A HIGH PERFORMANCE, MICROPROGRAMMED, NMOS-LSI PROCESSOR FOR 8- AND 16-BIT APPLICATIONS

Zvi Soha and William Pohlman

Western Digital Corp., Newport Beach, Ca.

The Western Digital Microprocessor Set (MPS) consists of three NMOS-LSI chips which are connected together by the Microinstruction Bus (MIB). The MIB is an 18-bit bi-directional bus designed to provide a communication path between the Data chip, the Control chip, and up to four Microprogram ROM (MICROM) chips. Communication between the MPS and other units in the system (memory, I/O controller) is executed via the Data Access port (see Figure 1). The primary features of the processor are:

- 8-bit internal organization
- 8/16-bit TTL compatible Data Access port
- DMA capability
- 26X8-bit register file
- Extensive microinstruction set with decimal operations included
- Maximum microprogram length of 2048 instructions
- Fast, efficient macroinstruction decode mapping arrays
- 300 nsec microcycle time; 22-bit microinstruction
- Single and double byte operations
- Microprogram subroutine capability
- Four external and three internal interrupts; programmable priorities

SYSTEM OPERATION:

During normal operation, a sequence of microinstruction addresses is generated by the Control chip. The addresses are transferred via the MIB to the Microm where the microinstructions are accessed and placed on the MIB. The microinstruction is read off the MIB by the Data and Control chips and stored in the Microinstruction Register MIR of both chips. At this point the microinstruction is ready for execution. The Data chip performs most of the execution and only the portion related to address sequence generation and Data Access port control is executed by the Control chip.

