NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.
SICOSAS, AN IBM 7090 COMPUTER ASSEMBLY PROGRAM
FOR THE CDC 160-A COMPUTER

J. D. Drinan

17 January 1963

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology; this work was supported by the U. S. Air Force under Air Force Contract AF 19(628)-500.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>GENERAL DESCRIPTION</td>
<td>2</td>
</tr>
<tr>
<td>SICOM ASSEMBLY PROGRAM (SICAP)</td>
<td>4</td>
</tr>
<tr>
<td>USE OF $ IN SICAP</td>
<td>5</td>
</tr>
<tr>
<td>SICAP LIBRARY SUBROUTINES</td>
<td>6</td>
</tr>
<tr>
<td>SICAP INFORMATION LAYOUT</td>
<td>7</td>
</tr>
<tr>
<td>MACHINE LANGUAGE ASSEMBLY PROGRAM (MLAP)</td>
<td>8</td>
</tr>
<tr>
<td>SICOSAS SPECIAL PURPOSE CARDS</td>
<td>9</td>
</tr>
<tr>
<td>SICAP</td>
<td>10</td>
</tr>
<tr>
<td>MLAP (MLAP, ..., OCTAL)</td>
<td>10</td>
</tr>
<tr>
<td>*(ASTERISK IN COLUMN 1)</td>
<td>10</td>
</tr>
<tr>
<td>NAME</td>
<td>10</td>
</tr>
<tr>
<td>ORG Y</td>
<td>10</td>
</tr>
<tr>
<td>DEC</td>
<td>11</td>
</tr>
<tr>
<td>OCT</td>
<td>11</td>
</tr>
<tr>
<td>BSS Y</td>
<td>11</td>
</tr>
<tr>
<td>EQU Y ± a</td>
<td>12</td>
</tr>
<tr>
<td>MASK</td>
<td>12</td>
</tr>
<tr>
<td>SQR</td>
<td>13</td>
</tr>
<tr>
<td>LOG</td>
<td>13</td>
</tr>
<tr>
<td>EXP</td>
<td>13</td>
</tr>
<tr>
<td>SIN</td>
<td>13</td>
</tr>
<tr>
<td>ATAN</td>
<td>13</td>
</tr>
<tr>
<td>CALL Y ± a, k</td>
<td>13</td>
</tr>
<tr>
<td>RETURN</td>
<td>13</td>
</tr>
<tr>
<td>END Y</td>
<td>13</td>
</tr>
<tr>
<td>SICOSAS LIMITATIONS AND RESTRICTIONS</td>
<td>13</td>
</tr>
<tr>
<td>ERROR PRINTOUTS</td>
<td>14</td>
</tr>
<tr>
<td>SICOSAS LOADER FOR THE 160-A</td>
<td>14</td>
</tr>
<tr>
<td>SICOSAS OPERATION NOTES</td>
<td>16</td>
</tr>
<tr>
<td>7090 PROCEDURE</td>
<td>16</td>
</tr>
<tr>
<td>160-A PROCEDURE</td>
<td>17</td>
</tr>
<tr>
<td>APPENDIX A: SICAP MNEMONIC OPERATION CODES</td>
<td>18</td>
</tr>
<tr>
<td>APPENDIX B: SICAP MNEMONICS ALPHABETIZED</td>
<td>33</td>
</tr>
<tr>
<td>APPENDIX C: MLAP MNEMONICS ALPHABETIZED</td>
<td>37</td>
</tr>
<tr>
<td>APPENDIX D: SAMPLE SICOSAS PROGRAM (SICAP ONLY)</td>
<td>43</td>
</tr>
<tr>
<td>APPENDIX E: SAMPLE SICOSAS PROGRAM (SICAP AND MLAP)</td>
<td>44</td>
</tr>
<tr>
<td>APPENDIX F: SICOSAS PROGRAM LISTING</td>
<td>45</td>
</tr>
</tbody>
</table>
ABSTRACT

SICOSAS is a symbolic assembly program designed to operate on the IBM 7090 for the purpose of merging many of the desirable features of two of CDC's programming systems, SICOM and OSAS-A, into a single programming system in order to make the CDC 160-A computer a more versatile tool for many applications. This memo is not intended as a self-sufficient primer. It is aimed at those who have had some background in the IBM 7090 computer and its associated SOS programming system as well as in the CDC 160-A computer with its OSAS-A and SICOM programming systems. Input cards are punched in the familiar SOS format. Output is an SOS-like program listing plus a binary copy of the assembled program on magnetic tape which is readily loaded and executed on the 160-A.
INTRODUCTION

The present Lincoln Laboratory CDC 160-A computer configuration offers the user a relatively fast (6.4 μ cycle time; 12.8 μ add time) machine with a 16K, 12-bit-word magnetic core memory. Buffering is available as well as indirect addressing and program interrupt. Input-output media include paper tape, magnetic tape (IBM compatible), and a typewriter. Additionally, real time data may be processed using existing input-output facilities. A scope is also connected for display purposes.

To those who are using or have used some of the Laboratory's larger machines, there may be some disappointing aspect to the 160-A facility among the following.

a. There are no machine-language multiply or divide instructions as such. These functions must be performed in other ways.
b. There are no machine index registers.
c. There is no floating point hardware.
d. The basic machine-language compiler OSAS-A*, while symbolic and mnemonic, is quite separate from the interpretive programming system with which it may interplay.
e. One of the two interpretive programming systems, SICOM†, is decimal but non-mnemonic, non-symbolic, requires paper tape input for compilation, and affords no program listing.
f. The versions of FORTRAN now available for the 160-A do not provide machine-language facility. [New versions of FORTRAN for the 160-A are under development‡, but even these new versions have certain inadequacies. The availability of an adequate version of FORTRAN (e.g., similar to 7090 FORTRAN) might well have obviated the current immediate need for SICOSAS.]

---

‡ Control Data Corporation Publication No. 505, "160-A FORTRAN/General Information Manual."
SICOM has been described in the CDC literature* as having the following characteristics.

"SICOM for the Control Data 160-A computer is a general-purpose interpretive system utilizing floating point arithmetic. It effectively converts the 160-A from a binary, fixed-point machine with a 12-bit word length, into a decimal, floating-point machine with 10-decimal digit, plus exponent, word length. SICOM provides full arithmetic, indexing, and logical capabilities... as well as additional features which provide functions and SICOM or machine-language subroutines."

Certainly the prospect of going from a machine with no multiply/divide capability and having no index registers to the same machine possessing full arithmetic, indexing, logical and machine-language-dropout-and-return capabilities even at the reduced speed of an interpretive programming scheme is an enjoyable one for most applications.

It has been the primary intent then of the SICOSAS system to retain all of the desirable operational characteristics of SICOM while making programming for SICOM less of a chore. This has been done in part by by-passing the SICOM compiler completely in assembling the SICOM part of a SICOSAS program. A secondary feature of the SICOSAS system has been the elimination of a separate compilation of the machine-language portion of the SICOM program which would normally be done by OSAS-A since this compilation is also done by SICOSAS at the same time as the compilation of the main SICOM program.

SICOSAS then is really a dual assembler, each mnemonic and symbolic, each capable of independent operation. Each assembler and its ground rules will be considered separately.

GENERAL DESCRIPTION

Although the two assemblers in SICOSAS, SICAP (SICOM Assembly Program) and MLAP (Machine Language Assembly Program) may each be invoked to the exclusion of the other, the SICOSAS programming system was designed primarily for those who require interplay between the SICOM interpretive system with its large decimal capacity (10 decimal digits) and machine-language, hand-coded routines which perform special functions that cannot be accomplished in the SICOM language.

---

*Control Data Corporation Publication No. 502, "Programming Systems."
Before SICOSAS, a person with such a problem would prepare the SICOM code by writing it in an absolute numeric format for punching by the flexowriter. His only listing of the program would be the flexowriter printout of the absolute code which would simply be a probably neater copy of the original manuscript. No commentary of any kind would be retained beyond the original manuscript. Thus, a portion of the document from which he would work would appear something as follows:

```
1221003
0251004
0240720
:   :
036Y000
4
```

Next he would prepare his machine-language subroutines, coding these symbolically and with machine code mnemonics in the OSAS-A system. His origins for these subroutines would be equated to the effective machine location transferred to by the SICOM "transfer to machine language" instructions. Any reference to the main SICOM program would be made in absolute or quasi absolute in these subroutines since two separate unrelated compilers would prepare these codes for machine loading. Finally, the subroutines would be compiled by OSAS-A (via paper tape or card images on magnetic tape) and a final paper tape containing these subroutines would evolve for machine loading.

Having loaded the subroutines, the next step would be to load the SICOM compiler/interpreter and proceed to compile the main SICOM program from a flexo tape in the format described above over the previously-entered, hand-coded subroutines. If the SICOM program needed library mathematical subroutines, these would finally be loaded over everything else in the manner prescribed in the SICOM manual.

To accomplish the same result in the SICOSAS system, one codes the entire problem together as a single program. Programming is virtually entirely symbolic. Machine-language mnemonics have been carried over entirely from OSAS-A for constructing hand-coded subroutines. MLAP additionally can be made either a decimal compiler or octal compiler and may be switched back and forth from card to card if need be. For SICOM commands, since no mnemonics existed, a full set has been constructed. All of the basic SICOM mathematical library subroutines are now an integral part of SICOSAS so that any or all may be incorporated into the final program automatically at compile time thereby obviating their separate loading at execute time.

In summary, the procedure to be followed to arrive at a SICOSAS program
operating on the 160-A is as follows:

a. Code the entire problem on card-room-provided coding forms.
b. Use the card room facility for punching, verifying, etc.
c. Compile the program on the 7090 either from tape (prestored) or from cards.
d. Print out the BCD listing tape and save the binary tape (map of 160-A bank 1 core storage).
e. At the 160-A call in assembled program for execution from magnetic tape with special SICOSAS loader.

This memo does not pretend to contain all the information required for a person to write a SICOSAS program. It is intended primarily for those who have at least a nodding acquaintance with the referenced literature, rather as a supplement describing a different approach to the solution of a certain class of problems. A limited number of SICOM Manuals are available for reference for those who wish to investigate this approach further. A listing of the SICOSAS program is included as Appendix F.

**SICOM ASSEMBLY PROGRAM (SICAP)**

The normal numeric format required by the bypassed SICOM compiler is of the following seven-digit form:

\[
K \text{OPADR}(1)
\]

where \( K \) = the index register: \( 0 \leq K \leq 9 \),
and \( \text{OP} \) = the Operation code,
and \( \text{ADDR} \) = the SICOM address.

Two departures from this format occur for those commands in which DR of (1) above is the operation code and for those in which \( K \) of (1) above is the operation code.

The basic SICAP symbolic card layout is as follows:

Cols. 1-6  Location Field containing all blanks or an alphabetic symbol to be associated with this card. Col. 1 must be non-blank if a symbol is used. At least one character in the symbol must be non-numeric, and none may be + or −.

Col. 7  Blank

Cols. 8-15  Operation Field containing a mnemonic operation code beginning in Col. 8.

Cols. 16 +  Variable Field starting in Col. 16 is composed
of the address field and the tag (or index) field. The address field is separated from the tag field by a comma (,). The address field may or may not include an additive subfield. If it does contain this subfield, it is separated from the pure address by a + or – sign. A blank (b) in variable field is a terminating character. Thus, typical cards appear.

1 7 8 1 6
X R A Y   C L A   A B L E - 1 0 , 7
S Y M B O L M P Y   P , 2
P   S T O   B O X + 2
D V P   P I
S P L I T   I D V P C   2 3 4 5

Thus, for card XRAY above, if the symbol ABLE had previously been equated to SICAP location 1000, say, SICAP would consider \( K = 7 \), \( OP = 22 \) and \( ADDR = 990 \). Hence the equivalent numeric code 7220990 would be decoded and inserted into the machine cells assigned to location XRAY.

Cols. b+1 to 72 Commentary. Any combination of Hollerith characters.

Appendix A lists all of the SICAP mnemonic operation codes along with a description of each. Appendix B is an alphabetic listing of all mnemonics recognized by SICAP including all of the entries in Appendix A plus the SICAP pseudo-operation codes.

Appendix D shows a SICOSAS program in which all coding is in the SICAP language. This program is the symbolic, mnemonic, and documented equivalent of the SICOM program appearing on page 42 of the SICOM Manual. The first line of the listing contains the author identification field (J. D. Drinan Nov. 28, 1962.). The second line identifies the various columns in the listing. The four columns headed by MLOC show the machine-language (octal) location of the SICAP information. SILOC, of course, is the SICAP location (decimal) corresponding to MLOC. The K, OP, ADDR columns depict the decoded seven-digit SICOM instruction. At the far right of the listing appears simply a tabulation of the cards in the symbolic deck. The remainder of the columns contains an image (Cols. 1-72) of each input card.

USE OF $ IN SICAP

In the original SICOM the notation Y000 was used to reference the SICOM accumulator. In SICAP this symbol has been replaced by $ (dollar sign), thus the SICAP
instruction at location ABLE

ABLE OFLT B $

would cause the contents of the accumulator to be output (in floating point followed by a tab) on the previously-selected device.

SICAP LIBRARY SUBROUTINES

The original SICOM system is composed in general of an interpreter/compiler program plus a library of relocatable arithmetic subroutines. Normally the subroutines required by the object program are manually loaded from paper tape into the absolute locations to which the object program will transfer. Control is then manually transferred to the SICOM dynamic start point in the main program for execution.

In SICOSAS a required subroutine is automatically incorporated into the object program upon the encounter of one of the five appropriate SUBROUTINE cards to be described here. Subroutine cards have the following format:

<table>
<thead>
<tr>
<th>Field</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location Field</td>
<td>A location symbol or all blank</td>
</tr>
<tr>
<td>Operation Field</td>
<td>Subroutine Designator</td>
</tr>
<tr>
<td>Variable Field</td>
<td>None</td>
</tr>
</tbody>
</table>

The subroutine designator is one of the following five specifying the square root, logarithmic, exponential, sin-cos, and arc tan subroutines, respectively.

<table>
<thead>
<tr>
<th>Designator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQR</td>
<td>Square root</td>
</tr>
<tr>
<td>LOG</td>
<td>Logarithmic</td>
</tr>
<tr>
<td>EXP</td>
<td>Exponential</td>
</tr>
<tr>
<td>SIN</td>
<td>Sin-cos</td>
</tr>
<tr>
<td>ATAN</td>
<td>Arc tan</td>
</tr>
</tbody>
</table>

The effect of meeting one of these cards is to cause SICAP to insert one of these five basic subroutines into the object program at the place in the program where the subroutine designator card was met. If a symbol is present in the location field of the SUBROUTINE card, it will be entered into the symbol table and equated to the value of the location counter at the time of meeting. In any case, SICAP inserts a symbol of its own at this location identical to the symbol of the subroutine designator; i.e., SQR, LOG, EXP, SIN or ATAN. Thus, to enter a given subroutine, one may transfer (TSR) either to the program symbol used to identify the start of the subroutine or in the absence of such a symbol to the subroutine designator itself.

The reader’s attention is invited to the table on page 55 of the SICOM Manual for information about arguments, entry points, etc., pertaining to these subroutines.
SICAP INFORMATION LAYOUT

The reader is directed to page 83 of the SICOM Manual for information regarding the location and format of the SICOM pseudo accumulator.

The machine layout for SICAP commands and numeric data is given below since a) it does not appear in the SICOM Manual, and b) it is "must" information for those SICOSAS programs which employ SICAP for arithmetic computations and MLAP for logical manipulations of the results.

Suppose the following two cards appear in a SICAP program:

```
ORG 1000
DATA DEC 9876.543210
```

The SICAP digit numbering convention is \( D^{10}_9 \ldots D^1_1 \) where \( D^9_{10} \) of DATA is 9 and \( D^1_1 \) is 0. Each \( D^n \) occupies 4 bits weighted 8, 4, 2 and 1 from left to right. A SICAP datum is stored in two SICAP locations which take up four machine locations. As a result of the above cards, this numeric datum will be assembled into SICAP locations DATA and DATA+1 (these, of course, are absolute SICAP locations 1000\(_{10}\) and 1001\(_{10}\) which occupy machine locations 3720\(_8\) through 3723\(_8\)) in the following format:

```
<table>
<thead>
<tr>
<th>Bit Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 7 3 0</td>
</tr>
</tbody>
</table>

DATA (3720\(_8\))
```

```
<table>
<thead>
<tr>
<th>D3</th>
<th>D2</th>
<th>D1</th>
</tr>
</thead>
<tbody>
<tr>
<td>D6</td>
<td>D5</td>
<td>D4</td>
</tr>
</tbody>
</table>
```

```
+1 (3722\(_8\))
```

```
<table>
<thead>
<tr>
<th>D9</th>
<th>D8</th>
<th>D7</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>EXPONENT</td>
<td>D10</td>
</tr>
</tbody>
</table>
```

The octal representation of this number in cells 3720 through 3723 appears

\[
1020 = D^3_3 D^2_2 D^1_1
\]
\[
2503 = D^6_6 D^5_5 D^4_4
\]
\[
4166 = D^9_9 D^8_8 D^7_7
\]
\[
2111 = S, \text{EXP}, D^{10}_{10}
\]

The machine cell assigned to contain the first part \( (D^3_3 D^2_2 D^1_1) \) of the datum must be an even SICAP location. This requirement is automatically handled by SICAP and is of no concern to the user.

The SICAP command requires one SICAP location (two machine locations), thus the following two cards:

```
ORG  100
START CLA START+1, 7
```
would cause information in the SICOM format KOPADDR to be assembled into the two equivalent machine locations 0310₈ and 0311₈ symbolically as follows:

\[
0310 = \begin{array}{cccccc}
& & & & & \\
& & & A & D & D & R & 0 \\
\end{array}
\]

\[
0311 = \begin{array}{cccc}
& & & K \\
& & & OP \\
\end{array}
\]

In the above format bit 11 of the "ADDR" cell and bits 11 and 10 of the "KOP" cell are concerned with bank assignment and are of no interest to the programmer.

Numerically in SICAP these two cells appear

\[
0310 = 0101 \\
0311 = 0722
\]

When examined in core storage, the two cells appear

\[
0310 = 0145 \\
0311 = 0722
\]

MACHINE LANGUAGE ASSEMBLY PROGRAM (MLAP)

In SICOM exists an instruction K74ADDR (transfer to machine language) which, when executed by the SICOM interpreter program, results in uninhibited control being transferred to the SICOM location effective ADDR.

SICAP uses mnemonic CALL Y ± a, k for the same purpose. That is, the interpreter relinquishes control and the machine-language program beginning at SICAP effective location Y ± a, k is free to operate at machine-language speed.

MLAP was included in the SICOSAS system to bypass the need of a separate OSAS-A compilation of the machine-language subroutines needed in many SICOM programs. Every machine-language operation mnemonic used in OSAS-A is present in MLAP. Certain control pseudo operations are different and these will be discussed.

In using MLAP in a SICOSAS program one simply heads the portions of the program that are machine language with an MLAP card (see section entitled SICOSAS Special Purpose Cards) and codes in what is effectively the OSAS-A language. Return to the SICAP interpreter is accomplished through the use of a single machine-language macro instruction RETURN. (See SICOSAS Special Purpose Cards.)

The symbolic card format for MLAP is identical to that for SICAP, excepting, of course, that there is no tag (index) field in MLAP. The field layout then is
with all SICAP format ground rules for each field in effect.

Appendix E shows a SICOSAS program in which there is interplay between the SICAP and MLAP languages.

Because of the inherent differences in the requirements of OSAS-A, the basic assembly system for the 160-A computer and MLAP, whose function is to insert machine-language instructions in a larger over-all interpretive system, certain of the OSAS-A pseudo-operation functions have, in some cases, been dropped. For those pseudo operations which do have a parallel in MLAP, the reader is referred to the section entitled SICOSAS Special Purpose Cards. In summary, the status of the OSAS-A pseudo operations in MLAP is as follows:

<table>
<thead>
<tr>
<th>OSAS-A</th>
<th>MLAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORG</td>
<td>ORG</td>
</tr>
<tr>
<td>PRG</td>
<td>None</td>
</tr>
<tr>
<td>CON</td>
<td>None</td>
</tr>
<tr>
<td>BLR</td>
<td>None</td>
</tr>
<tr>
<td>BSS</td>
<td>BSS</td>
</tr>
<tr>
<td>WA1</td>
<td>None</td>
</tr>
<tr>
<td>END</td>
<td>END</td>
</tr>
<tr>
<td>EQU</td>
<td>EQU</td>
</tr>
<tr>
<td>REM</td>
<td>*(Col. 1)</td>
</tr>
<tr>
<td>BNKX</td>
<td>None</td>
</tr>
<tr>
<td>SUPA</td>
<td>None</td>
</tr>
<tr>
<td>SUPB</td>
<td>None</td>
</tr>
<tr>
<td>BCD</td>
<td>None</td>
</tr>
<tr>
<td>BCDR</td>
<td>None</td>
</tr>
<tr>
<td>FLX</td>
<td>None</td>
</tr>
<tr>
<td>FLXR</td>
<td>None</td>
</tr>
</tbody>
</table>

Appendix C contains a complete alphabetized list of all mnemonics recognized by MLAP.

**SICOSAS SPECIAL PURPOSE CARDS**

The following cards are not of the command type listed in Appendix A. Some apply only when SICAP is controlling the assembly; others when MLAP is; still others when either is. All of these mnemonic, pseudo-operation codes occupy the operation field of the card unless stated otherwise.
**SICAP**

The SICAP card has no location field, no variable field. When a compilation is begun by SICOSAS, the assumption is that the initial cards conform to the SICAP mnemonic scheme. (See Appendix A.) When this mode is changed (by an MLAP card), the system is returned to the SICAP mode by a SICAP card. The SICAP card then specifies that the cards following it belong to the SICAP language, and hence, OSAS-A-type cards (recognized by MLAP) will be tagged as erroneous. A SICAP card read by SICAP has no effect.

