TM-1002

Overview of the Final Stage R&D of FGCS Project

by
S. Uchida

January, 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

Overview of the Final Stage R&D of FGCS Project

Shunichi Uchida

ICOT

Fifth Genration Computer Systems (FGCS) Project

- General Goal:

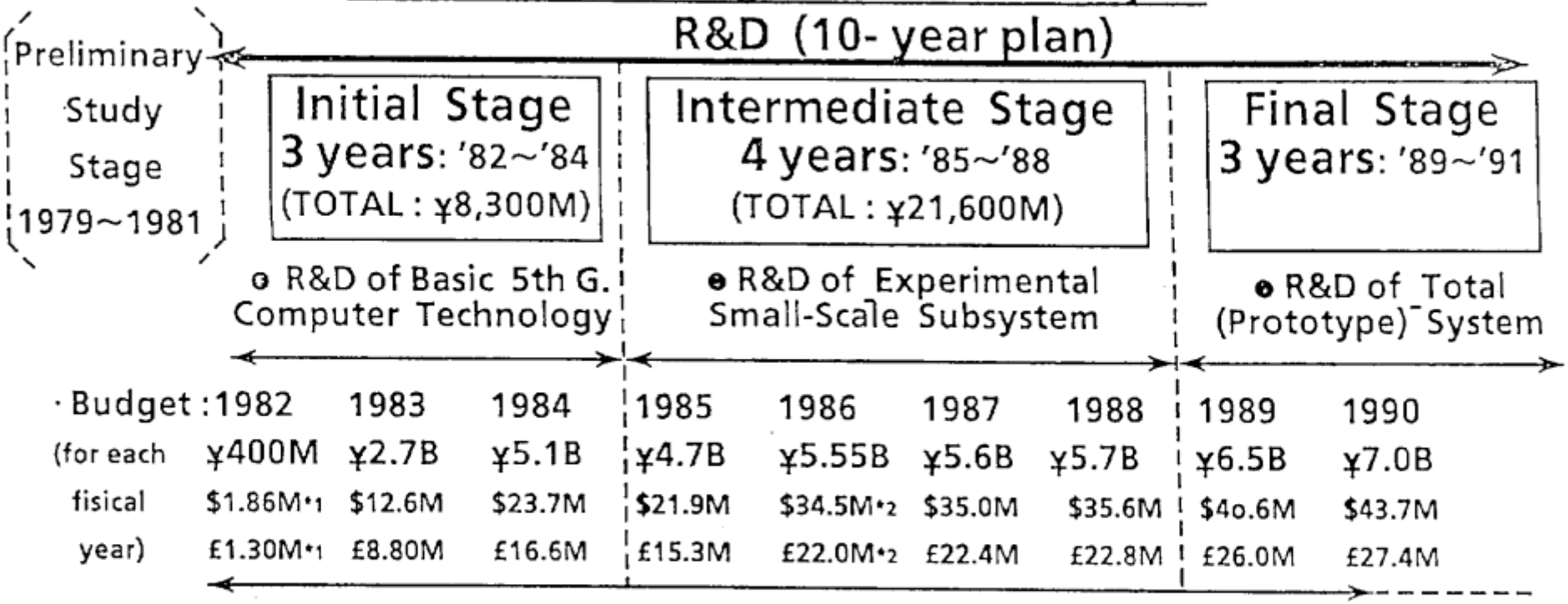
  R & D of Basic Technology for

  KIPS (Knowledge Info. Processing Sys.)

- Project Period: 10 years

	'82-'84 Initial Stage	8.3B Yen ($ 40M)
	'85-'88 Intermediate St.	21.3B Yen ($ 133M)
*	'89-'91 Final Stage	20.8B Yen ($ 149M)

Stages of & Budget for FGCS Project



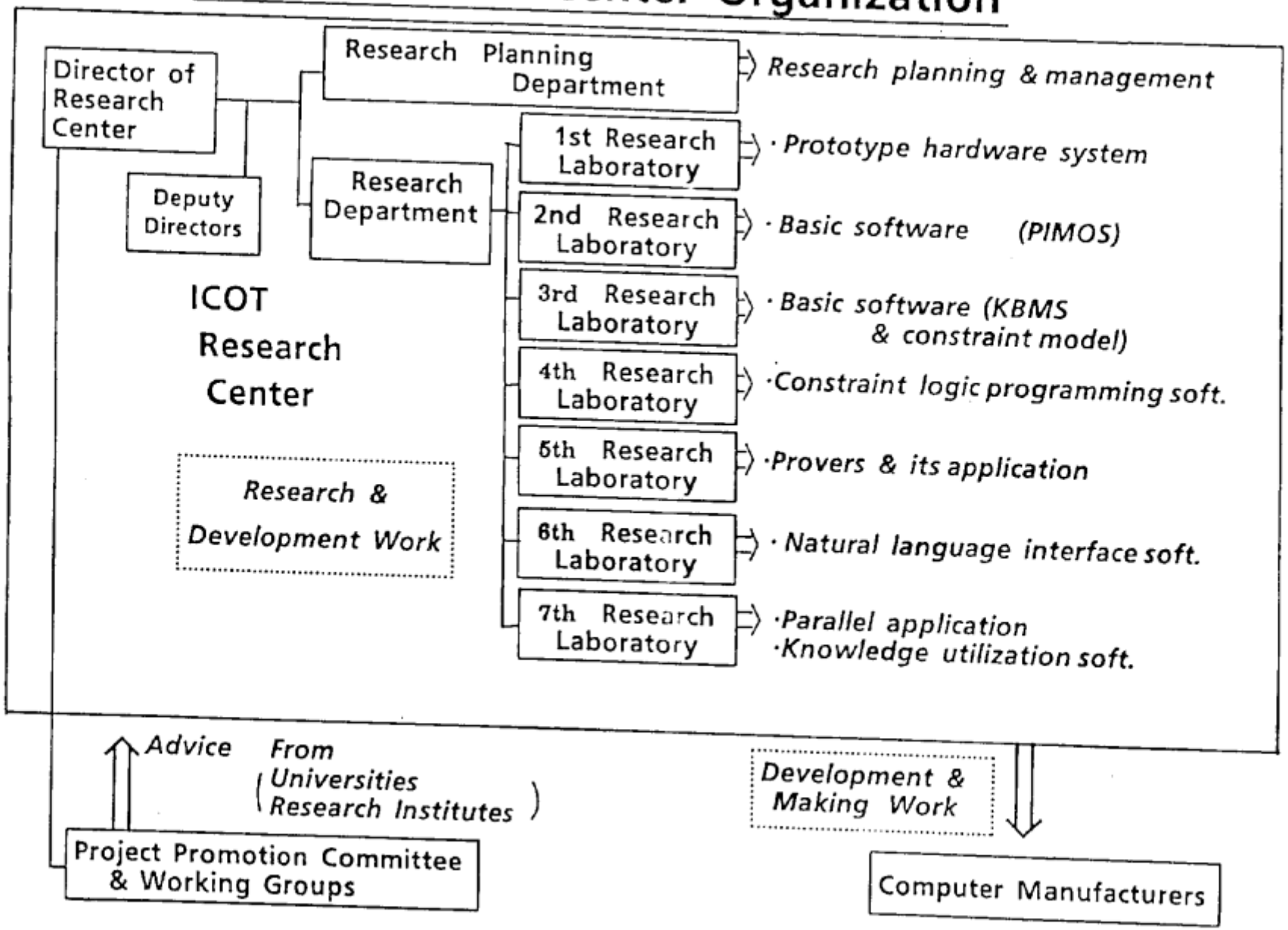
· R&D are carried out under the auspices of MITI.

(All budget are covered by MITY.)

