ICOT Technical Memorandum: TM-0996

TM-0996

Fifth Generation Computer Systems Project —Outline of Plan and Results

by
T. Kurozumi

Jan. 1991

ICOT

Mita Kokusai Bldg. 21F
4 28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 3456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Fifth Generation Computer Systems Project
— Outline of Plan and Results —

Takashi Kurozumi

Institute for New Generation Computer Technology

Abstract

Objectives of the Fifth Generation Computer Systems (FGCS) project, which has been supported by MITI as a national project, are to research and to develop new computer technologies, especially parallel processing and inference technologies, for knowledge information processing systems. This paper describes the planning of the FGCS project, the framework of the FGCS, the outline of the R&D plan and the state of the art of the FGCS project.

1. Framework of the Fifth Generation Computer System

The Fifth Generation Computer, or FGC, will process knowledge information by using inference processing as a basic mechanism. To achieve the necessary power of inference processing, the computer must be constructed based on highly parallel processing architecture. The hardware will be realized by using the latest VLSI technology. In order to have inference processing, logic programming language will be put into the interface between hardware and software, and research and development for hardware and software constitutions based on logic language will be conducted.

The objectives of the FGCS Project are to overcome the technical restrictions of conventional computers and to develop innovative computers capable of intelligent information processing. Such machines are becoming essential in the information-oriented society of the 1990s. Future computers will have to be intelligent, easy to use and productive in order to develop software.

Conventional computers have been classified into generations according to their constituent hardware elements: vacuum tubes, transistors, IC, and LSI. But they are all based on the same Von Neumann architecture, which is characterized by sequential processing and stored-program schemes. In

present-day computers, the characteristics of the architecture determine the type of machine language, and software based on the machine language is procedural. These computers have limited functions because of the enormous gap between the way they process and the knowledge-based way human beings think. Computers must follow pre-defined procedures, so it is difficult for them to process in a declarative way.

The concept of the FGCS aims at meeting the need for computers in the future, solving the bottleneck problems of current computers. A fundamental characteristic of human intelligent activity is inference by using every piece of stored knowledge, whether conscious or unconscious. Inference based on predicate logic is a method of procedures to extract unknown information using existing knowledge. (Figure 1.1)

In fifth generation computers, hardware and software are based on a programming method called logic programming. Programs, which are described in the form of a logic, can be executed as inference. The logic programming languages assigned to do this are called the kernel languages (KL) in this project. The new framework can be built as follows: firstly by specifying a predicate logic language as a new machine language, secondly by creating a hardware system that performs highly parallel inference processing based on the new language, and thirdly by creating a software system that performs a new type of processing, using the basic inference functions provided by the hardware system.

We estimate that fifth generation computers will require an inference speed 1,000 times greater than conventional computers. The high-level integration provided by advanced VLSIs enables us to make a reasonably compact and inexpensive computer with more than a thousand processors working in parallel. We measure speed in Logical Inferences Per Second, or LIPS. In this project, we aim at an inference execution speed of between 100M LIPS and 1G LIPS, for the prototype hardware consisting of one thousand processing elements. Software systems of FGC are assumed to have inference (problem-solving) function, knowledge base management function, intelligent (interactive) interface function and intelligent programming function. To realize these hardware and software, the most important keyword of FGC is inference and parallel processing. (Figure 1.2)

2. Preliminary stage for the FGCS project (Figure 2.1)

The Electro-technical Laboratory (ETL) led computer research activities in Japan since the 1950s, including the prototype manufacturing of a second generation computer (MARK III) in July 1956. In the 1970s it began studies on inference mechanism, led by Kazuhiro Fuchi as chief researcher, which included the study of Prolog. Through this research on Prolog as a language having a

structure closer to the structure of human thinking, they explored the possibilities of the development of logic programming and the creation of a new type of computer appropriate for inference.

In the meantime, MITI had put in effect a variety of policies aimed at enhancing the ability of research and development in the computer industry in Japan. In the 1970s MITI began to realize that the industry had grown to a level high enough to make such support from the government unnecessary. Amid this situation, MITI brought about a concept in May 1978 that said, "IBM's future system is expected to be a fourth generation computer. We will create the new fifth generation computer that goes one step further."

This concept is based on: 1) a question from user's point of view, "Present computers are difficult to use unless the user is professional. Can't we have a new computer that is easier to use?," and 2) an idea from the viewpoint of policy makers, "The design concept of computers has remained the same since Von Neumann. Isn't it possible to create a new non-Von Neumann type of computer?"

MITI asked for comments from research groups in Japan about whether it was possible to develop a non-Von Neumann type of computer. They knew that the Fuchi's group at ETL had the type of technical idea they needed and were convinced that the concept of a fifth generation computer could be realized.

It seemed quite tough for this project, which was aiming at a new, creative, and risky development, to acquire a budget as a government action. MITI thought that organizing a large-scale survey committee, including authorities from the academic society, would give strong support for acquiring a budget if the committee determined that this type of computer needs to be developed.

In 1979 the committee (Fifth Generation Computer Research Committee) and its sub-committees started their activities. It had 35 members, including professionals at universities, national and public institutes, computer manufacturers, and users. The committee had several sub-committees. The sub-committee for basic theory, organized with the ETL group as a center, brought forward the idea of "an innovative computer based on a non-procedural language, and a logic programming language as the language used for the computer." The sub-committee for computer architecture advocated "improving the computer along the line of the existing technology and then add non-Von Neumann functions to it." The committee as a whole produced a report in August 1980 that described opposite ideas, such as "improvers vs. innovators" and "realism vs. idealism."

Meanwhile, in February 1980 at ETL the committee gave a lecture to show the possibility of non-Von Neumann computers and to demonstrate Prolog.

The committee continued its work in 1980, and its membership increased to more than 100. They repeated vigorous investigations and discussions. Working groups were organized under the sub-committees to narrow down technical matters. Discussions at the committee were moving steadily toward supporting

non-Von Neumann computers. In its second annual report the committee stated its goal by saying. "The fifth generation computer is a knowledge information processing oriented computer system based on an innovative methodology and techniques, able to deal with advanced functions, including problem solving," and it described the concept as a non-Von Neumann inference machine. The report was translated into English and distributed to computer researchers abroad.

This report, titled "Proposal for the Research and Development of the Fifth Generation Computer," was produced, the members agreed to the concept, and MITI determined to start this project as a government policy action for new information processing. However, another process was still needed to acquire a budget for the project and formally make it a policy action of Japan. The committee continued its work in 1981, members increased to more than 120, and they further examined technical matters. At the same time, they started to prepare for an international conference on the fifth generation computer, which was to be held to discuss the planning of the project and results of their research activities with foreign researchers, and to receive their comments on the project.

In October 1981 the international conference was held. More than 300 attended from 15 countries, and the contents of the project drew much attention. Responding to the enthusiasm of MITI to acquire a budget in fiscal year 1982, the Ministry of Finance decided to admit the first-year budget for the project, on condition that a budget for the following years would be considered according to the results of the development activities of the project.

The efforts and enthusiasm of a large number of people, including the members of the committee and officers at MITI, and support from many other people, made it possible that this risky project for the research and development of the fifth generation computer, based on innovative ideas, became accepted and started as a national project.

Generally speaking, when many experts take part in a discussion of project planning and compile the results in a report, it is inevitable that different types of requirements are included, and that such arguments as "improvement vs. innovation" and "realism vs. idealism" arise. This project has been lucky to have understanding people and promoters, especially on the policy makers' side.

3. R&D stages of the FGCS project (Figure 3.1)

Because the FGCS project is very risky and involves a large number of unknown technologies, a relatively long period, ten years, has been scheduled for the project. This ten-year period is divided into three stages, three years for the initial stage, four years for the intermediate stage and three years for the final stage.

In the initial stage, R&D was conducted with emphasis on the basic FGC technologies.

In the intermediate stage, small-to-medium scale subsystems were developed as the basic components of a prototype system.

The aim of the final stage is to complete a FGCS prototype.

In addition, software development tools were developed by ourselves, because current computer systems have not enough functions for R&D of this project.

The budgets for each year, which are all covered by MITI, are shown in this figure. The total budget for the 3-year initial stage was about 8 billion yen, and the total budget for the 4-year intermediate stage was 22 billion yen, we spent 6.5 billion yen in 1989 and 7 billion yen in 1990. So, the total budget for this project is estimated to be more than 50 billion yen.

4. R&D results of the initial stage and the intermediate stage

R&D flow of system software, languages and hardware is summarized in figure 4.1. The top part shows sequential base, and the bottom part shows parallel base.

4.1 R&D results of the initial stage (FY 1982-1984) (Figure 4.2)

R&D in the initial stage was aimed at developing the basic technology required for fifth generation computers. Within the framework of this project, the R&D themes concerning knowledge information processing were analyzed and selected to achieve the goals of the initial stage. The specific subjects of R&D included an inference subsystem, knowledge base subsystem, basic software system and pilot models for software development. Goals were specified independently for each subject.

We became convinced that the basic framework of fifth generation computer was not mere hypothesis, but was viable and could be effected through the research of the basic technology required for fifth generation computers by developing and testing various experimental systems in the initial stage. The major results of each R&D subject are summed up as follows.