**MLAP (MLAP.... OCTAL)**

The MLAP card is a counterpart of the SICAP card. Upon meeting an MLAP card the mode of SICOSAS is changed to expect OSAS-A-type cards (see Appendix C) to follow until another SICAP card is met. The MLAP card has no location field. If the variable field (Cols. 16+) of the MLAP card is blank, the mode of MLAP is set to DECIMAL. If the variable field is alphabetic (such as OCTAL), the mode of MLAP is set to OCTAL. The variable fields of the cards following the MLAP (MLAP.... OCTAL) card are converted decimally or octally depending on the mode established. An MLAP card read by MLAP has no effect.

* (ASTERISK IN COLUMN 1)

A card so punched has no effect on SICOSAS other than to reproduce the entire card on the output listing. This then is a remarks card and may be used either with SICAP or MLAP in control. Such cards may appear anywhere in the symbolic deck.

**NAME**

The NAME card has no location field. The purpose of the card is to identify the output listing. This is done by inserting the contents of columns 16 through 51 of the NAME card into an appropriate place in the first line of every page of the output listing. Normally the program author would use his name and the date. A NAME card may be read by either SICAP or MLAP. A later NAME card will supplant one read earlier.

**ORG Y**

An origin (ORG) card has the effect of resetting the location counter to the value of the decimal integer Y in the variable field. It is to be remembered that one basic SICOM instruction requires two machine cells. For this reason, Y is always considered a SICOM location so that the location counter \( L_8 = 2Y_{10} \). Thus,
resets \( L \) to \( 3720_8 \).

Any number of ORG cards may appear in the program. SICAP and MLAP treat this card identically.

**DEC**

The pseudo operation DEC applies only to SICAP. It is used to introduce floating point decimal data into the SICAP portion of the SICOSAS program. The location and variable fields of the DEC card conform to the general rules for these fields. The variable field contains a signed mantissa (+ may be omitted), a mandatory decimal point (\( \cdot \)), and an optional signed exponent. Thus the following cards are valid examples of the DEC pseudo operation.

<table>
<thead>
<tr>
<th>Card</th>
<th>Will Convert To</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC 123.456E05</td>
<td>+ 1.23456 \times 10^8</td>
</tr>
<tr>
<td>DEC -1.23456E-1</td>
<td>- 1.23456 \times 10^0</td>
</tr>
</tbody>
</table>

It is to be remembered that in the variable field a blank is a terminating character so that

\[
\text{DEC 123.456E}b5 \text{ produces } + 1.23456 \times 10^3
\]

SICOSAS automatically stores DEC data according to the SICOM ground rules for this type of input.

**OCT**

The pseudo operation OCT applies only to MLAP. This card is used when the mode of MLAP has been established as DECIMAL (see MLAP card), but it is desired to introduce an octal integer into the program. The variable field has four columns beginning in Col. 16. This field must contain an octal integer \( I, 0 \leq I \leq 7777 \).

**BSS**

The "block starting with symbol" (BSS) pseudo operation is used to reserve a block of one or more words of core storage within the SICOSAS program. The variable field of the BSS card must be absolute. If SICAP is in control when the BSS is met, \( Y \) SICOM locations (2 x machine cells) are reserved. If MLAP meets the BSS, \( Y_{10} \) or \( Y_8 \) machine locations are set aside depending on the mode of MLAP.
**EQU Y = a**

The general rules for the location and variable field given previously apply to the pseudo operation EQU. Thus Y may be symbolic or absolute and a, if used, may be either octal or decimal. The EQU card, however, must appear in the symbolic deck ahead of any card using the symbol in the location field of the EQU card. Thus the following arrangement will compile correctly:

```
A EQU B
CLA A
;
B DEC 0.
```}

**MASK**

The MASK pseudo operation applies only to SICAP and is used in conjunction with the EXTR (K27 ADDR) instruction. The general rule for the location field applies. The variable field can be thought of as a floating point mask with up to ten sexadecimal numbers and an exponent completely analogous to the one in the DEC pseudo operation.

The sexadecimal numbers 0 through 9 correspond to the four-bit binary configurations 0000 through 1001.

U is 1010  X is 1101
V is 1011  Y is 1110
W is 1100  Z is 1111

Thus if a decimal datum is stored with the DEC pseudo operation

```
A DEC 123.5
CLA A
B EXTR MASKA
;
MASKA MASK Z.ZZE1
```}

The result of the extraction at location B would result in the number .235 appearing in the accumulator with $E = 02$; i.e., 23.5 normalized.

Notice that the use of $E$, ($\cdot$), and the blank are carried over from the DEC card.
These are SICAP Subroutine Designator cards explained under SICAP Library Subroutines.

**CALL Y ± a, k**

This is a SICAP command (explained elsewhere but mentioned here in context with the RETURN macro) which links the SICAP portion of the SICOSAS program with the machine-language portion (MLAP). Control is transferred to the machine-language subroutine starting at effective SICAP location Y ± a, k.

**RETURN**

RETURN is a macro-type, machine-language instruction that is the counterpart of the CALL command. RETURN may have a location field but has no variable field. When MLAP encounters a RETURN, it inserts the following three machine-language instructions into the program.

```
0040 SDC Set Direct Bank
2006 LDD Return Location
0030 IRJ Jump to SICOM Interpreter
```

**END Y**

The variable field of the END pseudo operation is a symbolic location in the SICOSAS program to which control will first be passed upon loading the SICOSAS program from tape on the 160-A. The END card must be present; however, if Y is all blank, the dynamic start point will be set to the first symbolic location encountered in the program.

**SICOSAS LIMITATIONS AND RESTRICTIONS**

a. The present version of SICOSAS was written with a two-bank (0 and 1) 160-A computer in mind. The SICOM compiler/interpreter program perforce occupies all of bank 0. The SICOSAS program is therefore limited to bank 1. This amount of storage is roughly equivalent to 1999 SICAP locations which must be shared with the machine-language portions of the program. SICOM, and therefore, SICAP, is enlargeable but at this writing this size limitation exists.

b. Neither SICAP nor MLAP has the facility directly to accept alphabetic data (which would normally be used for headings by the final 160-A program).
In the case of SICAP, the user may enter this type of data through an input device, while with MLAP the octal equivalents may be compiled directly.

c. Due to the manner in which the SICAP loader operates (see SICOSAS Loader), SICAP locations 0000 to 0005 (octal machine locations 0000 through 0013) may not be used directly by the SICOSAS program at compile time. This is to say that no program instruction or constant should be assigned to these cells; however, these locations may be freely referenced in the coding of the SICOSAS program through the use of the EQU card, for example, (see SICOSAS Special Purpose Cards) and freely used by the program in operation. While all other cells in bank one not explicitly containing the SICOSAS program will contain zeros, these first six SICAP locations will not.

ERROR PRINTOUTS

SICOSAS prints out error indications directly on the program listing as errors are encountered and continues with the assembly. Sources of error and the action taken are the following:

a. Undefined symbol: The symbol is equated to zero.
b. Multiply defined symbol: The symbol is assigned the value given it upon meeting it for the first time.
c. Illegal operation code: A NOP replaces the illegal code.
d. Out of range: In the 160-A machine instruction format EEXX, the XX which is evaluated as out of range is set to zero.
e. Illegal additive field: The additive field is set to zero.
f. OCT card variable field: Variable field is set to zero.
g. Decimal data: Zero is stored for the illegal datum.

SICOSAS LOADER FOR THE 160-A

As a result of the assembly process on the 7090 computer, there exists a binary tape which when read into the 160-A computer will result in bank 1 containing the information necessary to operate the SICOSAS program.

To facilitate the loading of this tape into the 160-A, SICOSAS assembles the following SICAP instructions into the 7090 map of the 160-A bank 1 core memory in
such a fashion that upon read-in they will occupy SICAP locations 0000 through 0005 as follows:

<table>
<thead>
<tr>
<th>Location</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>SELBIN 0</td>
<td>Select binary tape 0</td>
</tr>
<tr>
<td>01</td>
<td>ITAPE 0,1</td>
<td>Read in under control of index #1</td>
</tr>
<tr>
<td>02</td>
<td>TSM4 4</td>
<td>Error exit; transfer to 0004</td>
</tr>
<tr>
<td>03</td>
<td>TSM4 [START]</td>
<td>Read-in o.k.; transfer to execute</td>
</tr>
<tr>
<td>04</td>
<td>REW</td>
<td>Try again</td>
</tr>
<tr>
<td>05</td>
<td>TSM4 1</td>
<td>Reread the tape</td>
</tr>
</tbody>
</table>

When in the assembly process SICOSAS meets the END card, the absolute dynamic start point is determined and stored as the address of the TSM4 instruction in location 0003.

The problem with this tape at the 160-A is simply that of initiating its reading into location 0 of bank 1. This is accomplished by first reading in the SICOM interpreter into bank 0 and then have it call for paper tape input. (See Operation Notes.)

The paper tape which the SICOM interpreter reads contains the following equivalent SICAP instructions:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRPIN 0</td>
<td>Load following data starting in location 0</td>
</tr>
<tr>
<td>SELBIN 0</td>
<td>Select binary tape 0</td>
</tr>
<tr>
<td>ITAPE 0,1</td>
<td>Read tape 0 under I. R. 1 control</td>
</tr>
<tr>
<td>SIB 0,1</td>
<td>Set i base 0, I. R. 1</td>
</tr>
<tr>
<td>SIL 1000,1</td>
<td>Set i limit 1999, I. R. 1</td>
</tr>
<tr>
<td>ITYPE</td>
<td>Select typewriter input</td>
</tr>
<tr>
<td>OTYPE</td>
<td>Select typewriter output</td>
</tr>
<tr>
<td>TSM4 0</td>
<td>Transfer to location 0000</td>
</tr>
<tr>
<td>XEQ 2</td>
<td>Start automatic computation at location 0002</td>
</tr>
</tbody>
</table>

The instructions SELBIN through TSM4 0 are read by the SICOM interpreter/compiler and stored in locations 0000 through 0006 respectively. The XEQ 2 instruction transfers control to the SIB 0,2 instruction at location 0002 under the guidance of the interpreter. The instructions SIL 1999,1; ITYPE; and OTYPE are executed in turn whereupon control goes to location 0000 where the magnetic tape is selected and read-in to location 0000 and successive locations is initiated. The paper tape program is read over and the six order program at the beginning of the magnetic tape is in control. This program transfers to the SICOSAS program dynamic start point at the completion of a successful read-in, or, in the case of a faulty read-in, rewinds the magnetic tape and initiates another loading.
SICOSAS OPERATION NOTES

The procedure to be followed to have a SICOSAS program operate on the 160-A is a twofold one. First, one must run SICOSAS on the 7090 to obtain a) a printout of the listing of the assembled program, and b) a binary magnetic tape which contains a map of the assembled program as it will appear in bank 1 of the 160-A core memory. Second, one must load this binary magnetic tape into the 160-A for execution. These two phases will be discussed in the order outlined.

7090 PROCEDURE

SICOSAS (LA07 is the program identification number) is an SOS program which may be operated as such from the usual squoze deck under control of "Modify and Load". For purposes of minimizing loading time and the number of mediary tapes required for loading, a self-sufficient binary version of SICOSAS has been recorded on magnetic tape. For use with this binary version of LA07 there is available a loader operating through the card reader which calls in SICOSAS from tape drive B9 (high density) and transfers to its dynamic start point for execution.

Irrespective of the method of loading, SICOSAS tape requirements (aside from those required by SOS if this method of loading is elected or the B9 if the binary version is used) are as follows:

- A-7 (high density) BCD listing of assembled program.
- A-8 (high density) BCD input tape (if program to be assembled has been prestored on tape).
- B-7 (high density) Mediary tape.
- B-8 (low density) Binary map of assembled program (input to 160-A).

When SICOSAS is first loaded it halts on an HPR at location 355408 to allow sense switch settings to be made as follows:

- **UP** Input is on tape A-8.
- **SSW1 DOWN** Input is on cards.
- **UP** Print listing on-line (as well as off-line).
- **SSW3 (dynamic) DOWN** Do not print on-line.
- **UP** When finished with this assembly, rewind the listing tape.
- **SSW6 DOWN** Do not rewind the listing tape (stack output).

Upon the completion of an assembly, the binary map tape (B8) is rewound and must be replaced since no stacking is done on this tape. At this point the program
announces the completion of the job, halts and awaits the next run.

In addition to the initial HPR stop and the annotated "finished" stop, SICOSAS will stop on persistent input tape redundancies (HPR 1) and persistent output tape redundancies (HPR 2). The user is advised in either case to change tape and/or tape drive and start over. Pressing START at this point transfers to SICOSAS start point.

160-A PROCEDURE

The binary tape produced at the 7090 is mounted as tape 0 low density. The SICOM interpreter is now loaded (from SICOM master paper tape) into bank 0 as follows:

1. Clear and load to read in first block of paper tape. P = 0037, A = 7715, Z = 0.
2. Clear and run to read in remainder of interpreter program. P = 4562, A = 0005, Z = 7777.
3. SICOM is now ready to accept input via PETR. Place SICOSAS loader in reader, clear and run. The loader will now be loaded and will, in turn, load and execute the SICOSAS program.

JDD:pm
APPENDIX A
SICAP INSTRUCTIONS

The basic SICAP instruction card appears

1 6 7 8 15 16 b
SYMBOL  OPCODE Y ± a, k COMMENTARY

In general the symbol (SYMBOL, above) in the location field (Cols. 1-6), if used, may contain one to six alphanumeric characters at least one of which must be non-numeric. The symbol must not contain + or – and must start in Column 1. The mnemonic operation code (OPCODE, above) in the operation field (Cols. 8-15) must start in Column 8. The variable field (starting in Col. 16 and terminating with a blank) has an address field (Y ± a, above) and may have a tag (or index) field which is separated from the address field by a comma (,). K, where used, must be a decimal integer 0 ≤ K ≤ 9. The address field may contain an additive subfield (± a, above). The start of the additive field is signaled by the + or – sign. "a" must be a decimal integer 0 ≤ a ≤ 1999. The address field (Y, above) may either be symbolic or a decimal integer 0 ≤ Y ≤ 9999. If Y is absolute, no additive subfield is permitted.

The description of each instruction given below follows the ordering of instructions of the SICOM Command List on page (iv) of the SICOM manual. Attention is further invited to Appendix B which alphabetically lists all operations and pseudo operations recognized by SICAP along with references for each operation.

SIB        Y ± a, k
Set i Base
The base component of dimension i of index register k is set to the value Y ± a.

SID        Y ± a, k
Set i Difference
The difference component of dimension i of index register k is set to the value Y ± a.

SIL        Y ± a, k
Set i Limit
The limit component of dimension i of index register k is set to the value Y ± a.

DIB        Y ± a, k
Decrement i Base
This command operates on the i dimension of index register k. The base component is decreased by the amount of the difference component. The resulting base is then compared with the limit component. If the base is equal to or larger than the limit, Y ± a is then taken as the location of the next command to be executed. If the base is smaller, the command falls through, that is, the next command in
sequence is executed. During relative mode, operation is similar except \( Y \pm a \) indicates the number of locations forward to the next command.

**IIB \( Y \pm a, k \)**  
Increment i Base

This command operates on the i dimension of index register k. The base component is increased by the amount of the difference component. The resulting base is then compared with the limit component. If the base is smaller or equal to the limit, \( Y \pm a \) is then taken as the location of the next command to be executed. If the base is larger, the command falls through, that is, the next command in sequence is executed. During the relative mode, operation is similar except that \( Y \pm a \) indicates the number of locations forward to the next command.

**SJB \( Y \pm a, k \)**  
Set j Base

The base component of dimension j of index register k is set to the value \( Y \pm a \).

**SJD \( Y \pm a, k \)**  
Set j Difference

The difference component of dimension j of index register k is set to the value \( Y \pm a \).

**SJL \( Y \pm a, k \)**  
Set j Limit

The limit component of dimension j of index register k is set to the value \( Y \pm a \).

**DJB \( Y \pm a, k \)**  
Decrement j Base

This command operates similar to "Decrement i Base" except that the j dimension is involved.

**IJB \( Y \pm a, k \)**  
Increment j Base

This command operates similar to the command "Increment i Base" except that the j dimension is involved.

**CADM \( Y \pm a, k \)**  
Clear and Add Magnitude

The absolute value of contents of effective \( Y \pm a \) is copied into the accumulator.

**CLS \( Y \pm a, k \)**  
Clear and Subtract

The contents of effective \( Y \pm a \) are copied into the accumulator and the sign reversed.

**CLA \( Y \pm a, k \)**  
Clear and Add

The contents of effective \( Y \pm a \) are copied into the accumulator.

**IDVP \( Y \pm a, k \)**  
Inverse Divide

The contents of effective \( Y \pm a \) are divided by the contents of the accumulator.
DVP $Y \pm a, k$  
**Divide**
The contents of the accumulator are divided by the contents of effective $Y \pm a$.

MPY $Y \pm a, k$  
**Multiply**
The contents of the accumulator are multiplied by the contents of effective $Y \pm a$.

EXCH $Y \pm a, k$  
**Exchange**
The contents of the accumulator and the contents of effective $Y \pm a$ are interchanged.

EXTR $Y \pm a, k$  
**Extract**
The accumulator is extracted under control of the extractor in effective $Y \pm a$. All other accumulator bits are cleared. Each decimal digit is represented by four binary bits, therefore, any accumulator bit configuration may be detected by using sexadecimal numbers in the extractor. A "Z" digit in the extractor will extract the corresponding complete digit from the accumulator. The result of the extraction is in normalized (digitally) form with the proper exponent. To extract the sign, a negative extractor must be used. (See Appendix D, MASK pseudo operation.)

ADD $Y \pm a, k$  
**Add**
The contents of effective $Y \pm a$ are added to the contents of the accumulator.

SUB $Y \pm a, k$  
**Subtract**
The contents of effective $Y \pm a$ are subtracted from the contents of the accumulator.

ADM $Y \pm a, k$  
**Add Magnitude**
The absolute value of the contents of effective $Y \pm a$ are added to the contents of the accumulator.

ISUB $Y \pm a, k$  
**Inverse Subtract**
The contents of the accumulator are subtracted from the contents of the effective $Y \pm a$. The result is in the accumulator. Effective $Y \pm a$ is unchanged.

RAD $Y \pm a, k$  
**Replace Add**
The contents of the accumulator are added to the contents of effective $Y \pm a$ and the result is stored in effective $Y \pm a$.

CAS $Y \pm a, k$  
**Compare Accumulator with Storage**
The contents of the accumulator are compared with the contents of effective $Y \pm a$. (1) If effective $Y \pm a$ is the smaller of the two, control goes to the next command in
sequence. (2) If the two are equal, control goes to the second location down, i.e., if the command is at XRAY, an equal control will go to XRAY+2. (3) If effective \( Y \pm a \) is larger, control goes to the third command down, i.e., XRAY+3. This command cannot be executed from the last three locations in a bank (i.e., 1997, 1998, and 1999) due to the multiple exits.

- **OFLT** \( Y \pm a, k \)  
  Output Floating Point and Tab
  The contents of effective \( Y \pm a \) are put out as a floating point number on the previously selected device. The format is sign, period, 10 digits, exponent, followed by a tab.

- **OFLCR** \( Y \pm a, k \)  
  Output Floating Point and Carriage Return
  The contents of effective \( Y \pm a \) are put out as a floating point number on the previously selected device. The format is sign, period, 10 digits, exponent, followed by a carriage return.

- **OFXT** \( Y \pm a, k \)  
  Output Fixed Point and Tab
  The contents of effective \( Y \pm a \) are put out as a fixed point number on the previously selected device. The format is sign, \( m \) integer digits, period, \( n \) fractional digits, followed by a tab. \( m \) and \( n \) are set by "OFORM" Command.

- **OFXC** \( Y \pm a, k \)  
  Output Fixed Point and Carriage Return
  The contents of effective \( Y \pm a \) are put out as a fixed point number on the previously selected device. The format is sign, \( m \) integer digits, period, \( n \) fractional digits, followed by a carriage return. \( m \) and \( n \) are set by "OFORM" Command.

- **OFORM** \( Y \pm a, k \)  
  Set Output Format
  If effective \( Y \pm a, k \) is represented as a decimal integer ADDR, \( 0 \leq m \leq 18 \) is set to AD digits and \( n, 0 \leq n \leq 18 \) is set to DR digits. (See OFXTB and OFXC) AD plus DR cannot exceed 18. If DR is zero, the period is not put out.

- **OCRT** \( Y \pm a, k \)  
  Output Carriage Returns and Tabs
  If effective \( Y \pm a, k \) is represented as a decimal integer ADDR, AD carriage returns followed by DR tabs are put out.

- **OTBN** \( Y \pm a, k \)  
  Output Tabulating Number
  The value effective \( Y \pm a \) is put out without a period followed by a tab. The largest number which may be put out is 9999. This command is useful for automatic numbering of lines and columns of output.

- **OCMD** \( Y \pm a, k \)  
  Output a Command from Memory
  The command in location effective \( Y \pm a \) is put out in the standard form (K OP Y ± a) followed by a carriage return.
WRFILE  $Y \pm a, k$

Write File Number on Magnetic Tape
An end of file mark and a record containing the number effective $Y \pm a$ is written on the previously-selected tape drive. Effective $Y \pm a$ is the file number identifying the file which follows. The normal exit is $L + 2$. The conditional exit ($L + 1$) is utilized in the event that end of tape has been sensed during this or a previous operation. Since the search operation assumes no files are opened after the end of tape spot, it is recommended that if the end of tape spot is sensed, the file number be erased and the tape ended (backspace 2 records and write the end of tape code number). If bad tape or a drive malfunction prevents the writing of a file number anywhere between the initial position of the tape and the end of the tape, the drive will "hang up" on the conductive trailer at the end of the tape. File numbers must be in increasing size as you proceed to the end of the tape. Zero should not be used as a file number.

SRFILE  $Y \pm a, k$

Search Magnetic Tape for File Number
A search for file number effective $Y \pm a$ is initiated on the previously-selected tape drive. Upon the conclusion of a successful search, the next command executed is the second one in sequence ($L + 2$) (normal exit). The conditional exit ($L + 1$) is used if either of the following occur:

(1) Ten attempts to read a file number have failed due to a parity error. The accumulator is set to zero prior to exiting in this case.

(2) A reflective spot was sensed during the search operation (Load Point or End of Tape). The first file number after the spot is placed in the accumulator.

The search operation starts by backspacing to and reading the first file number back. A comparison of this "tape" file number with $Y \pm a$ determines the direction of the search.