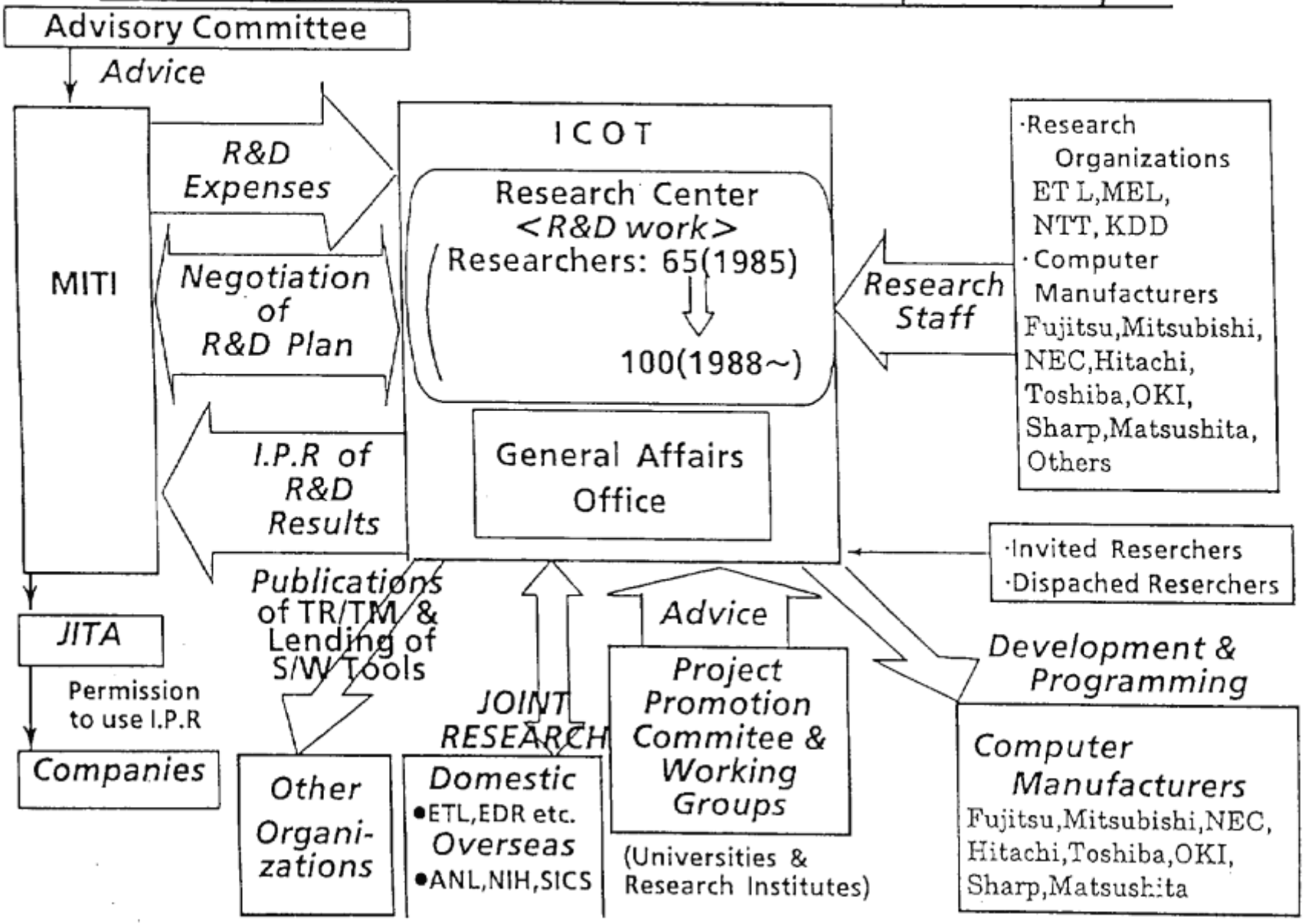
*1 $1 = ¥ 215, £1 = ¥ 307 (1982~1985)

*2 $1 = ¥ 160, £1 = ¥ 250 (1986~1989)

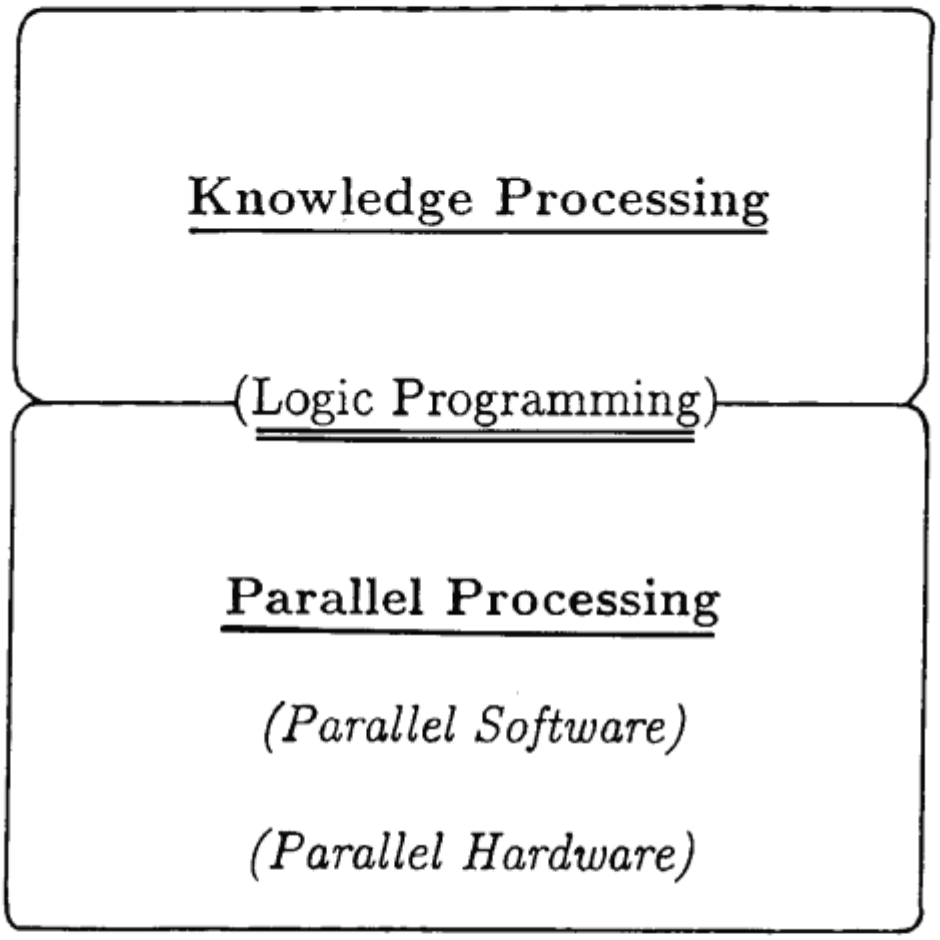
ICOT Research Center Organization



Organization of Fifth Generation Computer Project



Framework of FGCS



Sequential Inference Systems

1984 **PSI-I** 37 KLIPS (KL0/V1)

-ESP language and SIMPOS/V1

1986 **PSI-II** 330 KLIPS (KL0/V2)

-SIMPOS/V2 and Kappa-I

1988 -SIMPOS/V5 and **Kappa-II**

-Pseudo Multi-PSI and **PIMOS-S**

**About 300 PSI-II Machines and Network

1991 **PSI-III** $\Rightarrow$ 1.2 MLIPS (KL0/V2)

-SIMPOS/V7 + UNIX

-Domestic and international network link

to access PIM systems at ICOT

Knowledge Programming Software on PSI/SIMPOS

1987 - 1990

- Knowledge Representation Languages:
  CIL, CRL and Quixote
- KBMS based on Deductive and O-O DB
- Constraint Programming Languages: CAL
- Mathematical and Meta-programming Systems:
  CAP, EUODHILOS, ARGUS, etc
- NL Understanding Systems and Tools:
  DUALS and LTB
- Many Expert Systems:
  - VLSI CAD Systems
  - Go playing system
  - CASE Systems, etc.

Parallel Inference Systems

1985 **GHC**

1986 **Multi-PSI/V1** 1 KLIPS x 6PE (FGHC)

Parallel interpreter of FGHC

1988 **Multi-PSI/V2** 150 KLIPS x 64PE (KL1)

(2 - 5 MLIPS)

PIMOS/V1 and small benchmark programs

1990 **Final PIM** Chips and CPU Boards

PIMOS/V2 and many application programs

1992 **Final PIM System** 300-600 KLIPS/PE (KL1)

(PIM model/p ⇒ 200 MLIPS/512PE)