We developed sequential logic programming languages KL0 (Kernel Language version 0) and ESP (Extended Self-contained Prolog). KL0 is based on Prolog, and ESP is extended object-oriented modular programming functions and the macro expansion functions to KL0. The Personal Sequential Inference Machine, called PSI-I, was developed as a KL0 machine (Figure 4.3). Another model of sequential inference machine, called CHI, was also developed aiming at high-speed back-end use. The first version of the operating and programming

system for PSI, called SIMPOS, was also developed. SIMPOS is written entirely in ESP (Figure 4.4). PSI and SIMPOS were the world's first machine and fully-fledged OS based on logic programming language. About 100 PSI-Is were made for ICOT, and were the main tools with SIMPOS in the first half of the intermediate stage.

Several small-scale experimental software systems were trial-fabricated with functions for knowledge base management, semantic analysis and so on.

Parallel inference method was studied through trial-fabrication of experimental systems consisting of from eight to sixteen processors and through development of software simulators. In addition, an experimental relational database machine (Delta) capable of high-speed execution of relational algebraic operation was trial-fabricated.

A parallel logic programming language, called Guarded Horn Clauses (GHC), was proposed.

4.2 R&D results of the intermediate stage (FY 1985-1988) (Figure 4.5)

The objective of the intermediate stage was to create small-to-medium scale subsystems that would provide a basis for the fifth generation computers, while considering development of the individual research subjects and the integration of the subsystems in the final stage.

The focus of the first half of this stage was on the specifications of models, algorithms and the basic architecture on which software and hardware were to be built, based on the results of the initial stage. In the second half of this stage, experimental subsystems were implemented and evaluated. We believe that we can show the possibility of realization of fifth generation computer through the development of the parallel inference software environment in the intermediate stage. The major results of each R&D subject are summed up as follows.

Some of the aims of the intermediate stage were to perform design, experimental implementation and feasibility testing of KL1, and to research to refine the KL1 for the prototype machine.

Development of parallel hardware and software is similar to the chicken and egg problem. To develop parallel software, parallel hardware is required for the environment; and to design parallel hardware, we have to decide the hardware requirement from the software side. Therefore, in the intermediate stage we developed multi-PSI V1, which was constructed by 6 PSIs. Then, we developed a very small parallel OS to research the hardware requirement from the OS side. Next, we developed the multi-PSI V2 system, connecting 64 newly designed small PSI CPUs (Figure 4.6). This multi-PSI system is the first medium-scale experimental parallel inference machine in the world. Using this small CPU, front-end processor and PSI-II were also developed as the same system. Lastly, we developed the first version of a parallel OS,

called PIMOS, which can be used for software development of parallel application.

The fifth generation kernel languages play an important role in bridging hardware research and software research. There are two types of fifth generation kernel languages: sequential logic programming language (kernel language version 0 (KL0) which was developed in the initial stage) and parallel logic programming language (kernel language version 1 (KL1)). KL1 , whose core part is Flat GHC (FGHC) which is a subset of GHC, was designed for this hardware and software. (Figure 4.7)

KL1 and PIMOS can be used not only on a multi-PSI but also on a PSI-Ⅱ as a cross parallel programming environment. Now we have 7 multi-PSI systems, and more than 300 PSI-Ⅱs for a parallel software environment.

The basic design of Parallel Inference Machine (PIM), targeted for the final stage, was made.

Research on parallel algorithms was started through the development of parallel bench-marking programs and PIMOS.

As research on a knowledge base management mechanism, distributed knowledge base management software, called Kappa, was designed and implemented on the PSI/SIMPOS environment. This software manages and retrieves large-scale knowledge bases such as natural language dictionaries and knowledge bases for expert systems. An experimental parallel retrieval mechanism incorporating eight processor elements was trial-fabricated.

We investigated the following logic programming features important for knowledge programming and developing parallel programming techniques.

* Development of a processing system for constrained logic programming language (CAL)
* Research on the reflection function of GHC
* Research on partial evaluation and conversion techniques
* Research on a logic model of meta inference such as monotonic reasoning
* Development of a proof support system

We did research in discourse understanding and developed Japanese language processing tools.

* Development of the Language Tool Box (LTB) consisting of Japanese sentence analysis, and generation modules, semantic description language processing module and linguistic knowledge base

* Development of an experimental discourse understanding system (DUALS-3)

We also researched the various basic functions for constructing knowledge processing systems, and developed several experimental expert systems.

5. R&D plan and the status of the final stage

The objectives of R&D in the final stage (FY 1989-1991) are to build a prototype fifth generation computer system. This prototype system is based on parallel processing, as planned at the initiation of this project.

The prototype hardware system is constructed using highly parallel architecture that can perform high speed inference and knowledge base functions.

The prototype software system is divided into "Basic Software", "Knowledge Programming System" and "Experimental Application Systems."

Basic Software corresponds to the OS which has functions for the management and control of H/W resources, jobs, user, and so on.

The knowledge programming system is aimed to be able to program efficiently in a parallel logic programming for knowledge information processing.

The main aims of experimental parallel application systems are summed up as follows. One aim is R&D of parallel algorithms and load distribution through parallel applications. Another aim is to evaluate the various kinds of basic functions of the prototype system, and to clarify the examples of actual applications for the fifth generation computer. To achieve these objectives, we are, at present, gearing up to making a prototype system, using the results that were obtained up until the end of the intermediate stage.

Also, we are carrying out research on the basic technologies that may be needed in the future, besides the R&D themes that are needed for the final objectives (Figure 5.1).

5.1 Prototype hardware system

The prototype hardware system is designed to be a high-performance KL1 parallel processing machine suited for inference and knowledge base functions. As this prototype hardware system is intended to be used for development of parallel software, it is designed to be reliable and maintainable even though it is only a trial fabrication.

The most advanced VLSI technology, with sub-micron level integration is being applied to this hardware implementation, so one PE is able to be constructed by one board, and each cabinet can have 32 PEs. PE module performance for KL1 is estimated to be 300 to 600 KLIPS, so maximum system performance is expected to be several hundred MLIPS (Figure 5.2).

The prototype hardware system will be implemented as an integrated experimental environment, which is incorporated with several models of parallel inference machine. We have three large-scale models and two small-scale models. Each model is implemented to have different characteristics for various experimentation of hardware configuration techniques and so on (Figure 5.3) . For example, two models named PIM/p and PIM/c have multi-cluster construction: eight PEs are linked to a shared memory and have parallel cache in one cluster. Clusters in PIM/p (Figure 5.4) are connected in a double hyper-cube network, but the clusters in PIM/c are connected in a crossbar network. As PIM/m is implemented as a version-up model of the multi PSI system, PEs are

connected in a two-dimensional mesh-network. A processor element (PE) of PIM is expected to be three to four times faster than a multi-PSI PE, and one quarter the size. All these models are designed as KL1 machines, so all the parallel software including parallel OS can run on all these models.

5.2 Basic software system

Basic software will be implemented such OS functions as an efficient parallel S/W execution environment by controlling and managing the large-scale parallel processing hardware (such as PIM and multi-PSI).

Basic software consists of "Inference Control Module" called PIMOS, and "Knowledge base Management Module" called KBMS. All this basic software will be written in KL1 parallel processing language. Though, at present, this software is being developed by using "multi-PSI system", this software is designed to be independent from architectural details. So we think that this software will be easy to transplant from multi-PSI to each PIM (Figure 5.5).

As for PIMOS, we intend that it will have practical OS functions to be used for parallel software development environment. PIMOS already has basic OS functions such as resource management, job control, execution control of multi-PSI, remote access function and so on, and KL1 source codes of PIMOS is more than 100K lines. One characteristic of PIMOS is that PIMOS is a single OS showing a parallel machine as one system. This characteristic of a single parallel OS sharply distinguishes it from a distributed OS, which is an integration of sequential OSs. Another characteristic is that PIMOS has full hierarchical distributed control using tree structure. The aim of this is to avoid the bottleneck of resource and process control for the parallel machine. Also, this distributed control is able to increase parallelism in management, and to decrease communication traffic between PEs (Figure 5.6). As for extension and improvement of PIMOS, we plan to add multi-user and file-access functions, load-balancing library, and KBMS interface.

The aim of the knowledge base management module, KBMS, is easy storage and retrieval of complex knowledge data. The structure of knowledge data needs flexibility for data length, expression of variables, and various data structure. A goal of KBMS is deductive object-oriented data base, and data base structure is designed based on non-normal form relational data base. For evaluation of KBMS, natural language dictionary, gene-bank, and rules for application of laws are planned as knowledge bases. Though the multi-PSI system has no disks for storage, PIM is implemented to be able to connect disks for data base storage. KBMS will have distributed data base management functions and a parallel processing mechanism.

5.3 Knowledge programming system

The knowledge programming system is a group of utility software. The aims of this system are to provide various problem-solving functions and interactive interface functions, and to support construction and use of knowledge base.

The main research themes in problem-solving are parallel theorem provers and constraint-solving systems.
Theorem provers aim to prove first-order predicate logic.
For constraint-solving systems, we intend to design parallel constraint logic languages, and to implement constraint solvers for several application fields.
Research themes in the interactive interface are to develop general purpose Japanese language processing tools, and to research discourse understanding technology through development of an experimental discourse understanding system named DUALS. Another aspect of DUALS is to evaluate language processing tools by using the tools themselves. Japanese language processing tools consist of sentence analyzer, sentence generator, language knowledge base, and constraint semantic representation language processor.
Regarding technologies for knowledge base utilization support, we put emphasis on cooperative distributed technology, case-based reasoning, constraint models, and assumption-based reasoning (Figure 5.7).

