for Information
September 4-7	The 2nd International Symposium on Chemical Synthesis of Antibiotics and Related Microbial Products 15-F70-J180	Oiso, Japan	Faculty of Pharmaceutical Sciences University of Tokyo 7-3-1 Hongo Bunkyo-ku, Tokyo 113
September 10-14	The 17th Congress of the Collegium International Neuro-Psychopharmacologicum	Kyoto, Japan	The 17th CINP Congress c/o Simul International Inc. Kowa Building No. 1 1-8-10 Akasaka Minato-ku, Tokyo
September 16-22	The 15th IUMS Congress: Bacteriology & Mycology - Osaka, Japan - 1990 71-F2,000-J3,500	Osaka, Japan	Secretary General c/o Department of Microbiology Faculty of Medicine Kyoto University Yoshida, Konoe-cho Sakyo-ku, Kyoto 606
September 18-21	The 3rd Asia-Pacific Microwave Conference (APMC '90) 30-F150-J350	Tokyo, Japan	APMC '90 Secretariat c/o Business Center for Academic Societies Japan 3-23-1 Hongo Bunkyo-ku, Tokyo 113
September 19-22	The 2nd World Congress on Particle Technology N.A.-F100-J400	Kyoto, Japan	Secretariat: 2nd World Congress on Particle Technology c/o Society of Powder Technology, Japan Shibunkaku-kaikan 2-7 Tanakasekiden-cho Sakyo-ku, Kyoto 606
September 23-27	The 57th World Foundry Congress (WFC) 31-F400-J800	Osaka, Japan	Secretariat Japan Foundrymen's Society Toyokawa Building 8-12-13 Ginza Chuo-ku, Tokyo 104
September 24-27	The 6th International Congress on Polymers in Concrete	Shanghai, People's Republic of China	ICPIC-90 Secretariat c/o Associate Professor Tan Muhua Institute of Materials Science and Engineering Tongji University Shanghai
September 24-27	The 3rd International Aerosol Conference 29-F200-J300	Kyoto, Japan	Professor Kanji Takahashi c/o Institute of Atomic Energy Kyoto University Uji, Kyoto 611
September 24-28	The 3rd International Aerosol Conference	Kyoto, Japan	Professor Kanji Takahashi, General Secretary Institute of Atomic Energy Kyoto University Uji, Kyoto 611
September 24-28	The 12th International Conference: Boundary Element Method Conference (BEM 12)	Sapporo, Japan	Mr. Hiroshi Mizoguchi JASCHOME, KKE Inc. Dai-ichi Seimei Building 24F 2-7-1 Nishi-Shinjuku Shinjuku-ku, Tokyo 160
October 1-5	International Conference on Information Technology Commemorating the 30th Anniversary of the Information Processing Society of Japan (IPSJ) - InfoJapan '90 20-F200-J1,000	Tokyo, Japan	InfoJapan '90 Secretariat: IPSJ Hoshina Building 3F 2-4-2 Azabudai Minato-ku, Tokyo 106

