

THE EXPERIMENTAL LISP MACHINE

Kazuo Taki,** Yukio Kaneda* and Sadao Maekawa*

* Dept. of Systems Engineering, Kobe University, Rokkodai, Kobe, Japan

** OMIKA Works, HITACHI Ltd., Omika cho, Hitachi shi, Ibaraki, Japan

This machine is specially designed powerful enough to implement an efficient LISP interpreter in microcodes. A LISP processor module and a main memory module are connected to the unibus of a LSI-11 system. The LISP processor module is the microcoded device with a writable control storage that interprets a LISP program in the main memory. Four microprocessor slices(Am 2903) and one microprogram sequencer(Am 2910) are used.

1. INTRODUCTION

The LISP is widely used in the artificial intelligence research community. However LISP processing systems have several nontrivial problems to be solved. One of the most important problems is that the computing time of getting solutions of the problems is much slower than the case of numerical problems. The main reason for the slow speed of computation is due to an unfitness of LISP and a conventional computer architecture at present.

An approach to this problem is to design a machine whose architecture is LISP oriented and code structure more closely reflects that of the language. The machine proposed here is an experimental microprogramable computer system for LISP. This machine is specially designed powerful enough to implement an efficient LISP interpreter in microcodes.

2. THE LISP MACHINE SYSTEM

Our LISP machine is a stand alone personal computer, and Fig.1 illustrates the structure of the machine.

The LISP processor module and the shared main memory module are connected to the UNIBUS of the LSI-11 system(DEC Co). The LSI-11 acts as the master processor of the system and plays the roles of system initialization, input and output, and controls of the LISP processor module. The LISP processor module is a microprogramed device with a writable control storage(WCS) that interprets a LISP program stored in the main memory. The main memory module is a 64k words(32-bit words) semiconductor memory and its access

time is about 75 nsec. The main memory module is also accessible from the LSI-11 using a page mode access mechanism. A data transfer unit between the LISP processor module and the main memory module is a 32-bit word.

3. THE LISP PROCESSOR MODULE

The LISP processor module is implemented with a microprogrammed architecture(Fig.2).

Its kernel consists of an arithmetic and logic unit(ALU) and a computer control unit(CCU). The CCU has a WCS(4k of 56-bit words) to accommodate the specialized microcodes to support the LISP.

It has hardware for extracting specialized fields in a word, called field extractor, and the mapping memory which converts a main memory address into 3-bit code.

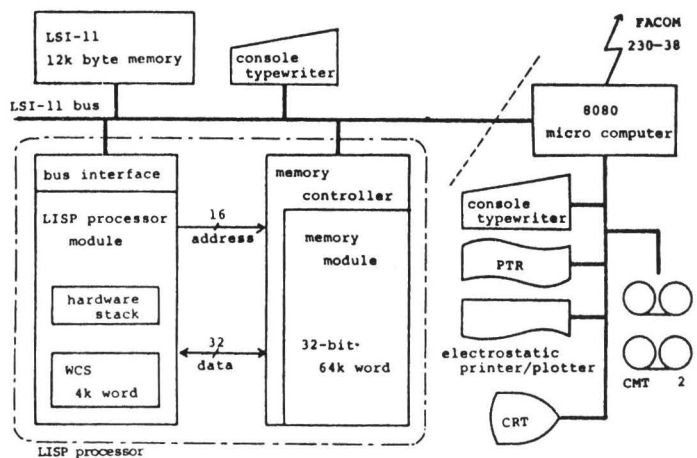


Fig.1 Hardware structure of the machine

It has not a cache, but has a high speed stack(4k of 16-bit words) acts as a high speed main memory area for a stack, which is where most of the memory references go in LISP. It has also three buses called A bus, Y bus and B bus. A bus is the data bus, Y bus is the destination bus and B bus is the data bus between the CCU and ALU through which a constant value is supplied to the ALU.

3.1 The ALU and CCU

The ALU is an array of four 4-bit expandable bipolar microprocessor slices(Am 2903). And the CCU is a control unit of the machine and consists of a microprogram sequencer(Am 2910), WCS and other circuits. The Am 2910 is an address sequencer intended for controlling the sequence of execution of microinstructions stored in the WCS(access time is about 150 nsec). Besides the capability of sequential access, the sequencer provides conditional branchings to any microinstructions within its 4096 microwords range. The flag register contains ALU's flags, the mapping memory flags and other flags. All these flags are available for conditional branching under the microprogram control.