If the "tape" number is larger than effective $Y \pm a$, the search will be in the direction of the load point.

If the "tape" number is smaller than $Y \pm a$ or if the load point is sensed, the search will be in the direction of the end of the tape. If a subsequent file number indicates that the direction should be changed again, the standard error indication of 0047 is given. Cycling the run switch after this indication will cause the search to restart.

Example: The search for 101 among consecutive numbers of 98, 100, and 103 would cause error 0047.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRPIN Y ± a, k</td>
<td>Group Input</td>
</tr>
<tr>
<td>SNGIN Y ± a, k</td>
<td>Single Input</td>
</tr>
<tr>
<td>IOCTAL Y ± a, k</td>
<td>Input Octal Tape</td>
</tr>
<tr>
<td>OALPHA Y ± a, k</td>
<td>Output Alphanumeric Data</td>
</tr>
<tr>
<td>LDANR Y ± a, k</td>
<td>Load Alpha Numeric Register (A. N. R.)</td>
</tr>
<tr>
<td>CMPANR Y ± a, k</td>
<td>Compare Alpha Numeric Register (A. N. R.)</td>
</tr>
<tr>
<td>MRGANR Y ± a, k</td>
<td>Merge into Alpha Numeric Register (A. N. R.)</td>
</tr>
<tr>
<td>EXTANR Y ± a, k</td>
<td>Extract Alpha Numeric Register (A. N. R.)</td>
</tr>
<tr>
<td>TZE Y ± a, k</td>
<td>Transfer on Accumulator Zero</td>
</tr>
<tr>
<td>TNZ Y ± a, k</td>
<td>Transfer on Accumulator Non-Zero</td>
</tr>
<tr>
<td>TPL Y ± a, k</td>
<td>Transfer on Accumulator Plus</td>
</tr>
</tbody>
</table>

- **Group Input**: Input is transferred to effective Y ± a and consecutive locations. (See SICOM Manual, paragraph II. B. 6, p. 14, "Normal Input".)
- **Single Input**: Input is transferred only to location effective Y ± a. One word of any type of input may be entered. The input is terminated by either a carriage return or a tab.
- **Input Octal Tape**: Tape punched by "OCTAL" Command is read into the receiving field starting at effective Y ± a. The field must fall entirely within a bank. The location of the receiving field may differ from the original source field.
- **Output Alphanumeric Data**: The contents of address effective Y ± a and effective Y ± a + 1 are put out as 8 alphanumeric characters on the previously-selected device. This instruction is not affected by the relative mode.
- **Load Alpha Numeric Register (A. N. R.)**: The contents of effective Y ± a are loaded into A. N. R. as alphanumeric data. (See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")
- **Compare Alpha Numeric Register (A. N. R.)**: (See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")
- **Merge into Alpha Numeric Register (A. N. R.)**: (See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")
- **Extract Alpha Numeric Register (A. N. R.)**: (See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")
- **Transfer on Accumulator Zero**: If the contents of the accumulator is zero, a mark 60 jump to effective Y ± a occurs. Otherwise, the normal sequence is continued.
- **Transfer on Accumulator Non-Zero**: If the contents of the accumulator is not zero, a mark 61 jump to effective Y ± a occurs. Otherwise, normal sequence continues.
- **Transfer on Accumulator Plus**: If the accumulator contains a positive quantity (including zero), a mark 62 jump to effective Y ± a occurs. Otherwise, normal sequence continues.
TMI Y ± a, k  Transfer on Accumulator Minus
If the accumulator contains a negative quantity, a mark 63 jump to effective Y ± a occurs. Otherwise, normal sequence continues.

TSM4 Y ± a, k  Transfer and Set Mark 4
The next command in sequence is marked for a "return to mark 4" return. Control goes to effective Y ± a.

TSM5 Y ± a, k  Transfer and Set Mark 5
The next command in sequence is marked for a "return to mark 5" return. Control goes to effective Y ± a.

TSM6 Y ± a, k  Transfer and Set Mark 6
The next command in sequence is marked for a "return to mark 6" return. Control goes to effective Y ± a.

TNR7 Y ± a, k  Transfer Non-Relative and Set Mark 7
The next command in sequence is marked for a "return to mark 7" return. Control goes to effective Y ± a. This command is not affected by the relative mode. This makes a jump from relative subroutine to fixed area possible.

TSR Y ± a, k  Transfer to Subroutine
This command operates the same as other unconditional transfer commands. It is also used to enter standard SICOM subroutines, thereby disturbing any 70 mark which may have been set previously.

TSJ1 Y ± a, k  Transfer on Selective Jump Switch 1 On
If selective jump switch number one is ON, the next consecutive command is marked for a return to selective jump one (RSJ1) return and control goes to the location effective Y ± a. If the jump switch is not ON, the normal sequence is continued.

TSJ2 Y ± a, k  Transfer on Selective Jump Switch 2 On
This command operates similar to the one above except that jump switch two is used and the return is "return to selective jump two" (RSJ2).

TSJ3 Y ± a, k  Transfer on Selective Jump Switch 1 and 2 Both On
If both jump switch one and two are ON, the next consecutive command is marked for a return to "return to selective jump three (RSJ3)". Control goes to effective Y ± a. If either jump switch if OFF, the next command in sequence is executed.

CALL Y ± a, k  Transfer to Machine Language Subroutine
Control is transferred to the machine language subroutine which starts at effective location Y ± a.
TBR  $Y \pm a, k$

Transfer Back Relative
Control goes backwards $Y \pm a$ effective locations regardless of the addressing mode. No mark is set.

STOANR  $Y \pm a, k$

Store A. N. R.
The contents of A. N. R. are copied into effective $Y \pm a$ and effective $Y \pm a + 1$ as alphanumeric data. A. N. R. is unchanged.

STO  $Y \pm a, k$

Store Accumulator
The contents of the accumulator are copied into effective $Y \pm a$ and effective $Y \pm a + 1$. The accumulator remains unchanged.

NOP

No Operation
This command may be used to reserve a location to be filled conditionally. No operation is performed. The computer proceeds to the next location.

STOP1

Selective Stop 1
If selective stop switch 1 is on, computation stops with Panel display $A = 0001$. Resume computation by cycling run switch. If the switch is off, the command has no effect. Computation continues in normal sequence regardless of switch setting.

STOP2

Selective Stop 2
Same as Selective Stop 1 except under control of Switch 2, Panel Display: $A = 0002$.

HALT3

Stop Display 3
Computation stops with Panel Display: $A = 0003$. It may be resumed at the next sequential location by cycling the run switch.

HALT4

Stop Display 4
Same as Stop Display 3 only $A = 0004$.

HPRIN

Halt and Await Input
If Flexowriter input has been selected, computation stops with $A = 0005$ and input is when the run switch is cycled. If typewriter input has been selected, input is gated. (See SICOM Manual, paragraph IV. a, p. 71, "Operating Modes.")

REL

Select Relative Mode
All subsequent commands are interpreted in the relative addressing mode. Termination is by execution of a "SELECT ABSOLUTE MODE" (ABS), a master clear and run or "Halt and Await Input".

25
ABS  Select Absolute Mode
Computation returns to the normal absolute addressing mode.

FAST  Select Non-Trace Mode
Tracing is discontinued. This command is normally used to avoid tracing through SICOM subroutines. During non-trace operation this command is the same as a "NO OPERATION". This also discontinues the step mode. Both operations are restored upon execution of "Reset Trace" or a master clear and run.

TRACE  Reset Trace Mode
Restores the trace and step modes which were in effect prior to the execution of a "SELECT NON-TRACE" (FAST). This command is normally placed at the end of a SICOM subroutine.

ZEROIR  Clear Index Registers
This command sets all components of all index registers to zero.

IFLEX  Select Flexowriter Input
All subsequent normal input commands will be via Flexowriter tape.

OFLEX  Select Flexowriter Output
All subsequent normal output will be punched on Flexowriter tape.

ITYPE  Select Typewriter Input
All subsequent normal input commands will be via the typewriter.

OTYPE  Select Typewriter Output
All subsequent normal output will be printed via the typewriter.

OPRINT  Select Printer
(Not Available.)

SELBCD  Select Tape to Printer
All subsequent normal output is put on magnetic tape drive 1 suitable for off line magnetic tape to printer listing. Only tape drive 1 and print channel 0 can be used. Each line to be printed is recorded as a 120 Binary Coded Decimal character record.

SELBIN  $Y \pm a, k$
Select Magnetic Tape Drive $Y \pm a, k$
If $k = 0$, the $j$ base of index $k$ is added to $Y \pm a$. This new value of $Y \pm a$ (or the original value if $k = 0$) is the number of the drive which will be used in all subsequent
tape operations. The accumulator is not disturbed. This command automatically selects odd parity. \( Y \pm a \) may have values of 1, 2, 3, or 4. If the power is off on drive \( Y \pm a \), or if no drive has been set to \( Y \pm a \), the computer hangs up with the select light on. The execution of 0 00 0017 "Select Tape to Printer" SELBCD voids all prior magnetic tape selections.

BSR \( Y \pm a \)  
Backspace \( Y \pm a \) Records  
If \( Y \pm a \neq 0 \), the previously-selected drive will be backspaced \( Y \pm a \) records. Note that an end of file mark and file numbers are each considered a record, and that no indication of load point or end of tape is given. \( Y \pm a \) may have values up to 99. Backspacing beyond the load point will cause the drive to "hang up" on the conductive leader.

REW  
Rewind  
The previously-selected drive will rewind to the load point. A rewind from the load point has no effect.

OTAPE \( 0, k \)  
Write on Magnetic Tape  
The field defined by index register \( k \) (i Base = 1st location, i Difference = length) is copied on the previously-selected tape drive as one record. The accumulator is not disturbed by either the normal (L + 2) or the conditional (L + 1) exit. The conditional exit indicates that the end of tape spot was sensed on this or a previous command.

If a parity error occurs during a write operation, the record is erased, 6 inches of tape skipped and the record is re-written. This process continues until the record is properly written or the conductive leader is reached. Due to the double exit, this command cannot be executed from the last two locations in a bank, i.e., 1998 and 1999.

ITAPE \( 0, k \)  
Read Magnetic Tape  
One record of information is read from the previously-selected tape drive. It is read into the field defined by index register \( K \) (i Base = starting location, i Difference = the length). If the tape record is longer than the field, the remainder of the record will be read but not stored in memory.

If the record is shorter than the field, all of the remainder of the field will be unchanged except the location immediately following the end of the record. This location will be altered. The normal exit (L + 2) does not disturb the accumulator. The conditional exit (L + 1) is utilized for the following two conditions:
(1) Ten attempts to read the record are unsuccessful due to a parity error. The accumulator is set to zero prior to exiting in this case. The record is stored in memory as it was read on the last attempt.

(2) The end of file was read. In this case, the next file number is placed into the accumulator ready for possible test to determine if the end of the tape has been reached. No end of tape indication as such is given.

An attempt to read beyond the last record on the tape will result in the drive "hanging up" on the conductive trailer. Due to the double exit, this command cannot be executed from the last two locations in a bank, i.e., 1998 and 1999.

**BLCPY** 0, k  
Block Copy

The source field defined by index register k is copied into a receiving field which starts at the address specified by the j base of index k. If the two fields overlap, the j base must be smaller than the i base. The two fields may be in different banks. (See SICOM Manual, paragraph II. B. 19, p. 30, "Block Operations."

**BLCLR** 0, k  
Block Clear

The field defined by index register k is cleared to zero. (See SICOM Manual, paragraph II. B. 19, p. 30, "Block Operations.")

**OOCTAL** 0, k  
Output Octal Tape

The field defined by index register k is punched, in bi-octal form, followed by two blank frames and the check sum (two frames). This tape may be read with the IOCTAL command. A one-inch trailer is also punched.

**OSPEC** 0, k  
Output Special Tape

The field defined by index register k is punched in bi-octal form in the format shown on page 31 of the SICOM Manual.

**ISPEC**  
Input Special Tape

Reads tape punched by the command OSPEC. Reading stops when a leader over 6" long is detected.

**RZE**  
Return to TZE

Control is returned to the location +1 of the most recently executed transfer on accumulator zero (TZE) instruction.

**RNZ**  
Return to TNZ

Control is returned to the location +1 of the most recently executed transfer on accumulator non-zero (TNZ) instruction.
RPL Return to TPL
Control is returned to the location +1 of the most recently executed transfer on accumulator positive (TPL) instruction.

RMI Return to TMI
Control is returned to the location +1 of the most recently executed transfer on accumulator negative (TMI) instruction.

RSM4 Return to TSM4
Control is returned to the location +1 of the most recently executed transfer and set Mark 4 (TSM4) instruction.

RSM5 Return to TSM5
Control is returned to the location +1 of the most recently executed transfer and set Mark 5 (TSM5) instruction.

RSM6 Return to TSM6
Control is returned to the location +1 of the most recently executed transfer and set Mark 6 (TSM6) instruction.

RNR7 Return to TNR7
Control is returned to the location +1 of the most recently executed transfer non-relative and set Mark 7 (TNR7) command.

RSR Return to TSR
Control is returned to the location +1 of the most recently executed transfer to subroutine (TRS) instruction.

RSJ1 Return to TSJ1
Control is returned to the location +1 of the most recently executed transfer if selective jump switch one is on (TSJ1) command.

RSJ2 Return to TSJ2
Control is returned to the location +1 of the most recently executed transfer if selective jump switch two is on (TSJ2) command.

RSJ3 Return to TSJ3
Control is returned to the location +1 of the most recently executed transfer if selective jump switch three is on (TSJ3) command.

OLOC Output Last Location
Location of last command executed is put out on the previously-selected device.
PUSSTOP

Punch Stop, Check and Leader

The following are punched on tape:

1. A stop code,

2. The check sum of all normal output punched since the last "Halt Select Manual Mode" or since the last "Punch Stop, Check, and Leader."

3. A one-inch leader.

This is useful for Flexowriter output exclusively.

CLAC* \( Y \pm a \) Clear and Add Constant

The value \( Y \pm a \) is loaded into the accumulator. The \( Y \pm a \) of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

CLSC* \( Y \pm a \) Clear and Subtract Constant

The value minus \( Y \pm a \) is loaded into the accumulator. The \( Y \pm a \) of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

IDVPC* \( Y \pm a \) Divide into Constant

The contents of the accumulator are divided into the value \( Y \pm a \). The \( Y \pm a \) of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

DVPC* \( Y \pm a \) Divide by Constant

The accumulator is divided by the value \( Y \pm a \). The \( Y \pm a \) of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

MPYC* \( Y \pm a \) Multiply by Constant

The accumulator is multiplied by the value \( Y \pm a \). The \( Y \pm a \) of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

ADDC* \( Y \pm a \) Add Constant

The value \( Y \pm a \) is added to the accumulator. The \( Y \pm a \) of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

SUBC* \( Y \pm a \) Subtract Constant

The value \( Y \pm a \) is subtracted from the accumulator. The \( Y \pm a \) of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

*The variable field must not contain a decimal point (').
ISUBC* Y ± a  Subtract from Constant

The accumulator is subtracted from the value Y ± a. The Y ± a of these commands is not an address but instead a numeric value between 0 and 9999. It cannot be a fraction. The result is always in the accumulator.

SHIFT* Y ± a  Shift Accumulator

If Y ± a is represented as a decimal integer ADDR, the accumulator is multiplied by 10 to the AD power (left shift) and divided by 10 to the DR power (right shift).

SETFWA Y ± a  Set First Address for Trace

An internal switch is set so that during the trace mode tracing starts after the execution of the command in location Y ± a.

SETLWA Y ± a  Set Last Address for Trace

The address at which tracing will terminate is set to Y ± a.

PXAI B 0, k  Place Index k (i Base) in Accumulator

The i base component of index register k replaces the contents of the accumulator as a positive floating point number.

PXAI D 0, k  Place Index k (i Difference) in Accumulator

The i difference component of index register k replaces the contents of the accumulator as a positive floating point number.

PXAI L 0, k  Place Index k (i Limit) in Accumulator

The i limit component of index register k replaces the contents of the accumulator as a positive floating point number.

PXAJ B 0, k  Place Index k (j Base) in Accumulator

The j base component of index register k replaces the contents of the accumulator as a positive floating point number.

PXAJ D 0, k  Place Index k (j Difference) in Accumulator

The j difference component of index register k replaces the contents of the accumulator as a positive floating point number.

PXAJ L 0, k  Place Index k (j Limit) in Accumulator

The j limit component of index register k replaces the contents of the accumulator as a positive floating point number.

*The variable field must not contain a decimal point (·).
PAXIB 0, k Place the Accumulator in the i Base Component of Index Register k

The integral portion of the accumulator is copied into the base component of index register k. The accumulator is unchanged.

PAXID 0, k Place the Accumulator in the i Difference Component of Index Register k

The integral portion of the accumulator is copied into the i difference component of index register k. The accumulator is unchanged.

PAXIL 0, k Place the Accumulator in the i Limit Component of Index Register k

The integral portion of the accumulator is copied into the i limit component of index register k. The accumulator is unchanged.

PAXJB 0, k Place the Accumulator in the j Base Component of Index Register k

The integral portion of the accumulator is copied into the j base component of index register k. The accumulator is unchanged.

PAXJD 0, k Place the Accumulator in the j Difference Component of Index Register k

The integral portion of the accumulator is copied into the j difference component of index register k. The accumulator is unchanged.

PAXJL 0, k Place the Accumulator in the j Limit Component of Index Register k

The integral portion of the accumulator is copied into the j limit component of index register k. The accumulator is unchanged.

XEQ Y ± a Execute

Automatic computation is begun starting at effective location Y ± a under the guidance of the SICOM interpreter.
APPENDIX B

Below are listed alphabetically all of the mnemonics recognized by SICAP. The + notation for the pseudo operations refers the reader to the section entitled SICOSAS SPECIAL PURPOSE CARDS. Where applicable, the original seven-digit SICOM numeric format is included as well as a page reference to the SICOM manual. Finally, for each mnemonic operation, a page in Appendix A is referenced where a complete description is given.

<table>
<thead>
<tr>
<th>SICAP MNEMONIC</th>
<th>SICOM FORMAT</th>
<th>SICOM MANUAL PAGE #</th>
<th>SICOSAS APPENDIX A PAGE #</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>0 00 0007</td>
<td>24</td>
<td>26</td>
</tr>
<tr>
<td>ADD</td>
<td>*K 30 ADDR</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>ADDC</td>
<td>6 01 ADDR</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>ADM</td>
<td>*K 32 ADDR</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>ATAN</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>BLCLR</td>
<td>K 00 0025</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>BLCPY</td>
<td>K 00 0024</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>BSR</td>
<td>0 00 AD19</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>BSS</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>CADM</td>
<td>*K 20 ADDR</td>
<td>09</td>
<td>19</td>
</tr>
<tr>
<td>CALL</td>
<td>*K 74 ADDR</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>CAS</td>
<td>*K 35 ADDR</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>CLA</td>
<td>*K 22 ADDR</td>
<td>09</td>
<td>19</td>
</tr>
<tr>
<td>CLAC</td>
<td>1 01 ADDR</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>CLS</td>
<td>*K 31 ADDR</td>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td>CLSC</td>
<td>2 01 ADDR</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>CMPANR</td>
<td>*K 55 ADDR</td>
<td>18</td>
<td>23</td>
</tr>
<tr>
<td>DEC</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>DIB</td>
<td>*K 05 ADDR</td>
<td>07</td>
<td>18</td>
</tr>
<tr>
<td>DJB</td>
<td>*K 15 ADDR</td>
<td>08</td>
<td>19</td>
</tr>
<tr>
<td>DVP</td>
<td>*K 24 ADDR</td>
<td>09</td>
<td>20</td>
</tr>
<tr>
<td>DVPC</td>
<td>4 01 ADDR</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>EQU</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>EXCH</td>
<td>*K 26 ADDR</td>
<td>09</td>
<td>20</td>
</tr>
<tr>
<td>EXP</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>EXTANR</td>
<td>*K 57 ADDR</td>
<td>19</td>
<td>23</td>
</tr>
<tr>
<td>EXTR</td>
<td>*K 27 ADDR</td>
<td>09</td>
<td>20</td>
</tr>
<tr>
<td>FAST</td>
<td>0 00 0008</td>
<td>24</td>
<td>26</td>
</tr>
<tr>
<td>GRPIN</td>
<td>K 50 ADDR</td>
<td>15</td>
<td>23</td>
</tr>
<tr>
<td>HALT3</td>
<td>0 00 0003</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>HALT4</td>
<td>0 00 0004</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>HPRIN</td>
<td>0 00 0005</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>IDVP</td>
<td>*K 23 ADDR</td>
<td>09</td>
<td>19</td>
</tr>
<tr>
<td>IDVPC</td>
<td>3 01 ADDR</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>IFLEX</td>
<td>0 00 0011</td>
<td>25</td>
<td>26</td>
</tr>
</tbody>
</table>

