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>ORGY</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.
SICOSAS, AN IBM 7090 COMPUTER ASSEMBLY PROGRAM
FOR THE CDC 160-A COMPUTER

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."

2
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
```

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:

\[ K \text{OP} \text{ADDR}(1) \]

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 alphanumeric 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 OFLTb $

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>Location Field</th>
<th>A location symbol or all blank</th>
</tr>
</thead>
<tbody>
<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.

```plaintext
SQR
LOG
EXP
SIN
ATAN
```

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 \( D_{10} D_9 \ldots D_1 \) where \( D_{10} \) of DATA is 9 and \( D_1 \) 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:

\[
\begin{array}{cccc}
11 & 7 & 3 & 0 \\
\hline
D_3 & D_2 & D_1 & \\
D_6 & D_5 & D_4 & \\
\hline
D_9 & D_8 & D_7 & \\
S & \text{EXPONENT} & D_{10} & \\
\end{array}
\]

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

\[
\begin{align*}
1020 &= D_3 D_2 D_1 \\
2503 &= D_6 D_5 D_4 \\
4166 &= D_9 D_8 D_7 \\
2111 &= S, \text{ EXP, } D_{10}
\end{align*}
\]

The machine cell assigned to contain the first part \((D_3 D_2 D_1)\) 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 0310₂ and 0311₂ symbolically as follows:

\[
0310 = \begin{array}{cccc}
11 & A & D & D & R & 0 \\
X & & & & &
\end{array}
\]

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

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>WAI</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 (·), 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>+ 123456 x 10^8</td>
</tr>
<tr>
<td>DEC -1.23456E-1</td>
<td>- 123456 x 10^0</td>
</tr>
</tbody>
</table>

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

**DEC 123.456Eb5** produces + 123456 x 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
- V is 1011
- W is 1100
- X is 1101
- Y is 1110
- 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, (·), and the blank are carried over from the DEC card.
These are SICAP Subroutine Designator cards explained under EXP SICAP Library Subroutines.

\textbf{CALL \textit{Y \pm a}, \textit{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 \( Y \pm a, k \).

\textbf{RETURN}

RETURN is a macro-type, machine-language instruction that is the counterpart of the \textit{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.

\begin{verbatim}
0040 SDC Set Direct Bank
2006 LDD Return Location
0030 IRJ Jump to SICOM Interpreter
\end{verbatim}

\textbf{END \textit{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.

\textbf{SICOSAS LIMITATIONS AND RESTRICTIONS}

\begin{enumerate}
\item 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.

\item 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).
\end{enumerate}
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>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>SELBIN 0</td>
<td>Select binary tape 0</td>
</tr>
<tr>
<td>0001</td>
<td>ITAPE 0, 1</td>
<td>Read in under control of index #1</td>
</tr>
<tr>
<td>0002</td>
<td>TSM4 4</td>
<td>Error exit; transfer to 0004</td>
</tr>
<tr>
<td>0003</td>
<td>TSM4 [START]</td>
<td>Read-in o.k.; transfer to execute</td>
</tr>
<tr>
<td>0004</td>
<td>REW</td>
<td>Try again</td>
</tr>
<tr>
<td>0005</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>Notes</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, renews the magnetic tape and initiates another loading.
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

\[ \begin{array}{ccccccc}
1 & \text{SYMBOL} & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & b & \text{OPCODE} & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & \text{COMMENTARY}
\end{array} \]

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 (\( Y \pm a \), 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 \( 0 \leq K \leq 9 \). The address field may contain an additive subfield (\( \pm a \), above). The start of the additive field is signaled by the + or - sign. "a" must be a decimal integer \( 0 \leq a \leq 1999 \). The address field (\( Y \), above) may either be symbolic or a decimal integer \( 0 \leq Y \leq 9999 \). 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.

\begin{tabular}{llll}
SIB & \( Y \pm a, k \) & Set i Base & The base component of dimension i of index register k is set to the value \( Y \pm a \).
SIB & \( Y \pm a, k \) & Set i Difference & The difference component of dimension i of index register k is set to the value \( Y \pm a \).
SIL & \( Y \pm a, k \) & Set i Limit & The limit component of dimension i of index register k is set to the value \( Y \pm a \).