5.4 Experimental applications

As I explained earlier, the main objectives of this project are to research technologies for parallel processing and knowledge information processing. We think that both technologies are important long-term research themes, and that it will be difficult to solve all problems of these technologies before the end of the project term.

There are no parallel machines suited for knowledge processing outside of ICOT, so we have to start from development and accumulation of concurrent algorithms and load distribution methods by developing application programs. Accumulation of experience in parallel knowledge programming is most important to research these technological problems, and to know what other problems there are. Even if programs are written in parallel language by using sequential algorithms, it is difficult to process them concurrently on parallel machines. So parallel programs have to be written based on concurrent algorithms. That is why research into concurrent algorithms is important.

The characteristics of parallel language such as KL1 are easiness and efficiency of parallel programming. For example, basic software and application programs are developed efficiently at ICOT, and these programs are developed without bugs caused by synchronization. The aims of developing application systems can be summarized as accumulating parallel technologies, and cultivating new parallel processing fields. Furthermore, application systems are useful for evaluating prototype hardware and software.

We started to develop experimental parallel programs from 1988. These programs are a natural language parser, a limited "Go" game solver, a puzzle solver, and a shortest-path problem solver. These programs were shown at the FGCS conference in 1988 held in Tokyo.

As the next step, we are exploring the following new parallel processing applications. One is wiring problem, layout, and logic simulation in LSI CAD. We chose these problems because they need heavy computation power. Theorem provers, genetic information processing, "Go" playing system, natural language processing, and legal inference based on case-based reasoning are other parallel applications.

These applications are distributed in various fields, aiming at the cultivation of new parallel processing fields (Figure 5.8).

6. Trends and future applications

Figure 6.1 shows the size of inference machines corresponding to LSI technologies. I would like to emphasize that highly parallel systems will become reasonable and natural in the near future as LSI technologies advance. Roughly speaking, the number of gates will quadruple every three years.

A CPU of the first inference machine, named PSI-I, developed in 1983, needed more than ten boards. The next CPU of PSI-II developed in 1986, needed three CPU boards, and one network board added for the multi-PSI system (Figure 6.2). Each CPU in the fifth generation prototype system is planned to have just one board (Figure 6.3). If a parallel machine is developed four years after the end of our project, one board would be able to admit up to several processors. By the year 2000, more than a thousand processors will be in one cabinet.

I explained several parallel application fields in this project, such as CAD, theorem provers, genetic information processing, natural language processing and legal inference. These applications are distributed in various fields, aiming at the cultivation of new parallel processing fields.

We believe that applications for fifth generation computers will extend to various areas in industry and society, because parallel machines similar to fifth generation computers will become common in the future. Figure 6.4 shows some expected application fields in the future. Though these fields appear to be almost the same as current computer applications, the functions and ranges of applications would be greatly advanced by the use of parallel knowledge processing technologies.

7. Promoting organization of FGCS project and ICOT activities

ICOT was established in 1982 as the non-profit core organization for promoting this project and it began R&D work on fifth generation computers, under the auspices of MITI. The structure of this project is shown in figure 7.1 .

The ICOT organization consists of a general affairs office and a research center. In the initial stage, the research center consisted of a research planning department and three laboratories. At present, it has a research planning department and a research department (seven laboratories). The responsibilities of each laboratory change occasionally depending on the progress that is made.

There were 50 staff at the beginning of the intermediate stage, and their number has increased every year. At the beginning of 1988, there were about 90; now there are more. Organization of the ICOT research center was changed almost every year according to the advance of R&D of this project. Organization of the ICOT research center (at FY 1990) is shown in figure 7.2. All of the ICOT researchers are lent from 18 research organizations, mainly from the national research centers, the eight big computer companies and NTT, for a period of three to four years, but key researchers do not rotate so often, and some researchers are working at ICOT throughout the full period.

Concerning R&D work, ICOT executes the core part of R&D work. And ICOT makes contracts with the 8 computer companies for experimental production of hardware and developmental software. Consequently, ICOT handles all the R&D works including developmental work made by computer companies towards the goal of this project.

ICOT has set up committee and working groups for discussion and receiving of advice on research subjects.

Since the entire cost for the R&D activities of this project is born by the government budget, intellectual property right, including patents for the R&D results, belongs to the government. Intellectual property right is managed by AIST (Agency of Industrial Science and Technology), and any company wishing to use one of them can be granted its patent by paying a fee. The PSI (Personal Sequential Inference Machine) and its OS, SIMPOS, have been commercialized by companies licensed by MITI. This approach is expected to contribute to the establishment of a basis for the diffusion of fifth generation computer technology.

But software, which has not yet been handled by intellectual property right by AIST, can be used for research at outside institutes. This scheme, including language specifications, is also one of the new research promoting schemes for fifth generation computers.

We think these ICOT activities are very important, because this project is a national project, and we believe that ICOT can contribute to knowledge information processing research all over the world by presenting the results of the research conducted at ICOT and exchanging information, including preliminary ideas, with outside researchers, and that technological knowledge should be shared with researchers all over the world. By promoting the diffusion of R&D

results and studies on fifth generation computer systems in Japan and elsewhere through these activities, we hope R&D of the FGC technologies will be continued in many places throughout the world after the completion of the project. For this, diffusion of R&D results is also important. Since the beginning of the project, ICOT has sought to have research cooperation and discussions with many researchers who have research themes similar to those of ICOT. For example, GHC, which is the core parallel language of parallel system, has been aided by discussions with researchers working on Parlog and Concurrent Prolog.

Figure 7.3 shows ICOT activities in international cooperation and in diffusion of R&D results.

One important way to present research results is distribution of technical papers. So far, more than 1400 papers have been released (among them about 500 papers were written in English). These titles are shown in a quarterly journal, called ICOT Journal, which is distributed to more than 1100 locations.

Another way is to hold conferences and symposia. ICOT has held an international conference on FGCS three times. The next conference is now planned for June, 1992.

We also hold joint-sponsored workshops with several countries (a Japan-Sweden-Italy workshop, a Japan-France AI symposium, a Japan-U.S.A. AI symposium and a Japan-U.K. workshop).

Every year we invite several renowned researchers from abroad for a short period. Based on the memorandums with National Science Foundation (NSF) in the United States, the Institute National de Recherche en Informatique et Automatique (INRIA) in France, and Department of Trade and Industry (DTI) in United Kingdom, we receive researchers from those countries for six months to one year.

Also, discussions between ICOT researchers and researchers of other organizations are frequently made by exchange visits.

Recently, joint research with other organization such as Argonne National Laboratory (ANL), National Institute of Health (NIH), Lawrence Berkeley Laboratory (LBL) and Swedish Institute of Computer Science (SICS) were started based on parallel logic programming.

To promote the project, these activities are very useful as they help people understand the technologies of the project, and let us hear their evaluation of our results. Also these activities are an effective way to form the basis of FGC technologies.

ACKNOWLEDGEMENTS

This project has been carried out through the efforts of the researchers at ICOT, and with the support of MITI and many others outside ICOT proper. We

wish to extend our appreciation to them all for the direct and indirect assistance and cooperation they have provided.

REFERENCE

1. Moto-oka, et al — Proceedings of the International Conference of Fifth Generation Systems, 1981, JIPDEC
2. K. Kawanobe, et al — ICOT research and Development, Proceedings of the International Conference on Fifth Generation Computer Systems 1984, ICOT
3. T. Kurozumi, et al — Fifth Generation Computer Systems Project, 1987, ICOT, TM-0303
4. T. Kurozumi, et al — ICOT Research and Development, Proceedings of the International Conference on Fifth Generation Computer Systems 1988, ICOT

Figure 1.1 **Framework of Fifth Generation Computer**

●Computer for
Knowledge Information
Processing (KIPS)
(Intelligent Assistant for Intelligent
Human Activities)

○Basic Functions→
★Inference using Knowledge base
★Ease of Use

○Basic Mechanisn of H/W & S/W→
★Logical Inference Processing
(based on **Logic Programming**)
★Highly **Parallel Processing**

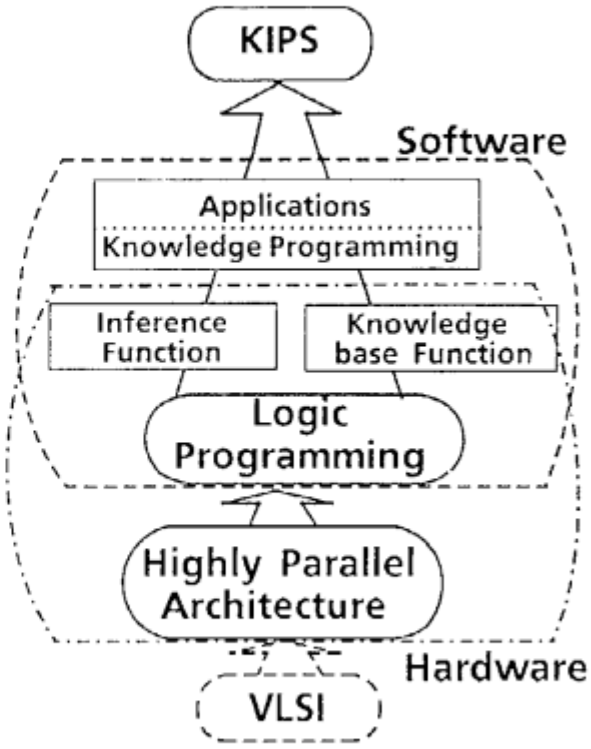


Figure 1.2 What is the Fifth Generation Computer ?