*Affected by relative mode.*
<table>
<thead>
<tr>
<th>SICAP</th>
<th>SICOM FORMAT</th>
<th>SICOM MANUAL PAGE #</th>
<th>SICOSAS APPENDIX A PAGE #</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIB    *K 06 ADDR</td>
<td>08</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>IJB    *K 16 ADDR</td>
<td>08</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>IOCTAL K 52 ADDR</td>
<td>17</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>ISPEC 0 00 0029</td>
<td>31</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>ISUB    *K 33 ADDR</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>ISUBC   8 01 ADDR</td>
<td>34</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>ITAPE   K 00 0021</td>
<td>28</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>ITYPE   0 00 0013 25A</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDANR   *K 54 ADDR</td>
<td>18</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>LOG                          +</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MASK                          +</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLAP                          +</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPY    *K 25 ADDR</td>
<td>09</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>MPYC   5 01 ADDR</td>
<td>33</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>MRGANR *K 56 ADDR</td>
<td>19</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>NAME              +</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP    0 00 0000</td>
<td>23</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>OALPHA K 53 ADDR</td>
<td>17</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>OCMND   K 45 ADDR</td>
<td>12</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OCRRTB  K 43 ADDR</td>
<td>12</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OFLCR   K 37 ADDR</td>
<td>11</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OFLEX   0 00 0012 25A</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OFLTB   K 36 ADDR</td>
<td>11</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OFORM   K 42 ADDR</td>
<td>12</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OFXCR   K 41 ADDR</td>
<td>11</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OFXTB   K 40 ADDR</td>
<td>11</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OLOC    0 00 0075</td>
<td>32</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>OOCTAL  K 00 0027</td>
<td>30</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>OPRINT  0 00 0016 25A</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORG                          +</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSPEC  K 00 0028</td>
<td>31</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>OTAPE  K 00 0020</td>
<td>27</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>OTBNO   K 44 ADDR</td>
<td>12</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OTYPE   0 00 0014 25A</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAXIB   K 00 0223</td>
<td>29</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>PAXID   K 00 0323</td>
<td>29</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>PAXIL   K 00 0423</td>
<td>29</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>PAXJB   K 00 1223</td>
<td>29</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>PAXJD   K 00 1323</td>
<td>29</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>PAXJL   K 00 1423</td>
<td>29</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>PUSTOP  0 00 0079</td>
<td>32</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>PXAIB   K 00 0222</td>
<td>29</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>PXAID   K 00 0322</td>
<td>29</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>PXAIL   K 00 0422</td>
<td>29</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>PXAJB   K 00 1222</td>
<td>29</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>PXAJD   K 00 1322</td>
<td>29</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>PXAJL   K 00 1422</td>
<td>29</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>RAD    *K 34 ADDR</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>REL    0 00 0006</td>
<td>24</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>

* Affected by relative mode.
<table>
<thead>
<tr>
<th>SICAP MNEMONIC</th>
<th>SICOM FORMAT</th>
<th>SICOM MANUAL PAGE.#</th>
<th>SICOSAS APPENDIX A PAGE.#</th>
</tr>
</thead>
<tbody>
<tr>
<td>REW</td>
<td>0 00 0019</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>RMI</td>
<td>0 00 0063</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RNR7</td>
<td>0 00 0067</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RNZ</td>
<td>0 00 0061</td>
<td>31</td>
<td>28</td>
</tr>
<tr>
<td>RPL</td>
<td>0 00 0062</td>
<td>31</td>
<td>29</td>
</tr>
<tr>
<td>RSJ1</td>
<td>0 00 0071</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RSJ2</td>
<td>0 00 0072</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RSJ3</td>
<td>0 00 0073</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RSM4</td>
<td>0 00 0064</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RSM5</td>
<td>0 00 0065</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RSM6</td>
<td>0 00 0066</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RSR</td>
<td>0 00 0070</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>RZE</td>
<td>0 00 0060</td>
<td>31</td>
<td>28</td>
</tr>
<tr>
<td>SELBCD</td>
<td>0 00 0017</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>SELBIN</td>
<td>K 00 0018</td>
<td>27</td>
<td>26</td>
</tr>
<tr>
<td>SETFWA</td>
<td>Y 01 ADDR</td>
<td>34</td>
<td>31</td>
</tr>
<tr>
<td>SETLWA</td>
<td>Z 01 ADDR</td>
<td>34</td>
<td>31</td>
</tr>
<tr>
<td>SHIFT</td>
<td>9 01 ADDR</td>
<td>34</td>
<td>31</td>
</tr>
<tr>
<td>SIB</td>
<td>K 02 ADDR</td>
<td>07</td>
<td>18</td>
</tr>
<tr>
<td>SICAP</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SID</td>
<td>K 03 ADDR</td>
<td>07</td>
<td>18</td>
</tr>
<tr>
<td>SIL</td>
<td>K 04 ADDR</td>
<td>07</td>
<td>18</td>
</tr>
<tr>
<td>SIN</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SJB</td>
<td>K 12 ADDR</td>
<td>08</td>
<td>19</td>
</tr>
<tr>
<td>SJD</td>
<td>K 13 ADDR</td>
<td>08</td>
<td>19</td>
</tr>
<tr>
<td>SJL</td>
<td>K 14 ADDR</td>
<td>08</td>
<td>19</td>
</tr>
<tr>
<td>SNGIN</td>
<td>K 51 ADDR</td>
<td>17</td>
<td>23</td>
</tr>
<tr>
<td>SQR</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRFILE</td>
<td>K 47 ADDR</td>
<td>13</td>
<td>22</td>
</tr>
<tr>
<td>STO</td>
<td>*K 77 ADDR</td>
<td>22</td>
<td>25</td>
</tr>
<tr>
<td>STOANR</td>
<td>*K 76 ADDR</td>
<td>22</td>
<td>25</td>
</tr>
<tr>
<td>STOP1</td>
<td>0 00 0001</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>STOP2</td>
<td>0 00 0002</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>SUB</td>
<td>*K 31 ADDR</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>SUBC</td>
<td>7 01 ADDR</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>TBR</td>
<td>K 75 ADDR</td>
<td>22</td>
<td>25</td>
</tr>
<tr>
<td>TMI</td>
<td>*K 63 ADDR</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>TNR7</td>
<td>K 67 ADDR</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>TNZ</td>
<td>*K 61 ADDR</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>TPL</td>
<td>*K 62 ADDR</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>TRACE</td>
<td>0 00 0009</td>
<td>24</td>
<td>26</td>
</tr>
<tr>
<td>TSJ1</td>
<td>*K 71 ADDR</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>TSJ2</td>
<td>*K 72 ADDR</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>TSJ3</td>
<td>*K 73 ADDR</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>TSM4</td>
<td>*K 64 ADDR</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>TSM5</td>
<td>*K 65 ADDR</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>TSM6</td>
<td>*K 66 ADDR</td>
<td>21</td>
<td>24</td>
</tr>
</tbody>
</table>

*Affected by relative mode.
<table>
<thead>
<tr>
<th>SICAP MNEMONIC</th>
<th>SICOM FORMAT</th>
<th>SICOM MANUAL PAGE #</th>
<th>SICOSAS APPENDIX A PAGE #</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSR</td>
<td>K 70 ADDR</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>TZE</td>
<td>*/K 60 ADDR</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>WRFILE</td>
<td>K 46 ADDR</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>XEQ</td>
<td>X 01 ADDR</td>
<td>34</td>
<td>32</td>
</tr>
<tr>
<td>ZEROIR</td>
<td>0 00 0010</td>
<td>25</td>
<td>26</td>
</tr>
</tbody>
</table>

*(Col. 1)

*Affected by relative mode.
APPENDIX C

Below is an alphabetized list of mnemonic operation codes and pseudo operation codes recognized by MLAP (Machine Language Assembly Program).

The referenced page number for the machine instructions is to Control Data 160-A Computer, Programming Manual, # 145A.

The * notation for the "relative" instructions refers the reader to page 27 of CDC Publication # 507 OSAS/A, The 160-A Assembly System.

The + notation for the pseudo operations recognized by MLAP refers the reader to the section entitled SICOSAS SPECIAL PURPOSE CARDS.

In the Y ± a notation, Y may be either symbolic or absolute. If absolute, Y will be considered either decimal or octal depending on the current mode of MLAP. (See MLAP card in Appendix D.) The additive field "a" may be used only if Y is symbolic and must be an absolute integer either octal or decimal again depending on the current mode of MLAP.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACJ</td>
<td>Y ± a</td>
<td>Set direct, indirect, and relative bank control and jump 36</td>
</tr>
<tr>
<td>ADB</td>
<td>Y ± a</td>
<td>Add backward. 30</td>
</tr>
<tr>
<td>ADC</td>
<td>Y ± a</td>
<td>Add constant. 30</td>
</tr>
<tr>
<td>ADD</td>
<td>Y ± a</td>
<td>Add direct. 30</td>
</tr>
<tr>
<td>ADF</td>
<td>Y ± a</td>
<td>Add forward. 30</td>
</tr>
<tr>
<td>ADI</td>
<td>Y ± a</td>
<td>Add indirect. 30</td>
</tr>
<tr>
<td>ADM</td>
<td>Y ± a</td>
<td>Add memory. 30</td>
</tr>
<tr>
<td>ADN</td>
<td>Y ± a</td>
<td>Add no address. 30</td>
</tr>
<tr>
<td>ADR</td>
<td>Y ± a</td>
<td>Add relative. *</td>
</tr>
<tr>
<td>ADS</td>
<td></td>
<td>Add specific. 30</td>
</tr>
<tr>
<td>AOB</td>
<td>Y ± a</td>
<td>Replace add one backward. 32</td>
</tr>
<tr>
<td>AOC</td>
<td>Y ± a</td>
<td>Replace add one constant. 32</td>
</tr>
<tr>
<td>AOD</td>
<td>Y ± a</td>
<td>Replace add one direct. 32</td>
</tr>
<tr>
<td>AOF</td>
<td>Y ± a</td>
<td>Replace add one forward. 32</td>
</tr>
<tr>
<td>AOI</td>
<td>Y ± a</td>
<td>Replace add one indirect. 32</td>
</tr>
<tr>
<td>AOM</td>
<td>Y ± a</td>
<td>Replace add one memory. 32</td>
</tr>
<tr>
<td>AOR</td>
<td>Y ± a</td>
<td>Replace add one relative. *</td>
</tr>
<tr>
<td>AOS</td>
<td></td>
<td>Replace add one specific. 32</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>ATE</td>
<td>Y ± a</td>
<td>A to buffer entrance register.</td>
</tr>
<tr>
<td>ATX</td>
<td>Y ± a</td>
<td>A to buffer exit register.</td>
</tr>
<tr>
<td>BLS</td>
<td>Y ± a</td>
<td>Block store.</td>
</tr>
<tr>
<td>BSS</td>
<td>Y ± a</td>
<td>Block starting with symbol.</td>
</tr>
<tr>
<td>CBC</td>
<td>Y ± a</td>
<td>Clear buffer controls</td>
</tr>
<tr>
<td>CIL</td>
<td>Y ± a</td>
<td>Clear interrupt lockout.</td>
</tr>
<tr>
<td>CTA</td>
<td>Y ± a</td>
<td>Bank controls to A.</td>
</tr>
<tr>
<td>DRJ</td>
<td>Y ± a</td>
<td>Set direct and relative bank control and jump.</td>
</tr>
<tr>
<td>END</td>
<td>Y ± a</td>
<td>End compilation</td>
</tr>
<tr>
<td>EQU</td>
<td>Y ± a</td>
<td>Equality</td>
</tr>
<tr>
<td>ETA</td>
<td>Y ± a</td>
<td>Buffer entrance register to A</td>
</tr>
<tr>
<td>ERR</td>
<td>Y ± a</td>
<td>Error stop.</td>
</tr>
<tr>
<td>EXC</td>
<td>Y ± a</td>
<td>External function constant.</td>
</tr>
<tr>
<td>EXF</td>
<td>Y ± a</td>
<td>External function forward.</td>
</tr>
<tr>
<td>HLT</td>
<td>Y ± a</td>
<td>Halt</td>
</tr>
<tr>
<td>HWI</td>
<td>Y ± a</td>
<td>Half write indirect.</td>
</tr>
<tr>
<td>IBI</td>
<td>Y ± a</td>
<td>Initiate buffer input.</td>
</tr>
<tr>
<td>IBO</td>
<td>Y ± a</td>
<td>Initiate buffer output.</td>
</tr>
<tr>
<td>INA</td>
<td>Y ± a</td>
<td>Input to A.</td>
</tr>
<tr>
<td>IRJ</td>
<td>Y ± a</td>
<td>Set indirect and relative bank control and jump.</td>
</tr>
<tr>
<td>JFI</td>
<td>Y ± a</td>
<td>Jump forward indirect.</td>
</tr>
<tr>
<td>JPI</td>
<td>Y ± a</td>
<td>Jump indirect.</td>
</tr>
<tr>
<td>JPR</td>
<td>Y ± a</td>
<td>Return jump.</td>
</tr>
<tr>
<td>LCB</td>
<td>Y ± a</td>
<td>Load complement backward.</td>
</tr>
<tr>
<td>LCC</td>
<td>Y ± a</td>
<td>Load complement constant.</td>
</tr>
<tr>
<td>LCD</td>
<td>Y ± a</td>
<td>Load complement direct.</td>
</tr>
<tr>
<td>LCF</td>
<td>Y ± a</td>
<td>Load complement forward.</td>
</tr>
<tr>
<td>LCI</td>
<td>Y ± a</td>
<td>Load complement indirect.</td>
</tr>
<tr>
<td>LCM</td>
<td>Y ± a</td>
<td>Load complement memory.</td>
</tr>
<tr>
<td>LCN</td>
<td>Y ± a</td>
<td>Load complement no address.</td>
</tr>
<tr>
<td>LCR</td>
<td>Y ± a</td>
<td>Load complement relative.</td>
</tr>
<tr>
<td>Instruction</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>LCS</td>
<td>Load complement specific.</td>
<td>28</td>
</tr>
<tr>
<td>LDB</td>
<td>Y ± a Load backward.</td>
<td>27</td>
</tr>
<tr>
<td>LDC</td>
<td>Y ± a Load constant.</td>
<td>27</td>
</tr>
<tr>
<td>LDD</td>
<td>Y ± a Load direct.</td>
<td>27</td>
</tr>
<tr>
<td>LDF</td>
<td>Y ± a Load forward.</td>
<td>27</td>
</tr>
<tr>
<td>LDI</td>
<td>Y ± a Load indirect.</td>
<td>27</td>
</tr>
<tr>
<td>LDM</td>
<td>Y ± a Load memory.</td>
<td>27</td>
</tr>
<tr>
<td>LDN</td>
<td>Y ± a Load no address.</td>
<td>27</td>
</tr>
<tr>
<td>LDR</td>
<td>Y ± a Load relative.</td>
<td>*</td>
</tr>
<tr>
<td>LDS</td>
<td>Load specific.</td>
<td>27</td>
</tr>
<tr>
<td>LPB</td>
<td>Y ± a Logical product backward.</td>
<td>34</td>
</tr>
<tr>
<td>LPC</td>
<td>Y ± a Logical product constant.</td>
<td>34</td>
</tr>
<tr>
<td>LPD</td>
<td>Y ± a Logical product direct.</td>
<td>34</td>
</tr>
<tr>
<td>LPF</td>
<td>Y ± a Logical product forward.</td>
<td>34</td>
</tr>
<tr>
<td>LPI</td>
<td>Y ± a Logical product indirect.</td>
<td>34</td>
</tr>
<tr>
<td>LPM</td>
<td>Y ± a Logical product memory.</td>
<td>34</td>
</tr>
<tr>
<td>LPN</td>
<td>Y ± a Logical product no address.</td>
<td>34</td>
</tr>
<tr>
<td>LPR</td>
<td>Y ± a Logical product relative</td>
<td>*</td>
</tr>
<tr>
<td>LPS</td>
<td>Logical product specific.</td>
<td>34</td>
</tr>
<tr>
<td>LSB</td>
<td>Y ± a (SCB) Logical sum backward.</td>
<td>35</td>
</tr>
<tr>
<td>LSD</td>
<td>Y ± a (SCD) Logical sum direct.</td>
<td>35</td>
</tr>
<tr>
<td>LSF</td>
<td>Y ± a (SCF) Logical sum forward.</td>
<td>35</td>
</tr>
<tr>
<td>LSI</td>
<td>Y ± a (SCI) Logical sum indirect.</td>
<td>35</td>
</tr>
<tr>
<td>LSN</td>
<td>Y ± a (SCN) Logical sum no address.</td>
<td>35</td>
</tr>
<tr>
<td>LSR</td>
<td>Y ± a (SCR) Logical sum relative.</td>
<td>*</td>
</tr>
<tr>
<td>LS1</td>
<td>Left shift one.</td>
<td>32</td>
</tr>
<tr>
<td>LS2</td>
<td>Left shift two.</td>
<td>32</td>
</tr>
<tr>
<td>LS3</td>
<td>Left shift three.</td>
<td>32</td>
</tr>
<tr>
<td>LS6</td>
<td>Left shift six.</td>
<td>32</td>
</tr>
<tr>
<td>MLAP</td>
<td>Set SICOSAS to MLAP mode.</td>
<td>+</td>
</tr>
<tr>
<td>MUH</td>
<td>Multiply A by one hundred.</td>
<td>30</td>
</tr>
<tr>
<td>MUT</td>
<td>Multiply A by ten.</td>
<td>29</td>
</tr>
<tr>
<td>NAME</td>
<td>(Name card)</td>
<td>+</td>
</tr>
<tr>
<td>NJB</td>
<td>Y ± a Negative jump backward.</td>
<td>37</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>NJF</td>
<td>Negative jump forward.</td>
<td>37</td>
</tr>
<tr>
<td>NJR</td>
<td>Negative jump relative.</td>
<td>*</td>
</tr>
<tr>
<td>NOP</td>
<td>No operation.</td>
<td>45</td>
</tr>
<tr>
<td>NZB</td>
<td>Non-zero jump backward.</td>
<td>37</td>
</tr>
<tr>
<td>NZF</td>
<td>Non-zero jump forward.</td>
<td>37</td>
</tr>
<tr>
<td>NZR</td>
<td>Non-zero jump relative.</td>
<td>*</td>
</tr>
<tr>
<td>ORG</td>
<td>Origin</td>
<td>+</td>
</tr>
<tr>
<td>OTA</td>
<td>Output from A.</td>
<td>43</td>
</tr>
<tr>
<td>OTN</td>
<td>Output no address.</td>
<td>43</td>
</tr>
<tr>
<td>OUT</td>
<td>Output</td>
<td>41</td>
</tr>
<tr>
<td>PJB</td>
<td>Positive jump backward.</td>
<td>37</td>
</tr>
<tr>
<td>PJF</td>
<td>Positive jump forward.</td>
<td>37</td>
</tr>
<tr>
<td>PJR</td>
<td>Positive jump relative.</td>
<td>*</td>
</tr>
<tr>
<td>PTA</td>
<td>Transfer P to A.</td>
<td>24</td>
</tr>
<tr>
<td>RAB</td>
<td>Replace add backward.</td>
<td>31</td>
</tr>
<tr>
<td>RAC</td>
<td>Replace add constant.</td>
<td>31</td>
</tr>
<tr>
<td>RAD</td>
<td>Replace add direct.</td>
<td>31</td>
</tr>
<tr>
<td>RAF</td>
<td>Replace add forward.</td>
<td>31</td>
</tr>
<tr>
<td>RAI</td>
<td>Replace add indirect.</td>
<td>31</td>
</tr>
<tr>
<td>RAM</td>
<td>Replace add memory.</td>
<td>31</td>
</tr>
<tr>
<td>RAR</td>
<td>Replace add relative.</td>
<td>*</td>
</tr>
<tr>
<td>RAS</td>
<td>Replace add specific.</td>
<td>31</td>
</tr>
<tr>
<td>RETURN</td>
<td>Return to SICAP</td>
<td>+</td>
</tr>
<tr>
<td>RS1</td>
<td>Right shift one.</td>
<td>32</td>
</tr>
<tr>
<td>RS2</td>
<td>Right shift two.</td>
<td>32</td>
</tr>
<tr>
<td>SICAP</td>
<td>Set SICOSAS to SICAP mode.</td>
<td>+</td>
</tr>
<tr>
<td>SSB</td>
<td>Subtract backward.</td>
<td>30</td>
</tr>
<tr>
<td>SBC</td>
<td>Subtract constant.</td>
<td>30</td>
</tr>
<tr>
<td>SBD</td>
<td>Subtract direct.</td>
<td>30</td>
</tr>
<tr>
<td>SBF</td>
<td>Subtract forward.</td>
<td>30</td>
</tr>
<tr>
<td>SBI</td>
<td>Subtract indirect.</td>
<td>30</td>
</tr>
<tr>
<td>SBM</td>
<td>Subtract memory.</td>
<td>30</td>
</tr>
<tr>
<td>SBN</td>
<td>Subtract no address.</td>
<td>30</td>
</tr>
<tr>
<td>SBR</td>
<td>Subtract relative.</td>
<td>*</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>SBS</td>
<td>Subtract specific.</td>
<td></td>
</tr>
<tr>
<td>SBU</td>
<td>Set buffer bank control.</td>
<td></td>
</tr>
<tr>
<td>SCB</td>
<td>Selective complement backward.</td>
<td></td>
</tr>
<tr>
<td>SCC</td>
<td>Selective complement constant.</td>
<td></td>
</tr>
<tr>
<td>SCD</td>
<td>Selective complement direct.</td>
<td></td>
</tr>
<tr>
<td>SCF</td>
<td>Selective complement forward.</td>
<td></td>
</tr>
<tr>
<td>SCI</td>
<td>Selective complement indirect.</td>
<td></td>
</tr>
<tr>
<td>SCM</td>
<td>Selective complement memory.</td>
<td></td>
</tr>
<tr>
<td>SCN</td>
<td>Selective complement no address.</td>
<td></td>
</tr>
<tr>
<td>SCR</td>
<td>Selective complement relative.</td>
<td></td>
</tr>
<tr>
<td>SCS</td>
<td>Selective complement specific.</td>
<td></td>
</tr>
<tr>
<td>SDC</td>
<td>Set direct bank control</td>
<td></td>
</tr>
<tr>
<td>SIC</td>
<td>Set indirect bank control.</td>
<td></td>
</tr>
<tr>
<td>SID</td>
<td>Set indirect and direct bank control.</td>
<td></td>
</tr>
<tr>
<td>SLJ</td>
<td>Selective jump.</td>
<td></td>
</tr>
<tr>
<td>SLS</td>
<td>Selective stop.</td>
<td></td>
</tr>
<tr>
<td>SJS</td>
<td>Selective stop and jump.</td>
<td></td>
</tr>
<tr>
<td>SRB</td>
<td>Shift replace backward.</td>
<td></td>
</tr>
<tr>
<td>SRC</td>
<td>Shift replace constant.</td>
<td></td>
</tr>
<tr>
<td>SRD</td>
<td>Shift replace direct.</td>
<td></td>
</tr>
<tr>
<td>SRF</td>
<td>Shift replace forward.</td>
<td></td>
</tr>
<tr>
<td>SRI</td>
<td>Shift replace indirect.</td>
<td></td>
</tr>
<tr>
<td>SRJ</td>
<td>Set relative bank control and jump.</td>
<td></td>
</tr>
<tr>
<td>SRM</td>
<td>Shift replace memory.</td>
<td></td>
</tr>
<tr>
<td>SRR</td>
<td>Shift replace relative.</td>
<td></td>
</tr>
<tr>
<td>SRS</td>
<td>Shift replace specific</td>
<td></td>
</tr>
<tr>
<td>STB</td>
<td>Store backward.</td>
<td></td>
</tr>
<tr>
<td>STC</td>
<td>Store constant.</td>
<td></td>
</tr>
<tr>
<td>STD</td>
<td>Store direct.</td>
<td></td>
</tr>
<tr>
<td>STE</td>
<td>Store buffer entrance register at Location 6X and transfer A to buffer entrance register.</td>
<td></td>
</tr>
<tr>
<td>STF</td>
<td>Store forward.</td>
<td></td>
</tr>
<tr>
<td>STI</td>
<td>Store indirect.</td>
<td></td>
</tr>
<tr>
<td>STM</td>
<td>Store memory.</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>STP</td>
<td>$Y \pm a$ Store P at location 5X.</td>
<td>26</td>
</tr>
<tr>
<td>STR</td>
<td>$Y \pm a$ Store relative.</td>
<td>*</td>
</tr>
<tr>
<td>STS</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>ZJB</td>
<td>$Y \pm a$ Zero jump backward.</td>
<td>37</td>
</tr>
<tr>
<td>ZJF</td>
<td>$Y \pm a$ Zero jump forward.</td>
<td>37</td>
</tr>
<tr>
<td>ZJR</td>
<td>$Y \pm a$ Zero jump relative.</td>
<td>*</td>
</tr>
<tr>
<td><em>(Col. 1)</em></td>
<td>Remark Card.</td>
<td>+</td>
</tr>
</tbody>
</table>
APPENDIX D

