A Universal Dump Program for Minicomputer Software Debugging

C. C. Klimasauskas
Communications Systems Research Section

Low-cost minicomputers in wide variety are finding application in control and monitoring tasks ranging from laboratory testing to network operation. One significant problem which arises from this circumstance is that a significantly larger minicomputer system than is needed to perform the primary tasks must be acquired to do convenient software development. Consequently, work has been underway for some time to facilitate software development for the minimal configuration minicomputer using the Medium-Scale Xerox Data Systems Sigma 5. This article describes a general-purpose memory display program which runs on the Sigma 5 to dump memory images of minicomputer software to a printer or other man-readable device. The dump is formatted, as specified by control-card options, into machine-language instructions, character strings, or virtually whatever word/byte/field format is meaningful to the current problem.

I. Introduction

Low-cost minicomputers in wide variety are finding application in control and monitoring tasks ranging from laboratory testing to network operation. One significant problem which arises from this circumstance is that a significantly larger minicomputer system than is needed to perform the primary tasks must be acquired to do convenient software development. Consequently, work has been underway for some time to facilitate software development for the minimal configuration minicomputer using the Medium-Scale Xerox Data Systems Sigma 5. A general introduction to this activity may be found in Ref. 1. We expect ultimately to provide direct-coupled support where the minicomputer software is generated on the larger machine and is then loaded into the minicomputer and exercised via intercomputer communication links. One debugging tool which is useful, even if tedious to use, is a memory dump or display. This article describes a general-purpose memory display program which runs on the Sigma 5 to dump images of minicomputer software to a printer or other man-readable device. The dump is formatted, as specified by control-card options, into machine-language instructions, character strings, or virtually whatever word/byte/field format is meaningful to the current problem.

II. An Overview of the Processor

The universal dump program (UDP) looks at each record of the core dump (CD) as a single bit string. The user partitions this bit string into smaller units called words. Each word is partitioned further into fields (contiguous bit strings). Associated with each field is a field number and with that number a field conversion mode. Each word is indexed by an address which is specified dynamically by telling the UDP where to find the first address of the word or statically through specification. The address may be incremented by one or the number of bytes per word as specified through the addressing mode. The user has the option of indicating the conversion mode for the address, and the word, in addition to each field.

The first step in this process is description of the target machine. This phase generates tables which describe word size, byte size, field characteristics, conversion modes and all other data required to generate the dump. This phase produces syntax error messages and warnings. The following sections give a detailed description of the syntax of target machine description and listing format specification.

The next phase analyzes the tables for consistency and interpolates where possible to produce a complete set of consistent tables to drive the actual dump generator. This phase may find fatal specification errors which will prevent further execution. It also produces warning messages for nonfatal errors. A following section contains a complete description of all error messages.

Phase three is the actual routine for producing the dump listing. This program acts as a set of table-driven subroutines under the control of a single control routine. Figure 1 describes these program steps diagramatically. The program is implemented in Xerox Data Systems META-SYMBOL on the Sigma 5 computer to be run under the Batch Processing Monitor. On the right-hand side of Fig. 1 is the overlay structure of the program. Each of the files to produce this resides on magnetic tape in the standard Xerox Data Systems 7-track labeled-tape format. The source file for each overlay is followed by its corresponding binary file (in parentheses).

III. Conventions for Syntax Definition

This section presents some general comments which describe the meaning of the constructions used in defining syntax.

(1) The bracket construction:

[ ] indicates that those items contained within the brackets are optional

[...] indicates indefinite continuation of the optional construction in which this is imbedded.

(2) The brace construction:

{ option 1 / option 2 / . / . / . / option } indicates that exactly one of the options indicated *must* be chosen.

(3) Upper case literals and symbols:

These are key words and delimiters which *must* occur exactly as shown.

(4) Lower case phrases and letters:

These are representative of other constructions which must be supplied according to elsewhere defined syntax. For instance, the phrase "usgc" occurs many times in the description of the syntax. This phrase represents a numeric constant and has the syntax described at the beginning of Section IV.

(5) Blanks or spaces:

Unless specifically noted, blanks are *not* allowed between characters. Three exceptions:

(a) ␣ signifies that exactly one blank is mandatory.

(b) ^ signifies that one or more blanks are optional.

(c) The blank occurs as part of a string as defined for the syntax of the particular statement. For example, see "constant literal field" under PRTDEF in Section IV.

IV. The Syntax of Machine Description and Listing Format Specification

Before describing the syntax, some general comments about constants, control cards, and comment cards are presented.

There are two classes of constants: (1) explicit or specified and (2) implicit. The explicit or specified constant will be denoted by "ussc" and the implicit constant by "usgc." The syntax of each of these is given below:

usgc::

[blank]	decimal digit [decimal digit [...]]
D	decimal digit [decimal digit [...]]
O	octal digit [octal digit [...]]
X	hexadecimal digit [hexadecimal digit [...]]

$$\text{ussc::} \left\{ \begin{array}{l} \text{decimal digit [decimal digit [...]]} \\ \text{octal digit [octal digit [...]]} \\ \text{hexadecimal digit [hexadecimal digit [...]]} \end{array} \right\}$$

as specified by the syntax of the statement.

Control cards must always have an asterisk in column one. Blanks are not allowed in control cards except where specifically noted. If the definition of a control card requires more than one physical card, continuation cards must not have an asterisk in column one. All other cards will be printed exactly as punched and may be used for documentation purposes.

As we consider in detail the syntax of each statement, we will first define its purpose and make general comments, then give the actual syntax and describe the various options which are possible. As we proceed, it may be helpful to study the decks and resulting dumps shown in Figs. 2 and 3.

A. WORD Specification (Target Machine Description)

This statement simply describes the length in bits of each word of the target machine. Since these are partitions of a continuous bit string, they are referred to as p-words or "partion words." (When each p-word is printed, first the address of the word is printed, then the contents, then the alphanumeric representation of the contents, and finally the user-defined fields). The syntax is as follows:

$$\text{*WORD} = \text{usgc}$$

where usgc is the number of bits per target machine word.

B. BYTE Specification (Target Machine Description)

This card is optional and is used for the purpose of computing the target machine address of the current p-word. This card must be used if byte addressing is specified. (See ADDRESS specification.) The syntax is as follows:

$$\text{*BYTE} = \text{usgc}$$

where usgc is the number of bits per target machine byte.

C. ADDRESS Specification (Target Machine Descriptor and Listing Format Specification)

This card defines the addressing mode of the target machine for the purpose of generating a pseudoaddress for the current p-word on the dump listing. Two modes are available: Byte and Word (default) addressing. Byte addressing increments the listing location counter by the number of bytes per word for each successive instruction. The conversion mode for the address can be optionally specified. The syntax is as follows:

$$\text{*ADDRESS} = \left\{ \begin{array}{l} \text{BYTE} \\ \text{WORD} \end{array} \right\} \left[, \left\{ \begin{array}{l} \text{HEX} \\ \text{DEC} \\ \text{OCT} \end{array} \right\} \right]$$

D. CHAR and SPECIAL Specifications (Target Machine Descriptors)

The CHAR card defines the actual width of a character and the number of right-justified significant bits needed to decode the character. For instance, ASCII characters are eight bits wide; however, the right seven bits actually contain all the character information. Three widely used character codes have been incorporated into the program. Provision is made for user-defined character conversion codes. The syntax of the CHAR specification is as follows:

$$\text{*CHAR} = \left\{ \begin{array}{c} \text{ANSCII} \\ \text{ASCII} \\ \text{EBCDIC} \\ \text{BCD} \\ \text{SPCL} \end{array} \right\} [(\text{usgc}_1,\text{usgc}_2)]$$

where

$\text{usgc}_1 \equiv$ the width of a character in bits.

$\text{usgc}_2 \equiv$ the number of right-justified significant bits per character.

If SPCL is specified, the actual binary-to-character conversion table must be specified. This is done through the SPECIAL card. The syntax is as follows:

$$\text{*SPECIAL} = \text{usgc}$$
$$\text{␣ character string}_1$$
$$\text{␣ character string}_2$$
$$\vdots$$
$$\text{␣ character string}_4$$

where

usgc ≡ the number of characters per character string. Because of the physical limitations of a card, this is at most 79.

␣ ≡ a required space. The first column must be blank.

character string$_i$ ≡ the ith contiguous string of characters in the binary-to-alphanumeric conversion vector. The jth character[1] in the character conversion vector should be (a) a period, if no valid extended alphanumeric character corresponds to the target machine's internal binary representation of j or if the character is not printable, or (b) the actual character if the target machine's internal binary representation of j corresponds to that character. (For example: in EBCDIC, for $j = 253$, the internal binary representation of j is 1111 1101 and does not correspond to a valid character, so the 253rd character in the conversion vector is a period. For $j = 240$, the internal binary representation of j is 1111 0000. This corresponds to the character "Ø"; hence, the 240th character in the conversion vector is Ø.)