◎ **Von Neumann Type Computer** (Stored-program Schemes)

- ·1st Generation (Vacuum tubes)
- ·2nd Generation (Transistors)
- ·3rd Generation (ICs)
- ·4th Generation (VLSIs)

◎ **5th Generation Computer** (Parallel processing,
Knowledge information processing)

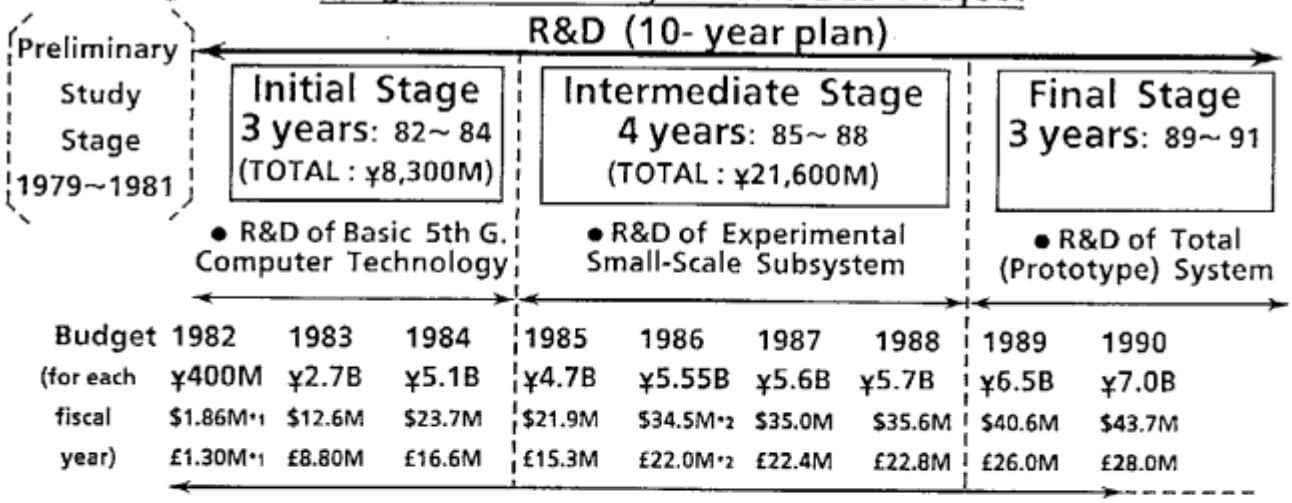
1st - 4th Generation	5th Generation
<Von Neumann Type>	<Non - Von Neumann Type>
· Low-level Machine Lang. (Procedural Programming)	· Logic Programming Lang. (Declarative Programming)
· Numerical Processing	· Inference & Knowledge base Processing
· Sequential Processing	· Parallel Processing

Figure 2.1 Preliminary Stage of the FGCS Project

1. Conception of FGCS
   - MITI
     - Conception of the R&D Project for Innovatory Computer
     - New computer technology that is easier to use.
     - New Non-Von Neumann type computer for a new market
   - ETL
     - Research on inference mechanism
     - Possibility of new computer for inference

2. Planning stage for the Project
   - Fifth Generation Computer Committee to discuss needs and possibility for new computer
   - International Conference on FGCS
   - Negotiation for the Budget of the FGCS Project.

Figure 3.1 Stages of & Budget for FGCS Project



- R&D are carried out under the auspices of MITI.

(All budget are covered by MITI.)

*1 $1 = ¥215, £1 = ¥307 (1982~1985)

*2 $1 = ¥160, £1 = ¥250 (1986~1990)

Figure 4.1
R&D Steps of System Software, Kernel Languages & Hardware

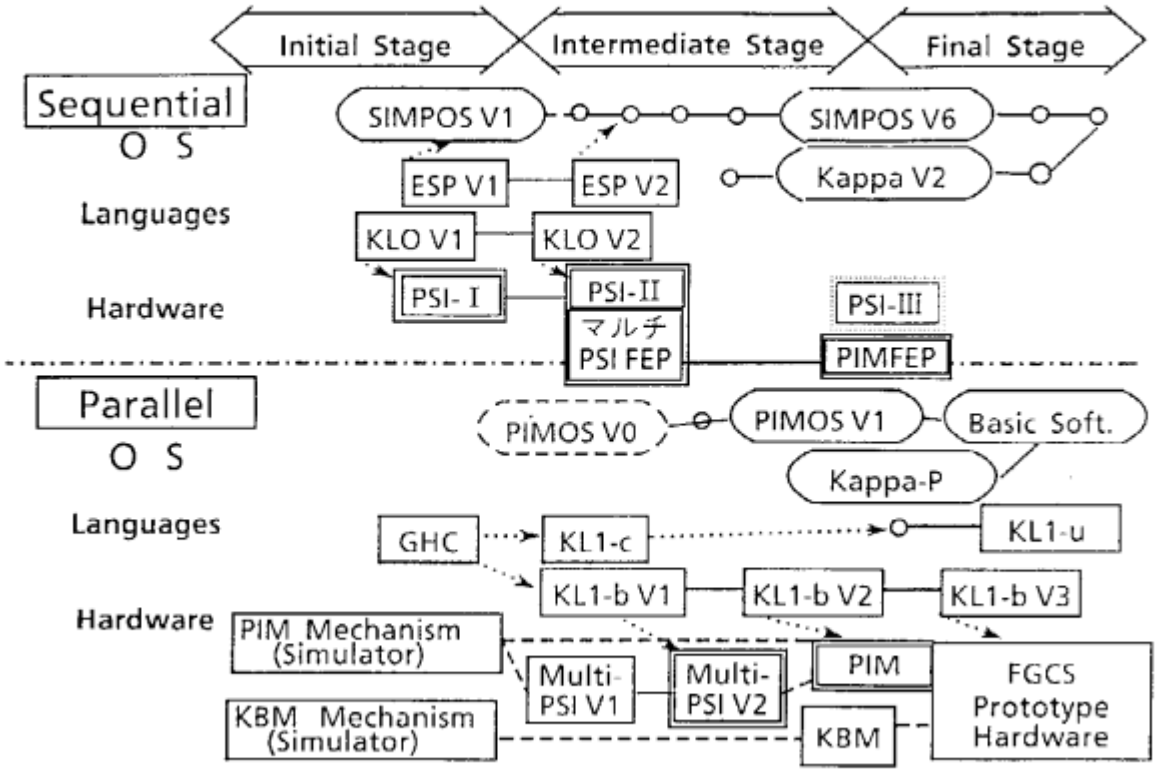


Figure 4.2 **Major R&D Results in the Initial Stage (1982-1984)**

○ Development of Sequential Inference Machine and its Software

- Personal Sequential Inference Machine (PSI- I)
- Cooperative High-performance Inference Machine (CHI)
- Sequential Inference Machine Programming Operating System (SIMPOS)
  : Written by ESP (Extended Self-contained Prolog)
  : ESP : Prolog(KL0) + modularization function(based on object-oriented)

○ R&D of Experimental Hardware and Software for Basic Technologies

- **Software & Hardware Simulators of Parallel Inference Mechanism**
- **Experimental RDB Machine with 4 Relational Algebraic Engines**
- **Parallel Logic Programming Language : GHC (Guarded Horn Clauses)**
- **Experimental Software Systems Written by Logic Programming Language**
  RDBM Software(KAISER), Discourse Understanding System (DUALS), Logic Program Verification System, and others

Figure 4.3 Personal Sequential Inference Machine(PSI-I)

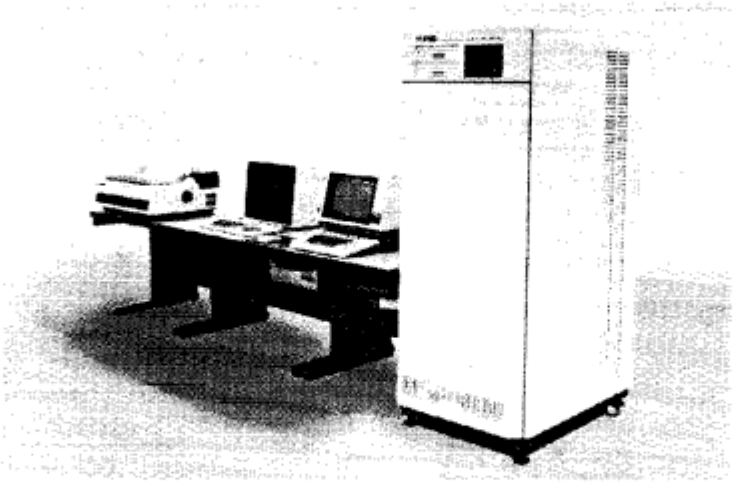


Figure 4.4 SIMPOS Configuration

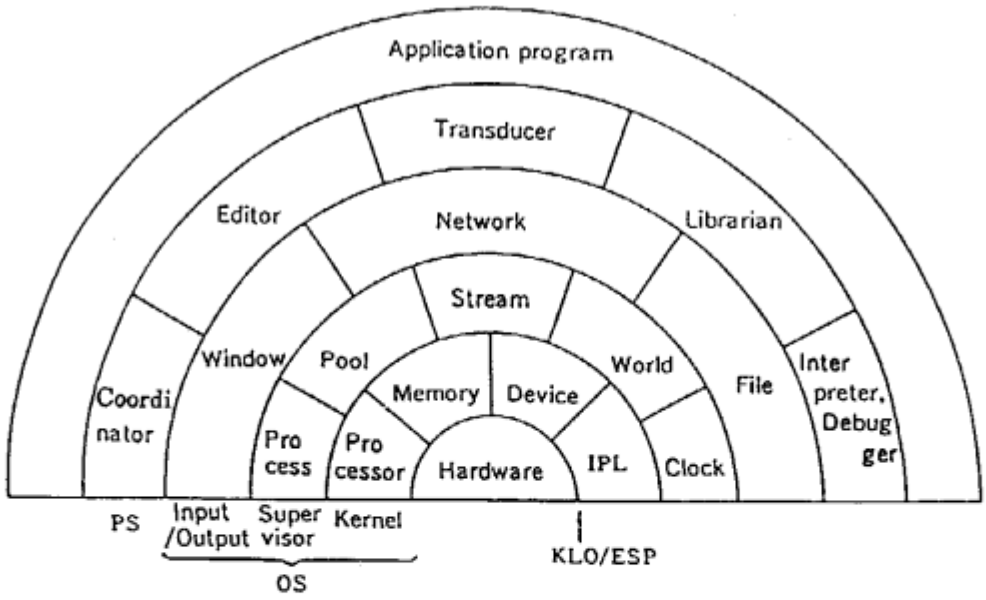


Figure4.5 Major R&D Results in the Intermediate Stage (1985-1988)