DIB & \( Y \pm a, k \) & 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, \( Y \pm a \) 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
\end{tabular}
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.

**OFLTB** $Y \pm a, k$

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.

**OFXTB** $Y \pm a, k$

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.

**OFXCR** $Y \pm a, k$

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, m, $0 \leq m \leq 18$ is set to AD digits and n, $0 \leq n \leq 18$
is set to DR digits. (See OFXTB and OFXCR) AD plus DR
cannot exceed 18. If DR is zero, the period is not put out.

**OCRTB** $Y \pm a, k$

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.

**OTBNO** $Y \pm a, k$

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.

**OCMND** $Y \pm a, k$

Output a Command from Memory

The command in location effective $Y \pm a$ is put out in the
standard form (K OP $Y \pm a$) 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.
GRPIN $Y \pm a, k$  
**Group Input**  
Input is transferred to effective $Y \pm a$ and consecutive locations. (See SICOM Manual, paragraph II. B, 6, p. 14, "Normal Input").

SNGIN $Y \pm a, k$  
**Single Input**  
Input is transferred only to location effective $Y \pm a$. One word of any type of input may be entered. The input is terminated by either a carriage return or a tab.

IOCTAL $Y \pm a, k$  
**Input Octal Tape**  
Tape punched by "OOCTAL" Command is read into the receiving field starting at effective $Y \pm a$. The field must fall entirely within a bank. The location of the receiving field may differ from the original source field.

OALPHA $Y \pm a, k$  
**Output Alphanumeric Data**  
The contents of address effective $Y \pm a$ and effective $Y \pm a + 1$ are put out as 8 alphanumeric characters on the previously-selected device. This instruction is not affected by the relative mode.

LDANR $Y \pm a, k$  
**Load Alpha Numeric Register (A. N. R.)**  
The contents of effective $Y \pm a$ are loaded into A. N. R. as alphanumeric data. (See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")

CMPANR $Y \pm a, k$  
**Compare Alpha Numeric Register (A. N. R.)**  
(See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")

MRGANR $Y \pm a, k$  
**Merge into Alpha Numeric Register (A. N. R.)**  
(See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")

EXTANR $Y \pm a, k$  
**Extract Alpha Numeric Register (A. N. R.)**  
(See SICOM Manual, paragraph II. B. 9, p. 18, "Manipulating Alphanumeric Data.")

TZE $Y \pm a, k$  
**Transfer on Accumulator Zero**  
If the contents of the accumulator is zero, a mark 60 jump to effective $Y \pm a$ occurs. Otherwise, the normal sequence is continued.

TNZ $Y \pm a, k$  
**Transfer on Accumulator Non-Zero**  
If the contents of the accumulator is not zero, a mark 61 jump to effective $Y \pm a$ occurs. Otherwise, normal sequence continues.

TPL $Y \pm a, k$  
**Transfer on Accumulator Plus**  
If the accumulator contains a positive quantity (including zero), a mark 62 jump to effective $Y \pm a$ occurs. Otherwise, normal sequence continues.