E. ROTATE Specification (Target Machine Preprocessing Descriptor)

In certain modes of operation, the halfwords may be swapped due to the method of transmission. This occurs, for instance, in the I/O processes of the PDP-11. It may also be desirable to rotate the contents of each word before breaking it into fields and decoding it. The ROTATE specification causes each p-word to be rotated right the number of bits specified. This occurs before any other procession on a record is done. The syntax is as follows:

*ROTATE:usgc

where

usgc is the number of bits to right-rotate each p-word before processing.

F. INSDEF Specification Card (Listing Format Specification)

INSDEF partitions the word into contiguous fields of bits. Each field has a corresponding field number which associates it with a print field (see PRTDEF). The syntax is as follows:

*INSDEF: ^ $f_1f_2f_3 \ldots f_r$

where

f_i is the ith contiguous string of bits corresponding to a print field; f_i itself is a contiguous string of digits (all the same) in the form dddd ... d, where d takes on values 0–9. When $d = 0$, the field is ignored. When $d \neq 0$, the field corresponds to the dth print field and will be connected for printing according to the attributes of the dth print field. If $\ell(f_i)$ represents the number of bits in field

$$i, \sum_{i=1}^{r} \ell(f_i)$$

must equal the number of bits per p-word. Hence $f_1f_2 \ldots f_r$ partitions the p-word into bit fields.

^ is an optional string of blanks.

G. PRTDEF Specification Card (Listing Format Specification)

PRTDEF specifies the type of conversion to perform when printing the bit fields of a p-word. It allows for inclusion of unchanging literals, conversion mode specifies, and zero width field delimiters. The syntax is as follows:

*PRTDEF: $f_1f_2 \ldots f_r$

where

f_i is a field of one of three types:

(1) An unchanging literal field. This field is always printed exactly as it appears on the PRTDEF card, with the exceptions noted under the SKIP and NULL attributes described later. This field may *not* contain any of the following characters: M,X, O,D,B,C,L,:.

(2) A conversion mode field. The jth conversion mode field is the dth print field ($j = d$) described under INSDEF. This field describes the mode of conversion (X-hexadecimal, O-octal, D-decimal, B-binary, C-character, M-mnemonic, L-literal) and the number of print positions allocated to printing the field. (Note: the rightmost characters of the field are always printed. Truncation always occurs to the left.) The general format of the field is $S_1S_2 \ldots S_n$ where $S_1 = S_2 = \ldots = S_n = S$, and S

[1]The index j starts at zero and goes through 2** (number of significant bits/character) −1 in steps of one.

defines the conversion mode as above; n is the width of the field in characters. For all but literal and mnemonic conversion, the maximum width is 12. For literals and mnemonics, the maximum is 4. There may be only *one* mnemonic field.

(3) A zero-width field delimiter. This is simply the character ':'. It serves to delimit two fields with the same conversion mode.

H. HEAD Specification Card (Listing Format Specification)

This card allows the user to put a heading at the top of each column of each page of listing. The header appears above the user-defined print fields. The first character after the colon appears immediately above the first character following the colon on the PRTDEF card (see PRTDEF). The syntax is as follows:

*HEAD: literal string

where 'literal string' is the user-defined header.

I. PARAM Specification Card (Listing Format Specification)

The PARAM card allows the user to modify the conversion mode of a print field when it is undesirable to print it. This card also is used to enter the literal table if the literal conversion mode was specified on the PRTDEF card. The syntax is as follows:

$$\text{*PARAM: usgc}_1\left[=\left(\left\{\begin{matrix}\text{SKIP}\\ \text{NULL}\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{SKIP}\\ \text{NULL}\end{matrix}\right\}\left[\left\{\begin{matrix}>\\ <\\ =\\ \neg\end{matrix}\right\}\left[\left\{\begin{matrix}>\\ <\\ =\\ \neg\end{matrix}\right\}\right]\text{usgc}_2\right]\right)\right]$$

$$[,\text{LIT}=\text{lit}_0,\text{lit}_1,\text{lit}_3,\ldots,\text{lit}_n]$$

where:

usgc_1 is the print field to be modified.

SKIP indicates that, following the current conversion mode field, blanks should be inserted to the end of the line and no further conversions made. This will occur if the SKIP-NULL condition is satisfied (see below); i.e., if the SKIP-NULL condition is satisfied, skip to the end of the line.

NULL indicates that the current conversion mode field and the preceding unchanging literal field should be blanked out. This will occur if the SKIP-NULL condition is satisfied.

SKIP-NULL condition: The relation operators in conjunction with usgc_2 form the SKIP-NULL condition. There may be a linear combination of at most two relational operators. If the relation thus defined between the contents of the current field and usgc_2 holds, the SKIP-NULL condition is '=0'. For the mnemonic conversion mode, the SKIP-NULL condition is implicitly satisfied if the operation code has not been assigned a mnemonic. In this case, an implicit NULL is performed, regardless of specification.

$\text{lit}_0,\text{lit}_1,\ldots,\text{lit}_n$ is the literal conversion vector for the current field if the literal conversion mode is specified. There are $n = 2^r - 1$ literals in this vector, where r is the bit width of the print field. Commas are used to delimit the literals. The ith literal corresponds to the internal binary representation of i (i takes on values 0 through $2^r - 1$).

J. OPS Specification (Listing Format Specification)

This card makes a one-to-one correspondence between machine operation codes and mnemonics. It builds the table for the mnemonic conversion field specified on the PRTDEF card. The syntax is as follows:

$$\text{*OPS} = \left\{\begin{matrix}\text{HEX}\\ \text{DEC}\\ \text{OCT}\end{matrix}\right\}$$

ussc_1, literal string_1

ussc_2, literal string_2

.

.

.

ussc_r, literal string_r

ussc_n, literal string_n

where

$ussc_i$ is the opcode for some instruction. $ussc_i$ is decoded as if it were in the mode specified by the 'OPS='card.

literal $string_i$ is the mnemonic corresponding to opcode $ussc_i$.

This card must follow the PRTDEF card.

K. BIN Specification (Listing Format Specification)

This card defines the mode of conversion for printing the contents of the entire p-word. (When each p-word is printed, first the address of the word, then its contents, in the mode specified by the BIN card, then the alphanumeric equivalent of the contents, and finally the user-defined fields are printed.) The syntax is as follows:

$$*\text{BIN} = \begin{Bmatrix}\text{HEX}\\ \text{DEC}\\ \text{OCT}\end{Bmatrix}$$

L. START and NEWLOC Specification (Listing Format Specification)

The purpose of START is to establish the initial value of the location counter for the address of the first p-word of the first record. If no NEWLOC card is present, the location counter is incremented for following records as if they were contiguous with the first. (This assumes that the core dump consists of more than one physical record.) This value may be explicitly stated or the program may be told where to find it in the first record. The syntax to do this is as follows:

$$*\text{START} = \begin{Bmatrix}\text{usgc}_1\\ \text{REC}[+\text{usgc}_2]:f_1f_2f_3[\{\pm\}\text{usgc}_3]\end{Bmatrix}$$

where

$usgc_1$ is the initial value of the location counter for listing purposes. REC indicates that the initial location will be found within the record. $usgc_2$ indicates the offset of the word in the record which contains the initial value of the location counter. (The word accessed is $usgc_2+1$.) $f_1f_2f_3$ forms a mask for extracting the location from the word. f_1 and f_3 are strings of zeros indicating bits to ignore. Either may be null. f_2 is the mask for extracting the bits which represent the initial location. It consists of a string of Ns. The total length of the three strings must be the number of bits per p-word. $usgc_3$ is an offset to be added to the address extracted, before using it as the initial value of the listing location counter.

NEWLOC performs exactly the same function as START and has the same syntax. However, NEWLOC applies to the second and each succeeding record. If the record mode is specified, it is assumed that each record is treated exactly the same as if it were the first record and START where specified with these parameters. If an absolute starting location is given under NEWLOC, this applies to the second record only, and each succeeding record is treated as if it were contiguous to the preceding record and the location counter is incremented appropriately. The syntax of NEWLOC is as follows:

$$*\text{NEWLOC} = \begin{Bmatrix}\text{usgc}_1\\ \text{REC}[+\text{usgc}_2]:f_1f_2f_3[\{\pm\}\text{usgc}_3]\end{Bmatrix}$$

where each symbol has the same meaning as for START.

Table 1 gives an incidence matrix which shows those cards that are necessary and any cards which they must precede. Those items in the left column must either precede, follow, or are optional with respect to the items across the top. This table should be helpful to the user for setting up a deck. If a card is optional, except in the presence of another, this is also noted. For instance, BYTE must precede the ADDRESS specification if used.