5 modules: model-p, m, c, k, i

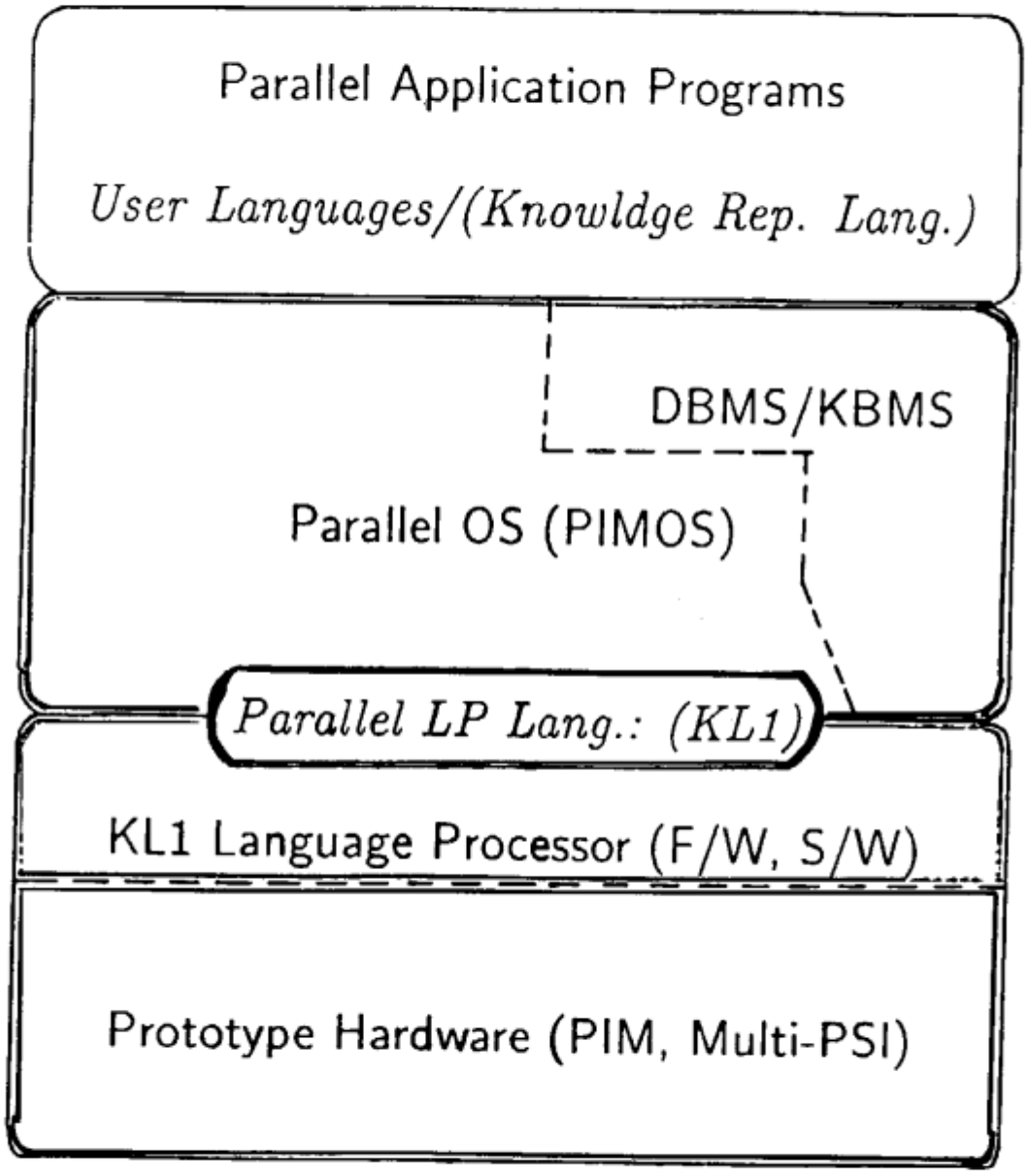
Total 1072PE's = 512 + 256 + 256 + 32 + 16

PIMOS/V3 + KBMS(Kappa-P)

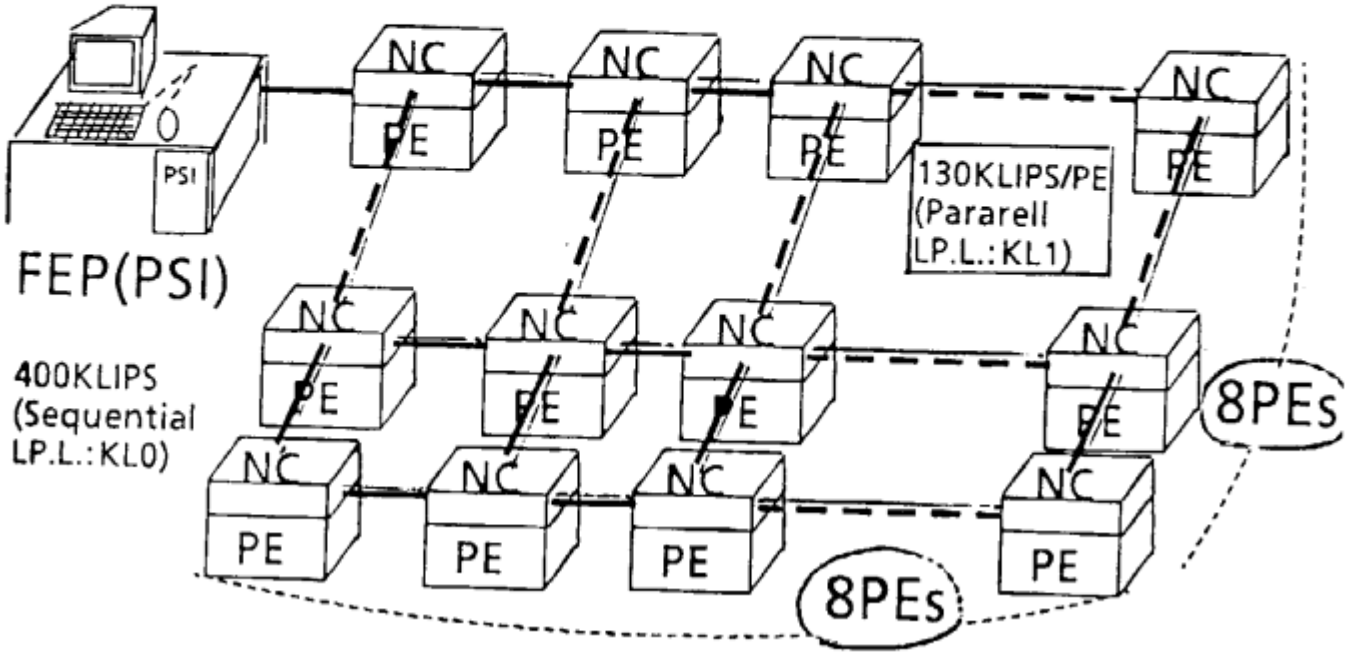
Parallel Application Systems

Prototype of the FGCS

⇒ Parallel Inference System



Multi-PSI/v2: For Parallel Software Development



- Machine language: KL1-b
- Max. 64PEs and two FEPs (PSI-II) connected to LAN
- Architecture of PE:
  - Microprogram control (64 bits/word)
  - Machine cycle: 200ns, Reg.file: 64W
  - Cache: 4 KW, set associative/write-back
  - Data width: 40 bits/word
  - Memory capacity: 16MW (80MB)
- Network:
  - 2-dimensional mesh
  - 5MB/s x 2 directions/ch with 2 FIFO buffers/ch
  - Packet routing control function

Main features of Kernel Language: KL1

- Born concurrent language based on GHC:

  Head :- Guard | Body.

  synchronization & condition execution

- Conditioning and synchronization

  ⇒ Dataflow scheme

- Two forms of ordering:

  – Strict ordering by data dependency ⇒ Dataflow

  – Prefered ordering for efficiency ⇒ Priority

- Monitoring, protection and controlling functions:

  ⇒ Meta-level control scheme NOT IN OS !

- For efficient implementation to be competitive with procedural language

  – Efficient incremental GC scheme (MRB)

  – Arrays with constant-time element updating

2. Basic Software

- Parallel OS (PIMOS) and

  Parallel programming environmnet for

  Parallel logic programming language (KL1)

- Parallel DBMS/KBMS based on

  Deductive and Object-oriented DB and

  A knowldge representation language (Quixote)

