

The Personal Sequential Inference TM-0001
Machine (SIM-P or PSI)
Outline of its Architecture and
Hardware System

by

Shunichi Uchida, Minoru Yokota, Akira Yamamoto,
Kazuo Taki, Hiroshi Nishikawa,
Takashi Chikayama and Takashi Hattori

November 1982

©ICOT, 1982

ICOT

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

(03) 456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

1. Introduction

In the framework of the fifth generation computer project, the development of software and hardware tools are planned. One of the most important tools is the sequential inference machine (SIM). SIM is considered as a personal computer and several SIMs are planned to be developed in the initial stage of the project.

Since SIM is planned to be used for software research such as a natural language understanding system and an expert system, it is desirable that SIM can run very fast and process large programs. For this kind of purposes, super personal machine should be appropriate. On the other hand, interactive use of a computer is often important in the development of various experimental programs. For this purpose, a medium performance machine, which is relatively cheap and can be copied to share it by a few users, is more appropriate.

In the course of the development, two types of SIM are planned. One of them is the personal sequential inference machine which is called "PSI" or "SIM-P". Another one is the super personal model of SIM which is called "Super PSI" or "SIM-C".

PSI is designed to be a medium performance personal machine which supports the logic programming language KL0 which is also called the version 0 of FGKL (Fifth Generation Kernel Language). And PSI is designed to attain about 20-30 K-LIPS (Logical Inference Per Second). On the other hand, Super PSI will be designed to be a high performance machine which is planned to attain 100K-1M LIPS.

In this document, functional specification of the machine architecture (PSI or SIM-P architecture) is described. As this document is a tentative report of the machine design, the content will be changed in the course of detail design and implementation processes.

2. Main Design Characteristics

As PSI is considered as a main computing tool in the last part of the initial stage and requested to complete its experimental hardware in early summer in 1983, main effort is to be made for designing its processing unit and memory unit. Some functions such as a virtual memory system are now considered as future extensions.

Functional design of PSI was begun in September 1982 and its design characteristics are summarized as follows.

2.1 Requirements for Functional Design

- An efficient research tool to promote software and hardware research by providing researchers with a good programming environment.
- A prototype of a personal computer to be used in the this project.
- Efficient support of logic programming languages by hardware and firmware.
- Support of sophisticated man-machine communications by providing such devices as a bit-map display, a pointing device, and Japanese character input and output devices.
- Connection with other machines via local area network to form a distributed computer system.

2.2 Main features of PSI architecture

- Firmware implementation of KL0 interpreter and kernel functions of operating system.
- Partial architectural support of process switching to implement a single-user multi-process system.
- Three machine modes; Kernel (supraGC), Supervisor and User modes.
- Employment of the tag architecture and microprogram control.
- Partial hardware support of important functions for KL0 interpretation such as unification and resolution by special hardware registers and stack manipulation mechanisms.
- Employment of a cache to improve memory access speed.
- A logical to physical address translation mechanism to implement multiple virtual stacks in main memory.
- Hardware and firmware support of interrupt handling and mode switching for efficient control of various input and output devices.
- Employment of a standard bus (IEEE-796 bus, or MULTIBUS*) for peripheral devices.
- Local area network support by ETHER-NET**.

-Ease of extension for functional units such as a arithmetic unit including floating point operations, main memory, peripheral devices and so forth.

-Such functions as a virtual memory system and a support of concurrent prolog are considered as future extensions.

-Special hardware and firmware support for collecting such statistical data of program's behavior as average number of variables appeared in one unification, mean time for one resolution and unification (it should be necessary to indicate the processing speed by our new unit: LIPS) statistics of memory usage, cache miss-hit ratio, and locality of address pattern.

Note:

* MULTIBUS is the TM of Intel corp.

** ETHER-NET is the TM of Xerox corp.

3. Performance goal and machine organization

3.1 Performance goal

PSI is planned as a medium performance personal machine, however, it must fulfil the requirement for computing power and memory capacity to run small to medium scale programs. To determine performance goal of machines, statistical data on the behavior of programs are necessary, however, we have very few data on logic programming languages.

Only data we knew is that Prolog programs requires much memory space especially in their debugging phase because they usually run using the interpreter. However, if they are once compiled, the necessary memory space is shrunk very much. It is not likely that the memory becomes full by program codes. Therefore, some appropriate maximum capacity, say 1MW, may exist for debugging phase.

For computing speed, the maximum can be determined by hardware architecture and component technology rather than the requirement for program's execution speed as far as experimental programs are concerned.

Roughly speaking, we estimated that the first experimental model of PSI should attain more than 10 KLIPS in computing speed and have more than 1MW for main memory.