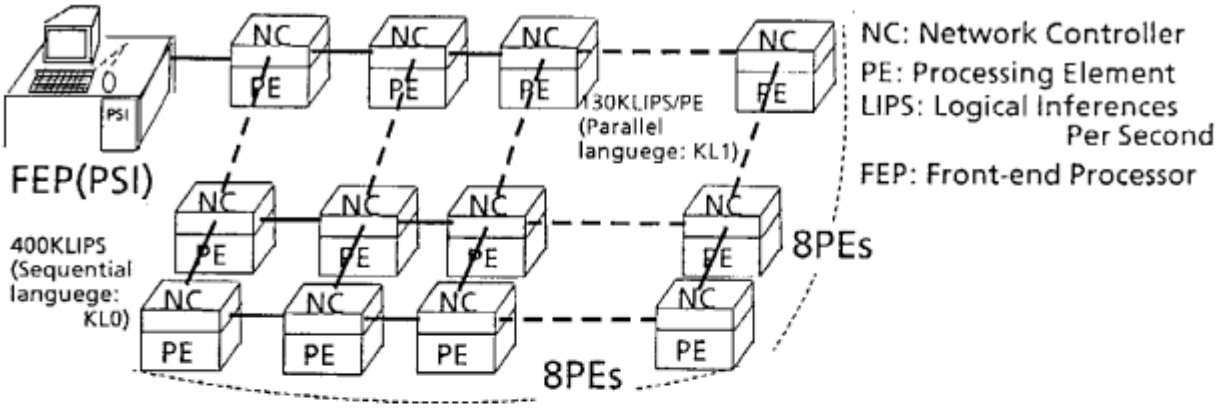
○Development of Experimental Parallel Inference Machine (PIM) for Parallel Software Development Environment

- Design and Implementation of KL1 (GHC→Flat GHC→KL1(KL1-b,KL1-c,KL1-p))
- Multi-PSI V1:Connecting 6 PSI-Is (1985~1986)
- Multi-PSI V2:Connecting 64 PSI-II CPUs(1987~1988) KL1 Based, 7Systems
- R&D of PIMOS(V1) on Multi-PSI V2, PIMOS-S (Pseudo-parallel system on PSI-II) and PDSS (Debug support system for KL1 on UNIX)

○R&D of H/W & S/W Technologies

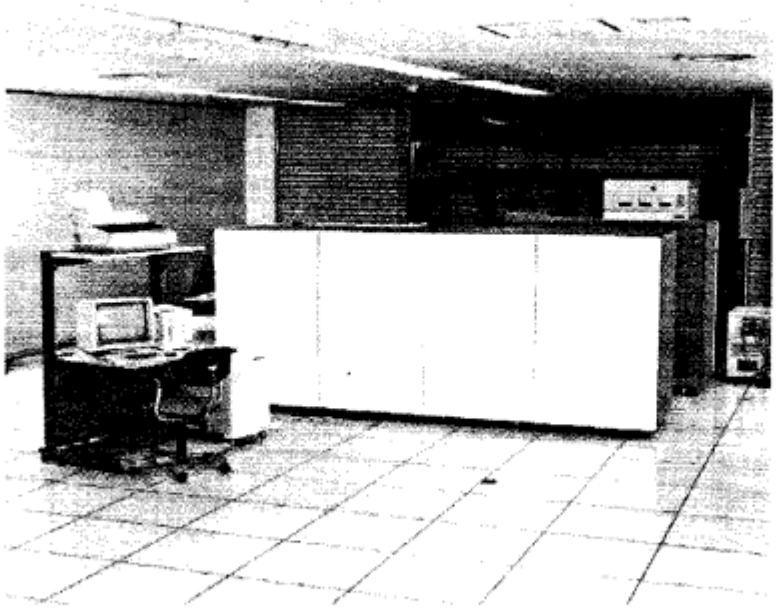
- Research on PIM : Basic Design of PIM, Simulation for architectural design,
- Research on Parallel algorithms through development of Experimental Parallel programs
- Multi-PSI FEP(= PSI-II), Improvement of SIMPOS
- Experimentation for Knowledge base (KB) Retrieval Mechanism
  - · Software simulator system and Hardware (8PEs) system,
  - · DB/KB Software system (Kappa) on SIMPOS and PSI
- General Purpose Japanese Language Tool Box (LTB) and Experimental Discourse Understanding System (DUALS V3)
- Constrained Logic Programming Language (CAL), Partial Evaluation Technology
- Proof Support System (CAP), Experimental Expert Systems

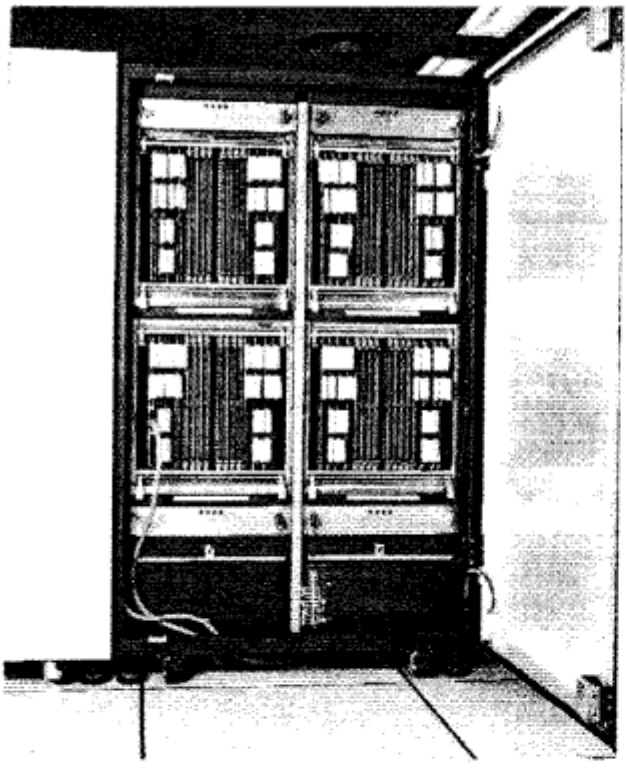
Figure 4.6-a Experimental Parallel Inference Machine: Multi-PSI/v2



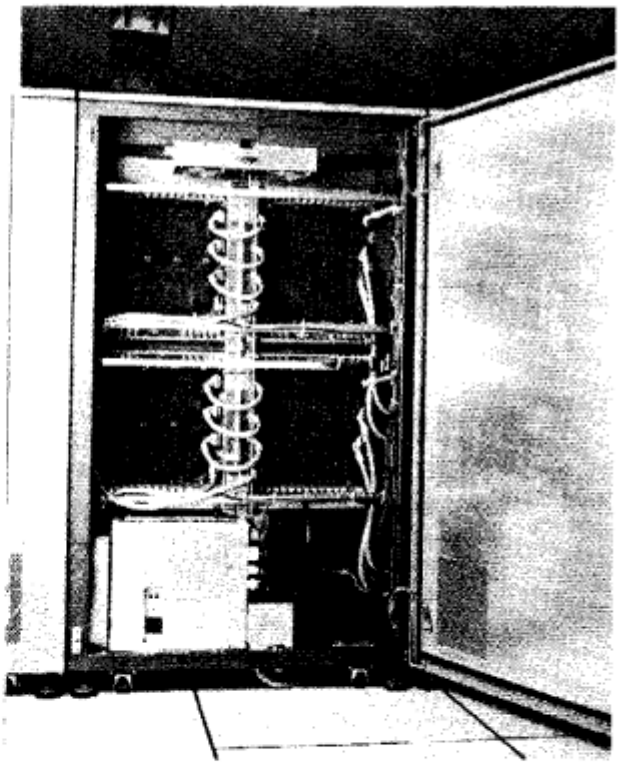
- 64 PEs max (PSI-II CPU each)
  2 ~ 5MLIPS/system (average)
- Machine language : KL1-b
- Memory : 16 Mw / PE (80MB)
- Network :
  2-dimensional mesh,
  -message exchange
  -routing functions
  -5MB/s x 2directions/ch

Figure 4.6-b Multi-PSI System (V2)