The microprocessor operates in a pipeline fashion in which the execution

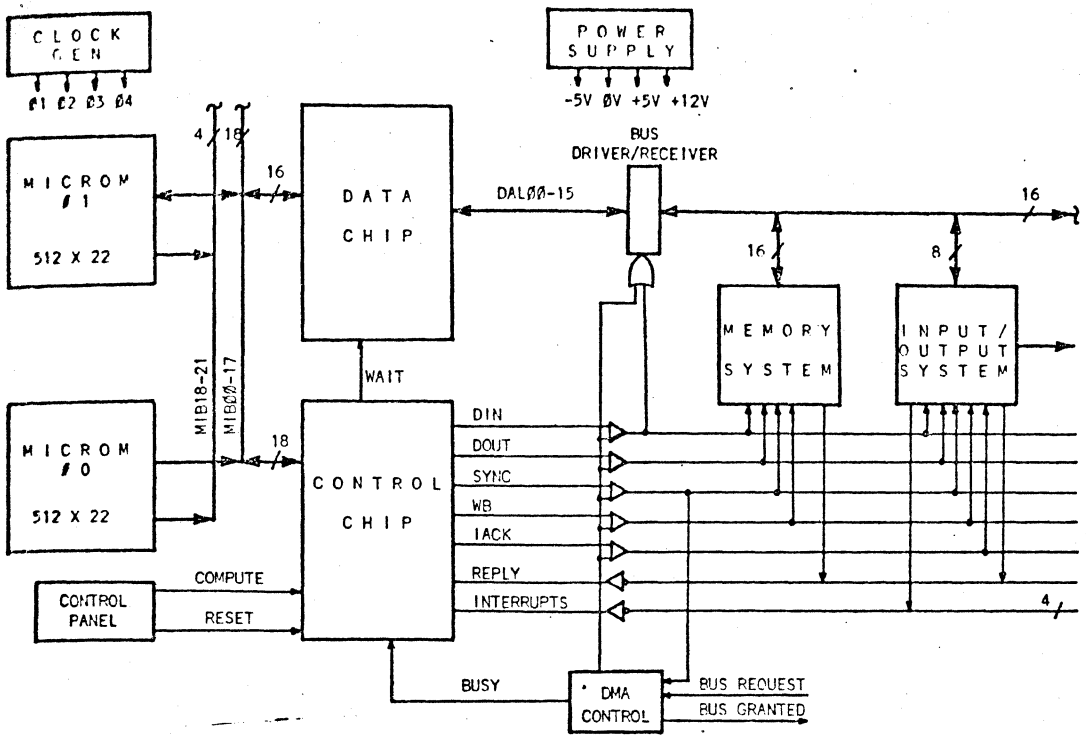
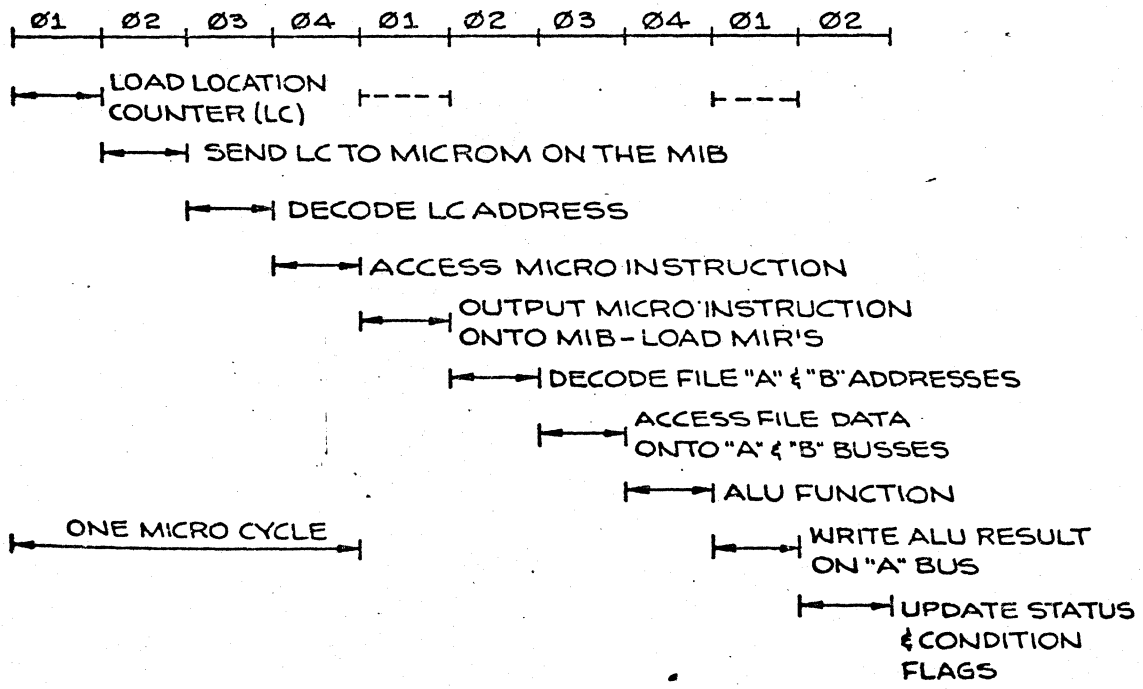


FIGURE 1 - MPS 1600 SYSTEM BLOCK DIAGRAM



MICRO INSTRUCTION PIPELINE TIMING

FIGURE NO. 2

of one microinstruction is overlapped with the fetching of the next microinstruction. Figure 2 shows the pipeline timing.

CONTROL CHIP:

The Control chip provides the microinstruction address sequence for the Microm and control for the Data Access port. It consists of a Programmable Translation Array (PTA), location Counter (LC), Return Register (RR), Data Access Control, and Interrupt flags (INT) as shown by Figure 3.

The 11-bit Location Counter is incremented by one after each access of the Microm, unless a new address is loaded into it by the MIR (Jump instruction), the RR (Return From Subroutine), or the PTA (Translation Branch).

The 11-bit Return Register is used to hold a subroutine return address. The RR stores the incremented contents of the LC when Bit 16 (LRR) of the microinstruction being accessed is a one. The contents of the RR is transferred to the LC under control of the PTA or Return From Subroutine instruction.