1990

Date	Title/Attendance	Site	Contact for Information
October 1-5	The 3rd International New Materials Conference (New Materials 90 Japan) 12-F100-J300	Osaka, Japan	Secretariat: New Materials 90 Japan c/o Inter Group Corp. Shohaku Building 6-23 Chayamachi Kita-ku, Osaka 530
October 9-12	Fracture and Fatigue of High-Performance and Multi-Phase Polymeric Materials 8-F25-J60	Undecided, Japan	Faculty of Engineering Yamagata University 4-3-16 Jonan Yonezawa, Yamagata 992
October 14-19	International Conference for New Smelting Reduction and Near Net Shape Casting Technologies for Steel	Pohang, Korea	Conference Department Institute of Metals 1 Carlton House Terrace London, SW1Y 5 5DB, U.K.
October 15-18	The 1st Asian-Pacific International Symposium on Combustion and Energy Utilization	Beijing, People's Republic of China	Professor Huang, Zhao Xiang and Song Jialin Institute of Engineering Thermophysics Chinese Academy of Sciences P.O. Box 2706, Beijing
October 15-19	The 4th International Symposium on Marine Engineering (ISME KOBE '90)	Kobe, Japan	ISME Organizing Committee c/o Kobe Shosen Daigaku 5-1-1 Fukae-Minami Higashinada-ku, Kobe 658
October 21-26	The 6th International Iron and Steel Congress 50-F300-J500	Nagoya, Japan	International Conference Department Iron and Steel Institute of Japan 3F, Keidanren Kaikan 1-9-4 Otemachi Chiyoda-ku, Tokyo 100
October 22-25	The 11th International Coal Preparation Congress N.A.-F250-J150	Tokyo, Japan	Secretariat 11th International Coal Preparation Congress c/o Simul International, Inc. Kowa Building, No. 9 1-8-10 Akasaka Minato-ku, Tokyo 107
October 22-26	International Conference on Information Technology in Connection with 30th Anniversary Celebration of Information Processing Society of Japan N.A.-F200-J1,000	Osaka, Japan	Secretariat: International Conference on Information Technology c/o Simul International, Inc. Kowa Building, No. 9 1-8-10 Akasaka Minato-ku, Tokyo 107
October 25-31	The 1st Japanese Knowledge for Knowledge-Based Systems Workshop (JKAW)	Kyoto, Japan	Assoc. Professor Riichiro Mizoguchi The Institute of Scientific and Industrial Research 8-1 Mihogaoka Ibaraki, Osaka 567
October 28-November 2	The 2nd International Conference: HSLA Steels	Beijing, People's Republic of China	Chinese Society of Metals 46 Dongsixi Dajie, Beijing 100711
October 29-November 1	Japan International Tribology Conference Nagoya - '90 N.A.-F100-J500	Osaka, Japan	Secretariat: Japan ITC Nagoya - '90 c/o Toyota Technological Institute 2-chome, Hisakata Tempaku-ku, Nagoya 468

1990

Date	Title/Attendance	Site	Contact for Information
November 4-8	International Symposium on Carbon, 1990: "New Processing and New Applications" 15-F50-J200	Tsukuba, Japan	The Carbon Society of Japan Saito Building 2F 2-16-13 Yujima Bunkyo-ku, Tokyo 113
November 14-16	Rare Metals '90 15-F100-J200	Kitakyushu, Japan	Mining and Materials Processing Institute of Japan (MMIJ) Nogizaka Building 9-6-41 Akasaka Minato-ku, Tokyo 107
November 26-29	The 3rd International Polymer Conference (3rd IPC) 5-F100-J200	Nagoya, Japan	IPC Secretariat c/o Society of Polymer Science, Japan 5-12-8 Ginza Chuo-ku, Tokyo 104
November 26-30	The 5th International Photovoltaic Science and Engineering Conference (International PVSEC-5)	Kyoto, Japan	Professor Junji Saraie Secretariat of International PVSEC-5 c/o Japan Convention Services, Inc. Nippon Press Center Building 2-2-1 Uchisaiwai-cho Chiyoda-ku, Tokyo 100
1990 (tentative)	Chemeca 1990 Applied Thermodynamics	New Zealand	Conference Manager The Institution of Engineers, Australia 11 National Circuit Barton, ACT 2600