Front-side View



Rear-side View

Figure 4.7 Language System of the Fifth Generation Computer

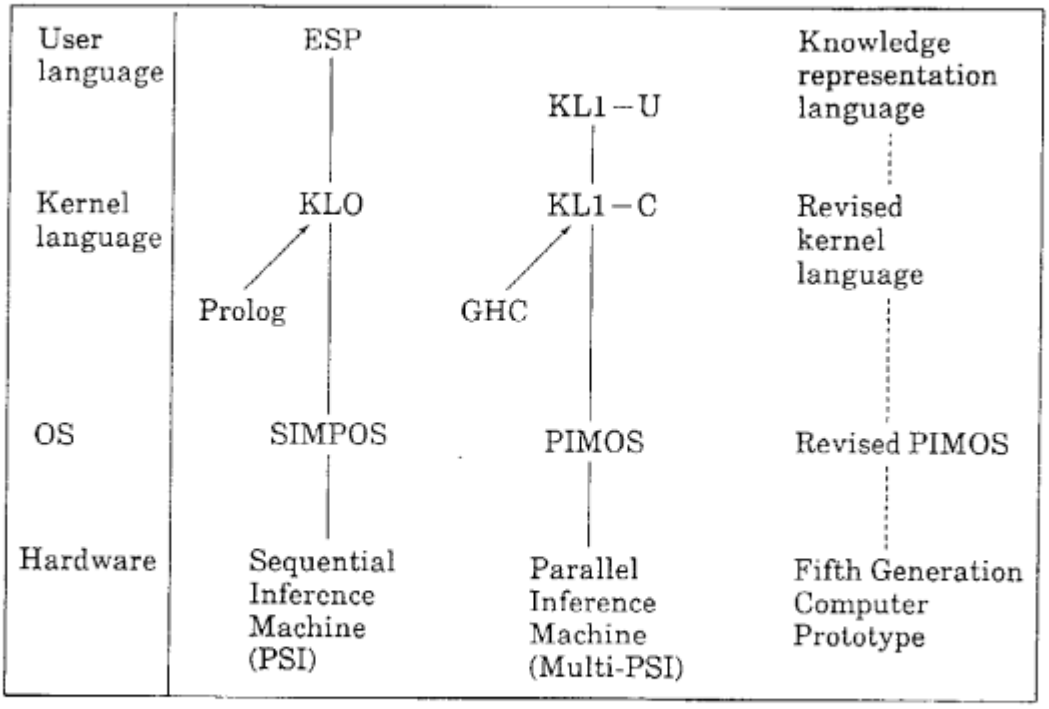


Figure 5.1 Outline of Final Stage

- R&D for 5G Prototype System
- H/W System with Highly Parallel Architecture for Inference & KB Functions
- S/W System based on Parallel L.P.Language for OS&Knowledge Prog.
- R&D of Parallel Algorithms and Load Distribution Through Parallel Applications
- Basic Research based on L.P.

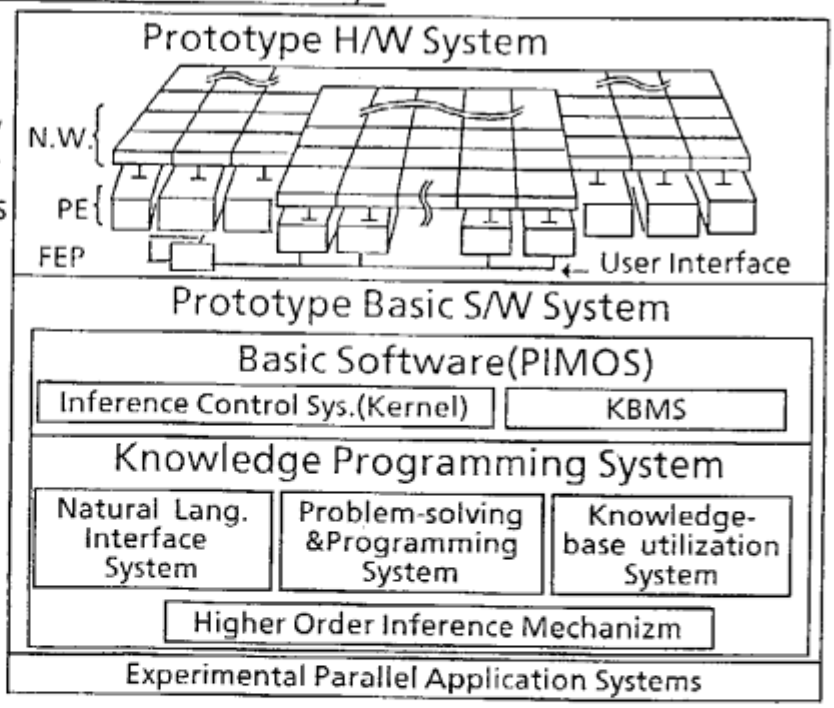


Figure 5.2 Prototype Hardware System

- ○ High speed parallel logic programming (KL1) machine
  suited for inference functions and knowledge base management functions
- ○ Large scale hardware implementation using most advanced VLSI technology
  (Up to 1,000 PEs / Loosely-coupled configuration,32 PEs/Cabinet, 1 PE/1 Board)
- ○ Reliable and maintainable system for parallel software development
- ○ Constructed by several models of Parallel Inference Machine
  - ● Large scale models (256~ 512PEs / Tightly-coupled configuration)
    - •PIM/p & PIM/c : Layered structure
      Intra-cluster (8 PEs with parallel cache and shared memory)
      Inter-cluster network (PIM/p:Double hypercube net, PIM/c:XB net.)
    - •PIM/m: Version-up model of Multi-PSI (Up to 256 PEs & mesh Network)
  - ● Small scale models
    Experimental models for architecture research: Inter-system connection and hierarchical parallel cache mechanisms

Figure 5.3-a Prototype System Configuration —Parallel Software Development Environment—

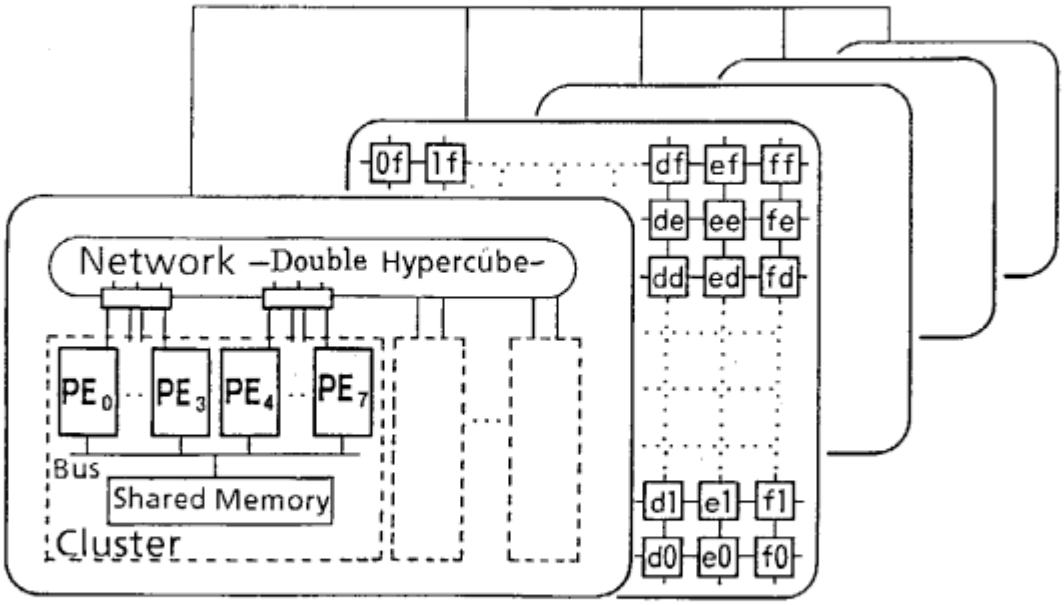
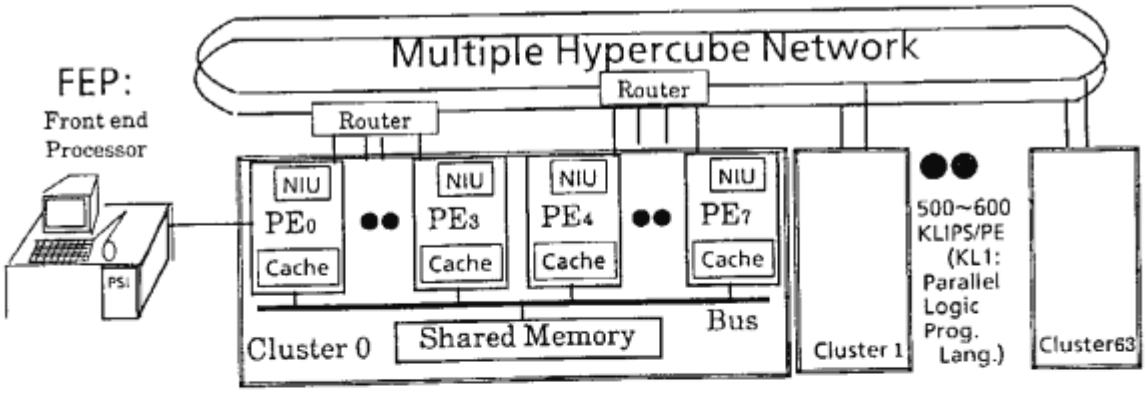


Figure 5.3-b Models of Parallel Inference Machine (PIM)
(Prototype Hardware System)