3.2 The Hardware Stack, Field Extractor, Mapping Memory and Bit Addressing Unit

The hardware stack is a 4096 16-bit word memory (access time of the memory chip is about 70 nsec). Two pointer registers STP and SAR are provided for stack operations.

The field extractor locates on the A bus and all data to the ALU flow through the unit. This unit is useful to extract a fixed specified field of data.

The mapping memory is a 1k 3-bit word memory. Whenever the main memory references are made the address is gated to the unit. 1k 3-bit word memory is considered as a tag table whose entries are associated with every 64 words sections of the main memory. The supplied address is converted into three bits code which indicates the data type of the object. By using this three bits code rapid conditional jump is possible in the microcodes.

The bit addressing unit is a functional unit which tests and sets a bit on a predefined bit table in the main memory. This unit is prepared for realizing the rapid garbage collection.

3.3 Microprogram Instructions

All operations to the ALU, CCU, hardware stack,

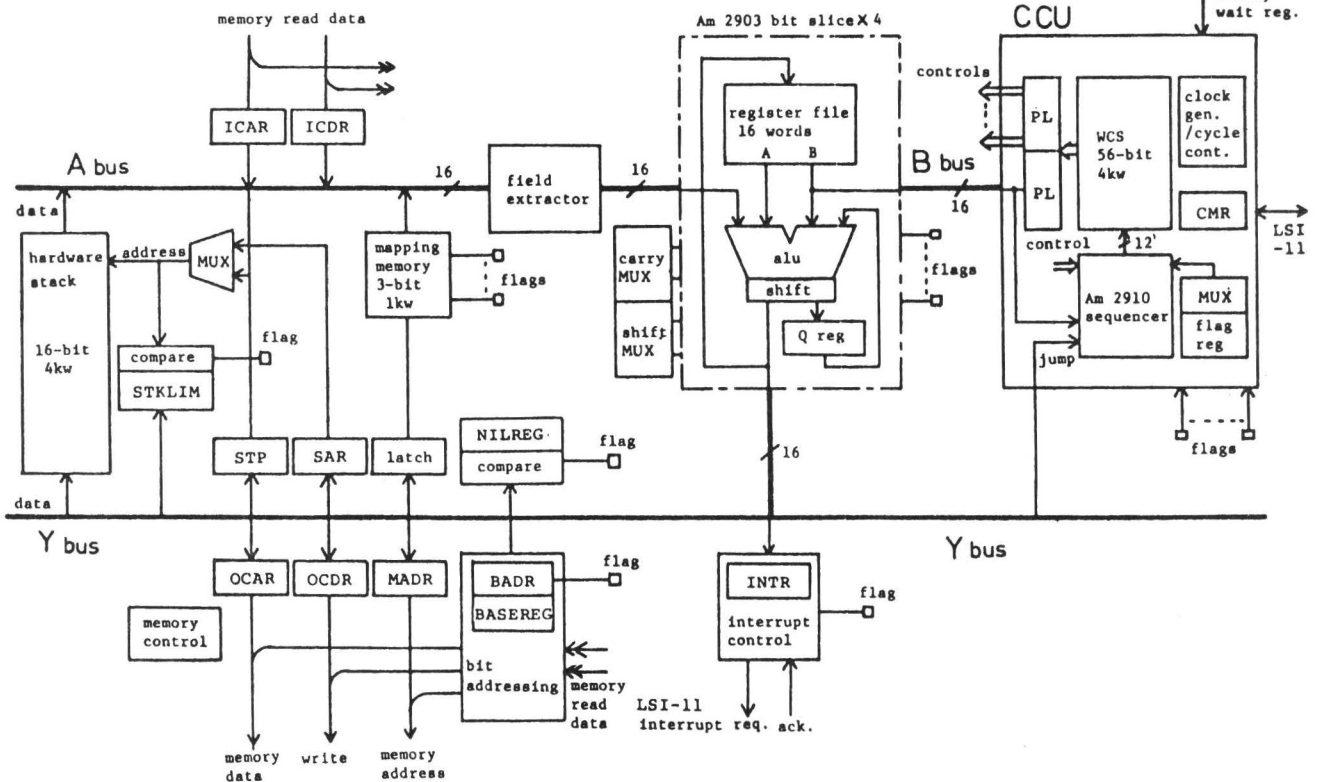


Fig. 2 LISP processor module.

mapping memory and other circuits are entirely under the control of microcode words of the WCS. The microinstruction is a 56-bit word divided into fields. Two types of microinstructions are provided. The type 1 instructions can provide a constant value to the ALU through the B bus, type 2 instructions are conditional branching instructions which can transfer controls to any microcode locations depend on condition flags.