3. Parallel Application Programs

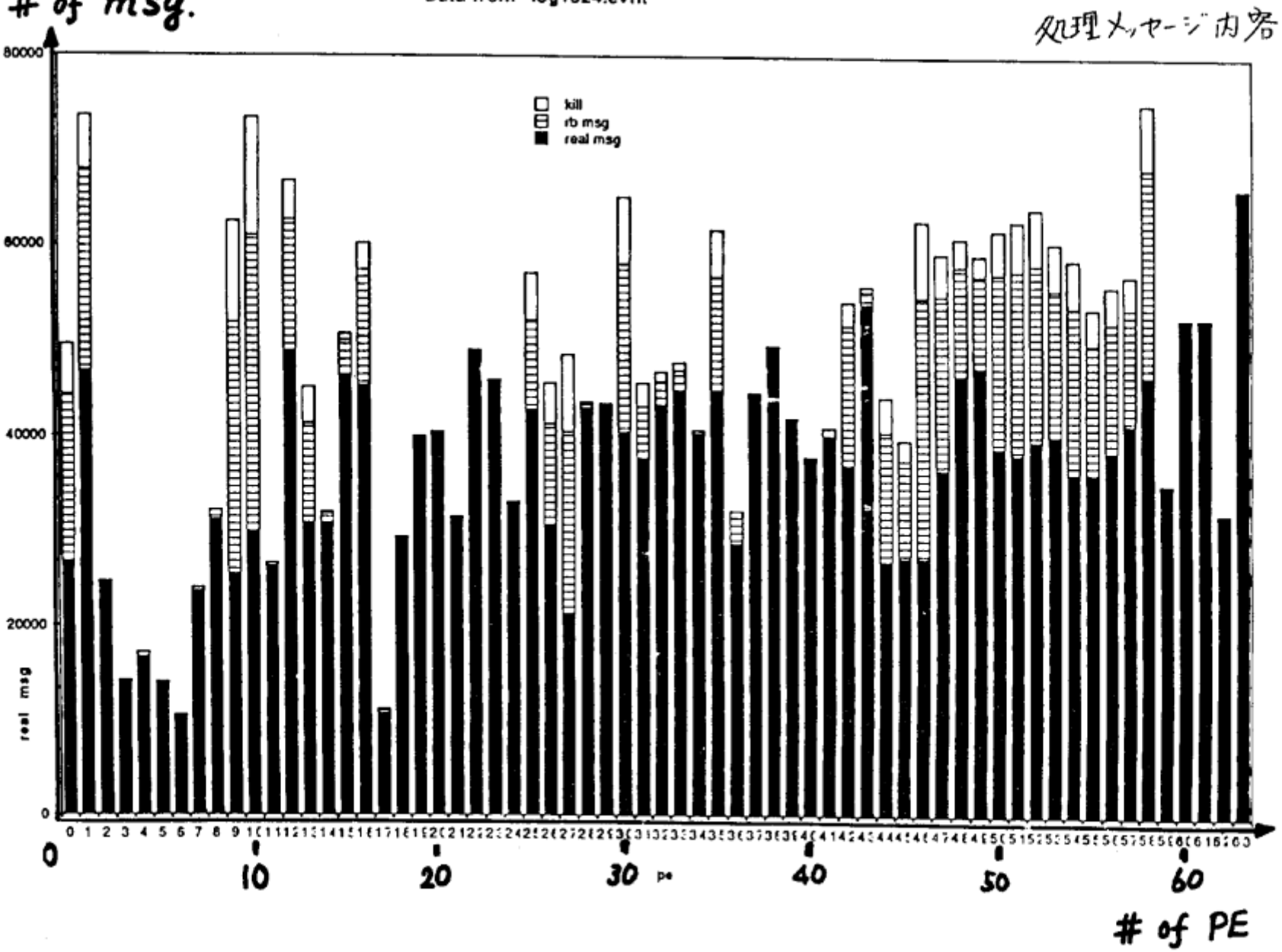
- Constraint LP System
- Parallel Theorem Provers
- NL Processing Programs
- VLSI CAD Programs
- Legal Reasoning Programs
- Genetic Information Analysis System

Parallel logic-level simulation

Objectives

- To examine efficiency of the Multi-PSI to a practical problem

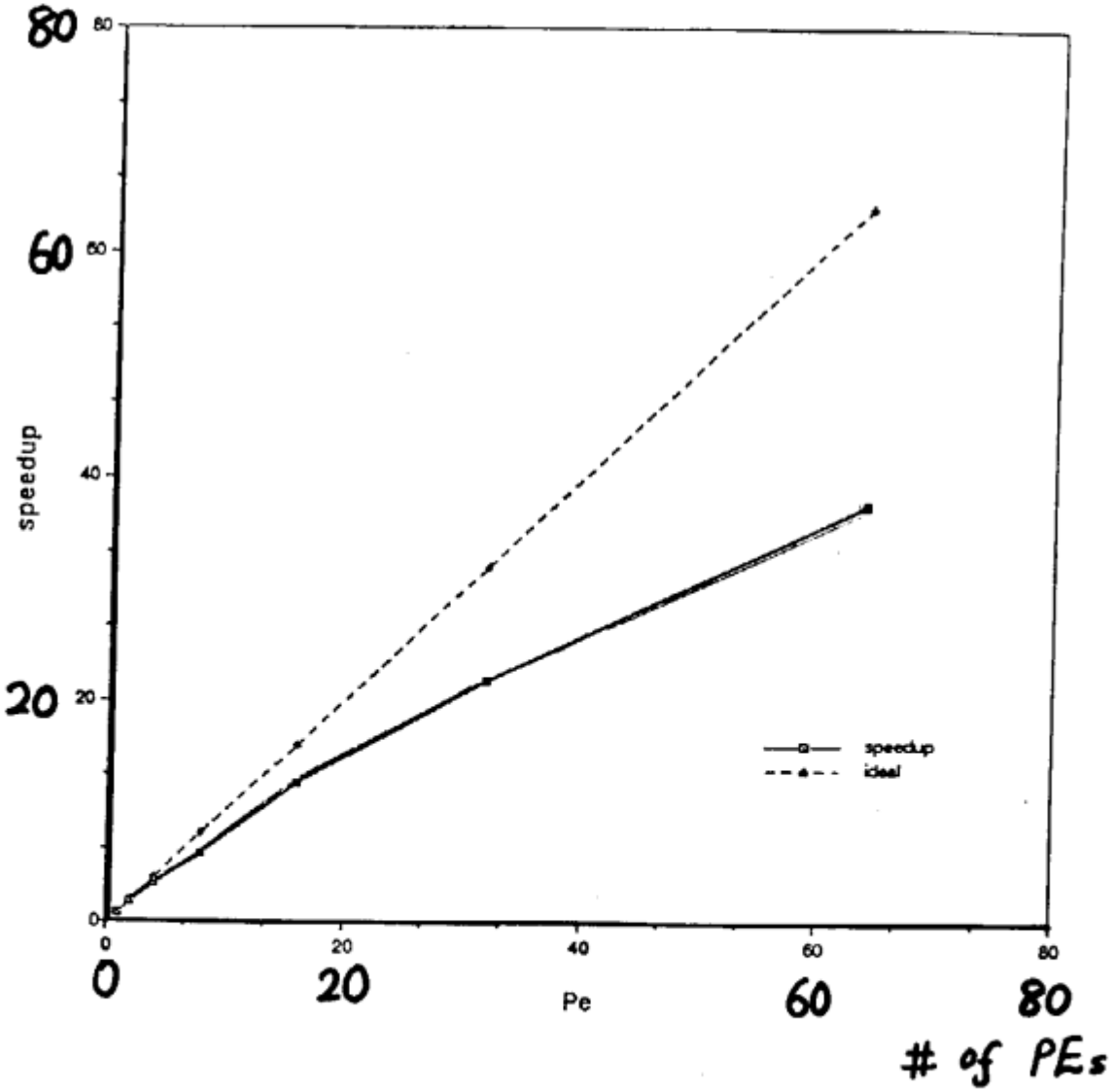
- To evaluate the virtual time mechanism


# of msg.
Data from "log1024.evnt"
処理メッセージ内容
kill
rb msg
real msg
80000
60000
40000
20000
0
real msg
0
10
20
30
40
50
60
PE
# of PE

Evaluation of the system performance

- Speed up ratio– 35~40 times faster using 64 processors than using 1 processor
- Total performance — 45k events /second using 64 processors (without indication modules)