The Programmable Translation Array provides a decoding mechanism for generating microinstruction addresses. The generated Jump address is a function of the current LC contents, the target macroinstruction stored in the Translation Register (TR), the contents of the Translation State Register (TSR), the Read Next Instruction (RNI) control signal, and the stage of the Interrupt flags.

The RNI signal (MIB17) is activated at the end of every macroinstruction execution sequence. It is used to invoke a translation as a function of the state of the Interrupt flags and to generate a Jump address to an Interrupt service subroutine. The absence of an Interrupt will cause a Jump address to an RNI sequence which will in turn load the TR with the next macroinstruction to be emulated.

The 3-bit Translation State Register (TSR) is used to store past translation information and thus reducing the total translation terms in the PTA needed for a given macroinstruction set.

The PTA consists of two Programmable Logic Arrays (PLAs) each of which consists of a set of AND gates and a set of OR gates in a ROM-type layout. The implementation of the PTA in two AND-OR logic levels reduces appreciably the size of the PTA when compared with a straight-forward ROM (one AND-OR level).

The Control chip recognizes seven Interrupt signals three of which are internal to the chip and can be set by firmware. The other four Interrupts are used by external units in the system for service requests. The Interrupts can be programmed in the PTA to any desired priority. Any one of the internal Interrupts can be programmed to act as an Interrupt Enable flag.

The Data Access control section provides control and timing signals for the Data Access port. The minimum WRITE cycle time is 600 nsec and the minimum READ cycle time is 900 nsec for bytes and 1200 nsec for words.

A Wait signal is provided by the Control chip to delay microinstruction execution until certain conditions are met. Normally, the Wait signal is used during asynchronous Data Access operations.

DATA CHIP:

The Data chip incorporates the paths, registers, and logic to execute microinstructions. It consists of a Register File, 8-bit ALU, 8 Condition flags, and a Data/Address port.

The Register File contains 26 8-bit dual-port file registers for storage of frequently required data. In particular the file provides storage for the target machine registers and working storage for the microprogram.