THIS IS A 7090 TRANSFORMATION OF A SICOM PROGRAM WRITTEN FOR THE 160-A COMPUTER BY J.O. DRINAN  DEC. 20, 1962

<table>
<thead>
<tr>
<th>MLOC</th>
<th>SILOC</th>
<th>K</th>
<th>OP</th>
<th>ADDR</th>
<th>SYMBOL</th>
<th>_OPCODE</th>
<th>VARIABLE FIELD</th>
<th>COMMENTARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000</td>
<td>0</td>
<td>02</td>
<td>0010</td>
<td>ORG</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td>0000</td>
<td>0</td>
<td>02</td>
<td>0010</td>
<td>START</td>
<td>ZEROR</td>
<td>ZERO ALL COMPONENTS OF ALL INDEX REGISTERS</td>
<td></td>
</tr>
<tr>
<td>0020</td>
<td>0008</td>
<td>1</td>
<td>22</td>
<td>1200</td>
<td>CYCLE</td>
<td>CLA</td>
<td>PICK UP X(I)</td>
<td></td>
</tr>
<tr>
<td>0022</td>
<td>0009</td>
<td>1</td>
<td>37</td>
<td>1202</td>
<td>DATA</td>
<td>DATA</td>
<td>ADD DATA+2,1</td>
<td>ADD Y(I)</td>
</tr>
<tr>
<td>0024</td>
<td>0012</td>
<td>2</td>
<td>24</td>
<td>1430</td>
<td>ADD</td>
<td>DVP</td>
<td>DIVIDE SUM X(I)+Y(I) BY Z(I)</td>
<td></td>
</tr>
<tr>
<td>0026</td>
<td>0011</td>
<td>3</td>
<td>19</td>
<td>1500</td>
<td>DVP</td>
<td>DIVISOR</td>
<td>STORE RESULT</td>
<td></td>
</tr>
<tr>
<td>0028</td>
<td>0012</td>
<td>3</td>
<td>06</td>
<td>0013</td>
<td>SD</td>
<td>RESULT</td>
<td>STORE RESULT</td>
<td></td>
</tr>
<tr>
<td>0030</td>
<td>0013</td>
<td>3</td>
<td>06</td>
<td>0080</td>
<td>IIB</td>
<td>NEXT</td>
<td>NEXT X(I)</td>
<td>STORE NEXT (X(I)+Y(I))</td>
</tr>
<tr>
<td>0032</td>
<td>0014</td>
<td>2</td>
<td>02</td>
<td>0300</td>
<td>SIB</td>
<td>CYCLE</td>
<td>CYCLE THROUGH DISCORS</td>
<td></td>
</tr>
<tr>
<td>0034</td>
<td>0015</td>
<td>2</td>
<td>16</td>
<td>0080</td>
<td>SIB</td>
<td>CYCLE</td>
<td>NEXT X(I)+Y(I)</td>
<td></td>
</tr>
<tr>
<td>0036</td>
<td>0016</td>
<td>1</td>
<td>02</td>
<td>2000</td>
<td>SIB</td>
<td>CYCLE</td>
<td>NEXT PORTION OF DATA</td>
<td></td>
</tr>
<tr>
<td>0040</td>
<td>0017</td>
<td>1</td>
<td>16</td>
<td>0008</td>
<td>IFR</td>
<td>CYCLE</td>
<td>DONE</td>
<td></td>
</tr>
<tr>
<td>0044</td>
<td>0018</td>
<td>2</td>
<td>02</td>
<td>0003</td>
<td>HALT3</td>
<td>TSM4</td>
<td>DUMMY OUT</td>
<td></td>
</tr>
<tr>
<td>0046</td>
<td>0019</td>
<td>2</td>
<td>64</td>
<td>1502</td>
<td>DATA</td>
<td>DEC</td>
<td>DATA IS STORED HERE</td>
<td></td>
</tr>
<tr>
<td>4540</td>
<td>1220</td>
<td>1</td>
<td>37</td>
<td>1220</td>
<td>ORG</td>
<td>DEC</td>
<td>1.5E+30</td>
<td>X1</td>
</tr>
<tr>
<td>4544</td>
<td>1222</td>
<td>2</td>
<td>14</td>
<td>1240</td>
<td>DEC</td>
<td>DEC</td>
<td>2.1E-29</td>
<td>Y1</td>
</tr>
<tr>
<td>4550</td>
<td>1224</td>
<td>3</td>
<td>34</td>
<td>1248</td>
<td>DEC</td>
<td>DEC</td>
<td>1.2E-28</td>
<td>X2</td>
</tr>
<tr>
<td>4554</td>
<td>1226</td>
<td>4</td>
<td>24</td>
<td>1260</td>
<td>DEC</td>
<td>DEC</td>
<td>2.1E-27</td>
<td>Y2</td>
</tr>
<tr>
<td>4560</td>
<td>1208</td>
<td>5</td>
<td>11</td>
<td>2600</td>
<td>DEC</td>
<td>DEC</td>
<td>5.1E+26</td>
<td>X3</td>
</tr>
<tr>
<td>4564</td>
<td>1210</td>
<td>6</td>
<td>1E</td>
<td>2500</td>
<td>DEC</td>
<td>DEC</td>
<td>6.1E+25</td>
<td>Y3</td>
</tr>
<tr>
<td>5050</td>
<td>1320</td>
<td>6</td>
<td>35</td>
<td>1300</td>
<td>ORG</td>
<td>DEC</td>
<td>7.9E+20</td>
<td>X4</td>
</tr>
<tr>
<td>5055</td>
<td>1322</td>
<td>7</td>
<td>9E</td>
<td>1900</td>
<td>DEC</td>
<td>DEC</td>
<td>8.8E-19</td>
<td>Y4</td>
</tr>
<tr>
<td>5060</td>
<td>1304</td>
<td>8</td>
<td>9E</td>
<td>1180</td>
<td>DEC</td>
<td>DEC</td>
<td>9.7E+18</td>
<td>X5</td>
</tr>
<tr>
<td>5064</td>
<td>1360</td>
<td>9</td>
<td>6E</td>
<td>1700</td>
<td>DEC</td>
<td>DEC</td>
<td>10.6E+17</td>
<td>Y5</td>
</tr>
<tr>
<td>5070</td>
<td>1306</td>
<td>10</td>
<td>5E</td>
<td>1600</td>
<td>DEC</td>
<td>DEC</td>
<td>11.5E-16</td>
<td>X6</td>
</tr>
<tr>
<td>5074</td>
<td>1312</td>
<td>11</td>
<td>4E</td>
<td>1500</td>
<td>DEC</td>
<td>DEC</td>
<td>12.4E+15</td>
<td>Y6</td>
</tr>
<tr>
<td>5360</td>
<td>1408</td>
<td>12</td>
<td>1E</td>
<td>2500</td>
<td>ORG</td>
<td>DEC</td>
<td>1.4E+20</td>
<td>X7</td>
</tr>
<tr>
<td>5365</td>
<td>1405</td>
<td>13</td>
<td>4E</td>
<td>2900</td>
<td>DEC</td>
<td>DEC</td>
<td>1.3E-9</td>
<td>Z1</td>
</tr>
<tr>
<td>5370</td>
<td>1404</td>
<td>14</td>
<td>4E</td>
<td>2080</td>
<td>DEC</td>
<td>DEC</td>
<td>1.4E+28</td>
<td>Z2</td>
</tr>
<tr>
<td>5374</td>
<td>1422</td>
<td>15</td>
<td>2E</td>
<td>3700</td>
<td>DEC</td>
<td>DEC</td>
<td>1.5E-37</td>
<td>Z3</td>
</tr>
<tr>
<td>5670</td>
<td>1500</td>
<td>16</td>
<td>1E</td>
<td>2000</td>
<td>ORG</td>
<td>Dec</td>
<td>1.5E+20</td>
<td>X(I)+Y(I)/Z(I)</td>
</tr>
<tr>
<td>5674</td>
<td>1502</td>
<td>17</td>
<td>02</td>
<td>0004</td>
<td>NNTJOBS HALT4</td>
<td>ORG</td>
<td>1500</td>
<td>RESULTS TO BE STORED HERE</td>
</tr>
<tr>
<td>0000</td>
<td>0000</td>
<td>18</td>
<td>02</td>
<td>0004</td>
<td>END</td>
<td>START</td>
<td>DUMMY END</td>
<td></td>
</tr>
</tbody>
</table>

* THIS IS A SAMPLE SICOMAS PROGRAM SHOWING SICAP COMMANDS ONLY 
* THE ORIGINAL PROGRAM CODED IN SICOM LANGUAGE APPEARS ON PAGE 42 OF 
* THE SICOM MANUAL. 

* NOTE THAT IT IS NOT NECESSARY TO SET THE I LIMIT OF INDEX REGISTER 3 
* BECAUSE THE NEXT COMMAND IN SEQUENCE IS TO BE EXECUTED REGARDLESS OF 
* THE INCREMENTING COMPARISON 

43
APPENDIX E

THIS IS A 7090 TRANSFORMATION OF A SICOM PROGRAM WRITTEN FOR THE 160-A COMPUTER BY ANYONE U. WISH
MLOC SILOC K OP ADDR SYMBOL_OPCODE VARIABLE_FIELD COMMENTARY PAGE 1

* THIS SAMPLE PROGRAM DEMONSTRATES COMMUNICATION BETWEEN A MAIN PROGRAM
* AND ITS SUBROUTINES, ONE OF WHICH (OUTPJ) IS IN MACHINE LANGUAGE.

0310 0100 ORG 170
0310 0100 0 64 0122 BEGIN TSM4 INPUT PICK UP A,E AND V
0312 0101 0 62 0123 TZE DONE FINISHED IF AC IS ZERO
0314 0102 0 65 0105 TSM5 SUBR COMPUTE RHQ USING A,E AND V
0316 0103 0 74 0214 CALL OUTPUT MACHINE LANGUAGE OUTPUT SUBROUTINE
0320 0104 0 66 0100 TSM6 BEGIN RECYCLE
0322 0105 0 21 0210 SUBR CLS E ECCENTRICITY
0324 0126 0 25 0210 MPY E -E**2 IN AC
0326 0107 6 01 0001 ADDC 1 1-E**2
0330 0108 0 25 0220 MPY A SEMI-MAJOR AXIS
0332 0109 0 77 0118 STO DIVDND A(1-E**2)
0334 0110 0 22 0212 CLA V TRUE ANOMOLY
0336 0111 0 72 0126 TSR COSDEG FIND COSINE V
0340 0112 0 25 0210 MPY E E COS(V)
0342 0113 6 01 0001 ADDC 1 1+F COS(V)
0344 0114 0 23 0118 IDVP DIVDND DIVIDE INTO A(1-E**2)
0346 0115 0 77 0120 STO RHO STORE ANSWER
0350 0116 0 02 0065 RSM5 RETURN TO MAIN PROGRAM
0354 0118 DIVDND DEC ø.
0360 0120 RHQ DEC ø.
0364 0122 0 82 0064 INPUT RSM4 DUMMY SUBROUTINE TO PICK UP INPUT DATA
0366 0123 0 02 0004 DONE HALT4
0370 0124 0 66 0100 TSM6 BEGIN
0374 0126 COSDEG SIN SIN-COS SUBROUTINE ENTRY +O =COSINE
0640 0208 A DEC ø.
0644 0210 E DEC ø.
0650 0212 V DEC ø.
MLAP
0654 7500 OUTPUT EXC 2402 EXTERNAL DEVICE
0655 4542
0656 7301 OUT 1 OUTPUT LOCATION RHQ+3
0657 0363 RHQ+3
0660 RETURN BACK TO MAIN PROGRAM
0100 END BEGIN
* THIS PROGRAM (LA07) ACCEPTS SYMBOLIC CODING IN BOTH A PSEUDO SICOM *
* LANGUAGE AND 160-A MACHINE LANGUAGE AND PRODUCES A PROGRAM LISTING *
* THE 160-A COMPUTER FOR EXECUTION.
* PROGRAM TAPES...
* THIS PROGRAM (LA07) ACCEPTS SYMBOLIC CODING IN BOTH A PSEUDO SICOM *
* LANGUAGE AND 160-A MACHINE LANGUAGE AND PRODUCES A PROGRAM LISTING *
* AND A BINARY MAP OF THE PROGRAM ON MAGNETIC TAPE WHICH READS INTO *
* THE 160-A COMPUTER FOR EXECUTION.
* PROGRAM TAPES...

A7 HI DENSITY..PROGRAM LISTING..PRINT PROGRAM CONTROL.
B7 HI DENSITY.. PROGRAM MEDIARY TAPE.
A8 HI DENSITY.. INPUT TAPE IF INPUT IS PRESTORED.
B8 LO DENSITY.. BINARY MAP OF ASSEMBLED PROGRAM.

ORG 15200

START
HPR
GETPEND GETPEND
+1 STL 22
+4 SVM 1
+5 TRA SETOFF
+6 STZ ONOFF
+7 STZ MCDE
+8 CLA CARD
+9 STO INUNIT
+10 TRA NOWIN
SETOFF STL ONOFF
+1 CLA INTAPE
+2 STO INUNIT
IIMAGE INPUT,,28
NOWIN STL 43
OIMAGE OUTPUT,,28
+4 STL 22
IFILE READ1
+8 STL 43
+11 AXT 20,4
+12 CLA NAME+20,4
+13 STO HEAD1+20,4
+14 TRX -2,4,1
ISCRIB INUNIT
READ1 STL 43
IFILE READ2
+4 STL 43
+7 AXT 0,1
COUNT SYMBOLS
ICHAR ALLBNK,,2
NO SYMBOL
+8 STL 43
ICHAR ISSYM,,13
NOT BLANK IN COLS 1-6
+12 STL 43
+16 CLA ATE
+17 STO ICNTR
ISCRIB INUNIT
READ2 STL 43
IBCW WORK,,1,14
+4 STL 43
+8 STZ TEMP
OBCW WORK,,1,14

APPENDIX F
WHAT KIND OF CARD

TEST FOR COMMENT

IS COMMENT IGNORE IT

LOOK AT OPCODE

IS SICOM CARD

CHECK RETURN CARD

MACHINE MODE

46
+1 TRA *+2
+2 TRA ISORG
+3 CAS DECDAT
+4 TRA *+2
+5 TRA ISDEC
+6 CAS EXTSK
+7 TRA *+2
+8 TRA ISDEC
+9 CAS END
+10 TRA *+2
+11 TRA ISEND
+12 CAS ECU
+13 TRA *+2
+14 TRA READ2
+15 CAS BSS
+16 TRA *+2
+17 TRA ISBSS
+18 NZT MODE
+19 TRA ISOP
+20 AXT BTYPE-MINSTR,4
CASBTY CAS BTYPE,4
+1 TRA *+2
+2 TRA LITEIT
+3 TIX CASBTY,4,1
+4 TRA ISOP
LITEIT SLN 1
IINT TEMP+1,1,6
ISCP STL 43
ALLBNK CLA ICNTR
+1 ADD TWO
+2 ZET MODE
+3 SUB ONE
+4 STO ICNTR
+5 SLT 1
+6 TRA READ2
+7 TRA ALLBNK+1
IBCW SYMTAB,1,1,1
ISSYM STL 43
+4 CLA ICNTR
+5 STO LCCTAB,1
+6 TXI *+1,1,-1
+7 SXA SYMCNT,1
+8 CLA ICNTR
AGN ADD TWO
+1 ZET MODE
+2 SUB ONE
+3 STO ICNTR
+4 SLT 1
+5 TRA READ2
+6 TRA AGN
IBCW SYMTAB,1,1,1
ISBSS STL 43
+4 CLA ICNTR
+5 STO LCCTAB,1

LOOK FOR SYMBOL
MACHINE LOCATION COUNTER
WAS THIS DEC DATA CARD
NO
-SYMBOL COUNT
WAS DEC DATA
+46 TRA ATNFOR
+47 CAS SICCM
+48 TRA Z1
+49 TRA \#+2
+50 TRA Z1
+51 STZ MCODE
+52 STZ ICODE
+53 TRA CMTCRD

Z1 CAS GCBACK
+1 TRA Z2
+2 TRA FCART

FOUND A RETURN CARD

Z2 CAS 160A
+1 TRA Z3
+2 TRA \#+2
+3 TRA Z3
+4 STL MCODE
ICHAR 0
+5 STL 43
ICHAR YISOCT,,8
+9 STL 43
IINT NEWTEM,,16,6
+13 STL 43
+17 STZ IMODE
+18 TRA CMTCRD
ICHAR 0

YISOCT STL 43
+4 STL IMODE
+5 TRA CMTCRD

Z3 CAS OCTAL
+1 TRA Z4
+2 TRA CCTCRC

Z4 CAS BLANK
+1 TRA Z5
+2 TRA BNCRC

Z5 CAS NAME
+1 TRA \#+2
+2 TRA CYCLEX
+3 CAS ORIGIN
+4 TRA \#+2
+5 TRA ORGCRC
+6 CAS DECCAT
+7 TRA \#+2
+8 TRA DATCRC
+9 CAS EXTMSK
+10 TRA BSSCAS
+11 TRA \#+2
+12 TRA BSSCAS
+13 STL MASKER
+14 TRA DATCRC

CYCLEX CLA NUMBER
+1 SUB ONE
+2 STO NUMBER
+3 TRA CYCLE
BSSCAS CAS BSS
+1 TRA **2
+2 TRA BSSCRD
+3 CAS EQU
+4 TRA **2
+5 TRA ECUCRD
+6 CAS END
+7 TRA **2
+8 TRA ENDCRD
+9 NZT MODE
+10 TRA STABLE
+11 AXT LSTINS-MINSTR,1
GOLOOK CAS LSTINS,1
+1 TRA **2
+2 TRA THANX
+3 TIX GCLOOK9191
+4 TRA BADCOD
THANX CLA LSTINS,1
+1 STO RITEOP
+2 CLA LSTKEY,1
+3 STO RITEKY
+4 LAC RITEKY,2
+5 TRA* AFORK,2
AFORK PZE ØTYPE
+1 PZE 1TYPE
+2 PZE 2TYPE
+3 PZE 3TYPE
+4 PZE 4TYPE
+5 PZE 5TYPE
STABLE AXT ENDCP-OPCODE,1
CMPCOD CAS ENDCP,1
+1 TRA **2
+2 TRA FNDOP
+3 TIX CMPCOD,1,1
+4 TRA BADCOD
FNDOP CLA ENDCP,1
+1 STO RITEOP
+2 CLA ENDC1N,1
+3 STO RITEKY
+4 LAC RITEKY,2
+5 TRA* FORK,2
FCRK PZE TYPEØ
+1 PZE TYPE1
+2 PZE TYPE2
+3 PZE TYPE3
+4 PZE TYPE4
+5 PZE TYPE5
+6 PZE TYPE6
ICHAR YESSION,18
TYPE0 STL 43
IINT SIADDR,,16,6
+4 STL 43
ICHAR Ø
+8 STL 43
+12 TSX VARFRM,4
EQU CARD
END CARD
IS MACHINE CODE
FIND CORRECT CODE IN TABLE
MATCHED
KOPADDR E.G. CLA MPY ETC
OP=DR E.G. NOP TMK5
K D18 SEL MAG TAPE
K DR E.G. BLOCK COPY
AD19 BACKSPACE
KOAD2R PXA PAX
KOP=OP EG CLAC GO ETC

51
+13  NOP
+14  TRA   NCPE
           ICHAR   0
YESSYM  STL   43
+4  TSX  FNSYM,4
+5   ARS   1
+6   STO  SIADDR
+7   SLT   2
+8  TRA   NCPE
+9   CAL  ACLIST
+10  ORS  RITEKY
+11  SLN   2
+12  TRA  TSXS
NOPE  CAL   INDEXR
+1  ORS  RITEKY
TSXS  TSX  STOKOP,4
ALMSTO TSX  CNVSAD,4
ASIFØ TSX  OKTODR,4
     +1  TSX  LCCMCH,4
     +2  TSX  LCCSIC,4
     +3  TSX  WRLIST,4
     +4   SLT   3
     +5  TRA  GCAHEG
     +6   CLA  RETURN
     +7   ADD   ONE
     +8   STA  **+1
     +9  TRA  **0
GOAHEG TSX  STOCMD,4
REBACK  CLA  ICNTR
     +1   ADD   TWO
     +2   ZET   MODE
     +3   SUB   ONE
     +4   STO  ICNTR
     +5   SLT   1
     +6  TRA   CYCLE
     +7  TRA  REBACK+1
TYPE1   AXT   5,4
     +1  STZ  K+5,4
     +2  TIX  *+1,4,1
     +3   CLA  ZERO
     +4  LDQ  RITEKY
     +5  RQL   6
     +6  LGL   6
     +7  STO  D2
     +8   CLA  ZERO
     +9  LGL   6
     +10  STO  R
     +11  TSX  OKTODR,4
     +12  TSX  LCCMCH,4
     +13  TSX  LCCSIC,4
     +14  TSX  WRLIST,4
     +15  LDQ  D2
     +16  MPY  DECMAL+2
     +17  XCA