M. END Specification

The END card delimits the description deck and calls in the consistency analyzer. This must be the last card in the description deck. The syntax is as follows:

*END

This concludes the discussion of the syntax of target machine description and listing format specification. Actual examples of each of the statements are shown in Figs. 2 and 3.

V. Program Limitations and Diagnostics

The Universal Dump Program is aimed at minicomputers, and this is reflected in some design limitations. Even so, it is very flexible, and should provide no insurmountable limitations to the user. Table 2 describes its characteristics and limitations.

The initial phases of the program produce a considerable quantity of diagnostics designed specifically to aid the user in quickly identifying and correcting control-deck errors. Because of this, and the philosophy that a program should be executed if at all possible, most diagnostics do not prevent continued execution of the program. In the initial syntax analysis section, diagnostics do not prevent continued execution, although a FAILURE will terminate processing of that statement. All consistency checks will be made on the tables, even though fatal errors in consistency may be discovered. Fatal errors prevent execution of the listing phase and termination will occur. Nonfatal errors (WARNINGS) have no effect on continued execution and are merely to call the user's attention to unusual conditions. Table 3 lists possible errors, their probable cause, and fatality.

VI. Two Examples of Use

In this section we consider two examples to illustrate use of the program. The first decodes a dump of part of the monitor program for the XDS 930 computer, which is a second-generation machine that uses word addressing. The word is 24 bits, character code BCD, and usual listing mode OCTAL. The first record is a bootstrap which loads into location 2. Each following record is in absolute loadable binary form, with load location information in the second word. The instruction format is shown in Fig. 4. For a further description of the 930, see the *XDS 930 Computer Reference Manual.* Figure 2 illustrates how this information was used to generate a machine description deck. Figure 2 also shows the resulting dump.

The second example is from a considerably different machine, the PDP-11, manufactured by Digital Equipment Corporation. The PDP-11 is a medium-speed, third-generation minicomputer. It has 16-bit words, which are byte-addressable, and uses the ASCII character set. The usual listing mode is octal. The example here is a dump of an I/O routine in the standard PDP-11 absolute dump format. This format swaps the low- and high-order bytes in each word, making it necessary to reverse them before analyzing a record. This was done by means of the ROTATE command. Address information is contained in the third word of each record. The PDP-11 uses the first four bits of each word as an op-code, and for the op-codes '00' and '10', other bits in the word are used as modifiers. This condition is indicated by using the mnemonics SPCL and SPLB, respectively. The PDP-11 instruction format is shown in Fig. 5. For a further description of the PDP-11 instructions, see the *PDP-11/20 Processor Handbook* (Digital Equipment Corp.). Figure 3 shows the actual specification deck and dump listing for the I/O routine.

Reference

1. Layland, J. W., "An Introduction to Minicomputer Software Support," Klimasauskas, C. C., "The X930 Program Set for Sigma 5 Assembly," Erickson, D. E., The SAPDP Program Set for Sigma 5 Assembly," *The Deep Space Network Progress Report*, Technical Report 32-1526, Vol. VII, pp. 84–96, Jet Propulsion Laboratory, Pasadena, Calif., Feb. 15, 1972.

Table 1. The order of heirarchy of specification cards

O—may optionally precede or follow
P—must precede the statement on the top of the matrix
F—must follow the statement on the top of the matrix
blank—optional card

	WORD	BYTE	ADDRESS	CHAR	SPECIAL	ROTATE	INSDEF	PRTDEF	HEAD	PARAM	OPS	BIN	START	NEWLOC
WORD	–	P	P	P	P	P	P	P	O	P	P	O	P	P
BYTE		–	P[a]											
ADDRESS			–											
CHAR	O	O	O	–	O	O	O	O	O	O	O	O	O	O
SPECIAL				F[b]	–									
ROTATE						–								
INSDEF	F	O	O	O	O	O	–	O	O	P	P	O	O	O
PRTDEF	F	O	O	O	O	O	O	–	O	P	P	O	O	O
HEAD									–					
PARAM	F	O	O	O	O	O	F	F	O	–	O	O	O	O
OPS	F	O	O	O	O	O	F	F	O	O	–	O	O	O
BIN												–		
START	F	O	O	O	O	O	O	O	O	O	O	O	–	O
NEWLOC	F													–

[a]BYTE must be specified before address if byte addressing mode is used.

[b]SPECIAL must be used if and only if SPCL is specified for character type.

Table 2. Characteristics and limitations on the Universal Dump Program

Characteristic	Minimum	Maximum
p-word length (bits)	2	32
Binary, octal, decimal, hexadecimal, character field width (print positions)	1	12
Literal and mnemonic field width (print positions)	1	4
Location counter characters (print positions)	5	5
Buffer size (bits) (current implementation)	2×10^5	5×10^5

Table 3. Error messages

Syntax errors (syntax analysis phase: nonfatal)

Message	Meaning
FAILURE: UNEXPECTED END OF LINE ENCOUNTERED	Syntax scan expected to see a character, but came to end of card first
WARNING: CARD WAS NOT COMPLETELY PROCESSED	Syntax scan ended and released card, yet nonblank characters still remain
WARNING: I/O ERROR-ABORT**	An irrecoverable I/O error has occurred and program has aborted (fatal)
WARNING: END CARD SUPPLIED—NO EXTRA CHARGE	End-of-file on card input was encountered before an *END card
FAILURE: ILLEGAL SIZE: MUST BE (1-32)	Self-explanatory
FAILURE: CANNOT REDEFINE WORD SIZE—ERROR	Attempt to redefine size of a p-word—ignored
FAILURE: ILLEGAL BYTE LENGTH	Length of byte is zero, greater than the length of a p-word, or p-word length has not been defined
FAILURE: CANNOT REDEFINE BYTE LENGTH	Attempt to redefine byte length has been made
WARNING: WORD IS NOT BYTE MULTIPLE	The p-word is not an even byte multiple
FAILURE: BYTE LENGTH: UNDEFINED—IGNORED	The byte length must be defined before the ADDRESS card is used
FAILURE: ILLEGAL PARAMS: WORD OR BYTE EXPECTED	Self-explanatory diagnostic of ADDRESS card
FAILURE: ROTATE OPTION ALREADY IN EFFECT	Self-explanatory
FAILURE: ROTATE BITS > = WORD LENGTH; OR WORD LENGTH = 0	Self-explanatory
FAILURE: ATTEMPT TO REDEFINE START/NEWLOC	Self-explanatory
FAILURE: ILLEGAL FIELD FMT (:) EXPECTED	Self-explanatory
WARNING: MASK AND WORD ARE UNEQUAL MASK ASSUMED RT JUSTIFIED	Self-explanatory in context of START/NEWLOC card
FAILURE: ATTEMPT TO REDEFINE INSTRUCTION	One or more INFDEF cards have already been processed
FAILURE: MUST HAVE WORD LENGTH BEFORE INSDEF—YOU LOSE	WORD card must precede INSDEF statement
WARNING: DEFINED INS LENGTH AND ACTUAL INS LENGTH DIFFER	Self-explanatory in context of INSDEF statement
WARNING: ILLEGAL FIELD DEF CHARACTER —SCAN ENDED	Self-explanatory
FAILURE: ONLY ONE PRTDEF CARD ALLOWED —THIS IGNORED	One or more PRTDEF cards have been processed
FAILURE: UNKNOWN CHAR TYPE: ANSCII, ASCII, BCD, EBCDIC, SPCL, EXPECTED	Self-explanatory in context of CHAR card
FAILURE: CANNOT REASSIGN CHARACTER TYPE	One or more CHAR cards have been processed
FAILURE: CHARACTERS ARE LONGER THAN ONE WORD	Self-explanatory
FAILURE: WORD IS NOT AN EVEN CHARACTER MULTIPLE	Self-explanatory
FAILURE: ILLEGAL CHARACTER, COMMA EXPECTED	Self-explanatory
FAILURE: MASK IS LARGER THAN CHARACTER	Self-explanatory
FAILURE: ILLEGAL CHAR, RT PAREN EXPECTED	Self-explanatory
FAILURE: NO SPECIAL CHARACTER TABLE NEEDED	SPCL was not an option on the CHAR card
FAILURE: MAX OF 79 CHAR/CARD EXCEEDED	Self-explanatory
FAILURE: NO CHAR NEEDED—WIDTH IS ZERO	SPCL card has occurred before CHAR card
FAILURE: UNEXPECTED END OF CARD	A control card was encountered before the character conversion vector was completed

Table 3 (contd)

Syntax errors (syntax analysis phase: nonfatal)