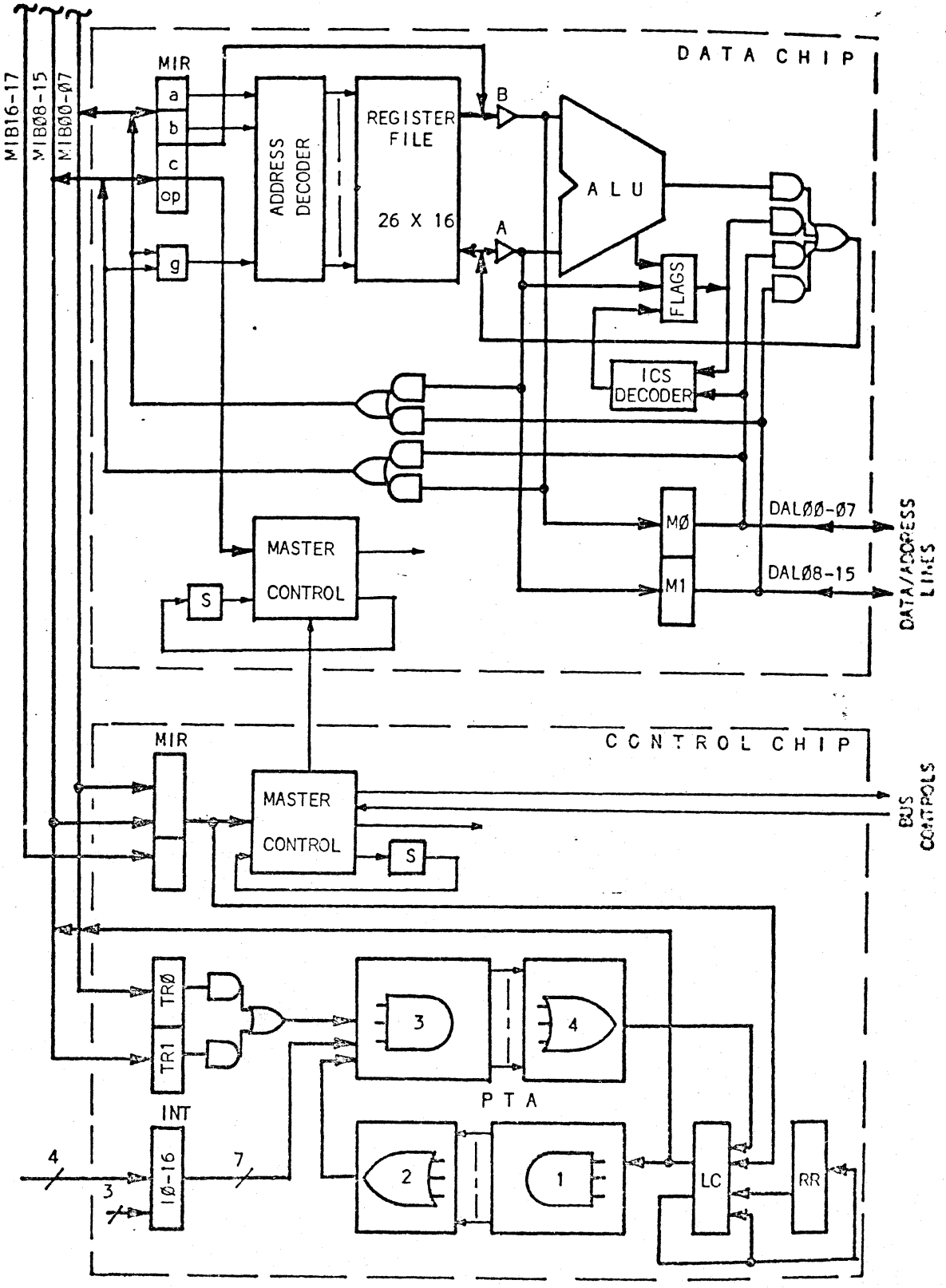


FIGURE 3 DATA AND CONTROL CHIPS

For the purposes of addressing, the file registers are divided into three sections. The first section consists of 10 file registers which are directly addressed by the "a" and "b" fields of the microinstruction. The second section consists of four file registers which are either directly addressed by the "a" and "b" fields or indirectly addressed via a 3-bit "G" register. The third section consists of twelve file registers which can only be addressed indirectly via the "G" register.

The G Register holds an indirect register designator and is used by the File Address Decoder whenever the "a" or "b" fields of the microinstruction contain a Ø or a 1. The G Register can be loaded directly from the register designator fields of the macroinstruction during instruction fetch time.

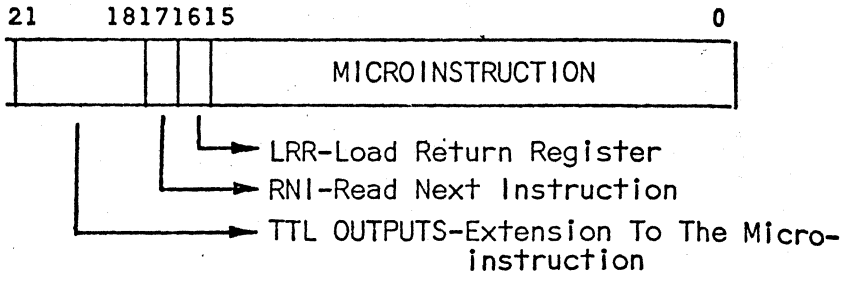
When executing two-cycle instructions, the low-order bit of the "a" and "b" fields of the microinstruction are complemented for the second cycle. This allows a single instruction to operate on a pair of file registers containing word operands. Normally this requires that both "a" and "b" fields be even during the first cycle. An odd "b" field during the first cycle of an arithmetic operation will cause the most significant byte of the B operand to be the extended sign of the least significant byte of B.

The ALU section performs the arithmetic and logic operations necessary for instruction execution. The two inputs to the ALU are read simultaneously from the file and the result is stored back into the file. The status of the result is monitored by the Condition flags.