Fig. speed-up ratio and ideal ratio

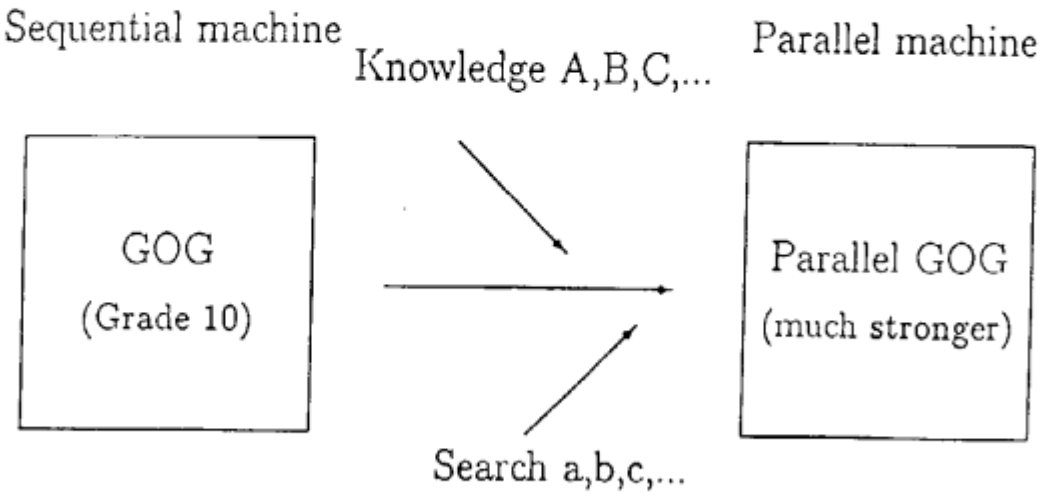


Experimental version

Parallel computer Go system

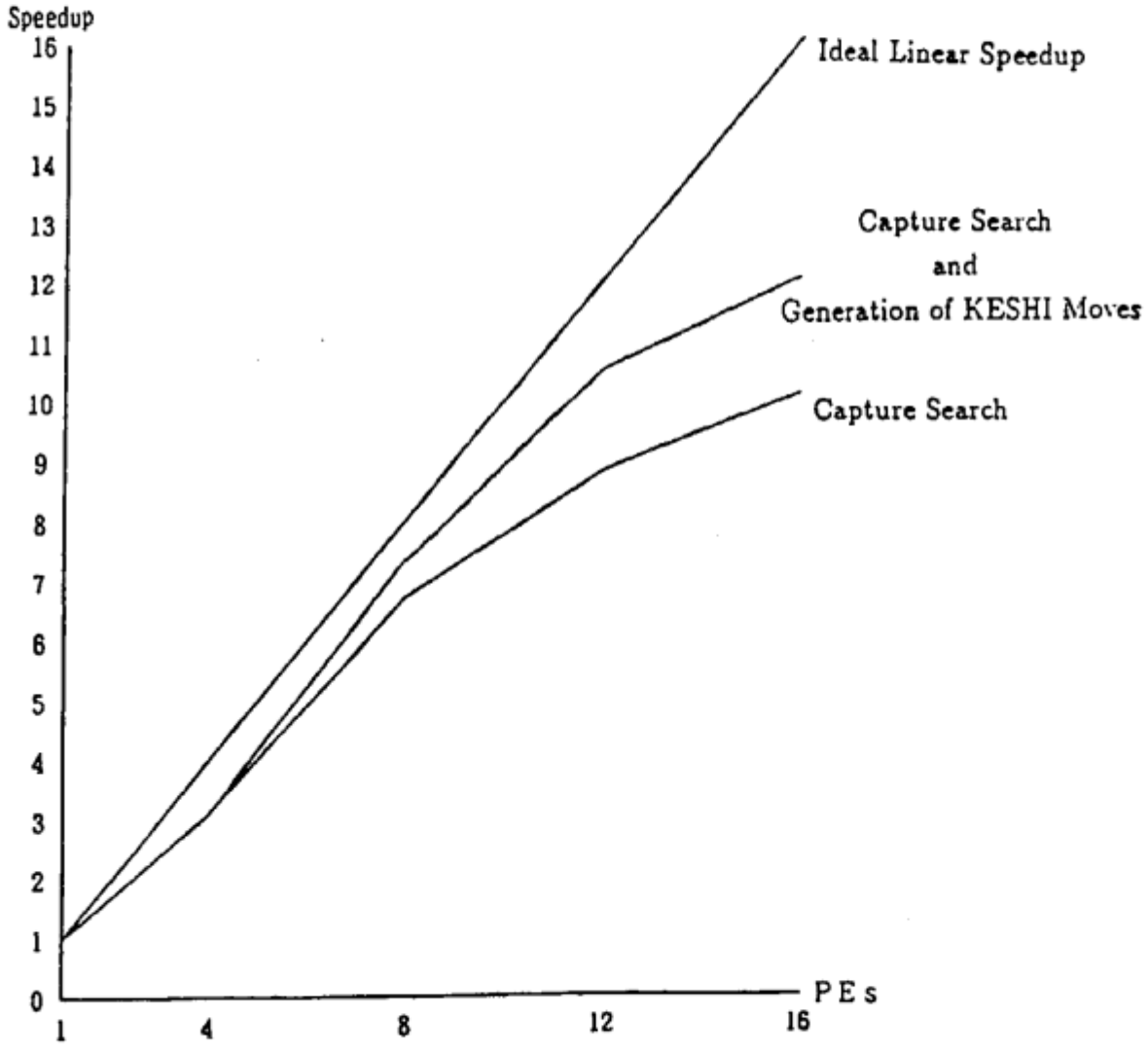
"GOG"

ICOT 7 th Laboratory


Sequential machine
Knowledge A,B,C,...
Parallel machine
GOG
(Grade 10)
Parallel GOG
(much stronger)
Search a,b,c,...

Execution Time and Speedup

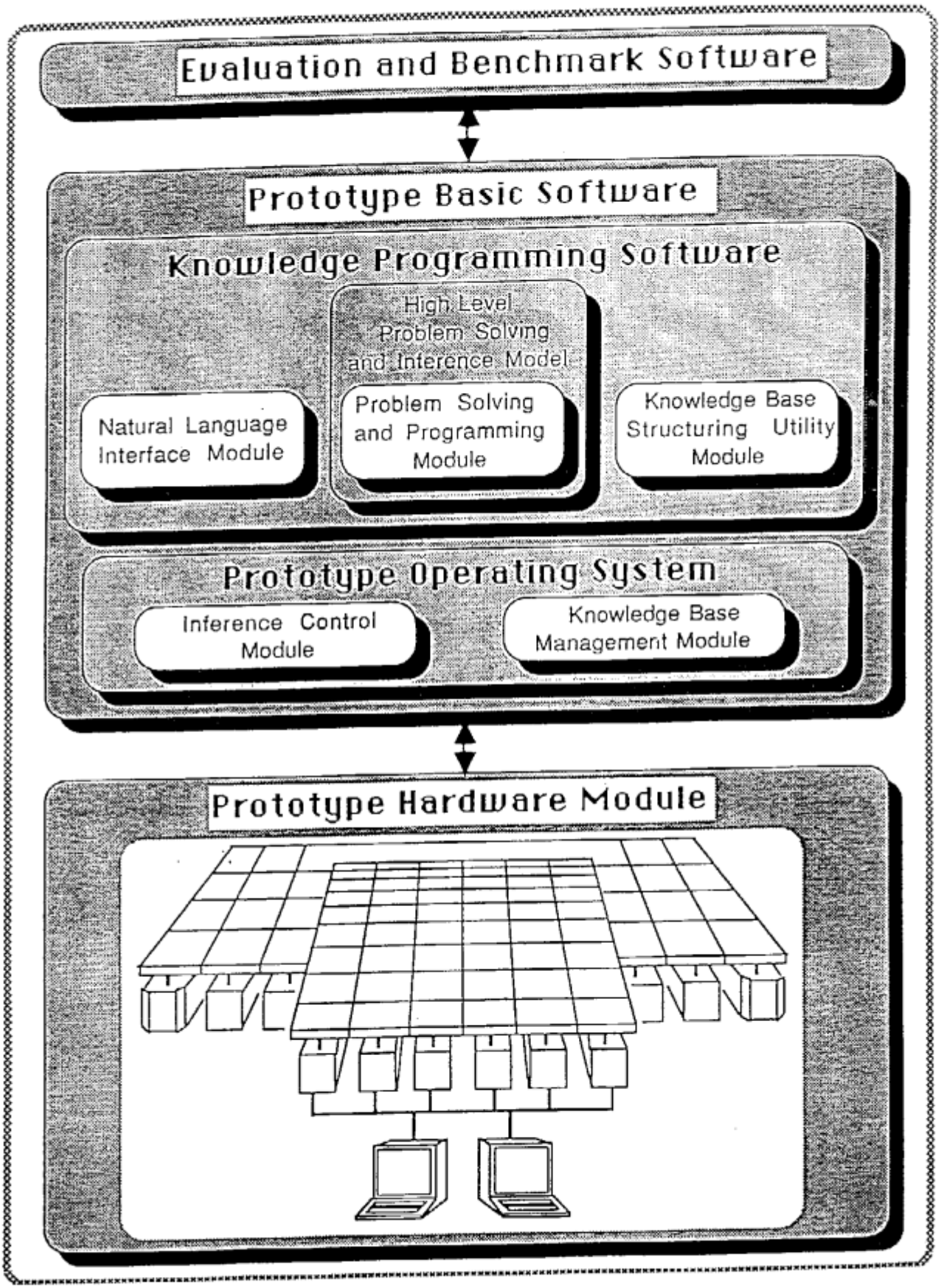
PEs	Search	Speedup	Search & KESHI	Speedup
1	2'21"	1.0	2'48"	1.0
4	0'46"	3.1	0'54"	3.1
8	0'21"	6.7	0'23"	7.3
12	0'16"	8.8	0'16"	10.5
16	0'14"	10.1	0'14"	12.0



