ICOT Technical Memorandum: TM-

Outline of the Personal Sequential Inference Machine:PSI

by

Shunichi Uchida,Minoru Yokota,

Akira Yamamoto,Kazuo Taki,

Hiroshi Nishikawa

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

ABSTRACT

PSI is a personal computer system being developed as a tool for providing researchers with an efficient programing environment. It directly supports a logic programming language, KL0 (Fifth Generation Kernel Language, Version 0), with firmware and hardware.

Its interpreter is implemented in the firmware and several hardware mechanisms are provided to attain almost the same level of performance as the DEC-10 Prolog on DEC2060. It also provides the user with a large memory space, 40 bits X 2 to 16 MW, which is essential for developing actual application programs like an expert system.

To make efficient man-machine interaction possible, such input and output devices as bit-map display, pointing device and key-board are provided. A local area network is also being developed to build a distributed system.

1. Introduction

In the fifth generation computer project, a variety of software and hardware tools is planned to be developed. One of the most important tools is the sequential inference machine (SIM). SIM is considered a personal computer supporting logic programming language KL0 (Fifth Generation Kernel Language, Version 0). [1]

Several models will be planned for SIM in the future. The first experimental model is named PSI (Personal Sequential Inference machine). It is under development and is planned to complete firmware, hardware and the subset of its operating system in the initial stage of the project. [2],[3]

PSI is designed to be a medium performance personal machine which can attain about 20 K-LIPS (Logical Inference Per Second). It provides large memory space (40 bits X 2 to 16 MW) which is essential for developing experimental application programs like an expert system.

To make efficient man-machine interaction possible, input and output devices such as bit-map display, pointing device and key-board are provided. A local area network is also being developed to build a distributed system. [4]

The outline of the PSI architecture and hardware system is described in this paper.

2. Design of the machine architecture

Machine language of PSI is called KL0. It is a Prolog-like language, however, it is intended to be used in writing the operating system. Thus, a variety of built-in predicates is added in KL0 to describe the kernel of the operating system and device control.

The internal representation of KL0 is obtained by compiling user-described KL0 programs. [1] Ooptimization, techniques such as tail recursion optimization, are made by the compiler. The KL0 interpreter, which interprets internal representation, is implemented in firmware. The architecture of PSI is primarily designed to support the interpretation efficiently.

Considering the experience of using DEC-10 Prolog on DEC2060, for the performance and memory space, we estimate the first experimental model of PSI should attain more than 10 KLIPS in computing speed and should have more than 1MW for main memory.

The machine is designed to attain about 20 KLIPS and is implemented by TTL ICs. It employs tag architecture and microprogram control. Each data cell in the 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 an independent stack or a heap area. Maximum size of the area is 16 MW (24 bits).

The operating system of PSI is planned to implement a single-user multi-process support mainly for experiments of intelligent man-machine interface and interactive use of the machine.

Furthermore, it is required that devices such as bit-map display and pointing device can be smoothly controlled by KL0 programs. KL0 language includes several primitives to handle interrupts, exceptions, process creation, deletion and syncronization.

Thus, architectural support is essential. This includes support for process switching, separation of the address space assigned for each process, input/output control and so forth.

The main design characteristics of PSI are summerized as follows.

-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.

-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) for peripheral devices. Also, local area network support is also planned to be implemented.

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

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

3. Machine organization

Roughly speaking, PSI consists of three hardware modules, namely, a processor module, a memory module and an input and output interface module. The organization of PSI is shown in Fig.1.

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.). In the first experimental model, a general purpose microcomputer (LSI-11) is attached as a supervisor processor for hardware debugging and also for the collection of various data to evaluate machine design.

The data path includes several registers, busses and functional units. Most of them are 32 bit wide, however, some of them are 40 bit wide to handle both 8 bit tag field and 32 bit data field. The internal data busses are designed to make unification and resolution as fast as possible. Larger register files (total capacity about 3 KW) than the conventional machines are included to keep data such as stack-top frames to decrease memory access operations. They are also used to keep process descriptors for fast process switching.

For arithmetic and logic operations, a fast 32 bit adder unit, a mask and shift unit are provided. A floating point arithmetic unit is considered as one of the optional units.

2) Memory module

The 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 data in the main memory every two micro cycles. The address translation mechanism employs two level mapping method and is designed to efficiently implement multiple virtual stacks in the 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 is divided into 1 KW pages, and it can be extended upto 16 MW. However, 2 to 4 MW is sufficient for usual software experiments.