Message	Meaning	Message	Meaning
FAILURE: ILLEGAL TYPE: HEX,DEC,OCT EXPECTED	Self-explanatory in context of BIN card	FAILURE: NO MNEMONIC FIELD FOR INSTRUCTION.	PRTDEF card did not have a mnemonic field or has not been encountered yet
FAILURE: UNIDENTIFIED PARAMETER JUST ENCOUNTERED	SKIP/NULL expected while scanning PARAM card—not found	WARNING: OP OUT OF RANGE—IGNORED	An op-code greater than maximum found while processing OPS card
WARNING: MISSING PRNTDEF FIELD—WIDTH OF 4 ASSUMED	PARAM card precedes the PRTDEF card	FAILURE: DYNAMIC STORAGE ALL ALLOCATED—NONE LEFT	Insufficient core available to complete processing of description deck
WARNING: INCORRECT NUMBER OF LITERALS	The number of literals on the PARAM card is less than needed to complete the binary-to-literal conversion vector	WARNING: THIS IS THE END	Normal message indicating completion of syntax phase

Syntax errors (table analysis phase: FAILURES are fatal)

Message	Meaning	Message	Meaning
WARNING: ZERO BYTES/WRD—ONE ASSUMED	Self-explanatory	FAILURE: ZERO FIELD WIDTH—FIELD:XX	PRTDEF field, but no corresponding field on INSDEF card (i.e., too many PRTDEF fields)
FAILURE: ZERO WIDTH CHARACTERS FAILURE: IMPROPER CHARACTER FIELD WIDTH	An irrecoverable error occurred while processing the CHAR card, or field width specifications are in error	FAILURE: NO CONVERSION TYPE—FIELD:XX	Too many INSDEF fields and no corresponding PRTDEF field