WAS IT AC SYMBOL
NO
$00000

FOR BINARY ROUTINE

SET UP HOLLERITH ADDR
MOVE K OP ADDR TO OUTRAN OUTPUT REGION
PUT MACHINE LOCATION IN OUTRAN REGION
PUT SICOM LOCATION IN OUTPUT REGION
WRITE ONE LINE ON LISTING TAPE

ZERO K O P A D FOR TYPE 1 INSTR

MOVE $00000DR TO OUTPUT REGION

52
DONE THIS CARD
SEL MAC TAPE

TYPE 2
STZ G
STZ P
STZ A
CLA ONE
STC D2
CLA ATE
STO 8
TSX VARFRM,4
NOP
IMASK 0
STL 43
LDQ D1
MPY DECIMAL+1
STQ TEMP
LDQ C2
MPY DECIMAL+2
XCA
ADD R
ADD TEMP
STO SIADDR
TRA ASIF0

TYPE 3
TSX WIPE,4
TSX VARFRM,4
NOP
LDQ RITEKY
CLA ZERC
RGL 6
LGL 6
STC C2
CLA ZERC
LGL 6
STO R
LDQ C2
MPY DECIMAL+2
XCA
ADD R
STC SIADDR
TRA ASIF0

TYPE 4
TSX WIPE,4
CLA ONE
STO D2
CLA NINE
STO R
INT AD,, 16
STL 43
LDQ AD
CLA ZERC
CVH DECIMAL+2
STQ A
STO D1

53
+14 XCA
+15 ALS 3
+16 STO K
+17 LDQ D1
+18 MPY DECIMAL+1
+19 XCA
+20 ADD D19
+21 STO SIADDR
+22 TRA ASIFØ

TYPES  TSX WIPE,4   PAX,PXA 22 OR 23
+1 TSX VARFRM,4
+2 NOP
+3 CLA TWO
+4 STO D2
+5 LDQ RITEKY
+6 CLA ZERO
+7 LGL 6
+8 STO A
+9 CLA ZERO
+10 LGL 6
+11 STO D1
+12 CLA ZERO
+13 LGL 6
+14 STO R
+15 AXT 3,4

LDQA  LDQ A+3,4
+1 MPY DECIMAL+3,4
+2 STQ TEMP+3,4
+3 TIX LDQA+3,4,1
+4 CLA R
+5 ADD TEMP
+6 ADD TEMP+1
+7 ADD TEMP+2
+8 STO SIADDR
+9 TRA ASIFØ

TYPE6  TSX WIPE,4   KOP = OP ADDR
+1 AXT 3,4
+2 LDQ RITEKY

LOOP  CLA ZERO
+1 LGL 6
+2 STO K+3,4
+3 TIX LOOP,4,1
ICHAR NOTK,,8 ADDR MAY BE SYMBOLIC OR A CONSTANT
+4 STL 43
ICHAR SIADDR,,16,6
+8 STL 43
ICHAR 0
+12 STL 43
+16 TRA SEEIF

NOTK  TSX FNDSYM,4 ADDR IS SYMBOLIC
+1 ARS 1
+2 STO SIADDR
ICHAR 0
+3 STL 43

54
SUBROUTINE LINKAGE

2000 OR LESS
NUMBER OF 2000 NDS

TEST FOR 8000

IS 8000
4000

SUBROUTINE LINKAGE

SET UP HOLLERITH K,O,P

SUBROUTINE LINKAGE

CONVER BINARY SICOM ADDR TO BCD
SUBROUTINE LINKAGE

CONVER LOCATION COUNTER IN OUTPUT REGIO

OCTAL 10000

SUBROUTINE LINKAGE

DECIMAL 10000

SUBROUTINE LINKAGE

SUBROUTINE LINKAGE
FOUND A COMMENT CARD ON 2ND PASS
WRITE THIS DEC ISCAN ROUTINE

SUBROUTINE LINKAGE
* THIS ROUTINE WILL SET UP TWO REGISTERS COMPRISING A SICOM COMMAND
BEGIN 1,7
STOCMD TXL **5,0,0
+8 CLA WHRMAP
+9 ADD ICNTR
+10 STA STOREV
+11 ADD ONE
+12 STA STOROD
+13 CLA SIADDR
STOREV STO **0
+1 SLT 2
+2 TRA NOTAC
+3 CLA ABLNK
ALS3 ALS 3
+1 ORA 0
+2 ALS 3
+3 ORA P
STOROD STO **0
RETURN STOCMD
+1 TRA STOCMD+1
NOTAC CLA RITEKY
+1 TMI **3
+2 CLA K
+3 TRA ALS3
+4 SSP
+5 ARS 30
+6 SUB OCT12
+7 TRA ALS3
OCT12 OCT 12
BEGIN 1,7
FNDSYM TXL **5,0,0
+8 TSX VARFRM,4
+9 TRA **2
+10 TRA NORMAL
+11 CLA MYOWN
+12 ALS 1
RETURN FNDSYM
+13 TRA FNDSYM+1
NORMAL AXT 0,4
+1 LAC SYMCNT,1
+2 CLA AFIELD
+3 CAS ACCODE
+4 TRA **2
+5 TRA ISAC

59
LCCKAT  CAS SYMTAB,4
+1  TRA  *+2
+2  TRA  FOUND
+3  TXI  *+1,4,-1
+4  TIX  LCOKAT,1,1
+5  STO  UNDEF
       SYMBOL IS UNDEFINED
+6  STL  22
       CREADY
OBCW  UNDEF,,1,20
+8  STL  22
+12  TSX  ERROR,4
+13  CLA  ZERO
+14  TRA  FOUNDX
ISAC  SLN  2
 +1  CLA  ACNUM
       RETURN  FNDSYM
 +2  TRA  FNDSYM+1
FOUND  CLA  LCCTAB,4
 +1  STO  EQUSYM
 +2  CLA  AFIELD
 +3  TXI  *+1,4,-1
 +4  TIX  FINE,1,1
CMPSYM  CAS  SYMTAB,4
 +1  TRA  *+2
 +2  TRA  STUPID
 +3  TXI  *+1,4,-1
 +4  TIX  CMPSYM,1,1
FINE  CLA  ECUSYM
 +1  ADD  ADDIT
       RETURN  FNDSYM
FOUNDX  TRA  FNDSYM+1
STUPID  STO  DCUBLE
       CREADY
 +1  STL  22
OBCW  DCUBLE,,1,20
 +3  STL  22
 +7  TSX  ERROR,4
 +8  TRA  FINE
       BEGIN  1,4
ERROR  TXL  *+3,0,0
       SUBROUTINE LINKAGE
OSCRIB  CUTFAPE,,20,1
 +4  STL  22
 +8  SWT  3
       RETURN  ERROR
 +9  TRA  ERROR+1
       CREADY
 +10  STL  22
OSCRIB  PRINTR,,20
 +12  STL  22
       RETURN  ERROR
 +16  TRA  ERROR+1
       BEGIN  2,7
VARFRM  TXL  *+5,0,0
       SUBROUTINE LINKAGE
IMASK  ,,31
+8 STL 43
+12 AXT 15,4
ICOLR 16
+13 STL 43
RAVEL STZ DATA+15,4
IBCC DATA+15,4,,1
+1 STL 43
+5 TIX RAVEL,4,1
IMASK 0
COLS 16-30 UNPACKED
+6 STL 43
+10 STZ CCMCOL
+11 STZ ADDIT
+12 STZ ADDCOL
+13 STZ INDEXR
+14 AXT 15,4
+15 AXT 0,1
+16 CLA ISBLNK
LOG1 CAS DATA+15,4
+1 TRA *+2
+2 TRA FNDBNK
+3 TXI *+1,1,1
+4 TIX LOG1,4,1
+5 AXT 0,1
FNDBNK SXA BNKCOL,1
+1 LXA BNKCOL,4
+2 CLA BNKCOL
+3 ADD WHRDAT
+4 STA REFADD
+5 AXT 0,1
LOG2 CLA ISPOS
+1 CAS* REFADD
+2 TRA *+2
+3 TRA FNDADD
+4 CLA ISNEG
+5 CAS* REFADD
+6 TRA *+2
+7 TRA FNDADD
+8 TXI *+1,1,1
+9 TIX LCG2,4,1
+10 AXT 0,1
FNDADD SXA ADDCOL,1
+1 LXA BNKCOL,4
+2 AXT 0,1
+3 CLA ISCOM
LOG3 CAS* REFADD
+1 TRA *+2
+2 TRA FNDCOM
+3 TXI *+1,1,1
+4 TIX LCG3,4,1
+5 AXT 0,1
FNDCOM SXA CCMCOL,1
+1 ZET CCMCOL
+2 TRA YSCOMA
ZETADD ZET ADDCOL

61
+1 TRA YESADC WAS AN ADDITIVE FIELD
IBCW AFIELD,,16,1 SYMBOL AS IT STANDS IS OK
+2 STL 43
RETURN VARFRM NO ADD NO COMMA
+6 TRA VARFRM+1
YESADD CLA SIXX = MAYBE COMMA BUT ADDITIVE FIRST
+1 SUB ADDCCL ISOLATE SYMBOL FROM + OR -
INTP LTL **+2
+1 CLA ZERC ERROR
+2 XCA
+3 MPY SIXX
+4 XCA
+5 STA SHFTR (6-C)X6 = NO BITS TO RIGHT OF SYMBOL
+6 LXA ADDCCL,1 NO CHARACTERS IN SYMBOL
+7 AXT 0,2
+8 CLA ZERC
ROAT ALS 6
+1 CRA DATA,2 6 BCD BLANKS
+2 TXI **+1,2,-1 SYMBOL UNDRESSED STORED FOR LOOK UP
+3 TIX RCTAT,1,1
+4 LDQ BLANK
SHFTR LGL **0
+1 SLW AFIELD NOW GET ADDITIVE FIELD
+2 SLT 2 I MACROS
+3 TRA **+2
+4 TRA ISTAG
+5 CLA ADDCCL
+6 ADD CCLDIF
+7 STA IFDEC+3
+8 STA IFOCT+3
+9 CLA CCMCCL
+10 ZET CCMCCL
+11 TRA **+2
+12 CLA BNKCCL
+13 SUB ADDCCL
+14 ALS 18
+15 STD IFDEC+3 ZERO = DEC MODE
+16 STD IFOCT+3
+17 ZET IMODE
+18 TRA ACKOCT
ICHAR NONOCT,,8
+19 STL 43
IINT ADDIT,,**,**
IFDEC STL 43
+4 ZET MODE
+5 TRA MCVEON
+6 CLA ADDIT
+7 ALS 1
+8 STO ADDIT
MCVEON NZT CCMCCL IS THERE A TAG FIELD
RETURN VARFRM SYMBOL SET IN AFIELD +OR- IN ADDIT
+1 TRA VARFRM+1
+2 TRA ISTAG ICHAR NONOCT,,8
ACKOCT 
STL 43
I OCTAL ADDIT,**,**

IFOCT 
STL 43
RETURN VARFRM
+4 TRA VARFRM+1
ICHAR 0

NONOCT 
STL 43
OREADY
+4 STL 22
OBCW BADVAR,,1,20
+6 STL 22
+10 TSX ERROR,4
+11 STZ ADDIT
RETURN VARFRM
+12 TRA VARFRM+1

YSCOMA 
ZET ADDCOL
+1 TRA YESADD
+2 TRA HARD1
ISTAG 
CLA CCCOOL
+1 ADD COLDIF
+2 ADD ONE
+3 STA PROTAG+3
IINT K,**,1

PROTAG 
STL 43
+4 CLA K
+5 ALS 30
+6 SLW INDEXR
+7 SLT 1
RETURN VARFRM
+8 TRA VARFRM+1
RETURN VARFRM,1
+9 AXT 1,4
ICHAR 0

HARD1 
STL 43
ICHAR ISALPH,,8
+4 STL 43
+8 CLA CMCOOL
+9 ALS 18
+10 STD TRYIT+3
IINT MYOWN,,16,**

TRYIT 
STL 43
+4 SLN 1
+5 TRA ISTAG
ICHAR 0

ISALPH 
STL 43
+4 SLN 2
+5 CLA CMCOOL
+6 STO ADDCOL
+7 CLA SIXX
+8 SUB CMCOOL
+9 TRA INTPL

CMCOOL PZE
BNKCOL PZE
ADDCOL PZE

IS THERE ADDITIVE BESIDED TAG
PROCES TAG FIELD LATER
FIRST FIELD MAY NUMERIC

THIS CASE HAS NO ADD BUT TAG + MAYBE N
ICOSP0

INEXOR PZE
SIXX DEC 6
ADDIT PZE
COLDIF DEC 16
PYcht PZE
BEGIN 1,7

SCANIT TXL *+5,0,0
+8 CLA WHRMAP
+9 ADD ICNTR
+10 STA D890
+11 ADD ONE
+12 STA D567
+13 ADD ONE
+14 STA D234
+15 ADD ONE
+16 STA EXPD1
IMASK ,,31
+17 STL 43
+21 AXT 15,4
ICOLR 16
+22 STL 43

UNPACK STZ DATA+15,4
IBGC DATA+15,4,,1
+1 STL 43
+5 TIX UNPACK,4,1
IMASK 0
+6 STL 43
+10 CLA EXP0
+11 STO EXPCNT
+12 AXT 10,4
+13 STZ DIG+10,4
+14 TIX +,-1,4,1
+15 STZ MANSGN
+16 STZ EPART
+17 STZ ISTNZ
+18 STZ DPTIND
+19 AXT 15,1
LOOK AT 15 COLUMNS
+20 AXT 0,2
COLUMN COUNTER
+21 AXT 0,4
DIGIT STORER
+22 STZ DPTIND

LCCKC CLA DATA+15,1
+1 CAS ISBLNK
+2 TRA ++2
+3 TRA GETCUT
FOUND A BLANK
+4 CAS TEN
+5 TRA GR10
+6 HPR
+7 TNZ MARKIT
GR0 IS BETWEEN 1 AND 9
+8 ZET 1STNZ
IS THIS A LEADING 0
+9 TRA ++2
NO
+10 TRA MCRCLM
YES IGNORE IT
MARKIT NZT 1STNZ
+1 TRA 1STONE
STODIG STO DIGIT,4
+1 TXI ++1,4,1
MORCLM TXI ++1,2,1
+1 TIX LCOKC,1,1
+2 TRA ILEGAL
GR10 CAS ISPOS
+1 TRA ++2
+2 TRA YESPOS IS A + SIGN
+3 CAS ISNEG - SING
+4 TRA ++2
+5 TRA YESNEG E
+6 CAS ISE
+7 TRA ++2
+8 TRA YESE
+9 CAS ISDCPT
+10 TRA ++2
+11 TRA YESDPT
+12 NZT MASKER
+13 TRA ILEGAL
+14 SUB DELTA
+15 ZET 1STNZ
+16 TRA STODIG
+17 TRA 1STONE
YESPOS STZ MANGSN
+1 TRA MORCLM
YESNEG STL MANGSN
+1 TRA MORCLM
YESDPT SXA DECCLM,2 E FOUND WILL TERMINATE SEARCH
+1 STL DPTIND
+2 TRA MORCLM
1STONE SXA NZCOLM,2
+1 STL 1STNZ
+2 TRA STODIG
YESE SXA ECOLMN,2
+1 STL EPART
+2 LAC ECOLMN,4 NO
ICOLR 17,4
+3 STL 43
ICCHAR ILEGAL,,8
+6 STL 43
IINT EEXP,,,3
+10 STL 43
GETOUT CLA DECCLM
+1 CAS NZCOLM
+2 TRA DECR
+3 HPR 100
+4 CLA NZCOLM NON ZERO COLUMN IS LARGER
+5 SUB DECCLM
+6 SUB ONE OH REALLY
+7 SSM
TESTE NZT EPART WAS THERE AN EPART
+1 TRA ++2 NO
+2 ADD EEXP YES ADD IN
+3 ADD EXPONT OCT 100
STO EXP#NT
+5 TPL ASSM#L
+6 CLA EXP#¥
+7 SUB EXP#NT
+8 STO EXP#NT
+9 TRA ASSM#L
DECGR CLA DECL#M
+1 SUB N#COL#M
+2 TRA TES#TE

ASSEMBLE SIGN EXP + D1
NEG MANTISSA

ASSM#L CLA Z#RC
+1 ZET MANSGN
+2 CLA ONE
+3 ALS 7
+4 CRA EXP#NT
+5 ALS 4
+6 ORA DIGIT

EXP#1 STO **0
+1 CLA DIGIT+1
+2 ALS 4
+3 CRA DIGIT+2
+4 ALS 4
+5 CRA DIGIT+3

D234 STO **0
+1 CLA DIGIT+4
+2 ALS 4
+3 CRA DIGIT+5
+4 ALS 4
+5 ORA DIGIT+6

D567 STO **0
+1 CLA DIGIT+7
+2 ALS 4
+3 CRA DIGIT+8
+4 ALS 4
+5 CRA DIGIT+9

D890 STO **0
+1 STZ MASKER
RETURN SCANIT
+2 TRA SCANIT+1

ILEGAL

OREADY
+4 STL 22
GBCW BADDEC,,1,2Ø
+6 STL 22
+10 TSX ERROR,4
+11 STZ MASKER
RETURN SCANIT
+12 TRA SCANIT+1

EVEN#P CAL ICN#T
+1 ANA 2#BITS
+2 SSM
+3 ADD D4
+4 ANA 2#BITS
+5 ADD ICN#T

66
SUBROUTINE LINKAGE

+6 STO ICNTR
+7 TRA 1,4
2BITS PZE 3
D4 PZE 4
BEGIN 1,7
RITOUT TXL *+5,0,0
REWB REWB 8
+1 SDLB 8
+2 AXT 12,4
+3 CLA RIMAP+12,4
+4 STO SIMAP+12,4
+5 TIX *-2,4,1
+6 AXT 1365,1
+7 AXT 0,2
CLASIM CLA SIMAP,2
+1 ALS 12
+2 CRA SIMAP+1,2
+3 ALS 12
+4 CRA SIMAP+2,2
+5 SLW OREGIN+1365,1
+6 TXI *+1,2,3
+7 TIX CLASIM,1,1
+8 WTBB 8
+9 RCHB Scribe
+10 TGOB *
+11 TRCB REWB
+12 REWB 8
RETURN RITOUT
+13 TRA RITCUT+1
$0TYPE CAL RITEKY
+1 ARS 21
+2 STO FOROUT
+3 ANA M7777
+4 STC FCRMAP
+5 TSX OUTCOD,4
+6 TSX LCCMCH,4
+7 TSX MAPIT,4
+8 TSX WRLIST,4
+9 CLA ICNTR
+10 ADD ONE
+11 STO ICNTR
ICHTAR 0
BNKCRD STL 43
ICHTAR DILLY,8
+4 STL 43
+8 NZT IMODE
+9 TRA QDECM
IOTAL TEMP,,16,4
+10 STL 43
+14 TRA PICKUP
INT TEMP,,16,4
QDECM STL 43
PICKUP CLA TEMP
+1 ANA M7777

EG BLS 0100 YYYY
+2 TRA STOFOR
  ICHAR 0

CILLY STL 43
  +4 TSX FNSYM,4
  +5 ANA M7777
  +6 TRA STOFOR

1TYPE CAL RITEKY
  +1 ARS 21

STCFCR STO FCRCUT
  +1 ANA M7777
  +2 STO FCRCUT
  +3 TSX OLTCOD,4
  +4 TSX LCMCCH,4
  +5 TSX MAPIT,4
  +6 TSX WRLIST,4
  +7 TRA REBACK

  ICHAR 0

EG PTA 0101 NO SYMBOLIC FIELD

2TYPE STL 43
  ICHAR SILLY,8
  +4 STL 43
  +8 STL 43

CLAM7 CLA M7
  +1 ANS TEMP

CALRIT CAL RITEKY
  +1 ARS 21
  +2 CRA TEMP
  +3 TRA STOFOR

  ICHAR 0

SILLY STL 43
  +4 TSX FNSYM,4
  +5 STO TEMP
  +6 TRA CLAM7

  ICHAR 0

EG LCDN 04XX

3TYPE STL 43
  ICHAR NILLY,8
  +4 STL 43
  +8 NZI IMODE
  +9 TRA QCDEC
  +10 STI 43
  +14 TRA CLAM77

  ICHAR 0

EG LCDN 04XX

QCDEC STL 43

CLAM77 CLA M77
  +1 CAS TEMP
  +2 TRA CALRIT
  +3 TRA CALRIT

STZTEM STZ TEMP
  +1 TSX OCRNG,4
  +2 TRA CALRIT

  ICHAR 0

NILLY STL 43
  +4 TSX FNSYM,4

68
I

JCD
+6 STZ TEMP
+7 TRA CALRIT

JCD STO TEMP
+1 CAL RITEKYY
+2 ANA TGMASK
+3 TZE CLAM77
+4 SUB ITAG
+5 TNZ ISBACK
+6 CLA TEMP
+7 SUB ICNTR

TMIS TMI STZTEM
+1 STO TEMP
+2 TRA CLAM77

ISBACK CLA ICNTR
+1 SUB TEMP
+2 TRA TMIS

ICHAR 0

4TYPE STL 43
ICHAR TILLY,,8
+4 STL 43
IINT TEMP,,16,1
+8 STL 43
+12 CLA TEMP
+13 ALS 3
+14 STO TEMP
+15 TRA CALRIT

TILLY TSX FNDSYM,4
+1 TNZ +3
+2 STZ TEMP
+3 TRA CALRIT
+4 ALS 3
+5 STO TEMP
+6 TRA CLAM77

5TYPE TSX FNDSYM,4
+1 SUB ICNTR
+2 TMI ISREV
+3 STO TEMP
+4 TRA CLAM77

ISREV SSP
+1 STO TEMP
+2 CAL TGMASK
+3 ANA RITEKYY
+4 ALS 12
+5 ACL RITEKYY
+6 SLW RITEKYY
+7 TRA CLAM77

BEGIN 1,4

CUTCCD TXL +3,0,0
OCOTAL FOROUT,,20,4
+4 STL 22
RETURN CUTCCD
+8 TRA OUTCCD+1
BEGIN 1,4

EG SLJ 77X0

RELATIVE MODE

BACKWARD REFERENCE

SUBROUTINE LINKAGE
MAPIT TXL  *+3,0,0
+4 CLA  WRRMAP
+5 ADD  ICNTR
+6 STA  *+2
+7 CLA  FCMAP
+8 STO  **0
RETURN MAPIT
+9 TRA  MAPIT+1
BEGIN  1,4