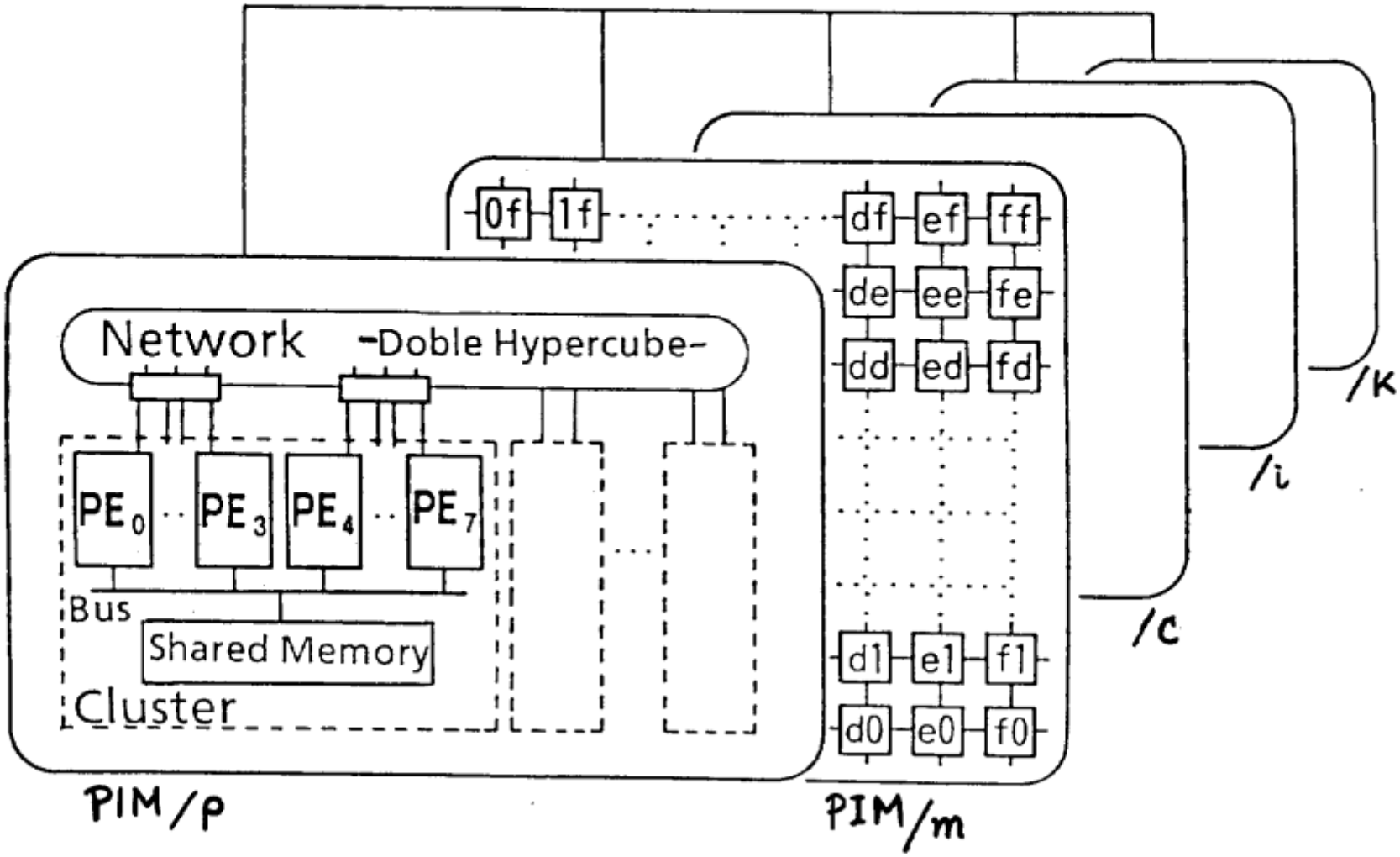
Main research targets in the Final Stage

"To establish a parallel logic programming environment to be able to open for large scale application problems"

1. Large-scale high-speed PIM modules
2. Reliable multi-user operating system
3. Parallel DBMS
4. Collection of parallel programming tools and skills through the development of application programs

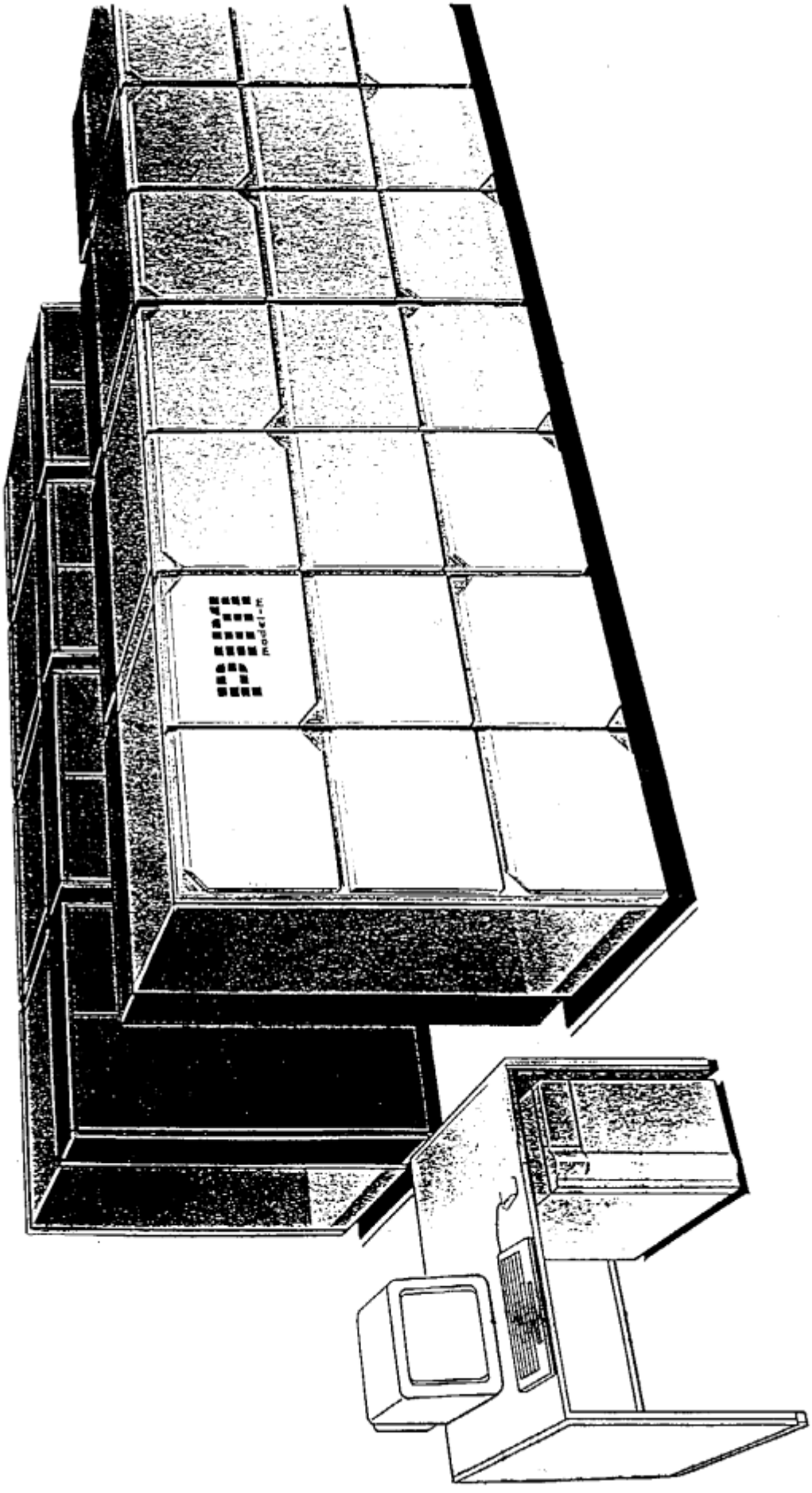

Evaluation and Benchmark Software
Prototype Basic Software
Knowledge Programming Software
High Level Problem Solving and Inference Model
Natural Language Interface Module
Problem Solving and Programming Module
Knowledge Base Structuring Utility Module
Prototype Operating System
Inference Control Module
Knowledge Base Management Module
Prototype Hardware Module

Prototype System Configuration
−Parallel Software Developement Environment−



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	60 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)	Approx. 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



Main features of Hardware Architecture

- Design of the abstract machine instruction set:

  KL1-b $\Leftarrow$ **WAM** like format

  - Optimization by compiler
  - **Architecture support**

- Design of PE

  - Data typing and dereferencing

    $\Rightarrow$ Tag architecture and stack mechanism

  - Multi-stage pipeline for instruction execution harmonized with the tag architecture

    (4-5 stage pipe line)

  - Direct support for the incremental GC:

    MRB

  - Special interrupt mechanism for KL1-b

- Efficient communication within a cluster

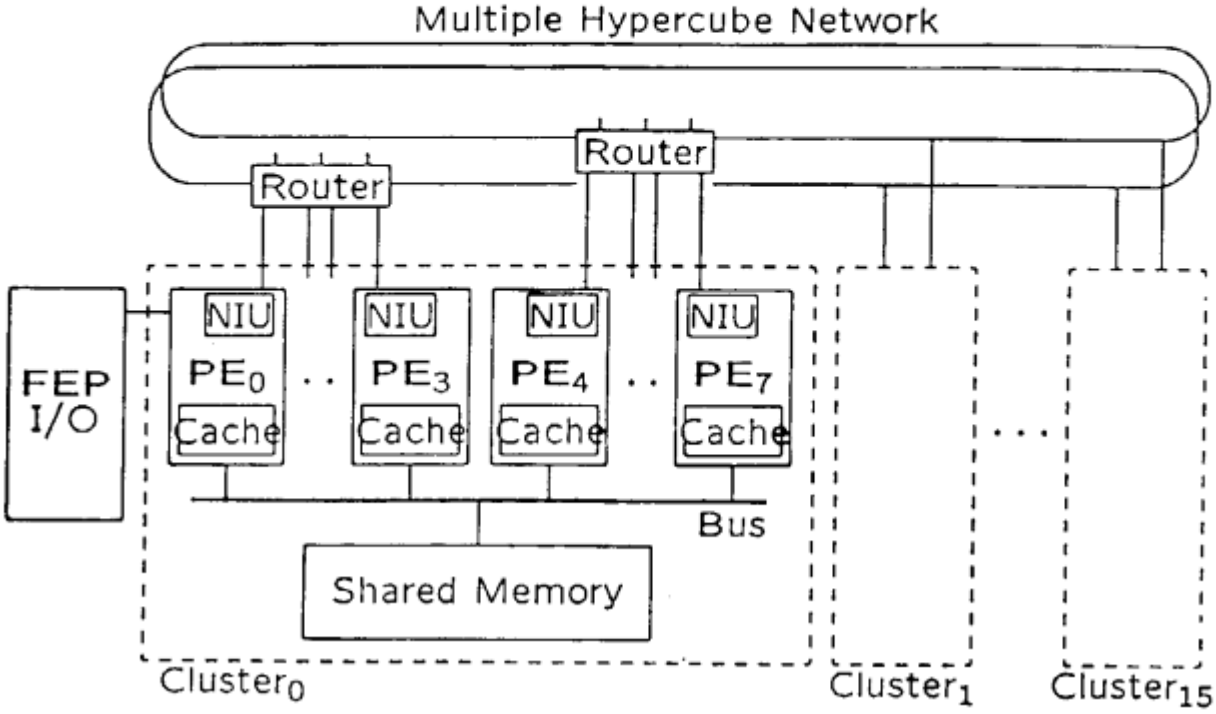
  ⇒ Coherent local cache for KL1-b

  - Specialized cache protocol designed for KL1-b
  - Low cost hardware lock

- High performance inter-cluster network

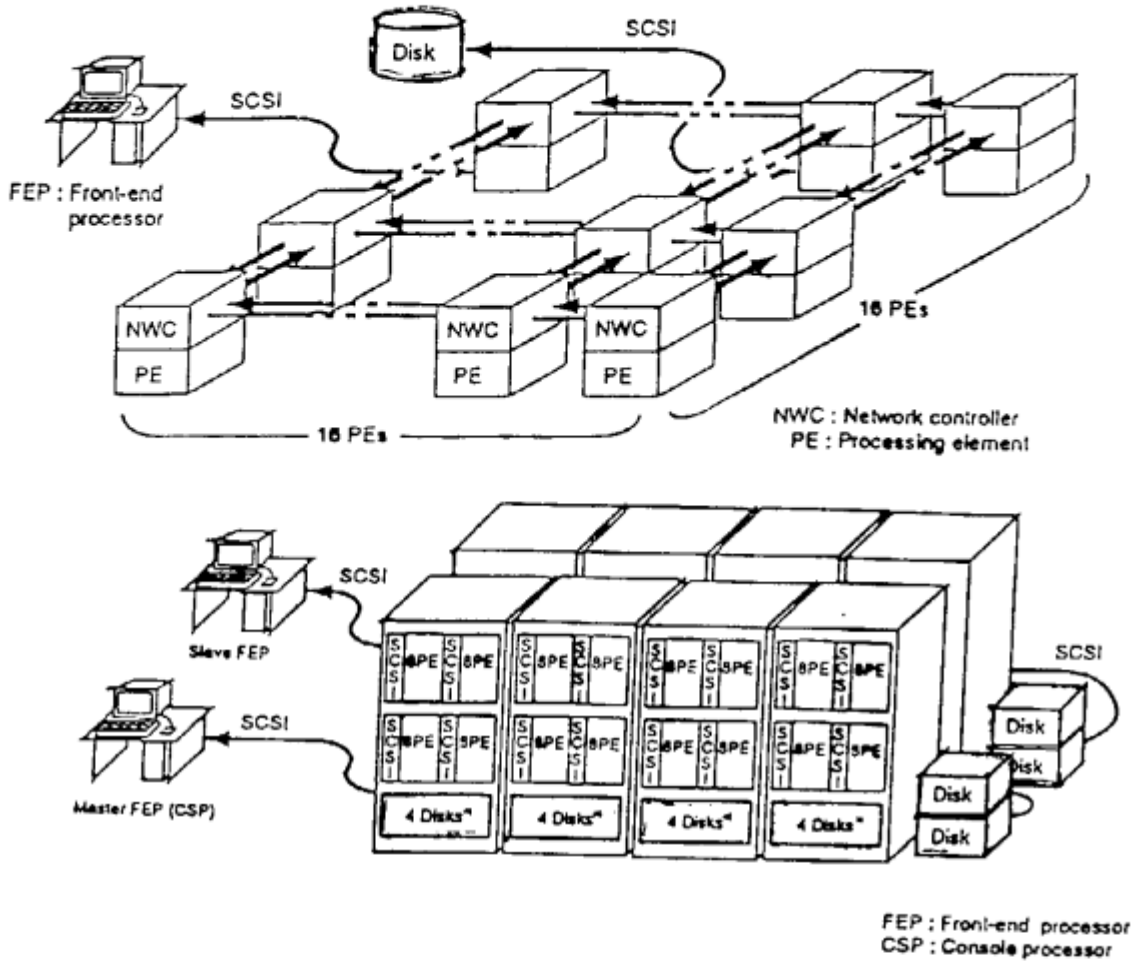
  - Efficient handling of both short and long messages
  - One network port for each PE to avoid bottleneck of message I/O channels

PIM model-P Architecture



- Machine language: KL1-b
- Architecture of PE and cluster
  - RISC + HLIC(Microprogrammed)
  - Machine cycle: 60ns, Reg.file: 40bits x 32W
  - 4 stage pipeline for RISC inst.
  - Internal Inst. Mem: 50 bits x 8 KW
  - Cache: 64 KB, 256 column, 4 sets, 32B/block
  - Protocol: Write-back, Invalidation
  - Data width: 40 bits/word
  - Shared Memory capacity: 256 MB
- Max. 512 PEs, 8 PE/cluster and 4 clusters/cabinet
- Network:
  - Double hyper-cube (Max 6 dimensions)
  - Max. 20MB/sec in each link

PIM model-M Architecture



FEP : Front-end processor
CSP : Console processor

- Machine language: KL1-b
- Architecture of PE:
  - Microprogram control (64 bits/word x 32 KW)
  - Data width: 40 bits/word
  - Machine cycle: 60ns, Reg.file: 40 bits x 64W
  - 5 stage pipeline
  - Cache: 1 KW for Inst., 4 KW for Data
  - Memory capacity: 16MW x 40 bits (80 MB)
- Max. 256 PEs, 32 PE/cabinet
- Network:
  - 2-dimensional mesh
  - 4.2MB/s x 2 directions/ch