MICROM CHIP:

The Microprogram ROM (MICROM) is a 512X22-bit word, high-speed ROM with maximum addressing capability of 2048 words.

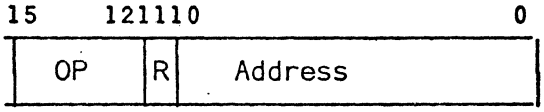
The 22-bit instruction format is as follows:



MICROINSTRUCTION FORMAT:

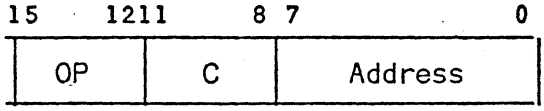
There are four basic microinstruction formats:

JUMP --



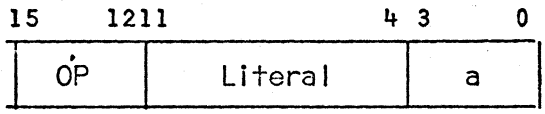
This format provides 11-bit Jump address or a Return From Subroutine control (R).

CONDITIONAL JUMP --



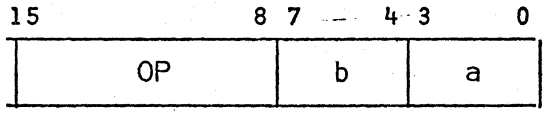
This format provides a Jump address within a 256-word page and 16 Jump conditions.

LITERAL FORMAT --



This format provides 8-bit literal data. The "a" field usually specifies a file register.

REGISTER FORMAT --



The "a" and "b" fields of this format specify file registers. Every operation can either by a byte or a word operation. A word operation is executed in two processor cycles in which case the "a" and "b" specify file register pairs. Bit 8 of the microinstruction causes four condition flags to be updated. The microinstruction set is sufficient to emulate macroinstruction sets in the order of magnitude of the PDP-11/40 efficiently.

PDP-11 EMULATION:

The MPS has been microprogrammed by Digital Equipment Corp., Maynard, Massachusetts, to emulate the PDP-11 minicomputer. The complete basic PDP-11 macroinstruction set has been programmed in two Microms. Fixed point multiply and divide and the floating point instructions have been programmed in a third Microm. The emulator runs on the average only ten percent slower than the PDP-11/05.

The PDP-11 macroinstruction set posed some interesting emulation problems. The machine has eight general purpose registers, two of which are dedicated to program counter and stack pointer functions. It has both single and double operand instructions with eight different addressing modes for each operand. Of course, there are many more areas of interest, but these two can be described briefly here.

In the PDP-11 macroinstruction formats the general purpose registers are specified by one or two three-bit fields for single or double operand operations, respectively. In the MPS system, the register file was designed to be addressable either directly via the "a" and "b" fields of the microinstruction or indirectly by the "G" register as described previously. The program counter and stack pointer registers were designed to be both directly addressable at the microprogram level and indirectly addressable with the "G" register since they need to be directly controlable at the microlevel but, however, can be treated as GP registers at the macrolevel. This technique permitted the number of registers in the dual-port file to be increased from 16 to 26 without increasing the width of the microinstruction which was limited by pinout considerations.

To optimize the speed of instruction decoding in a microprogrammed machine it is necessary to use fast PLA mapping techniques rather than

firmware decoding. Furthermore, it is desirable to do mapping functions in parallel with microinstruction fetching and execution so that useful operations can occur simultaneously with decoding operations wherever possible. The Programmable Translation Arrays (PTA) implemented on the Control chip permitted efficient mapping on the many addressing modes provided in the PDP-11 instruction format. The fundamental instruction decoding process using the mapper is shown in Figure No. 4. The mapper permits the microprogrammer to do multi-way conditional branches from any address to any other address. This mapping process is completed within one microcycle and does not require any interruption of microinstruction flow in the pipeline as would a Jump microinstruction. (When a Jump is executed the next instruction undergoing Fetch must be discarded.)