1991

Date	Title/Attendance	Site	Contact for Information
February 7-12	The 10th International Conference on Offshore Mechanics and Arctic Engineering	Seoul, Korea	Korea Cmt for Ocean Resources and Engineering Dong-A University 840 Sahagu Pusan, Korea
February 10-15	POLYMER '91: International Symposium on Polymer Materials	Melbourne, Australia	Dr. G.B. Guise P.O. Box 224 Belmont, VIC 3216, Australia
May 7-13	Beijing Essen Welding '91	Beijing, People's Republic of China	Messe Essen Nobert Street D-4300 Essen Federal Republic of Germany
June 10-14	The 4th International Conference on Nucleus-Nucleus Collisions 20-F200-J200	Kanazawa, Japan	Institute of Physical and Chemical Research (RIKEN) 2-1 Hirosawa Wako, Saitama 351-01
June (tentative)	International Conference on Stainless Steels 20-F50-J100	Tokyo, Japan	Secretariat: STAINLESS STEELS '91 The Iron and Steel Institute of Japan Keidanren Kaikan 1-9-4 Otemachi Chiyoda-ku, Tokyo 100
June (tentative)	JIMIS-6: Intermetallic Compound - Properties and Applications	Tokyo, Japan	Professor Osamu Waizumi Institute for Materials Research 2-1-1 Katahira Sendai 980

1991			
Date	**Title/Attendance**	**Site**	**Contact for Information**
July 7-12	The 16th International Conference on Medical and Biological Engineering (ICMBE) 45-F600-J1,400	Kyoto, Japan	Japan Society of Medical Electronics and Biological Engineering 2-4-16 Yayoi Bunkyo-ku, Tokyo 113
July 7-12	The 9th International Congress on Medical Physics (ICMP) 54-F1,000-J1,500	Kyoto, Japan	c/o Division of Physics National Institute of Radiological Science 4-9-1 Anagawa Chiba 260
July 24-26	The 3rd International Conference on Residual Stresses (ICRS-3) 30-F150-J200	Tokushima, Japan	Society of Materials Sciences, Japan 1-101 Yoshida Izumidono-cho Sakyo-ku, Kyoto 606
July 24-30	The 17th International Conference on the Physics of Electronic and Atomic Collisions	Brisbane, Australia	Dr. W.R. Newell Department of Physics University College of London Gower Street London WC1E 6BT UK
July 29-August 2	The 6th International Conference on Mechanical Behavior of Materials (ICM-6) 30-F300-J300	Kyoto, Japan	Society of Materials Sciences, Japan 1-101 Yoshida Izumidono-cho Sakyo-ku, Kyoto 606
August 25-31	International Congress on Analytical Science-1991 (ICAS '91) 25-F500-J1,000	Chiba, Japan	The Japan Society for Analytical Chemistry Rm 304 Gotanda Sun Heights 1-26-2 Nishi Gotanda Shinagawa-ku, Tokyo 141
August (tentative)	The 16th International Conference on Medical and Biological Engineering (ICMBE)	Kyoto, Japan (tentative)	Japan Society of Medical Electronics and Biological Engineering 2-4-16 Yoyogi Bunkyo-ku, Tokyo 113
September 29-October 4	The 5th Asian Pacific Congress of Clinical Biochemistry (5th APCCB) 20-F300-J600	Kobe, Japan	Secretariat: 5th APCCB c/o Central Laboratory for Clinical Investigation Osaka University Hospital 1-1-50 Fukushima Fukushima-ku, Osaka 553
October 28-31	International Conference on Fast Reactors and Fuels Cycles 8-F150-J350	Kyoto, Japan	Power Reactor & Nuclear Fuel Development Corp. 1-9-13 Akasaka Minato-ku, Tokyo 107
Undecided 1991	The 9th International Conference on Hot Carriers in Semiconductors 10-F50-J100	Nara, Japan	Department of Electronics Osaka University 2-1 Yamada-Oka Suita, Osaka 565