Item	PIM/p	PIM/c	PIM/m	PIM/i	PIM/K
Machine instructions	RISC-type + macro instructions	Horizontal microinstructions	Horizontal microinstructions	RISC-type	RISC-type
Target cycle time	50 nsec	50 nsec	50-60 nsec	100 nsec	100 nsec
LSI devices	Standard cell	Gate array	Cell base	Standard cell	Custom
Process Technology (line width)	About 1 μm	0.8 μm	0.8 μm	1.2 μm	1.2 μm
Machine configuration	Multicluster connections (8 PEs linked to a shared memory) in a hypercube network	Multicluster connections (8 PEs + CC linked to a shared memory) in a crossbar network	Two-dimensional mesh network connections	Shared memory connections through a parallel cache	Two-level parallel cache connections
Number of PEs connected	512 PEs	256 PEs	256 PEs	8 PEs x 2	16 PEs x 2

Figure 5.4 A Model of Parallel Inference Machine --PIM/p



- Layered structure
  - 8PEs / cluster
  - 64 clusters / system (512 PEs total)
- Machine language: KL1(-b)
  RISC + macro Instructions
- VSLI processors: 4-stage pipeline
- Memory : 32 Mw / cluster
- Network :
  Double hypercube
  - Message exchange
  - Routing functions
  - 20MB/s x 2directions/ch

Figure 5.5 Basic Software (OS)

★*Description in the KL1 (Parallel Logic Language)*
★*Efficient Control of Large-scale Parallel Processing Hardware*

- Independent from architectural details. (Multi-PSI → PIM)

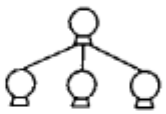
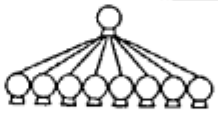
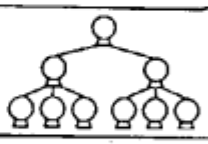
(1) PIMOS (PIM operating system):Inference Control Module

- ● A practical OS for large scale S/W experiments.
  - •Full-fledged OS functions
  - •Emphasis on ease of use
- ● A single Parallel OS showing a parallel machine as one system.
- ● Full hierarchical distributed control
- ○ Basic Functions:100K lines in KL1 source codes. (1990.4)
- ○ Improvement of PIMOS :
  - ● 1990FY: Remote access, Multi-user, File access
  - ● Future Plan:KBMS interface,Load balancing library etc.(200K lines in KL1 source codes)

(2) KBMS:Knowledge-base Management Module

- ● Easy storage of complex knowledge data
  - •Full-fledged non-normal form relational DBMS functions
- ● Distributed database management and parallel processing mechanism

Figure 5.6
Diagram Illustrating PIMOS Features

	Parallel Version of Sequential-Processing OS	PIMOS
Example of Management Procedure	Centralized Control Task management tables.	Full distributed control Tree-structure of task
Small-Scale Parallel Processing		
Large-Scale Parallel Processing		
Parallelism in Management	Small→Bottleneck	Large
Communication Traffic	Access concentration →Large	Distributed processing →Small

Each level manages only the main components of the lower level; there is no attempt to manage all details of the entire system from a single point

Figure 5.7 Knowledge Programming System
— Technology and Tools —

● Problem-solving
  - Parallel theorem-proving technologies (and provers)
  - Parallel constraint logic languages (and solving system)

● Natural Language Interface
  - Japanese language processing system
    → Sentence analyzer, Sentence generator, Language KB,
      Language for constraint semantic representation, ···
  - Experimental discourse understanding system (DUALS)

● Knowledge-base Utilization Support
  - Cooperative distributed technologies
  - Case-based reasoning
  - Constraint models

Figure 5.8 Experimental Applications in Parallel Knowledge Processing

○ Aims: Cultivation of a new field for Parallel Processing
& accumulation of Parallel Programming Technologies
  - *Development of concurrent algorithms and load distribution method*
  - *Accumulation of concurrent programming paradigms*
    - Several typical AI problems containing much computation
    - Collection of different type of program structures and dynamic characteristics

○ The experimental parallel application programs (1988 — 1989)
1. PAX : a natural language parser
2. Tsume-go : a board game solver
3. A packing piece puzzle (Pentomino)
4. A shortest-path finding problem

○ Exploration of new themes in parallel processing applications (1989 — End)
- Continuous research on
  Concurrent programming and Load distribution / Scheduling methods
- More practical and larger application programs
1. Computation intensive problems in LSI CAD
2. Theorem Provers
3. Genetic information processing
4. "GO" game solver (playing system)
5. Natural language processing
6. Legal inference based on case based reasoning
etc.

Figure 6.1 TRENDS of LSI TECHNOLOGIES & 5G MACHINES

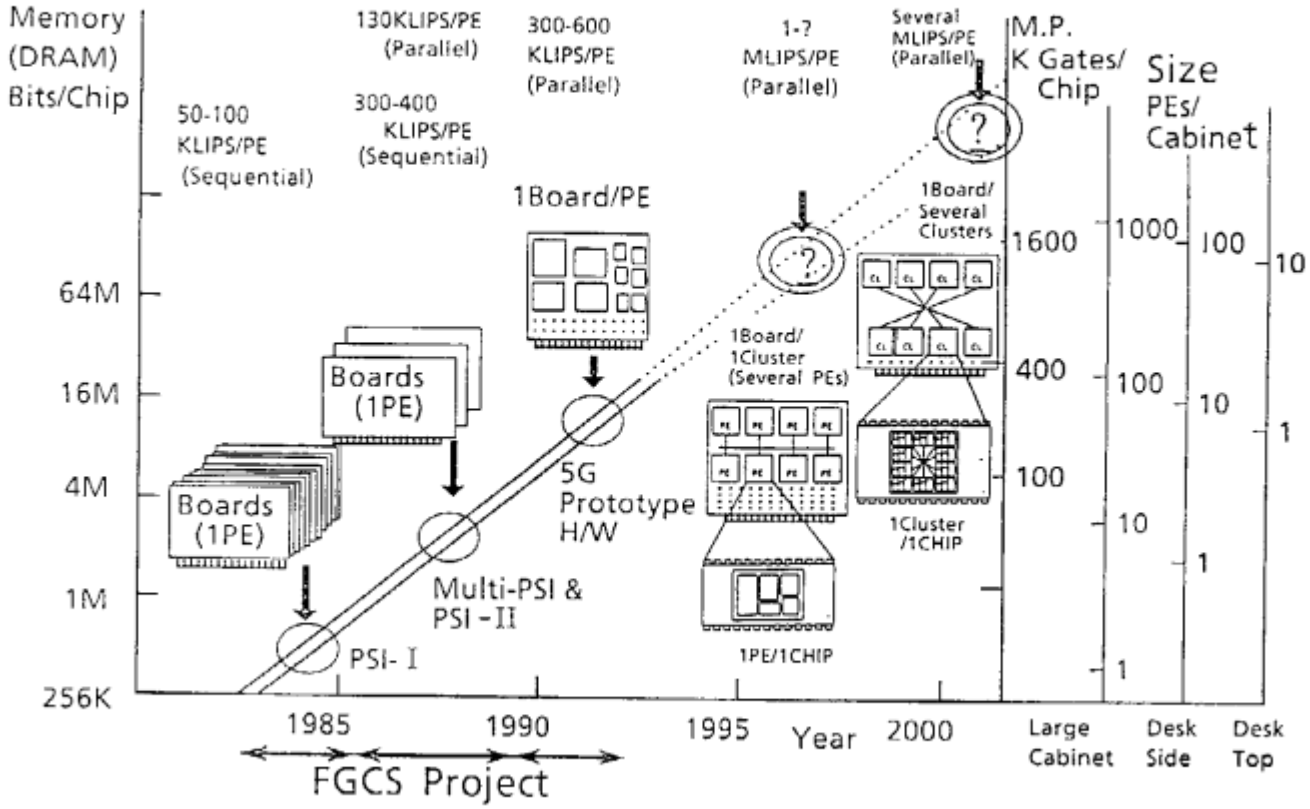


Figure 6.4 **Impact of the 5G Computer System**

5G Computers are expected to be used in many areas in industry and society

Examples of Applications

Industrial areas	**·Design:**	*CAD, CAE*
	·Production Processes:	*Intelligent robots, CAM*
	·Development:	*Expert Systems(ES) for efficient development*
	·Management:	*ES for Decision support*
	·Office Work:	*Intelligent OA*
	·Maintenance:	*Remote diagnosis*
Social areas	**·Education:**	*CAI*
	·Clinical:	*Medical consultation, Automated nursing*
International areas	**·Translation:**	*Machine translation*