4. DATA AND FUNCTION TYPES

There are four data types, small integer, integer, symbolic atom and list. Small integer, +8191 to -8192, doesn't occupy any memory words. Integers are represented in two's complement notation. Symbolic atom are stored as four consecutive words, each of which is divided into two 16-bit cells termed the function body cell, attribute cell, argument cell, top-value-cell, property list cell, print name cell and two unused cells.

Three types microcoded subroutine functions, MSUBR, MRSUBR and MRFSUBR are provided. MRSUBR and MRFSUBR are recursive functions and MSUBR is not.

MRSUBR: EVAL, APPLY, MAP, GET, EQUAL, etc.
MRFSUBR: COND, PROG, AND, DE, SETQ, etc.
MSUBR: CAR, CDR, CONS, EQ, NULL, ATOM, etc.

5. INTERPRETER

In our system the user program is reconstructed into a binary tree structure and interpretation of the program is made by traversing the tree in the endorder fashion.

The functional calling is the basic main control structure in LISP. As mentioned above, the LISP machine function calling is made fast through the use of the hardware stack and microcodes. The stack is formatted into frames; each frame contains a header, arguments, local variable value slots, and a push down list for intermediate results. The header gives the function which owns the frame, a pointer to the function body, links this frame to previous frames, and remembers the program counter and microprogram counter.

6. SYSTEM REALIZATION AND EVALUATIONS

The hardware of the LISP processor module and the main memory module consist of about 650 IC chips and 17 universal printed circuit boards(8x7 square

inches). The LISP interpreter of the machine is entirely microcoded and stored in the WCS. The program size of the first version is about 1200 microinstruction steps.

In order to evaluate our LISP machine, we executed the benchmark programs of the second LISP contest of IPSJ(Information Processing Society of Japan). Our interpreters execution times of these programs were shorter than those of any participants of the contest(almost all typical LISP systems in Japan such as HLISP installed HITAC-8800 and OLISP installed NEC system 800-2 were attended).

Examples of typical two benchmarks(e.g. TARAI and TPU) are shown in Table 1. TPU is a theorem prover based on the unit binary resolution principle written by C.L. Chang and TARAI is a very short program which does almost recursions only except the very simple functions GREATERP and SUB1.

These measurements showed that the architecture of our machine nicely fits in the control structure of the LISP language and realizes a very rapid execution of LISP programs.

ACKNOWLEDGMENTS

The authors wish to express their thanks to Messrs. Toshio Shimada and Yoshinori Yamaguchi of Electrotechnical Laboratory for their useful suggestions and discussions.

REFERENCES

[1] Taki, K., Kaneda, Y. and Maekawa, S. "The Experimental LISP Machine." Tec. Report of the Professional Group on Computer Architecture of IPSJ, 32-3, Sept., 1978 (in Japanese).

[2] Bawden, A. et al "LISP Machine Progress Report." MIT AI Lab. Memo., No. 444, Aug., 1977.

	-----INTERPRETER-----			-----COMPILER-----		
	FAST-LISP KOBE UNIV.	HLISP	OLISP	HLISP	OLISP	
TARAI-3	55	78	197	17	36	
TARAI-4	1,013	1,443	3,727	330	670	
TARAI-5	27,538	39,068	100,734	8,905	18,934	
TARAI-6	1,011,732	---	---	322,347	---	
TPU-1	904	4,790	2,262	1,029	658	
TPU-2	3,426	13,411	10,262	2,871	2,064	
TPU-3	1,365	5,684	3,932	1,227	850	
TPU-4	1,815	8,025	5,042	1,707	1,161	
TPU-5	238	866	759	181	132	
TPU-6	6,728	22,849	20,241	4,893	3,617	

Table 1 Execution times of the benchmark programs of the 2nd LISP contest of IPSJ (unit time is msec.)