OCRNG TXL  *+3,0,0
OREADY
+4 STL  22
OBCW  RANGERT,1,20
+6 STL  22
+10 TSX  ERRCR,4
RETURN  OCRNG
+11 TRA  OCRNG+1
ICHAR  GOOF,,8

OCTCRD STL  43
IOCTAL  TEMP,,16,4
+4 STL  43
+8 CLA  TEMP
+9 TRA  STOFOR
OREADY

GOOF STL  22
OBCW  GOON,,1,20
+2 STL  22
+6 TSX  ERROR,4
+7 CLA  ZERO
+8 TRA  STOFOR

FORART AXT  3,1
+1 TSX  LCMCH,4
+2 TSX  WRLIST,4

CLABAC CLA  BACKTO+3,1
+1 STO  FCMAP
+2 TSX  MAPIT,4
+3 CLA  ICNTR
+4 ADD  ONE
+5 STO  ICNTR
+6 TXI  CLABAC,1,1
+7 TRA  CYCLE

INSTR EQU  ICNTR
DETAIL
DETAIL
QSUBRA  SSCR,BB1,ENDSQR,DD1

FORSQR TSX  EVENUP,4
+1 CLA  SSQR
+2 STO  SYMTAB,1
+3 CLA  ICNTR
+4 STO  LCCTAB,1
+5 TXI  **1,1,1
+6 SXX  SYMCNT,1
ICHAR  BB1,,8
+7 STL  43
+11 TSX WRLIST,4
+12 CLA ENDSQR
+13 ADD INSTR
+14 STO INSTR
+15 TRA CYCLE

LIST
QSUBRB ENDSQR
LOGFCR TSX EVENUP,4
QSUBRB ENDEXP
EXPFOR TSX EVENUP,4
QSUBRB ENDSIN
SINFOR TSX EVENUP,4
QSUBRB ENDSIN
ATNFOR TSX EVENUP,4
SSQR BCI 1,SQR
SLCG BCI 1,LOG
SEXP BCI 1,EXP
SSIN BCI 1,SIN
SATN BCI 1,ATAN
IMODE PZE
EQUSYM PZE

DOUBLE BCI 9, HAS BEEN MULTIPLY DEFINED.... THE VALUE ASSIGNED
+9 BCI 9, TO THE SYMBOL ON FIRST ENCOUNTER WILL BE USED....
+18 BCI 2,************

LIST
MODE PZE
EXTMSK BCI 1, MASK
MASKER PZE
DELTA OCT 52
SICOM BCI 1, SICAP
GOBACK BCI 1, RETURN
160A BCI 1, MLAP
THREE DEC 3
FOROUT PZE
FORMAP PZE
M7777 OCT 7777
M7 OCT 7

BADVAR BCI 9, THIS CARD HAS VIOLATED THE RULES FOR THE ADDITIVE SUBF
+9 BCI 9, IELD.... IT WILL BE SET TO ZERO....
+18 BCI 2,************
RANGER BCI 9, THE FOLLOWING CARD HAS AN OUT OF RANGE CONDITION. THE
+9 BCI 9, XX OF THE EEXX FORMAT HAS BEEN REPLACED BY ZERO....
+18 BCI 2,************

M77 OCT 77
GOON BCI 9, THE VARIABLE FIELD OF THE FOLLOWING OCT CARD IS IN ERR
+9 BCI 9, OR.... ZERO HAS BEEN SUBSTITUTED....
+18 BCI 2,************
ALLDUN BCI 9, THIS IS THE END OF THIS ASSEMBLY. REMOVE BINARY B8.
+9 BCI 9, REMOVE LISTING TAPE A7 (UNLESS STACKING). PRESS START
+18 BCI 2, FOR NEXT JOB
BACKTO OCT 40
+1 OCT 2006
+2 OCT 30
MINSTR BCI 1, BLIS

TYPING A TWO REGISTERS
BTYPE  BCI  1,PTA
    +1 BCI  1,ETA
    +2 BCI  1,CTA
    +3 BCI  1,LD S
    +4 BCI  1,LC S
    +5 BCI  1,STS
    +6 BCI  1,MUT
    +7 BCI  1,MUH
    +8 BCI  1,ADS
    +9 BCI  1,SBS
   +10 BCI  1,RAS
   +11 BCI  1,AOS
   +12 BCI  1,LS1
   +13 BCI  1,LS2
   +14 BCI  1,LS3
   +15 BCI  1,LS6
   +16 BCI  1,RS1
   +17 BCI  1,RS2
   +18 BCI  1,SRS
   +19 BCI  1,LPS
   +20 BCI  1,SCS
   +21 BCI  1,CBC
   +22 BCI  1,CIL
   +23 BCI  1,INA
   +24 BCI  1,OTA
   +25 BCI  1,ERR
   +26 BCI  1,HLT
   +27 BCI  1,HPR

THIS TYPE (B) HAS NO VARIABLE FI
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+28</td>
<td>BCI 1, STP</td>
</tr>
<tr>
<td>+29</td>
<td>BCI 1, STE</td>
</tr>
<tr>
<td>+30</td>
<td>BCI 1, SRJ</td>
</tr>
<tr>
<td>+31</td>
<td>BCI 1, SIC</td>
</tr>
<tr>
<td>+32</td>
<td>BCI 1, IRJ</td>
</tr>
<tr>
<td>+33</td>
<td>BCI 1, SDC</td>
</tr>
<tr>
<td>+34</td>
<td>BCI 1, DRJ</td>
</tr>
<tr>
<td>+35</td>
<td>BCI 1, SID</td>
</tr>
<tr>
<td>+36</td>
<td>BCI 1, ACJ</td>
</tr>
<tr>
<td>+37</td>
<td>BCI 1, SBU</td>
</tr>
<tr>
<td>+38</td>
<td>BCI 1, NOP</td>
</tr>
<tr>
<td>+39</td>
<td>BCI 1, SLS</td>
</tr>
<tr>
<td>+40</td>
<td>BCI 1, LDN</td>
</tr>
<tr>
<td>+41</td>
<td>BCI 1, LDD</td>
</tr>
<tr>
<td>+42</td>
<td>BCI 1, LDI</td>
</tr>
<tr>
<td>+43</td>
<td>BCI 1, LDF</td>
</tr>
<tr>
<td>+44</td>
<td>BCI 1, LDB</td>
</tr>
<tr>
<td>+45</td>
<td>BCI 1, LCN</td>
</tr>
<tr>
<td>+46</td>
<td>BCI 1, LCD</td>
</tr>
<tr>
<td>+47</td>
<td>BCI 1, LCI</td>
</tr>
<tr>
<td>+48</td>
<td>BCI 1, LCF</td>
</tr>
<tr>
<td>+49</td>
<td>BCI 1, LCB</td>
</tr>
<tr>
<td>+50</td>
<td>BCI 1, STD</td>
</tr>
<tr>
<td>+51</td>
<td>BCI 1, STI</td>
</tr>
<tr>
<td>+52</td>
<td>BCI 1, STF</td>
</tr>
<tr>
<td>+53</td>
<td>BCI 1, STB</td>
</tr>
<tr>
<td>+54</td>
<td>BCI 1, HWI</td>
</tr>
<tr>
<td>+55</td>
<td>BCI 1, ADN</td>
</tr>
<tr>
<td>+56</td>
<td>BCI 1, ADD</td>
</tr>
<tr>
<td>+57</td>
<td>BCI 1, ADI</td>
</tr>
<tr>
<td>+58</td>
<td>BCI 1, ADF</td>
</tr>
<tr>
<td>+59</td>
<td>BCI 1, ADB</td>
</tr>
<tr>
<td>+60</td>
<td>BCI 1, SBN</td>
</tr>
<tr>
<td>+61</td>
<td>BCI 1, SBD</td>
</tr>
<tr>
<td>+62</td>
<td>BCI 1, SBI</td>
</tr>
<tr>
<td>+63</td>
<td>BCI 1, SBF</td>
</tr>
<tr>
<td>+64</td>
<td>BCI 1, SBB</td>
</tr>
<tr>
<td>+65</td>
<td>BCI 1, RAD</td>
</tr>
<tr>
<td>+66</td>
<td>BCI 1, RAI</td>
</tr>
<tr>
<td>+67</td>
<td>BCI 1, RAF</td>
</tr>
<tr>
<td>+68</td>
<td>BCI 1, RAB</td>
</tr>
<tr>
<td>+69</td>
<td>BCI 1, AOD</td>
</tr>
<tr>
<td>+70</td>
<td>BCI 1, ACI</td>
</tr>
<tr>
<td>+71</td>
<td>BCI 1, ACF</td>
</tr>
<tr>
<td>+72</td>
<td>BCI 1, AOB</td>
</tr>
<tr>
<td>+73</td>
<td>BCI 1, SRD</td>
</tr>
<tr>
<td>+74</td>
<td>BCI 1, SRI</td>
</tr>
<tr>
<td>+75</td>
<td>BCI 1, SRF</td>
</tr>
<tr>
<td>+76</td>
<td>BCI 1, SRB</td>
</tr>
<tr>
<td>+77</td>
<td>BCI 1, LPN</td>
</tr>
<tr>
<td>+78</td>
<td>BCI 1, LPD</td>
</tr>
<tr>
<td>+79</td>
<td>BCI 1, LPI</td>
</tr>
<tr>
<td>+80</td>
<td>BCI 1, LPF</td>
</tr>
<tr>
<td>+81</td>
<td>BCI 1, LPB</td>
</tr>
</tbody>
</table>

TYPE C FILL IN LO ORDER OCTAL DIGIT

TYPE D FILL IN TWO LO ORDER OCTAL DIGIT
+82 BCI 1,SCN
+83 BCI 1,SCD
+84 BCI 1,SCI
+85 BCI 1,SCF
+86 BCI 1,SCB
+87 BCI 1,SCF
+88 BCI 1,NZF
+89 BCI 1,PJF
+90 BCI 1,NJF
+91 BCI 1,ZJB
+92 BCI 1,NJB
+93 BCI 1,PJB
+94 BCI 1,NJB
+95 BCI 1,JPI
+96 BCI 1,JFI
+97 BCI 1,INP
+98 BCI 1,CUT
+99 BCI 1,CTN
+100 BCI 1,EXF
+101 BCI 1,SJS
+102 BCI 1,SLJ
+103 BCI 1,ACR
+104 BCI 1,ACR
+105 BCI 1,LCR
+106 BCI 1,LDR
+107 BCI 1,LPR
+108 BCI 1,LSR
+109 BCI 1,NJR
+110 BCI 1,NZR
+111 BCI 1,PJR
+112 BCI 1,RAR
+113 BCI 1,SBR
+114 BCI 1,SCR
+115 BCI 1,SRR
+116 BCI 1,STR
+117 BCI 1,ZJR
+118 BCI 1,LSB
+119 BCI 1,LSD
+120 BCI 1,LSF
+121 BCI 1,LSI
+122 BCI 1,LSN

<table>
<thead>
<tr>
<th>LSTINS</th>
<th>PZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFD</td>
<td>015/100,21/0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MCCDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFD</td>
</tr>
<tr>
<td>VFD</td>
</tr>
<tr>
<td>VFD</td>
</tr>
<tr>
<td>VFD</td>
</tr>
<tr>
<td>VFD</td>
</tr>
<tr>
<td>VFD</td>
</tr>
<tr>
<td>No.</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>+6</td>
</tr>
<tr>
<td>+7</td>
</tr>
<tr>
<td>+8</td>
</tr>
<tr>
<td>+9</td>
</tr>
<tr>
<td>+12</td>
</tr>
<tr>
<td>+11</td>
</tr>
<tr>
<td>+12</td>
</tr>
<tr>
<td>+13</td>
</tr>
<tr>
<td>+14</td>
</tr>
<tr>
<td>+15</td>
</tr>
<tr>
<td>+16</td>
</tr>
<tr>
<td>+17</td>
</tr>
<tr>
<td>+18</td>
</tr>
<tr>
<td>+19</td>
</tr>
<tr>
<td>+20</td>
</tr>
<tr>
<td>+21</td>
</tr>
<tr>
<td>+22</td>
</tr>
<tr>
<td>+23</td>
</tr>
<tr>
<td>+24</td>
</tr>
<tr>
<td>+25</td>
</tr>
<tr>
<td>+26</td>
</tr>
<tr>
<td>+27</td>
</tr>
<tr>
<td>+28</td>
</tr>
<tr>
<td>+29</td>
</tr>
<tr>
<td>+30</td>
</tr>
<tr>
<td>+31</td>
</tr>
</tbody>
</table>
+33 VFD 015/112,21/1
+34 VFD 015/113,21/1
+35 VFD 015/3300,21/1
+36 VFD 015/3700,21/1
+37 VFD 015/5300,21/1
+38 VFD 015/5700,21/1
+39 VFD 015/102,21/1
+40 VFD 015/103,21/1
+41 VFD 015/110,21/1
+42 VFD 015/111,21/1
+43 VFD 015/114,21/1
+44 VFD 015/115,21/1
+45 VFD 015/4700,21/1
+46 VFD 015/1300,21/1
+47 VFD 015/1700,21/1
+48 VFD 015/104,21/1
+49 VFD 015/120,21/1
+50 VFD 015/7600,21/1
+51 VFD 015/7677,21/1
+52 VFD 015/121,21/1
+53 VFD 015/7700,21/1
+54 VFD 015/7700,21/1
+55 VFD 015/150,21/2
+56 VFD 015/160,21/2
+57 VFD 015/10,21/2
+58 VFD 015/20,21/2
+59 VFD 015/30,21/2
<table>
<thead>
<tr>
<th>VFD</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>015/40,21/2</td>
<td>VFD 015/50,21/2</td>
<td></td>
</tr>
<tr>
<td>015/60,21/2</td>
<td>VFD 015/70,21/2</td>
<td></td>
</tr>
<tr>
<td>015/140,21/2</td>
<td>VFD 015/14100,21/2</td>
<td></td>
</tr>
<tr>
<td>015/500,21/3</td>
<td>VFD 021/200001,15/3</td>
<td></td>
</tr>
<tr>
<td>015/400,21/3</td>
<td>VFD 021/230002,15/3</td>
<td></td>
</tr>
<tr>
<td>015/2000,21/3</td>
<td>VFD 021/260001,15/3</td>
<td></td>
</tr>
<tr>
<td>015/2400,21/3</td>
<td>VFD 021/270002,15/3</td>
<td></td>
</tr>
<tr>
<td>015/2500,21/3</td>
<td>VFD 021/420001,15/3</td>
<td></td>
</tr>
<tr>
<td>015/4000,21/3</td>
<td>VFD 021/430002,15/3</td>
<td></td>
</tr>
<tr>
<td>015/6000,21/3</td>
<td>VFD 021/300001,15/3</td>
<td></td>
</tr>
<tr>
<td>015/7000,21/3</td>
<td>VFD 021/310002,15/3</td>
<td></td>
</tr>
<tr>
<td>015/8000,21/3</td>
<td>VFD 021/320001,15/3</td>
<td></td>
</tr>
<tr>
<td>015/9000,21/3</td>
<td>VFD 021/330002,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/700,21/3</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/3400,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/3500,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/360001,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/370002,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/5000,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/5100,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/520001,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/530002,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/5400,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/5500,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/560001,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/570002,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/4400,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/4500,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/460001,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/470002,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/200,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1000,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1100,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/120001,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/130002,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/300,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1400,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1500,21/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/160001,15/3</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/170002,15/3</td>
<td></td>
</tr>
</tbody>
</table>

79
<table>
<thead>
<tr>
<th></th>
<th>VFD</th>
<th>Code</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>+114</td>
<td>021/630001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+115</td>
<td>021/610001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+116</td>
<td>021/620001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+117</td>
<td>021/630001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+118</td>
<td>021/640002,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+119</td>
<td>021/650002,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+120</td>
<td>021/660002,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+121</td>
<td>021/670002,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+122</td>
<td>021/680002,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+123</td>
<td>015/7000,21/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+124</td>
<td>021/710001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+125</td>
<td>021/720001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+126</td>
<td>021/730001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+127</td>
<td>021/740001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+128</td>
<td>021/750001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+129</td>
<td>021/760001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+130</td>
<td>021/770001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+131</td>
<td>021/780001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+132</td>
<td>021/790001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+133</td>
<td>021/800001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+134</td>
<td>021/810001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+135</td>
<td>021/820001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+136</td>
<td>021/830001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+137</td>
<td>021/840001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+138</td>
<td>021/850001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+139</td>
<td>021/860001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+140</td>
<td>021/870001,15/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VFD  C21/160001,15/5
+141
VFD  C21/460001,15/5
+142
VFD  C21/420001,15/5
+143
VFD  C21/610003,15/5
+144
VFD  C21/170002,15/3
+145
VFD  C21/140000,15/3
+146
VFD  C21/160001,15/3
+147
VFD  C21/150000,15/3
+148
VFD  C21/300000,15/3
+149
VFD  C15/1,21/1

LSTKEY

VFD

ENCSQR

PZE

*SCRXXX

LCGXXX

OCT

81
+88 OCT 3600,101,6307,2313,4326,2314,4327,2006
+88 OCT 70,5402,400,4334,400,4337,6607,0
+96 OCT 40,2302,4024,6614,3044,465,502,2004
+104 OCT 4827,4620,4446,2003,421,421,421,2001
+112 OCT 4401,2547,1064,1761,2600,3531,1470,2021
+120 OCT 1571,4071,2405,6001,2044,1220,3430,2022
+128 OCT 3511,1425,4450,6023,2470,1050,2006,2025
+136 OCT 110,120,423,2023,2544,442,23,2002
+144 OCT 1447,3626,2560,2021,3174,52,3326,52
ENDATN PZE =-ATANXX
CREGIN BSS 1400,0
SCRIBE IOCDB BEGIN,1365
RETURN SCANIT
 +1 TRA SCANIT+1
UNDEF BCI 9, IS AN UNDEFINED SYMBOL. IT WILL BE EQUATED TO
+9 BCI 3, ZERO.
+12 BCI 8, ******************************************************
NEWTEM BSS 5,H
ACCODE BCI 9, IS AN ILLEGAL OPCODE. A NOP HAS BEEN USED.
+9 BCI 3,
+12 BCI 8, ******************************************************
XXX BCI 1, XXXXXX
C19 DEC 19
RITECP BCI 1,
RITEKY PZE 0,
ACCODE BCI 1,$
AFIELD BCI 1,
DATA BSS 15,C
DIGIT BSS 10,C
MANSIGN PZE
EXPTN PZE
EXP0 OCT 100
EXPNSGN PZE
ISPOS BCI 1,00000+
ISNEG BCI 1,00000-
ISE BCI 1,00000E
ISBLNK BCI 1,00000
ISTNZ PZE
MANSET PZE
ZER0= FIRST NON ZERO COLUMN NOT MET
ZER0 = MANTISSA SIGN NOT SET
ZER0 = NO E PART
EPART PZE
DECGLMN PZE
NZCOLUM PZE
TEN DEC 10
ECCLMN PZE
ISDCPT BCI 1,00000.
OPTIND PZE
C17 DEC 17
EEXP DEC 0
E PART OF EXPONENT
BADDEC BCI 9, THE FOLLOWING DEC CARD VIOLATES THE RULES FOR DATA IN
+9 BCI 9, PUT... A ZERO HAS BEEN STORED.
+18 BCI 2,
MCHLCC PZE 0
NAME BCI 9, THIS IS A 7090 TRANSFORMATION OF A SICOM PROGRAM WRIT
+9 BCI 9, TEN FCR THE 160-A COMPUTER BY SOMEONE WHO CHOOSES TO

83
Page 84

HEADING 1
RSD 20.0

HEAD 2
BCI 9, MLOC SILOC K OP ADDR SYMBOL OPCODE VARIABLE FIE
+9 BCI 9,LD COMMENTARY PAGE

PAGE BCI 2,
NAME BCI 50,1

PAGENO PZE CASE 1

CLA 22
+1 SUB ONE
+2 STA SWTCH+1
HEAD CASE1,,1,55,1
+3 STL 22
OREADY
+8 STL 22
CBCW HEAD1,,1,20
+10 STL 22
OSCRIB CUTAPE,,20,1
+14 STL 22
OREADY
+18 STL 22
+20 CLA PAGENO
+21 ADD ONE
+22 STO PAGENO
CBCW HEAD2,,1,20
+23 STL 22
CINT PAGENO,,109,2
+27 STL 22
OSCRIB CUTAPE,,20,1
+31 STL 22
OREADY
+35 STL 22
CBLANK 120,,1
+37 STL 22
OSCRIB CUTAPE,,20,1
+40 STL 22

SWTON SWT 3
+1 TRA **0
+2 WPDA
+3 SPRA 1
OREADY
+4 STL 22
OBCW HEAD1,,1,20
+6 STL 22
OSCRIB PRINTR,,20
+10 STL 22
OREADY
+14 STL 22
OBCW HEAD2,,1,20
+16 STL 22
CINT PAGENO,,109,2
+20 STL 22
OSCRIB PRINTR,,20
+24 STL 22
OREADY
NORMAL SICOM COMMAND = 7 CHARACTERS