Figure 6.2 CPU boards of the Multi-PSI System

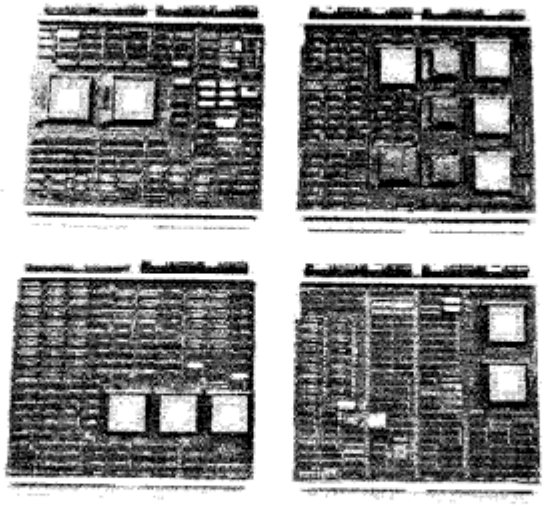


Figure 6.3 CPU board (example) of the 5G Prototype System

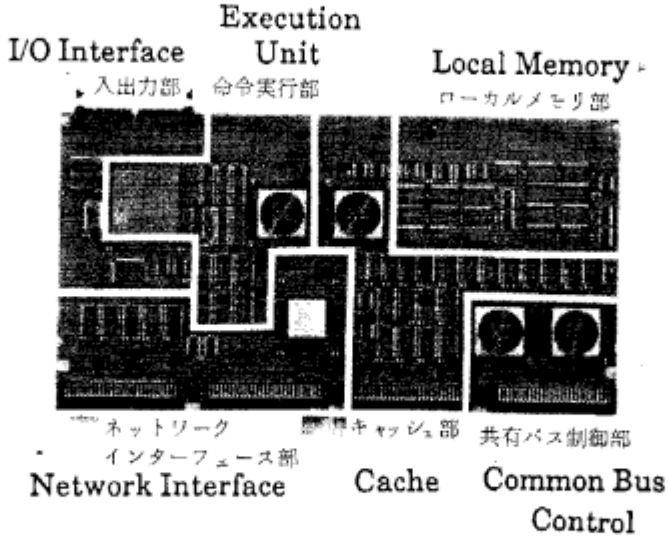


Figure 7.1 Organization of Fifth Generation Computer Project

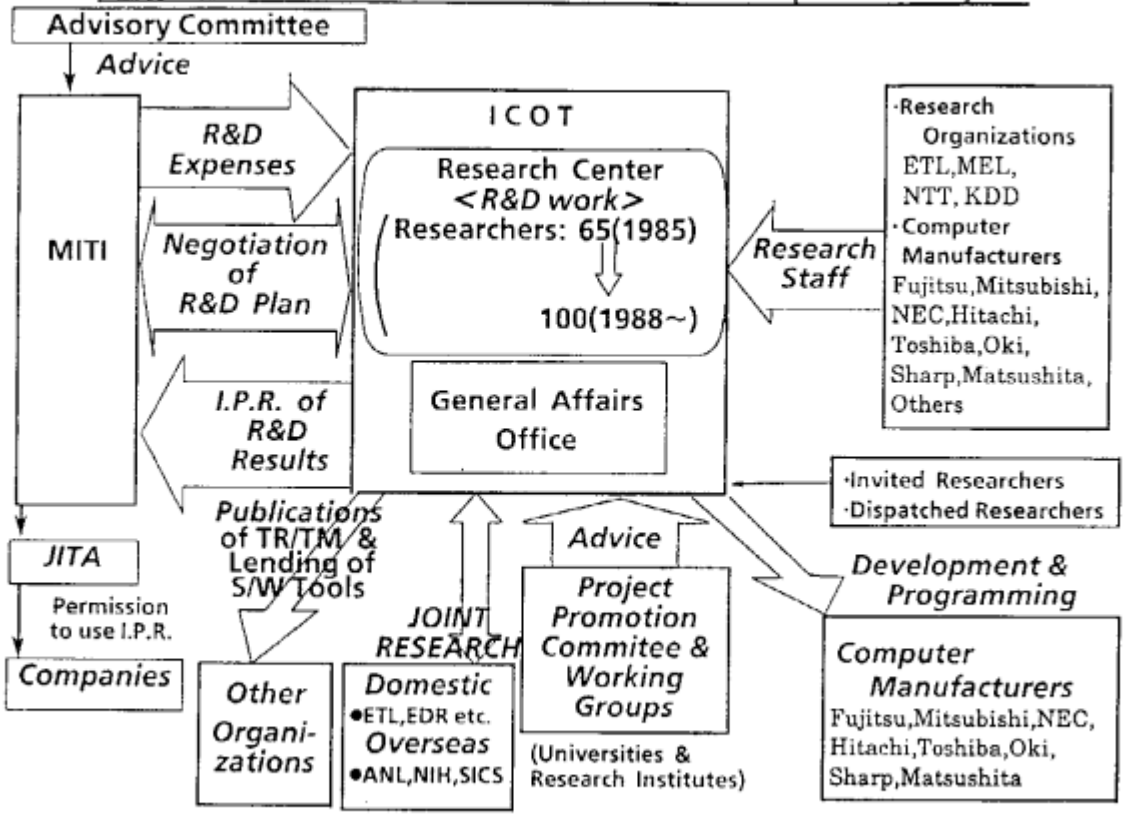


Figure 7.2 ICOT Research Center Organization

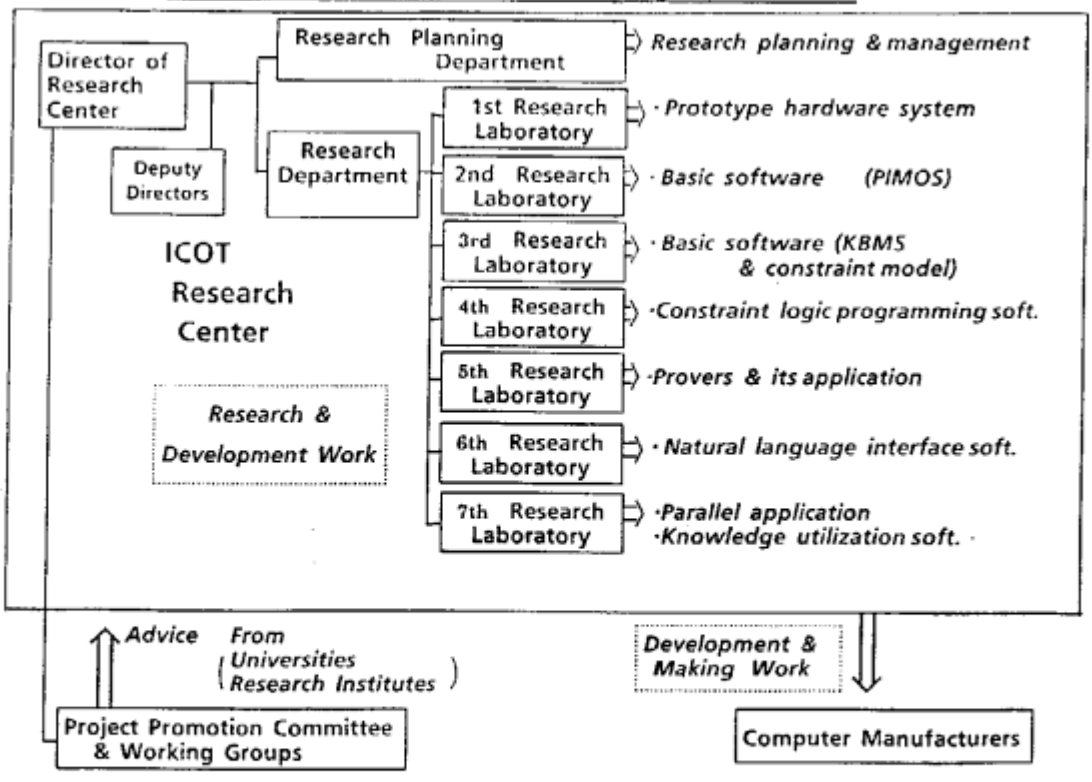


Figure.7.3 International Co-operation & Diffusion Activities of R&D Results

- **Preparation and Distribution of Technical Papers**
  - ICOT Journal: *Distributed to 600 overseas locations,*
  - Technical Reports and Memoranda *(1,500TR/TMs)): Sent to 30 organizations (regularly), and sent to many researchers /organizations (on request)*

- **Conferences and Symposia**
  - Sponsorship of the International Conference on FGCS (1981 , 1984 , November 1988, 1st-5th June 1992(planning))
  - Joint Symposia and Workshops(co-sponsorship)
    - *U.S.-Japan AI symposium*: Dec.1987(Tokyo), Oct.1989(Chicago), Sept.1990(Tokyo)
    - *France-Japan AI symposium*: Oct.1986(Tokyo), Nov.1987(Sophia-Antipolis), Nov.1989(Izu), Jan.1991(Rennes)
    - *Sweden-Japan Workshop*: Mar.1984, Jul.1984, Nov.1985, Jul.1986, & *Sweden-Italy-Japan Workshop*: Jun.1990(Pisa), Aug.1991(Stockholm)
    - *U.K.-Japan Workshop*: Jun.1990(London), Oct.1991(Tokyo)

- **Invitation of Experts** *(Researchers)*
  to ICOT for short periods for research exchange
  65 researchers(1982-Nov.1990): U.S.A(20), U.K.(15), France(4), F.R.G(7), Canada(5), Israel(5), Sweden(4), Italy(2), Australia(1), Austria(1), Holland(1)

- **Acceptance of Researchers** *for half to 1 year (based on agreement) from*
  U.S.*(NSF-backed)* : Dr.Tick(Oct.1988-Oct.1989), Dr.Stickel(Sep.-Nov.1991)
  France*(INRIA-backed)* : Dr.Autret/Dr.Devienne(1988), Dr.Burg/Dr.Dure/Dr.Helft(1989), Dr.Romanczuk(1990)
  & U.K.*(DTI-backed)*

- ***Dispatch of researchers (Universities, ICOT)***
  to International Conferences and Meetings to present technical papers

- **Acceptance of visitors to ICOT (Researchers, Journalists, etc.)**

- **Joint Research with other Organizations:** ***ANL , NIH , LBL, SICS***