1992

Date	Title/Attendance	Site	Contact for Information
February (tentative)	The 19th Australian Polymer Symposium	Perth, Australia	RACI Polymers Division P.O. Box 224 Belmont, VIC 3216
May 17-22 (tentative)	NETWORKS '92: The 5th International Network Planning Symposium 20-F200-J200	Kobe, Japan	NTT Telecommunication Networks Laboratories 3-9-11 Midori-cho Musashino-shi, Tokyo 180
August 30-September 4	The 9th International Congress on Photosynthesis	Nagoya, Japan	Professor Norio Murata Okazaki National Research Institute National Research for Basic Biology 38 Saigou-Naka, Miyoudaigi-cho-aza Okazaki, Aichi
October 26-30	The 14th International Switching Symposium (ISS '92) 60-F1,200-J800	Yokohama, Japan	NTT Communication Switching Laboratories 3-9-11 Midori-cho Musashino-shi, Tokyo 180
November 9-12	The 8th International Congress on Heat Treatment of Materials N.A.-FJ500	Osaka, Japan (tentative)	Secretariat of 8th International Congress on Heat Treatment of Materials c/o Research Institute for Applied Science 49 Tanaka Ohi-cho Sakyo-ku, Kyoto 606

1993

Date	Title/Attendance	Site	Contact for Information
May 23-28	The 18th International Mineral Processing Congress	Sydney, Australia	AUSIMM, Conference Department P.O. Box 122 Parkville, VIC 3052
1993 (tentative)	International Federation of Automatic Control Congress	Sydney, Australia	Conference Manager The Institution of Engineers, Australia 11 National Circuit Barton, ACT 2600

1994

Date	Title/Attendance	Site	Contact for Information
Tentative	XXX International Conference on Coordination Chemistry	Kyoto, Japan	Professor Hitoshi Ohtaki Coordination Chemistry Laboratories Institute for Molecular Science Myodaiji-cho, Okazaki 444
Tentative	The 10th International Conference on the Strength of Metals and Alloys (ICSMA-10)	Undecided, Japan	Professor Hiroshi Oikawa Faculty of Engineering Tohoku University Aoba, Aramaki Aza Sendai 980

1995			
Date	**Title/Attendance**	**Site**	**Contact for Information**
Tentative	The 13th International Vacuum Congress (IVC-13) The 7th International Conference on Solid Surfaces (ICSS-7)	Undecided, Japan	Vacuum Society of Japan 302 Kikai Shinko Kaikan Annex 3-5-22 Shiba-koen Minato-ku, Tokyo 105

Yuko Ushino is a technical information specialist for ONR Far East. She received a B.S. degree from Brigham Young University at Provo, Utah.

SCIENTIFIC INFORMATION BULLETIN INDEX, VOLUME 14

AUTHORS

SUBJECTS

SUBJECTS (continued)

SUBJECTS (continued)

SUBJECTS (continued)

SUBJECTS (continued)

SUBJECTS (continued)

RESEARCH FACILITIES/ INSTITUTIONS

RESEARCH FACILITIES/ INSTITUTIONS (continued)

•U.S. G.P.O. 1989-262-353:20000

NOTICE

The Office of Naval Research/Air Force Office of Scientific Research/Army Research Office, Liaison Office, Far East is located on the second floor of Bldg #1, Akasaka Press Center and bears the following mail identification:

Mailing address:	Office of Naval Reseach/Air Force Office of Scientific Research/Army Research Office Liaison Office, Far East APO San Francisco 96503-0007
Local address:	ONR/AFOSR/ARO Far East Akasaka Press Center 7-23-17, Roppongi Minato-ku, Tokyo 106
Telephone numbers:	Civilian 03-401-8924 Autovon 229-3236 Telefax 03-403-9670

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 12503 WASH., D.C.

POSTAGE WILL BE PAID BY DEPARTMENT OF THE NAVY

OFFICE OF NAVAL RESEARCH
LIAISON OFFICE, FAR EAST
APO SAN FRANCISCO 96503-0007

CHANGE REQUEST

This form is provided for your convenience to indicate necessary changes or corrections in mailing the ***Scientific Bulletin*** to you.

Please continue sending me the Scientific Bulletin _____

Please make the address change as indicated _____

Please discontinue sending me the ***Scientific Bulletin*** _____

Old Address _____

_____

_____

_____

New Address _____

_____

_____