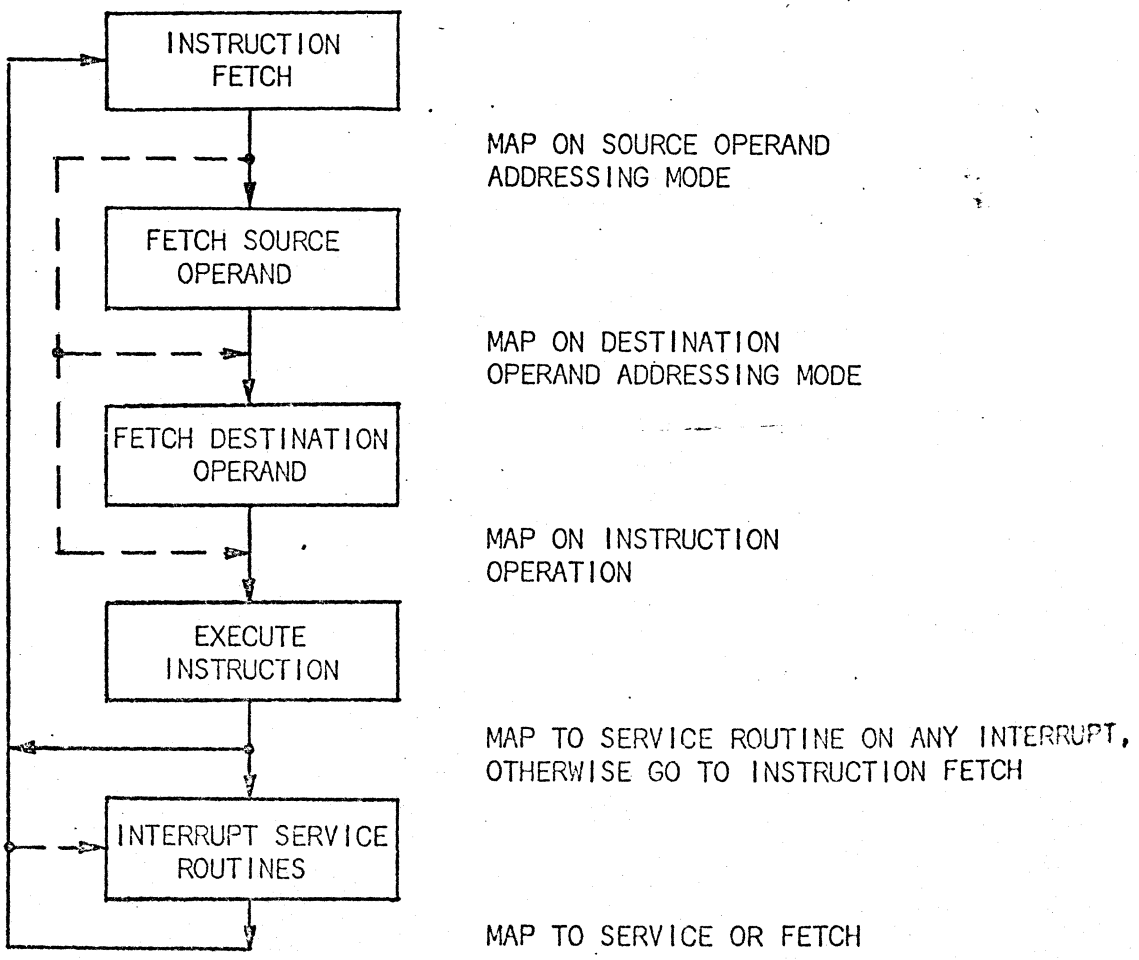


FIGURE NO. 4 BASIC PDP-11 INSTRUCTION DECODING PROCESS

CONCLUSION:

It is feasible to replace today's minicomputers with NMOS-LSI microprocessors and achieve comparable performance. But, in order to achieve performance in LSI emulation of target TTL machines special purpose hardware and mapping techniques are required to accommodate the idiosyncrasies of their macro languages. MOS-LSI microprocessors will permit the addition of features without large cost increases through microprogramming.