3) Input and output interface module

As the control of various input and output devices is one of the important tasks of PSI, flexible input and output interfaces are desirable. PSI has employed the IEEE-796 bus as its standard bus and prepared a high-speed port as an option. The standard bus is used to connect such usual devices as magnetic disk, printer, floppy disk, and LAN interface. A bit-map display is also connected to this bus.

As PSI microprogram directly controls input and output data transfer and thus no DMA transfer between PSI main memory and standard bus devices is provided, a small capacity memory module is attached to this bus for buffering input and output data.

4. Conclusion

PSI is now in the detail logic design process. However, we have found several interesting subjects to study, such as 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 many new ideas on logic programming in software and hardware will be born in the near future.

Acknowledgement

The authors express their thanks to Dr. Takashi Chikayama and Mr. Takashi Hattori for valuable advice and help in implementation of KL0 interpreter and mechanisms to support the operating system. They also express their thanks to Mr. Kazuhiro Fuchi, Director of ICOT and Mr. Toshio Yokoi for their suggestions and supports.

Reference

1. Chikayama, T., "ESP as a Preliminary Kernel Language of Fifth Generation Computers", New Generation Computing, Vol.1, No.1, (to appear), Jul. 1983.

2. Uchida, S., "Toward a New Generation Computer Architecture", in VLSI Architecture as Chap. 19, Prentice Hall (to appear), and also as Tech. Rep. of ICOT, TR-001, Aug. 1982.

3. Moto-oka, T., "Overview to the Fifth Generation Computer System Project", Proc. of 10th Int'l Sympo. on Computer Architecture (to appear), Jun. 1983.

4. Murakami, K., Kakuta, T., Miyazaki, N., Shibayama, S., Yokota, H., " A Relational Data Base Machine: First Step to Knowledge Base Machine", Proc. of 10th Int'l Sympo. on Computer Architecture (to appear), June 1983.

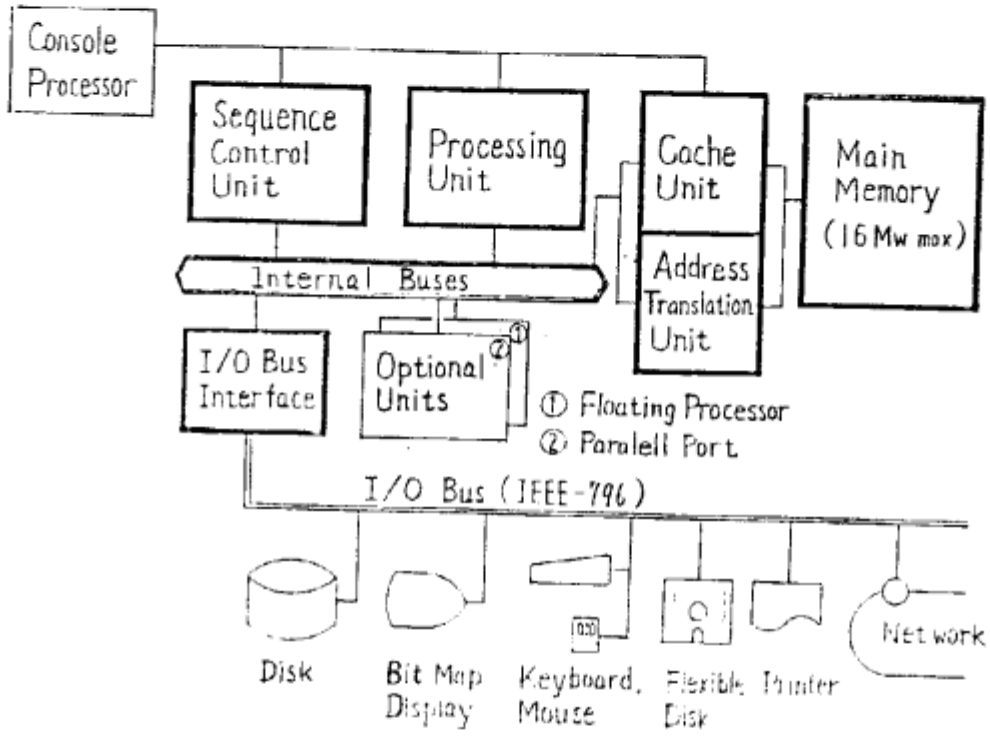

Fig. 1 The organization of PSI