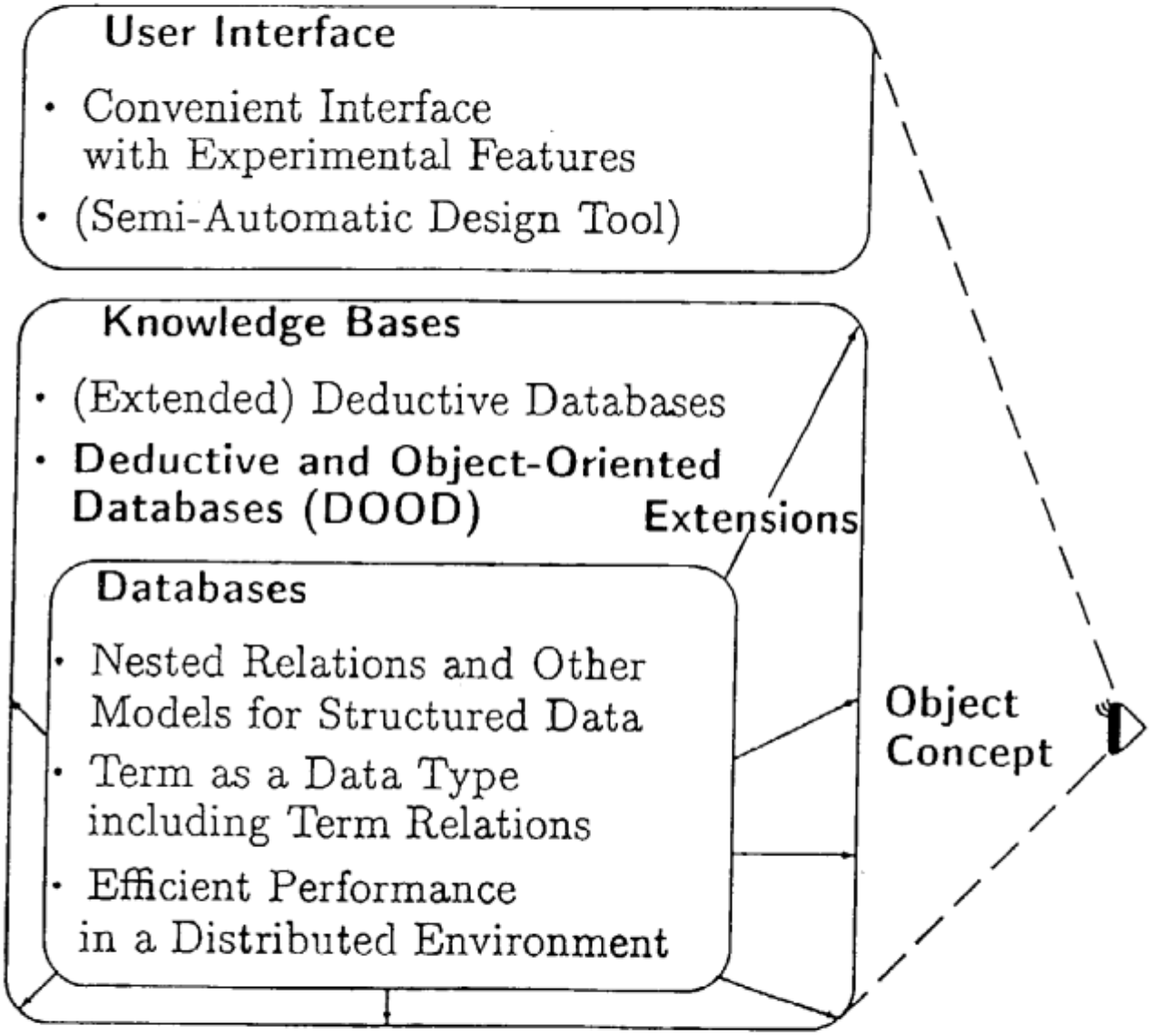
Main features of PIMOS

- A stand-alone self-contained OS written in KL1, (currently 100K lines and growing)
- Basic functions:
  - Resource management (PEs, Memories, Files and I/O) and User task management
  - Remote access control, Multi-user management
- Optimized implementation for parallel processing: Hierarchical decentralized management of the resources
- KL1 programming environment:
  - Debugging functions: Deadlock detection and Selective tracing
  - Performance debugging: Visualizing tools
  - Load distribution libraries (under development)

3. Parallel DBMS and its extension to KBMS

- Development of parallel DBMS, **Kappa-P** based on Nested Relational Model
- Implementation of relational algebraic operations on parallel and distributed environments, Multi-PSI and PIM
- Design and implementation of KBMS based on Deductive DB and Object-Oriented DB
- Design and implementation of a knowledge representation language, **Quixote** for the KBMS

Basic Policies



Current Status of the Kappa System

- Kappa-I was implemented in August, 1987
  - 60,000 Lines in ESP
  - Several β-Test Users

- Kappa-II was implemented in March, 1989
  - 125,000 Lines in ESP
  - More Efficient Performance Than Kappa-I
  - Main Memory Databases
  - User Definable Interface
  - Practical Constructors (List, Bag)
  - Widely Released

- Kappa-P will be implemented in March, 1991
  - Written in KL1
  - Parallel Version of Kappa-II
  - (More Efficient Performance Than Kappa-II?)

4. Application systems for parallel inference system including DBMS/KBMS

(a) Small benchmarking programs:
Best-path, Pentomino and Tume-Go programs

(b) Routing and logic simulation programs for VLSI CAD systems

(c) Parsers for lexical and syntactic analysis in NL understanding systems

(d) Parallel Theorem Provers

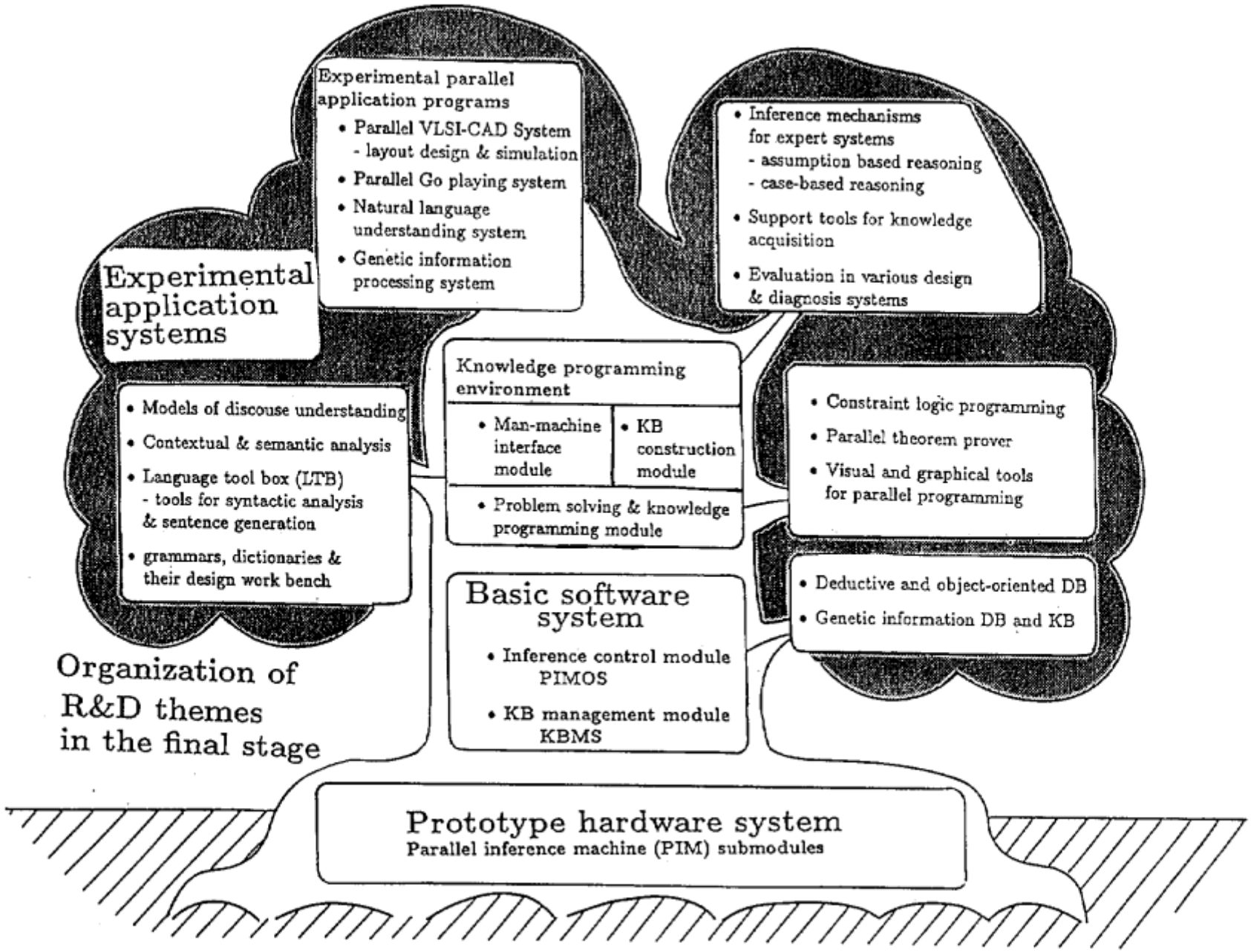
(e) Parallel implementations of Constraint Logic Programming Language, **GDCC**

(f) Parallel Go playing system, **GOG**

(g) Legal reasoning programs

(h) Programs for multiple sequence alignment of DNA or protein sequences for genetic information analysis

(i) Evaluation of Kappa-II using GenBank data and description of metabolic reactions using the knowledge representation language, Quixote


Experimental application systems
Experimental parallel application programs
• Parallel VLSI-CAD System - layout design & simulation
• Parallel Go playing system
• Natural language understanding system
• Genetic information processing system
• Inference mechanisms for expert systems - assumption based reasoning - case-based reasoning
• Support tools for knowledge acquisition
• Evaluation in various design & diagnosis systems
• Models of discouse understanding
• Contextual & semantic analysis
• Language tool box (LTB) - tools for syntactic analysis & sentence generation
• grammars, dictionaries & their design work bench
Knowledge programming environment
• Man-machine interface module
• KB construction module
• Problem solving & knowledge programming module
• Constraint logic programming
• Parallel theorem prover
• Visual and graphical tools for parallel programming
• Deductive and object-oriented DB
• Genetic information DB and KB
Basic software system
• Inference control module PIMOS
• KB management module KBMS
Organization of R&D themes in the final stage
Prototype hardware system
Parallel inference machine (PIM) submodules

TRENDS of LSI TECHNOLOGIES & 5G MACHINES