WARNING: NO CHR OPT—EBCDIC ASSUMED	No CHAR card in description deck	WARNING: LIT ENTRY, BUT CNVRSN TYPE NOT LIT—FIELD:XX	LIT option specified on PARAM card, but conversion type from PRTDEF was not literal mode
FAILURE: SPECIAL CHARACTER TABLE IS NULL	SPCL was specified on the CHAR card, but no SPECIAL card was validly processed		
WARNING: TOTAL PRINT FIELD > 33 CHR—TRUNCATED	The number of characters required to print the location, contents, and user fields exceeds 33 characters	FAILURE: PARAMETER ENTRY AFTER END OF FIELD IN PSN:XX	Field specified on a PARAM card is greater than any PRTDEF field
FAILURE: NO PRTDEF CARD PRESENT	Self-explanatory	WARNING: EXTRACT MSK, BUT NULL FIELD—FIELD:XX	INSDEF field is not related to any PRTDEF field
FAILURE: NO INSDEF CARD PRESENT	Self-explanatory		
FAILURE: LIT/MNE, BUT NO LIT ENTRY—FIELD: XX	L or M specification on PRTDEF, but no 'LIT' on PARAM card or OPS card respectively	WARNING: LITS, BUT NULL FIELD—FIELD:XX	PARAM card for undefined PRTDEF field has LIT option
		WARNING: PARAMS, BUT NULL FIELD: FIELD:XX	PARAM card for undefined field

XX is the field number (1–9).

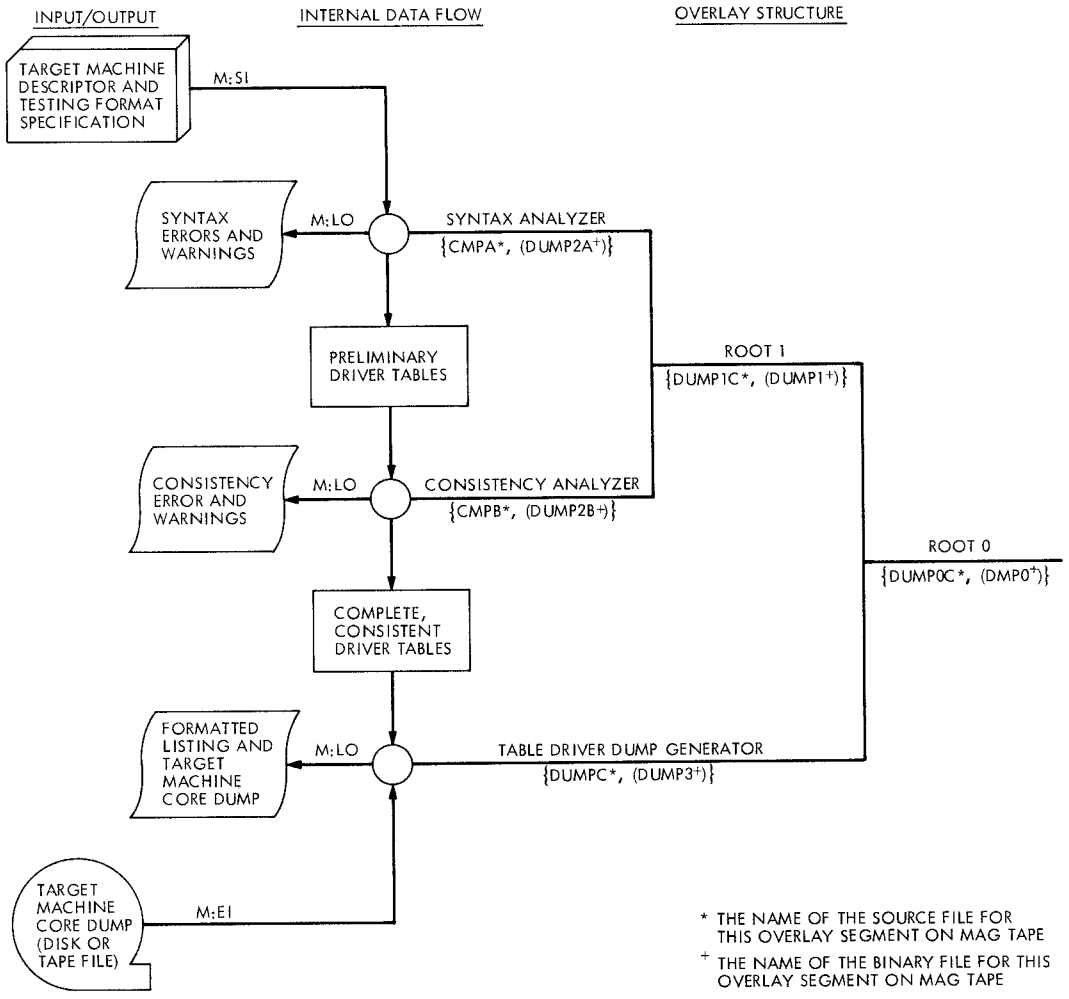
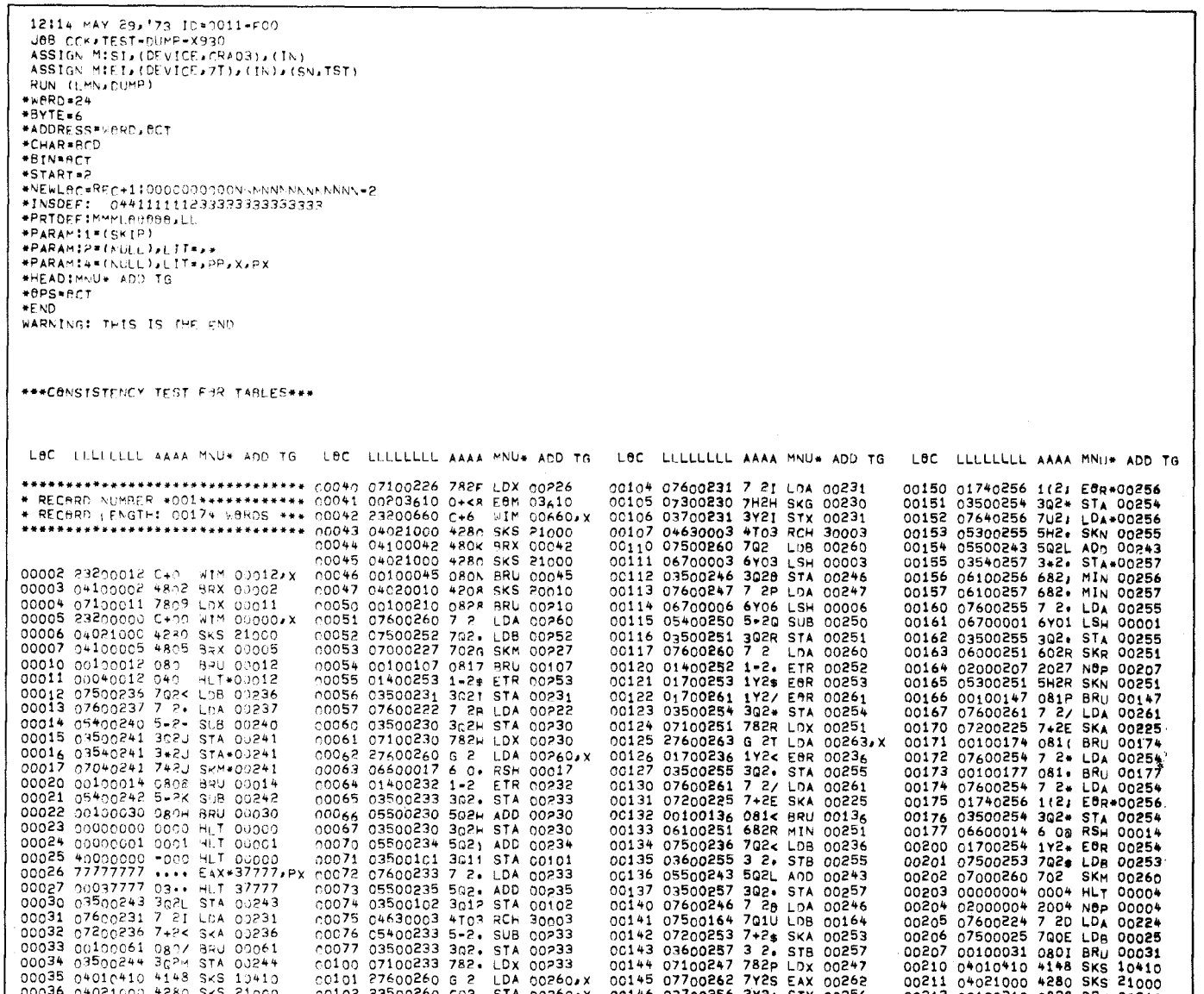
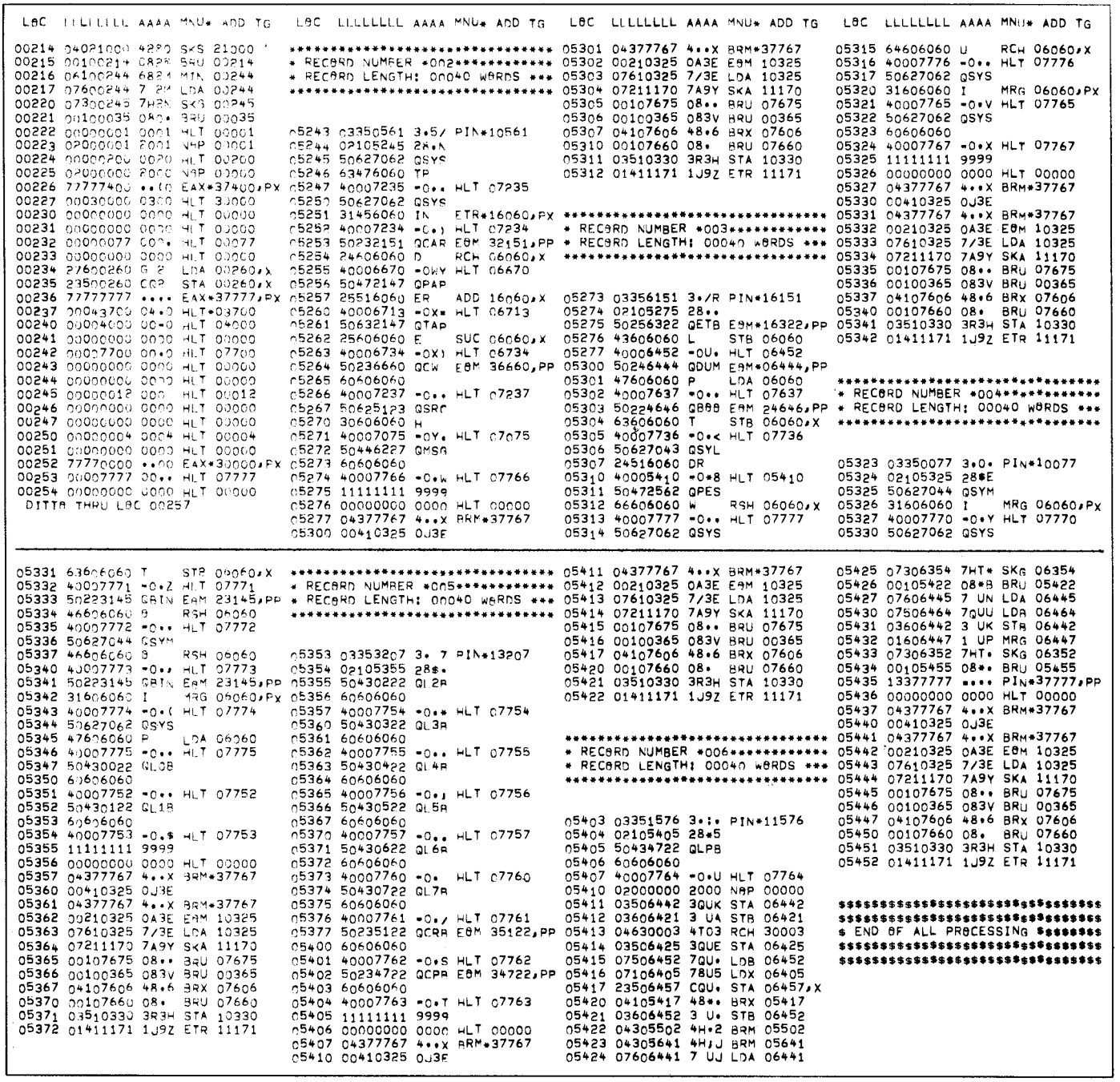