CARD READER
PRINTER
CARD OR TAPE

B7 FOR COPY CARD INPUT

ZERO = ON LINE CARD INPUT
LEADER
ENABLE OCT 2000
CCTAL BCI 1, OCT
NUMER PZE
SIMAP BSS 4096, 0
WHRMAP HTR SIMAP
WHRDAT HTR DATA
REFADD PZE 0, 4
RIMAP DEC 18
+1 OCT 0
+2 DEC 21
+3 OCT 100
+4 DEC 4
+5 OCT 64
G02 DEC 0
+1 OCT 64
+2 DEC 19
+3 DEC 0
+4 DEC 1
+5 OCT 64
BSS BCI 1, BSS
OPCODE BCI 1, SIB
+1 BCI 1, SID
+2 BCI 1, SIL
+3 BCI 1, DIB
+4 BCI 1, IIB
+5 BCI 1, SJB
+6 BCI 1, SJD
+7 BCI 1, SJL
+8 BCI 1, DJB
+9 BCI 1, IJB
+10 BCI 1, CADM
+11 BCI 1, CLS
+12 BCI 1, CLA
+13 BCI 1, IDVP
+14 BCI 1, DVP
+15 BCI 1, MPY
+16 BCI 1, EXCH
+17 BCI 1, EXTR
+18 BCI 1, ADD
+19 BCI 1, SUB
+20 BCI 1, ADM
+21 BCI 1, ISUB
+22 BCI 1, RAD
+23 BCI 1, CAS
+24 BCI 1, OFLTB
+25 BCI 1, OFLCR
+26 BCI 1, OFXTB
+27 BCI 1, OFXCR
+28 BCI 1, OFORM
+29 BCI 1, OCRTB
+30 BCI 1, OTBNC
+31 BCI 1, OCMND
+32 BCI 1, WRFILE
+33 BCI 1, SRFILE
+34 BCI 1, GRPIN
+35 BCI 1, SNGIN
+36 BCI 1, IOCTAL
+37 BCI 1, OALPHA
+38 BCI 1, LDANR
+39 BCI 1, CMPANR
+40 BCI 1, MRGANR
+41 BCI 1, EXTANR
+42 BCI 1, TZE
+43 BCI 1, TNZ
+44 BCI 1, TPL
+45 BCI 1, TMI
+46 BCI 1, TSMN
+47 BCI 1, TSM5
+48 BCI 1, TSM6
+49 BCI 1, TNR7
+50 BCI 1, TSR
+51 BCI 1, TSJ1
+52 BCI 1, TSJ2
+53 BCI 1, TSJ3
+54 BCI 1, CALL
+55 BCI 1, TBR
+56 BCI 1, STOANR
+57 BCI 1, STO
+58 BCI 1, NOP
+59 BCI 1, STOP1
+60 BCI 1, STOP2
+61 BCI 1, HALT3
+62 BCI 1, HALT4
+63 BCI 1, HPRIN
+64 BCI 1, REL
+65 BCI 1, ABS
+66 BCI 1, FAST
+67 BCI 1, TRACE
+68 BCI 1, ZEROIR
+69 BCI 1, IFLEX
+70 BCI 1, OFLEX
+71 BCI 1, IYPE
+72 BCI 1, OYPE
+73 BCI 1, OPRINT
+74 BCI 1, SELBOD
+75 BCI 1, SELBIN
+76 BCI 1, BSR
+77 BCI 1, Rew
+78 BCI 1, OTAPE
+79 BCI 1, ITAPE
+80 BCI 1, BLCPY
+81 BCI 1, BLCLR
+82 BCI 1, IOCTAL
+83 BCI 1, OSPEC
+84 BCI 1,ISPEC
+85 BCI 1,RZE
+86 BCI 1,RNZ
+87 BCI 1,RPL
+88 BCI 1,RMI
+89 BCI 1,RSM4
+90 BCI 1,RSM5
+91 BCI 1,RSM6
+92 BCI 1,RNR7
+93 BCI 1,RSR
+94 BCI 1,RSJ1
+95 BCI 1,RSJ2
+96 BCI 1,RSJ3
+97 BCI 1,LOC
+98 BCI 1,PUSTOP
+99 BCI 1,CLAC
+100 BCI 1,CLSC
+101 BCI 1,IDVPC
+102 BCI 1,DVPC
+103 BCI 1,ADDC
+104 BCI 1,SUBC
+105 BCI 1,ISUBC
+106 BCI 1,SHIFT
+107 BCI 1,XEQ
+108 BCI 1,SETFWA
+109 BCI 1,SETLWA
+110 BCI 1,MPYC
+111 BCI 1,PXAIB
+112 BCI 1,PXAID
+113 BCI 1,PXAIL
+114 BCI 1,PXAJB
+115 BCI 1,PXAJC
+116 BCI 1,PXAJL
+117 BCI 1,PAXIB
+118 BCI 1,PAXIC
+119 BCI 1,PAXIL
+120 BCI 1,PAXJB
+121 BCI 1,PAXJD
+122 BCI 1,PAXJL

ENDOP PZE
OPKEY BCI 1,002000
SID BCI 1,003000
SIL BCI 1,004000
DIB BCI 1,005000
IIB BCI 1,006000
SJB BCI 1,012000
SJD BCI 1,013000
SJL BCI 1,014000
CJB BCI 1,015000
IJ BCI 1,016000
CADM BCI 1,020000
CLS BCI 1,021000
CLA BCI 1,022000
JVP BCI 1,023000

THIS IS SIB 1ST OF TYPE 1 INSTRUCTIONS
END OF TYPE 0 INSTRUCTIONS

TYPE 1 INSTR OP CODE = DR
CLIR BCI 1,010001
RDIFW BCI 1,011001
WRFLW BCI 1,012001
RDTPW BCI 1,013001
WRTPW BCI 1,014001
WRPRN BCI 1,016001
SELTTP BCI 1,017001
SELMTD BCI 1,018002
ESR BCI 1,019001
REW BCI 1,019001
WRMT2 BCI 1,020003
RDMT2 BCI 1,021003
BLCPY BCI 1,024003
BLCLR BCI 1,025003
POT BCI 1,027003
PST BCI 1,028003
RCST BCI 1,029001
RZERC BCI 1,060001
RNZERO BCI 1,061001
RPCS BCI 1,062001
RNEG BCI 1,063001
RMK4 BCI 1,064001
RMK5 BCI 1,065001
RMK6 BCI 1,066001
RMK7 BCI 1,067001
RSR BCI 1,070001
RSH1 BCI 1,071001
RSH2 BCI 1,072001
RSH3 BCI 1,073001
CLCC BCI 1,075001
PLSTCP BCI 1,079001
CLAC BCI 1,010006
CLSC BCI 1,020006
IDVPC BCI 1,030006
IDVPC BCI 1,040006
ADD C BCI 1,050006
SUBC BCI 1,060006
ISUBC BCI 1,070006
SHFTAC BCI 1,080006
+1 BCI 1,090006
SETTRC BCI 1,100006
STPTRC BCI 1,110006
MPYC BCI 1,120006
PXIBA BCI 1,022005
PXIDA BCI 1,032005
PXILA BCI 1,042005
PXJBA BCI 1,122005
PXJDA BCI 1,132005
PXJLA BCI 1,142005
ATXIB BCI 1,023005
ATXID BCI 1,033005
TXID BCI 1,033005
ATXIL BCI 1,043005
ATXJB BCI 1,123005
ATXJD BCI 1,133005

TYPE 2 INSTR

TYPE 6
<table>
<thead>
<tr>
<th>ATXJL</th>
<th>BCI</th>
<th>1,143005</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCKEY</td>
<td>PZE</td>
<td>1</td>
</tr>
<tr>
<td>SYMTAB</td>
<td>BSS</td>
<td>1000,1</td>
</tr>
<tr>
<td>LCCTAB</td>
<td>BSS</td>
<td>1000,C</td>
</tr>
<tr>
<td></td>
<td>CRC</td>
<td>32256</td>
</tr>
<tr>
<td></td>
<td>CORE</td>
<td>START,-1</td>
</tr>
<tr>
<td>DLMP</td>
<td>STL</td>
<td>2169</td>
</tr>
<tr>
<td></td>
<td>CORE</td>
<td>43,43</td>
</tr>
<tr>
<td>+4</td>
<td>STL</td>
<td>2169</td>
</tr>
<tr>
<td></td>
<td>CORE</td>
<td>22,22</td>
</tr>
<tr>
<td>+8</td>
<td>STL</td>
<td>2169</td>
</tr>
<tr>
<td>+12</td>
<td>TRA</td>
<td>110</td>
</tr>
<tr>
<td>END</td>
<td>START</td>
<td></td>
</tr>
</tbody>
</table>

91
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDKR</td>
<td>0014</td>
</tr>
<tr>
<td>ENDEXP</td>
<td>0038</td>
</tr>
<tr>
<td>ENDEXT</td>
<td>0047</td>
</tr>
<tr>
<td>ENDELCG</td>
<td>0038</td>
</tr>
<tr>
<td>ENDELS</td>
<td>0038</td>
</tr>
<tr>
<td>ENDESL</td>
<td>0037</td>
</tr>
<tr>
<td>ENDESN</td>
<td>0013</td>
</tr>
<tr>
<td>ENDESQ</td>
<td>0028</td>
</tr>
<tr>
<td>EVENRP</td>
<td>0022</td>
</tr>
<tr>
<td>EXPFOR</td>
<td>0028</td>
</tr>
<tr>
<td>EXPONT</td>
<td>0039</td>
</tr>
<tr>
<td>EXPSGN</td>
<td>*0039</td>
</tr>
<tr>
<td>EXPXIK</td>
<td>0038</td>
</tr>
<tr>
<td>EXTRAN</td>
<td>*0045</td>
</tr>
<tr>
<td>EXTMSK</td>
<td>0028</td>
</tr>
<tr>
<td>FNDADD</td>
<td>0017</td>
</tr>
<tr>
<td>FNDNRC</td>
<td>0017</td>
</tr>
<tr>
<td>FNDNRM</td>
<td>0015</td>
</tr>
<tr>
<td>FORART</td>
<td>0026</td>
</tr>
<tr>
<td>FORATT</td>
<td>0027</td>
</tr>
<tr>
<td>FOREX</td>
<td>0027</td>
</tr>
<tr>
<td>FOREX2</td>
<td>0027</td>
</tr>
<tr>
<td>FORMAP</td>
<td>0028</td>
</tr>
<tr>
<td>FORMCH</td>
<td>0014</td>
</tr>
<tr>
<td>FOROUT</td>
<td>0028</td>
</tr>
<tr>
<td>FORSIN</td>
<td>0027</td>
</tr>
<tr>
<td>FORSCR</td>
<td>0026</td>
</tr>
<tr>
<td>FOUNX</td>
<td>0016</td>
</tr>
<tr>
<td>GETOUT</td>
<td>0021</td>
</tr>
<tr>
<td>GOAED</td>
<td>0008</td>
</tr>
<tr>
<td>GOSBCK</td>
<td>0028</td>
</tr>
<tr>
<td>GOLOC</td>
<td>0007</td>
</tr>
<tr>
<td>HLTGIN</td>
<td>*0045</td>
</tr>
<tr>
<td>HLTSH1</td>
<td>*0045</td>
</tr>
<tr>
<td>HLTSH2</td>
<td>*0045</td>
</tr>
<tr>
<td>ILLEGAL</td>
<td>0022</td>
</tr>
<tr>
<td>IMAGE1</td>
<td>0005</td>
</tr>
<tr>
<td>IMAGE2</td>
<td>*0005</td>
</tr>
<tr>
<td>INDEXR</td>
<td>0020</td>
</tr>
<tr>
<td>INTAPE</td>
<td>0041</td>
</tr>
<tr>
<td>INUNIT</td>
<td>0041</td>
</tr>
<tr>
<td>IREDNL</td>
<td>0012</td>
</tr>
<tr>
<td>ISALPH</td>
<td>0019</td>
</tr>
<tr>
<td>ISBACK</td>
<td>0025</td>
</tr>
<tr>
<td>ISBLNK</td>
<td>0049</td>
</tr>
<tr>
<td>ISDVT</td>
<td>0049</td>
</tr>
<tr>
<td>LEADER</td>
<td>0042</td>
</tr>
<tr>
<td>LITEIT</td>
<td>0003</td>
</tr>
<tr>
<td>LOMCH</td>
<td>0012</td>
</tr>
</tbody>
</table>
HALT 0045
HARD 0019
HEAD 0040
HEAD2 0040
IGNTR 0041
IDVP 0044
IDVPC 0046
IFDEC 0018
IFICT 0019
IIB 0044
JBD 0044
MODE 0028
INPUT 0041
INSTR 0026
INTPL 0040
ISAC 0040
SBS 003
ISCM 0020
ISDEC 0040
ISE 0039
ISEND 0040
ISNEG 0039
ISOP 0003
ISORG 0004
ISPOS 0004
ISREV 0039
ISE 0003
ISNEG 0039
ISOP 0003
ISORG 0004
ISPOS 0004
ISREV 0003
IUC 0004
IUCB 0004
JDD 0025
K 0041
KOP 0041
LDQA 0018
LOGL 0018
LOG2 0017
LOG3 0017
LOOK 0020
LOOP 0010
M 0028
M7 0028
M77 0028
MAPIT 0026
MCODE 0031
MODE 0028
MPY 0045
MPYC 0046
MYCOWN 0020
NAME 0039
NAMER 0040
NILLY 0040
NINE 0041
NCEND 0012
NCP 0045
NOPE 0008
NCT 0011
NCTAC 0015
NCTK 0010
NCGO 005
NCWIN 0001
C 0041
CCMND 0045
OCRTB 0045
CTL 0015
COTT 002
CTAL 0042
OFLCR 0045
OFLTB 0045
OFORM 0045
OFXCR 0045
OLC 0046
ONE 0042
OFF 0041
ORC 0026
CPKEY 0044
OTBNO 0045
P 0041
PAGE 0040
PCT 0046
PST 0046
PXIBA 0046
PXIDA 0046
PXILA 0046
PXJBA 0046
PXJO 0046
PXJLA 0046
CDECM 0023
CDECM 0024
CDEC 0024
CCDEC 0024
R 0041
RAT 0045
RAVEL 0017
RDFLW 0046
RDFMT 0046
RDST 0046
RDTPW 0046
READ1 0001
READ2 0001
RESTR 0045
REVRS 0013
REW 0046
REW8 0023
RIMAP 0042
RM4 0046
RM5 0046
RM6 0046
RM7 0046
RM8 0046
RS 0046
RS1 0046
RS2 0046
RS3 0046
RS4 0046
RS5 0046
RS6 0046
RS7 0046
RS8 0046
RS9 0046
RS10 0046
RS11 0046
RS12 0046
RS13 0046
RS14 0046
RS15 0046
RS16 0046
S 0046
SATN 0028
SATT 0046
SEIF 0011
SEP 0020
SFMT 0045
SFO 0045
SFLTR 0018
SIC 0020
SID 0044
SIL 0044
SILLY 0024
SIMAP 0042
SIUX 0020
SJRE 0044
SJO 0044
SLL 0044
SCL 0046
SLOG 0026
SNGIN 0045
SSIN 0028
SSQR 0028
SSS 0028
TEND 0033
TENTE 0021
THANX 0026
THREE 0026
TILLY 0025
TM 0045
TIONS 0025
TML 0045
TNC 0045
TFL 0045
TRIAL 0015
TSMT 0045
TSM 0045
TSM7 0045
TST 0045
TST1 0045
TST2 0045
TST3 0045
TST4 0045
TST5 0045
TST6 0045
TST7 0045
TST8 0045
UNDEF 0059
WFM 0045
WILE 0011
WORK 0041
WRFLW 0046
WRMT 0046
WRPRN 0046
WRTPW 0046
X 0039
Y 0021
Z 0039
Z 0021
Z2 0026
Z3 0026
Z4 0026
Z5 0026
ZERO 0041
<table>
<thead>
<tr>
<th>Division 2</th>
<th>Group 25</th>
<th>Group 34</th>
<th>Group 51</th>
<th>AFSSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. R. Wieser</td>
<td>M. Check</td>
<td>J. Craig</td>
<td>P. Fergus</td>
<td></td>
</tr>
<tr>
<td>S. H. Dodd</td>
<td>E. Harvey</td>
<td>P. Fleck</td>
<td>J. Frankovich</td>
<td></td>
</tr>
<tr>
<td>Group 21</td>
<td>E. Hofmann</td>
<td>T. Goblick</td>
<td>J. Heart</td>
<td></td>
</tr>
<tr>
<td>W. Crowther</td>
<td>B. Moriarty</td>
<td>S. McCarroll</td>
<td>S. Ornstein</td>
<td></td>
</tr>
<tr>
<td>A. Curtis (2)</td>
<td>W. Vecchia</td>
<td>R. Price</td>
<td>Group 52</td>
<td></td>
</tr>
<tr>
<td>J. Drinan</td>
<td>D. Wiggert</td>
<td>W. Smith</td>
<td>G. Bluestein</td>
<td></td>
</tr>
<tr>
<td>O. Fortier</td>
<td>H. Bostick</td>
<td>Group 35</td>
<td>S. Larkin</td>
<td></td>
</tr>
<tr>
<td>H. Frachtenman</td>
<td>Group 26</td>
<td>B. Fowler</td>
<td>F. Nagy</td>
<td></td>
</tr>
<tr>
<td>D. Hafford</td>
<td>D. Underwood</td>
<td>P. Miller</td>
<td>M. Wendell</td>
<td></td>
</tr>
<tr>
<td>F. Heart</td>
<td>H. Zieman</td>
<td>L. Tahlmoush</td>
<td>Group 55</td>
<td></td>
</tr>
<tr>
<td>A. Mathiasen</td>
<td></td>
<td></td>
<td>F. Slagle</td>
<td></td>
</tr>
<tr>
<td>R. Saliga</td>
<td></td>
<td></td>
<td>J. Slagle</td>
<td></td>
</tr>
<tr>
<td>P. Smith</td>
<td>J. Arnow</td>
<td>Group 37</td>
<td>Group 58</td>
<td></td>
</tr>
<tr>
<td>P. Stylos</td>
<td>D. Koniver</td>
<td>N. Holway</td>
<td>E. Freed</td>
<td></td>
</tr>
<tr>
<td>R. Teoste</td>
<td>H. Meily</td>
<td>Group 314</td>
<td>M. Goodman</td>
<td></td>
</tr>
<tr>
<td>S. White</td>
<td>H. Peterson</td>
<td></td>
<td>L. Klem</td>
<td></td>
</tr>
<tr>
<td>Group 22</td>
<td></td>
<td></td>
<td>C. McElwain</td>
<td></td>
</tr>
<tr>
<td>M. Andrews</td>
<td>G. P. Dinneen</td>
<td></td>
<td>R. Mitchell</td>
<td></td>
</tr>
<tr>
<td>A. Armenti</td>
<td>Group 32</td>
<td></td>
<td>G. Mueser</td>
<td></td>
</tr>
<tr>
<td>J. Avery</td>
<td>B. Benn</td>
<td></td>
<td>J. Smith</td>
<td></td>
</tr>
<tr>
<td>V. Brady</td>
<td>H. Briscoe</td>
<td></td>
<td>M. Smith</td>
<td></td>
</tr>
<tr>
<td>R. Carroll</td>
<td>R. Brown</td>
<td></td>
<td>A. Stowe</td>
<td></td>
</tr>
<tr>
<td>N. Ciampa</td>
<td>J. Burdette</td>
<td></td>
<td>F. Trask</td>
<td></td>
</tr>
<tr>
<td>D. Clapp</td>
<td>N. Donald</td>
<td></td>
<td>B. White</td>
<td></td>
</tr>
<tr>
<td>J. FitzGerald</td>
<td>R. Martinson</td>
<td></td>
<td>D. Yntema</td>
<td></td>
</tr>
<tr>
<td>A. Freed</td>
<td>J. Morriello</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R. Gauvreau (15)</td>
<td>G. Price</td>
<td></td>
<td>Group 71</td>
<td></td>
</tr>
<tr>
<td>A. Grometstein</td>
<td>K. Ralston</td>
<td></td>
<td>E. Boyce</td>
<td></td>
</tr>
<tr>
<td>J. Halberstein</td>
<td>H. Shimmin</td>
<td></td>
<td>W. Fanning</td>
<td></td>
</tr>
<tr>
<td>C. Hardy</td>
<td></td>
<td></td>
<td>C. Pappas</td>
<td></td>
</tr>
<tr>
<td>J. Hartogensis</td>
<td></td>
<td></td>
<td>D. Regillo</td>
<td></td>
</tr>
<tr>
<td>P. Heffernan</td>
<td></td>
<td></td>
<td>F. Wan</td>
<td></td>
</tr>
<tr>
<td>R. Holland</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C. Hopkins</td>
<td></td>
<td></td>
<td>Group 75</td>
<td></td>
</tr>
<tr>
<td>D. Kramer</td>
<td></td>
<td></td>
<td>D. Moore</td>
<td></td>
</tr>
<tr>
<td>B. Lee</td>
<td></td>
<td></td>
<td>P. Stetson</td>
<td></td>
</tr>
<tr>
<td>R. Leonard</td>
<td></td>
<td></td>
<td>Group 81</td>
<td></td>
</tr>
<tr>
<td>D. Lovenvirh</td>
<td></td>
<td></td>
<td>M. Durgin</td>
<td></td>
</tr>
<tr>
<td>A. MacGillivray</td>
<td></td>
<td></td>
<td>V. Mason</td>
<td></td>
</tr>
<tr>
<td>F. Mahoney</td>
<td></td>
<td></td>
<td>N. Rawson</td>
<td></td>
</tr>
<tr>
<td>M. Malone</td>
<td></td>
<td></td>
<td>S. Sillers</td>
<td></td>
</tr>
<tr>
<td>L. Meixsell</td>
<td></td>
<td></td>
<td>C. Work</td>
<td></td>
</tr>
<tr>
<td>B. Nanni</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J. Nolan</td>
<td></td>
<td></td>
<td>Group 84</td>
<td></td>
</tr>
<tr>
<td>J. O'Brien</td>
<td></td>
<td></td>
<td>M. Dresselhaus</td>
<td></td>
</tr>
<tr>
<td>F. Oliver</td>
<td></td>
<td></td>
<td>W. Mason</td>
<td></td>
</tr>
<tr>
<td>D. Reiner</td>
<td></td>
<td></td>
<td>Group 22 Central Depository</td>
<td></td>
</tr>
<tr>
<td>J. Rheinstein</td>
<td></td>
<td></td>
<td>J. FitzGerald</td>
<td></td>
</tr>
<tr>
<td>K. Russell</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B. Schafer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>