For the experiments of intelligent man-machine interface and interactive use of the machine, it is required that such devices as a bit-map display and a pointing device can smoothly be controlled by KL0 programs.

Adding to these conditions, short development period was considered to determine the performance goal of PSI. As the current result of functional design process, performance goal and main hardware features are determined as follows.

The machine is designed to attain 20-30 KLIPS and will be implemented by TTL ICs. It employs the tag architecture and each data cell in main memory consists of 8 bit tag and 32 bit data fields. Logical address is specified by 32 bits. Logical address space is divided into 256 areas by 8 bits. Each area can be used as independent stack or heap area. Maximum size of the area is 16 MW (24 bits).

As the first interpreter of KL0 is designed to use four stacks, namely a control stack, a local stack, a global stack and a trail stack, each process is to use four areas for the stacks. For the codes of KL0 programs, it is designed to be able to share them and thus one area may be commonly used by all the processes. Then, the maximum number of processes supported by the firmware is 64, however, it may be extended by software support made by the operating system.

The KL0 language includes several primitives to handle interrupts, exceptions, process creation, deletion and synchronization. Partial hardware and firmware supports are provided to implement these primitives efficiently. These supports are expected to be useful for implementing a concurrent prolog interpreter as well as the control of fast man-machine interactions.

3.2 Machine organization

Generally speaking, PSI consists of three hardware modules, namely, a processor module, a memory module and an input and output interface module. These modules are now considered for the first experimental model of PSI. Thus, the processor module includes some redundant circuits as room to accept new ideas in firmware and software implementations. Extension of the modules are especially considered in this model.

1) Processor module

The processor employs microprogram control. Its micro cycle is designed less than 200 ns and its micro instruction is 64 bit wide. Its microprogram is usually stored in RAM (16 KW Max.) but it has a small ROM to store a bootstrap program and other small control programs. For the first experimental model, a general purpose microcomputer is attached as a service processor for hardware debugging and collection of various performance data of the machine.

The data path includes several registers, busses and functional units. It is designed for 32 bit data. Some register files are 40 bit wide to store 8 bit tag field and 32 bit data. Some register is specialized to efficiently handle stack frames in the process of resolution. This register is useful for the call of a new clause. It also includes a 4 KW scratchpad file to keep process descriptors for fast process switching.

For arithmetic and logic operations, a fast 32 bit adder unit, a mask and shift unit and a multiply and divide unit are provided. A floating arithmetic unit is considered as one of the optional units. An interface to a fast data bus (32 bit wide) may be attached as one of the optional units.

2) Memory module

Memory module of PSI includes a cache and a logical-to-physical address translation mechanism. The cache has two 40 bit-4 KW buffers and enables us to access a data in the main memory every two micro cycles. The address translation mechanism employs two level mapping method and is designed to efficiently implement stacks in main memory. This mechanism observes the stack top and issues interrupt when the stack top is extended over the allocated limit of memory cells.

The main memory can be extended upto 16 MW, however, it is expected that it usually has 1-2 MW. The memory is divided into 1 KW pages and accessed through two address mapping tables.

3) Input and output interface module

As the control of various input and output devices is one of the important task of PSI, two external busses are provided. One is a standard bus (IEEE-796 bus or MULTIBUS*) and another is a high speed bus. Furthermore, a special bus interface for a service processor (for the experimental model, it is LSI-11) is

provided.

The standard bus is used to connect such usual devices as a magnetic disk, a pointing device, a floppy disk, and an ETHER-Net interface. A memory module is also attached to this bus for buffering input and output data. PSI microprogram directly control the input and output data transfer and thus no DMA transfer between PSI main memory and the standard bus devices is provided.

The high speed bus has a 32 bit data path and is used for fast data transfer which is controlled by the microprogram. The bit-map display is an example of the devices to be attached to this bus.

3.3 PSI and a local area network.

PSI is designed as an independent personal machine which provide its users with a powerful and versatile computing facility. however, it gives the users a more productive environment by being connected to a local area network.

PSI has an ETHER-Net interface and its operating system is planned to support distributed processing. PSI will have the capability of communicating with other machines such as other PSIs, super-PSIs (or SIM-C), relational database machines, LISP-machines.

4. Conclusion

PSI is now under the functional design process. however, we have already found several interesting research subjects concerning to interpretation methods, concurrent processing, memory management and garbage collection, interrupt and exception handling, implementation of virtual memory, and so forth.

Making full use of the flexibility of PSI firmware and hardware, it is expected that PSI will be a workbench of many new ideas on software and hardware research.