Fig. 1. Dump Program diagram



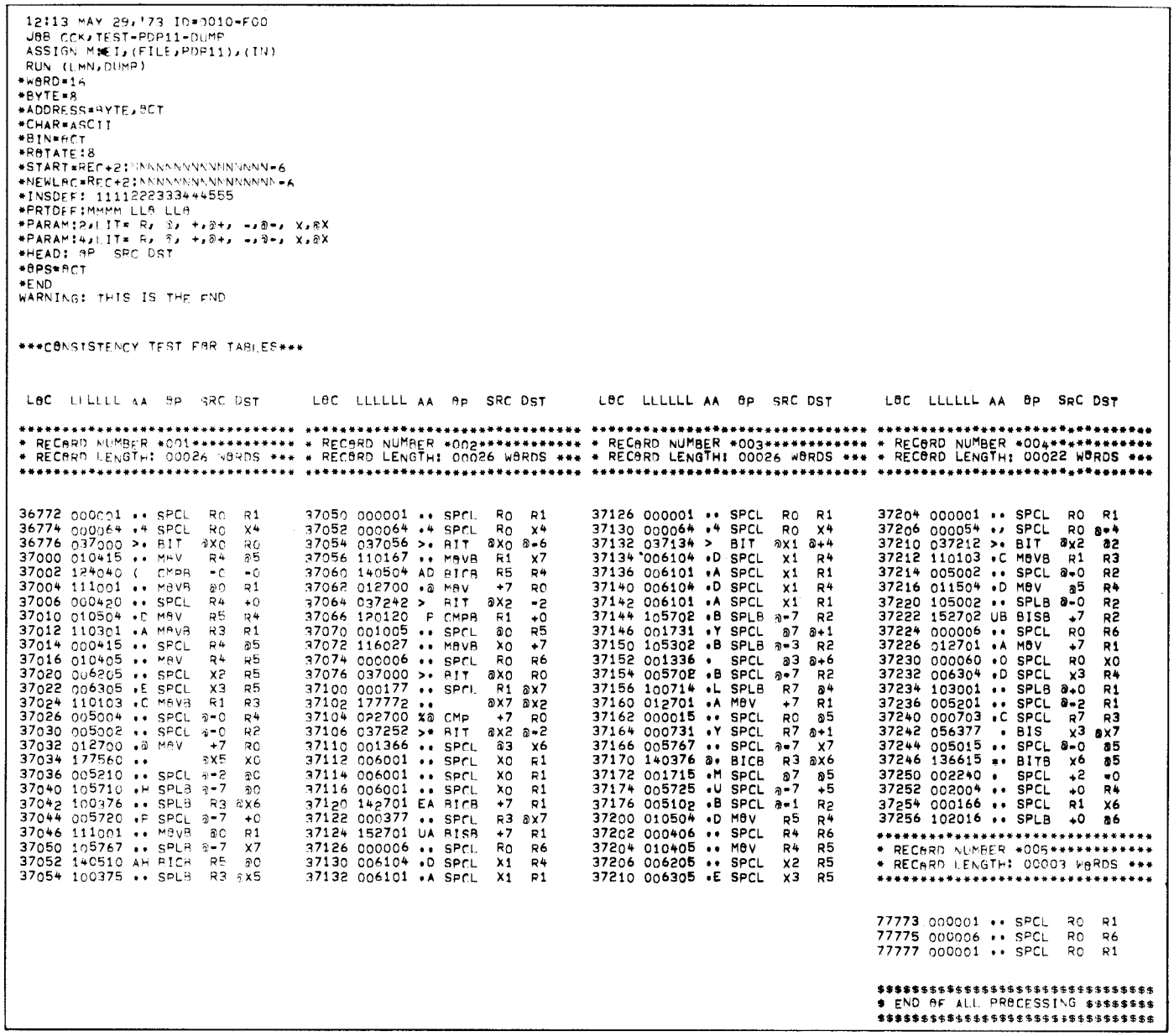
```
 12:14 MAY 29,'73 ID=0011-F00
 JOB CCK,TEST=DUMP=X930
 ASSIGN M:SI,(DEVICE,CRA03),(IN)
 ASSIGN M:EI,(DEVICE,7T),(IN),(SN,TST)
 RUN (LMN,DUMP)
*WORD=24
*BYTE=6
*ADDRESS=WORD,OCT
*CHAR=BCD
*BIN=OCT
*START=2
*NEWLOC=REC+1;000000000NNNNNNNNNNNN=2
*INSDEF:  04411111123333333333333
*PRTDEF:MMMLOWOB,LL
*PARAM:1=(SKIP)
*PARAM:P=(NULL),LIT=,,
*PARAM:4=(NULL),LIT=,PP,X,PX
*HEAD:MNU* ADD TG
*OPS=OCT
*END
WARNING: THIS IS THE END


***CONSISTENCY TEST FOR TABLES***


 LOC  LLLLLLLL AAAA MNU* ADD TG   LOC  LLLLLLLL AAAA MNU* ADD TG   LOC  LLLLLLLL AAAA MNU* ADD TG   LOC  LLLLLLLL AAAA MNU* ADD TG

****************************** 00040 07100226 782F LDX 00226    00104 07600231 7 2I LDA 00231    00150 01740256 1(2; EOR*00256
* RECORD NUMBER *001*********** 00041 00203610 0+<R EOM 03410    00105 07300230 7H2H SKG 00230    00151 03500254 3Q2* STA 00254
* RECORD LENGTH: 00174 WORDS *** 00042 23200660 C+6  WIM 00660,X 00106 03700231 3Y2I STX 00231    00152 07640256 7U2; LDA*00256
****************************** 00043 04021000 4280 SKS 21000    00107 04630003 4T03 RCH 30003    00153 05300255 5H2* SKN 00255
                               00044 04100042 4804 BRX 00042    00110 07500260 7Q2  LDB 00260    00154 05500243 5Q2L ADD 00243
                               00045 04021000 4280 SKS 21000    00111 06700003 6Y03 LSH 00003    00155 03540257 3*2* STA*00257
00002 23200012 C+0  WIM 00012,X 00046 00100045 080N BRU 00045    00112 03500246 3Q2B STA 00246    00156 06100256 682; MIN 00256
00003 04100002 4802 BRX 00002    00047 04020010 4208 SKS 20010    00113 07600247 7 2P LDA 00247    00157 06100257 682* MIN 00257
00004 07100011 7809 LDX 00011    00050 00100210 0828 BRU 00210    00114 06700006 6Y06 LSH 00006    00160 07600255 7 2* LDA 00255
00005 23200000 C+00 WIM 00000,X  00051 07600260 7 2  LDA 00260    00115 05400250 5*2Q SUB 00250    00161 06700001 6Y01 LSH 00001
00006 04021000 4280 SKS 21000    00052 07500252 7Q2* LDB 00252    00116 03500251 3Q2R STA 00251    00162 03500255 3Q2* STA 00255
00007 04100005 4805 BRX 00005    00053 07000227 702G SKM 00227    00117 07600260 7 2  LDA 00260    00163 06000251 602R SKR 00251
00010 00100012 080  BRU 00012    00054 00100107 0817 BRU 00107    00120 01400252 1-2* ETR 00252    00164 02000207 2027 NOP 00207
00011 00040012 040  HLT*00012    00055 01400253 1-2$ ETR 00253    00121 01700253 1Y2$ EOR 00253    00165 05300251 5H2R SKN 00251
00012 07500236 7Q2< LDB 00236    00056 03500231 3Q21 STA 00231    00122 01700261 1Y2/ EOR 00261    00166 00100147 081P BRU 00147
00013 07600237 7 2* LDA 00237    00057 07600222 7 2B LDA 00222    00123 03500254 3Q2* STA 00254    00167 07600261 7 2/ LDA 00261
00014 05400240 5-2- SUB 00240    00060 03500230 3Q2H STA 00230    00124 07100251 782R LDX 00251    00170 07200225 7*2E SKA 00225
00015 03500241 3Q2J STA 00241    00061 07100230 782H LDX 00230    00125 27600263 G 2T LDA 00263,X  00171 00100174 081( BRU 00174
00016 03540241 3*2J STA*00241    00062 27600260 G 2  LDA 00260,X  00126 01700236 1Y2< EOR 00236  00172 07600254 7 2* LDA 00254
00017 07040241 742J SKM*00241    00063 06600017 6 0* RSH 00017    00127 03500255 3Q2* STA 00255    00173 00100177 081* BRU 00177
00020 00100014 0808 BRU 00014    00064 01400232 1-2  ETR 00232    00130 07600261 7 2/ LDA 00261    00174 07600254 7 2* LDA 00254
00021 05400242 5-2K SUB 00242    00065 03500233 3Q2. STA 00233    00131 07200225 7*2E SKA 00225    00175 01740256 1(2; EOR*00256
00022 00100030 080H BRU 00030    00066 05500230 5Q2H ADD 00230    00132 00100136 081< BRU 00136    00176 03500254 3Q2* STA 00254
00023 00000000 0000 HLT 00000    00067 03500230 3Q2H STA 00230    00133 06100251 682R MIN 00251    00177 06600014 6 0@ RSH 00014
00024 00000001 0001 HLT 00001    00070 05500234 5Q2) ADD 00234    00134 07500236 7Q2< LDB 00236    00200 01700254 1Y2* EOR 00254
00025 40000000 -000 HLT 00000    00071 03500101 3G11 STA 00101    00135 03600255 3 2* STB 00255    00201 07500253 7Q2$ LDB 00253
00026 77777777 .... EAX*37777,PX 00072 07600233 7 2. LDA 00233    00136 05500243 5Q2L ADD 00243    00202 07000260 702  SKM 00260
00027 00037777 03.. HLT 37777    00073 05500235 5Q2. ADD 00235    00137 03500257 3Q2* STA 00257    00203 00000004 0004 HLT 00004
00030 03500243 3Q2L STA 00243    00074 03500102 3Q12 STA 00102    00140 07600246 7 2B LDA 00246    00204 02000004 2004 NOP 00004
00031 07600231 7 2I LDA 00231    00075 04630003 4T03 RCH 30003    00141 07500164 7Q1U LDB 00164    00205 07600224 7 2D LDA 00224
00032 07200236 7+2< SKA 00236    00076 05400233 5-2. SUB 00233    00142 07200253 7+2$ SKA 00253    00206 07500025 7Q0E LDB 00025
00033 00100061 0807 BRU 00061    00077 03500233 3Q2. STA 00233    00143 03600257 3 2. STB 00257    00207 00100031 0801 BRU 00031
00034 03500244 3Q2M STA 00244    00100 07100233 782. LDX 00233    00144 07100247 782P LDX 00247    00210 04010410 4148 SKS 10410
00035 04010410 4148 SKS 10410    00101 27600260 G 2  LDA 00260,X  00145 07700262 7Y2S EAX 00262    00211 04021000 4280 SKS 21000
00036 04021000 4280 SKS 21000    00102 23500260 CQ2  STA 00260,X  00146 03700256 3Y2; STX 00256    00212 00100210 0828 BRU 00210
00037 00100035 080. BRU 00035    00103 04100101 4811 BRX 00101    00147 07600254 7 2* LDA 00254    00213 00207630 0**H EOM 07630
```

Fig. 2. An application of the Universal Dump Program to the XDS 930 computer



```
LOC  LLLLLLLL AAAA MNU* ADD TG   LOC  LLLLLLLL AAAA MNU* ADD TG   LOC  LLLLLLLL AAAA MNU* ADD TG   LOC  LLLLLLLL AAAA MNU* ADD TG

00214 04021000 4220 SKS 21000 '  ****************************** 05301 04377767 4••X BRM*37767    05315 64606060 U    RCH 06060,X
00215 00100214 0828 SRU 00214    * RECORD NUMBER *002*********** 05302 00210325 0A3E E0M 10325    05316 40007776 -0•• HLT 07776
00216 04100244 6824 MIN 00244    * RECORD LENGTH: 00040 WORDS *** 05303 07610325 7/3E LDA 10325    05317 50627062 QSYS
00217 07600244 7 2M LDA 00244    ****************************** 05304 07211170 7A9Y SKA 11170    05320 31606060 I    MRG 06060,PX
00220 07300245 7M2N SKG 00245                                     05305 00107675 08•• BRU 07675    05321 40007765 -0•V HLT 07765
00221 00100035 0820 BRU 00035                                     05306 00100365 083V BRU 00365    05322 50627062 QSYS
00222 00000001 0001 HLT 00001    05243 03350561 3•5/ PIN*10561    05307 04107606 48•6 BRX 07606    05323 60606060
00223 02000001 2001 NOP 00001    05244 02105245 28•A              05310 00107660 08•  BRU 07660    05324 40007767 -0•X HLT 07767
00224 00000200 0020 HLT 00200    05245 50627062 QSYS              05311 03510330 3R3H STA 10330    05325 11111111 9999
00225 02000000 2000 NOP 00000    05246 63476060 TP                05312 01411171 1J9Z ETR 11171    05326 00000000 0000 HLT 00000
00226 77777400 ••00 EAX*37400,PX 05247 40007235 -0•• HLT 07235                                     05327 04377767 4••X BRM*37767
00227 00030000 0300 HLT 30000    05250 50627062 QSYS                                               05330 00410325 0J3E
00230 00000000 0000 HLT 00000    05251 31456060 IN   ETR*16060,PX ****************************** 05331 04377767 4••X BRM*37767
00231 00000000 0000 HLT 00000    05252 40007234 -0•) HLT 07234    * RECORD NUMBER *003*********** 05332 00210325 0A3E E0M 10325
00232 00000077 007• HLT 00077    05253 50232151 QCAR E0M 32151,PP * RECORD LENGTH: 00040 WORDS *** 05333 07610325 7/3E LDA 10325
00233 00000000 0000 HLT 00000    05254 24606060 D    RCH 06060,X  ****************************** 05334 07211170 7A9Y SKA 11170
00234 27600260 G 2  LDA 00260,X  05255 40006670 -0WY HLT 06670                                     05335 00107675 08•• BRU 07675
00235 23500260 C02  STA 00260,X  05256 50472147 QPAP                                              05336 00100365 083V BRU 00365
00236 77777777 •••• EAX*37777,PX 05257 25516060 ER   ADD 16060,X  05273 03356151 3•/R PIN*16151    05337 04107606 48•6 BRX 07606
00237 00043700 04•0 HLT*03700    05260 40006713 -0X= HLT 06713    05274 02105275 28••              05340 00107660 08•  BRU 07660
00240 00004000 00•0 HLT 04000    05261 50632147 QTAP              05275 50256322 QETB E9M*16322,PP 05341 03510330 3R3H STA 10330
00241 00000000 0000 HLT 00000    05262 25606060 E    SUC 06060,X  05276 43606060 L    STB 06060   05342 01411171 1J9Z ETR 11171
00242 00007700 00•0 HLT 07700    05263 40006734 -0X) HLT 06734    05277 40006452 -0U• HLT 06452
00243 00000000 0000 HLT 00000    05264 50236660 QCW  E0M 36660,PP 05300 50246444 QDUM E9M*06444,PP
00244 00000000 0000 HLT 00000    05265 60606060                   05301 47606060 P    LDA 06060   ****************************
00245 00000012 000  HLT 00012    05266 40007237 -0•• HLT 07237    05302 40007637 -0•• HLT 07637    * RECORD NUMBER *004*********
00246 00000000 0000 HLT 00000    05267 50625123 QSRC              05303 50224646 QB88 E9M 24646,PP * RECORD LENGTH: 00040 WORDS ***
00247 00000000 0000 HLT 00000    05270 30606060 H                 05304 63606060 T    STB 06060,X  ****************************
00250 00000004 0004 HLT 00004    05271 40007075 -0Y• HLT 07075    05305 40007736 -0•< HLT 07736
00251 00000000 0000 HLT 00000    05272 50446227 QMSG              05306 50627043 QSYL
00252 77770000 ••00 EAX*30000,PX 05273 60606060                   05307 24516060 DR                05323 03350077 3•0• PIN*10077
00253 00007777 00•• HLT 07777    05274 40007766 -0•W HLT 07766    05310 40005410 -0•8 HLT 05410    05324 02105325 28•E
00254 00000000 0000 HLT 00000    05275 11111111 9999              05311 50472562 QPES              05325 50627044 QSYM
 DITTO THRU LOC 00257            05276 00000000 0000 HLT 00000    05312 66606060 W    RSH 06060,X  05326 31606060 I    MRG 06060,PX
                                 05277 04377767 4••X BRM*37767    05313 40007777 -0•• HLT 07777    05327 40007770 -0•Y HLT 07770
                                 05300 00410325 0J3E              05314 50627062 QSYS              05330 50627062 QSYS

---

05331 63606060 T    STB 06060,X  ****************************** 05411 04377767 4••X BRM*37767    05425 07306354 7HT• SKG 06354
05332 40007771 -0•Z HLT 07771    * RECORD NUMBER *005*********** 05412 00210325 0A3E E0M 10325    05426 00105422 08•B BRU 05422
05333 50223145 QBIN E0M 23145,PP * RECORD LENGTH: 00040 WORDS *** 05413 07610325 7/3E LDA 10325    05427 07606445 7 UN LDA 06445
05334 46606060 B    RSH 06060    ****************************** 05414 07211170 7A9Y SKA 11170    05430 07506464 7QUU LDA 06464
05335 40007772 -0•• HLT 07772                                     05415 00107675 08•• BRU 07675    05431 03606442 3 UK STB 06442
05336 50627044 QSYM                                               05416 00100365 083V BRU 00365    05432 01606447 1 UP MRG 06447
05337 46606060 B    RSH 06060    05353 03353207 3•5 7 PIN*13207   05417 04107606 48•6 BRX 07606    05433 07306352 7HT• SKG 06352
05340 40007773 -0•, HLT 07773    05354 02105355 28$•              05420 00107660 08•  BRU 07660    05434 00105455 08•• BRU 05455
05341 50223145 QBIN E0M 23145,PP 05355 50430222 QL2R              05421 03510330 3R3H STA 10330    05435 13377777 •••• PIN*37777,PP
05342 31606060 I    MRG 06060,PX 05356 60606060                   05422 01411171 1J9Z ETR 11171    05436 00000000 0000 HLT 00000
05343 40007774 -0•( HLT 07774    05357 40007754 -0•• HLT 07754                                     05437 04377767 4••X BRM*37767
05344 50627062 QSYS              05360 50430322 QL3R                                               05440 00410325 0J3E
05345 47606060 P    LDA 06060    05361 60606060                   ****************************** 05441 04377767 4••X BRM*37767
05346 40007775 -0•• HLT 07775    05362 40007755 -0•• HLT 07755    * RECORD NUMBER *006*********** 05442 00210325 0A3E E0M 10325
05347 50430022 QL0B              05363 50430422 QL4R              * RECORD LENGTH: 00040 WORDS *** 05443 07610325 7/3E LDA 10325
05350 60606060                   05364 60606060                   ****************************** 05444 07211170 7A9Y SKA 11170
05351 40007752 -0•• HLT 07752    05365 40007756 -0•; HLT 07756                                     05445 00107675 08•• BRU 07675
05352 50430122 QL1B              05366 50430522 QL5R                                               05446 00100365 083V BRU 00365
05353 60606060                   05367 60606060                   05403 03351576 3•:• PIN*11576    05447 04107606 48•6 BRX 07606
05354 40007753 -0•$ HLT 07753    05370 40007757 -0•• HLT 07757    05404 02105405 28•5              05450 00107660 08•  BRU 07660
05355 11111111 9999              05371 50430622 QL6R              05405 50434722 QLPB              05451 03510330 3R3H STA 10330
05356 00000000 0000 HLT 00000    05372 60606060                   05406 60606060                   05452 01411171 1J9Z ETR 11171
05357 04377767 4••X BRM*37767    05373 40007760 -0•  HLT 07760    05407 40007764 -0•U HLT 07764
05360 00410325 0J3E              05374 50430722 QL7R              05410 02000000 2000 NOP 00000
05361 04377767 4••X BRM*37767    05375 60606060                   05411 03506442 3QUK STA 06442    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
05362 00210325 0A3E E0M 10325    05376 40007761 -0•/ HLT 07761    05412 03606421 3 UA STB 06421    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
05363 07610325 7/3E LDA 10325    05377 50235122 QCRB E0M 35122,PP 05413 04630003 4T03 RCH 30003    $ END OF ALL PROCESSING $$$$$$$$
05364 07211170 7A9Y SKA 11170    05400 60606060                   05414 03506425 3QUE STA 06425    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
05365 00107675 08•• BRU 07675    05401 40007762 -0•S HLT 07762    05415 07506452 7QU• LDB 06452    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
05366 00100365 083V BRU 00365    05402 50234722 QCPR E0M 34722,PP 05416 07106405 7BU5 LDX 06405
05367 04107606 48•6 BRX 07606    05403 60606060                   05417 23506457 CQU• STA 06457,X
05370 00107660 08•  BRU 07660    05404 40007763 -0•T HLT 07763    05420 04105417 48•• BRX 05417
05371 03510330 3R3H STA 10330    05405 11111111 9999              05421 03606452 3 U• STB 06452
05372 01411171 1J9Z ETR 11171    05406 00000000 0000 HLT 00000    05422 04305502 4H•2 BRM 05502
                                 05407 04377767 4••X BRM*37767    05423 04305641 4H;J BRM 05641
                                 05410 00410325 0J3E              05424 07606441 7 UJ LDA 06441
```

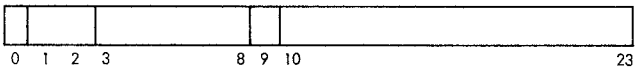
Fig. 2 (contd)



```
 12:13 MAY 29,'73 ID=0010-F00
 JOB CCK,TEST-PDP11-DUMP
 ASSIGN MT I,(FILE,PDP11),(IN)
 RUN (LMN,DUMP)
*WORD=16
*BYTE=8
*ADDRESS=BYTE,OCT
*CHAR=ASCII
*BIN=OCT
*ROTATE:8
*START=REC+2:NNNNNNNNNNNNNNNN=6
*NEWLOC=REC+2:NNNNNNNNNNNNNNNN=A
*INSDEF: 1111222333444555
*PRTDEF:MMMM LLA LLA
*PARAM:2,LIT= R, @, +,@+, -,@-, X,@X
*PARAM:4,LIT= R, @, +,@+, -,@-, X,@X
*HEAD: OP  SRC DST
*OPS=OCT
*END
WARNING: THIS IS THE END


***CONSISTENCY TEST FOR TABLES***


 LOC  LLLLLL AA  OP  SRC DST      LOC  LLLLLL AA  OP  SRC DST      LOC  LLLLLL AA  OP  SRC DST      LOC  LLLLLL AA  OP  SRC DST

********************************* ********************************* ********************************* *********************************
* RECORD NUMBER *001************ * RECORD NUMBER *002************ * RECORD NUMBER *003************ * RECORD NUMBER *004************
* RECORD LENGTH: 00026 WORDS *** * RECORD LENGTH: 00026 WORDS *** * RECORD LENGTH: 00026 WORDS *** * RECORD LENGTH: 00022 WORDS ***
********************************* ********************************* ********************************* *********************************


36772 000001 .. SPCL  R0  R1     37050 000001 .. SPCL  R0  R1     37126 000001 .. SPCL  R0  R1     37204 000001 .. SPCL  R0  R1
36774 000064 .4 SPCL  R0  X4     37052 000064 .4 SPCL  R0  X4     37130 000064 .4 SPCL  R0  X4     37206 000054 ., SPCL  R0 @+4
36776 037000 >. BIT  @X0  R0     37054 037056 >. BIT  @X0 @+6     37132 037134 >  BIT  @X1 @+4     37210 037212 >. BIT  @X2  @2
37000 010415 .. MOV   R4 @5      37056 110167 .. MOVB  R1  X7     37134 006104 .D SPCL  X1  R4     37212 110103 .C MOVB  R1  R3
37002 124040 (  CMPB  -0  -0     37060 140504 AD BICB  R5  R4     37136 006101 .A SPCL  X1  R1     37214 005002 .. SPCL @+0  R2
37004 111001 .. MOVB @0  R1      37062 012700 .@ MOV   +7  R0     37140 006104 .D SPCL  X1  R4     37216 011504 .D MOV   @5  R4
37006 000420 .. SPCL  R4 +0      37064 037242 >  BIT  @X2  -2     37142 006101 .A SPCL  X1  R1     37220 105002 .. SPLB @-0  R2
37010 010504 .D MOV   R5  R4     37066 120120  P CMPB  R1  +0     37144 105702 .B SPLB @-7  R2     37222 152702 UB BISB  +7  R2
37012 110301 .A MOVB  R3  R1     37070 001005 .. SPCL  @0  R5     37146 001731 .Y SPCL  @7 @+1     37224 000006 .. SPCL  R0  R6
37014 000415 .. SPCL  R4 @5      37072 116027 .. MOVB  X0  +7     37150 105302 .B SPLB @-3  R2     37226 012701 .A MOV   +7  R1
37016 010405 .. MOV   R4  R5     37074 000006 .. SPCL  R0  R6     37152 001336 .  SPCL  @3 @+6     37230 000060 .0 SPCL  R0  X0
37020 006205 .. SPCL  X2  R5     37076 037000 >. BIT  @X0  R0     37154 005702 .B SPCL @-7  R2     37232 006304 .D SPCL  X3  R4
37022 006305 .E SPCL  X3  R5     37100 000177 .. SPCL  R1 @X7     37156 100714 .L SPLB  R7  @4     37234 103001 .. SPLB @+0  R1
37024 110103 .C MOVB  R1  R3     37102 177772 ..       @X7 @X2     37160 012701 .A MOV   +7  R1     37236 005201 .. SPCL @-2  R1
37026 005004 .. SPCL @-0  R4     37104 022700 %@ CMP   +7  R0     37162 000015 .. SPCL  R0  @5     37240 000703 .C SPCL  R7  R3
37030 005002 .. SPCL @-0  R2     37106 037252 >* BIT  @X2 @-2     37164 000731 .Y SPCL  R7 @+1     37242 056377  . BIS   X3 @X7
37032 012700 .@ MOV   +7  R0     37110 001366 .. SPCL  @3  X6     37166 005767 .. SPCL @-7  X7     37244 005015 .. SPCL @-0  @5
37034 177560 ..       @X5 X0     37112 006001 .. SPCL  X0  R1     37170 140376 @. BICB  R3 @X6     37246 136615 *. BITB  X6  @5
37036 005210 .. SPCL @-2  @0     37114 006001 .. SPCL  X0  R1     37172 001715 .M SPCL  @7  @5     37250 002240  . SPCL  +2  -0
37040 105710 .H SPLB @-7  @0     37116 006001 .. SPCL  X0  R1     37174 005725 .U SPCL @-7  +5     37252 002004 .. SPCL  +0  R4
37042 100376 .. SPLB  R3 @X6     37120 142701 EA BICB  +7  R1     37176 005102 .B SPCL @-1  R2     37254 000166 .. SPCL  R1  X6
37044 005720 .P SPCL @-7  +0     37122 000377 .. SPCL  R3 @X7     37200 010504 .D MOV   R5  R4     37256 102016 .. SPLB  +0  @6
37046 111001 .. MOVB @0  R1      37124 152701 UA BISB  +7  R1     37202 000406 .. SPCL  R4  R6     *********************************
37050 105767 .. SPLB @-7  X7     37126 000006 .. SPCL  R0  R6     37204 010405 .. MOV   R4  R5     * RECORD NUMBER *005************
37052 140510 AH BICB  R5  @0     37130 006104 .D SPCL  X1  R4     37206 006205 .. SPCL  X2  R5     * RECORD LENGTH: 00003 WORDS ***
37054 100375 .. SPLB  R3 @X5     37132 006101 .A SPCL  X1  R1     37210 006305 .E SPCL  X3  R5     *********************************


                                                                                                  77773 000001 .. SPCL  R0  R1
                                                                                                  77775 000006 .. SPCL  R0  R6
                                                                                                  77777 000001 .. SPCL  R0  R1


                                                                                                  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
                                                                                                  $ END OF ALL PROCESSING $$$$$$$$
                                                                                                  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Fig. 3. An application of the Universal Dump Program to the PDP-11 computer

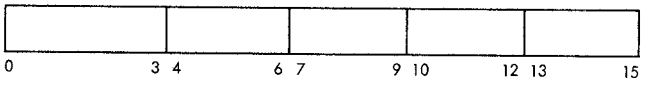
THE 930 INSTRUCTION:



BIT	FUNCTION
0	UNUSED
1-2	TAG FIELD: BIT 1 SET, INDEXING IS USED BIT 2 SET, OPERATOR IS PROGRAMMED
3-8	OPERATION CODE
9	INDIRECT ADDRESSING BIT
10-23	ADDRESS OF OPERAND

Fig. 4. Instruction format for the XDS 930 computer

THE PDP-11 INSTRUCTION:



BIT	FUNCTION
0-3	OPERATION CODE
4-6	ADDRESSING MODE FOR SOURCE OPERAND
7-9	SOURCE OPERAND REGISTER
10-12	ADDRESSING MODE FOR DESTINATION OPERAND
13-15	DESTINATION OPERAND REGISTER

Fig. 5. Instruction format for the PDP-11 computer