<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMI</td>
<td>Y ± a, k</td>
<td>Transfer on Accumulator Minus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the accumulator contains a negative quantity, a mark 63 jump to effective Y ± a occurs. Otherwise, normal sequence continues.</td>
</tr>
<tr>
<td>TSM4</td>
<td>Y ± a, k</td>
<td>Transfer and Set Mark 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The next command in sequence is marked for a &quot;return to mark 4&quot; return. Control goes to effective Y ± a.</td>
</tr>
<tr>
<td>TSM5</td>
<td>Y ± a, k</td>
<td>Transfer and Set Mark 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The next command in sequence is marked for a &quot;return to mark 5&quot; return. Control goes to effective Y ± a.</td>
</tr>
<tr>
<td>TSM6</td>
<td>Y ± a, k</td>
<td>Transfer and Set Mark 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The next command in sequence is marked for a &quot;return to mark 6&quot; return. Control goes to effective Y ± a.</td>
</tr>
<tr>
<td>TNR7</td>
<td>Y ± a, k</td>
<td>Transfer Non-Relative and Set Mark 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The next command in sequence is marked for a &quot;return to mark 7&quot; 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.</td>
</tr>
<tr>
<td>TSR</td>
<td>Y ± a, k</td>
<td>Transfer to Subroutine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>TSJ1</td>
<td>Y ± a, k</td>
<td>Transfer on Selective Jump Switch 1 On</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>TSJ2</td>
<td>Y ± a, k</td>
<td>Transfer on Selective Jump Switch 2 On</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This command operates similar to the one above except that jump switch two is used and the return is &quot;return to selective jump two&quot; (RSJ2).</td>
</tr>
<tr>
<td>TSJ3</td>
<td>Y ± a, k</td>
<td>Transfer on Selective Jump Switch 1 and 2 Both On</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If both jump switch one and two are ON, the next consecutive command is marked for a return to &quot;return to selective jump three (RSJ3)&quot;. Control goes to effective Y ± a. If either jump switch is OFF, the next command in sequence is executed.</td>
</tr>
<tr>
<td>CALL</td>
<td>Y ± a, k</td>
<td>Transfer to Machine Language Subroutine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Control is transferred to the machine language subroutine which starts at effective location Y ± a.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>TBR</strong></td>
<td>Transfer Back Relative</td>
<td></td>
</tr>
<tr>
<td>Y ± a, k</td>
<td>Control goes backwards Y ± a effective locations regardless of the addressing mode. No mark is set.</td>
<td></td>
</tr>
<tr>
<td><strong>STOANR</strong></td>
<td>Store A. N. R.</td>
<td></td>
</tr>
<tr>
<td>Y ± a, k</td>
<td>The contents of A. N. R. are copied into effective Y ± a and effective Y ± a + 1 as alphanumeric data. A. N. R. is unchanged.</td>
<td></td>
</tr>
<tr>
<td><strong>STO</strong></td>
<td>Store Accumulator</td>
<td></td>
</tr>
<tr>
<td>Y ± a, k</td>
<td>The contents of the accumulator are copied into effective Y ± a and effective Y ± a + 1. The accumulator remains unchanged.</td>
<td></td>
</tr>
<tr>
<td><strong>NOP</strong></td>
<td>No Operation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This command may be used to reserve a location to be filled conditionally. No operation is performed. The computer proceeds to the next location.</td>
<td></td>
</tr>
<tr>
<td><strong>STOP1</strong></td>
<td>Selective Stop 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td><strong>STOP2</strong></td>
<td>Selective Stop 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Same as Selective Stop 1 except under control of Switch 2, Panel Display: A = 0002.</td>
<td></td>
</tr>
<tr>
<td><strong>HALT3</strong></td>
<td>Stop Display 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computation stops with Panel Display: A = 0003. It may be resumed at the next sequential location by cycling the run switch.</td>
<td></td>
</tr>
<tr>
<td><strong>HALT4</strong></td>
<td>Stop Display 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Same as Stop Display 3 only A = 0004.</td>
<td></td>
</tr>
<tr>
<td><strong>HPRIN</strong></td>
<td>Halt and Await Input</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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, &quot;Operating Modes.&quot;)</td>
<td></td>
</tr>
<tr>
<td><strong>REL</strong></td>
<td>Select Relative Mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All subsequent commands are interpreted in the relative addressing mode. Termination is by execution of a &quot;SELECT ABSOLUTE MODE&quot; (ABS), a master clear and run or &quot;Halt and Await Input&quot;.</td>
<td></td>
</tr>
</tbody>
</table>
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:
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.

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 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.
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.

PXAIB 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.

PXAIN 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.

PXAIL 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.

PXAJB 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.

PXAJD 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.

PXAJL 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>BLCOPY</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 MNEPOMIC</th>
<th>SICOM FORMAT</th>
<th>SICOM MANUAL PAGE #</th>
<th>SICOSAS APPENDIX A PAGE #</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIB</td>
<td>*K 06 ADDR</td>
<td>08</td>
<td>19</td>
</tr>
<tr>
<td>IJB</td>
<td>*K 16 ADDR</td>
<td>08</td>
<td>19</td>
</tr>
<tr>
<td>IOCTAL</td>
<td>K 52 ADDR</td>
<td>17</td>
<td>23</td>
</tr>
<tr>
<td>ISPEC</td>
<td>0 00 0029</td>
<td>31</td>
<td>28</td>
</tr>
<tr>
<td>ISUB</td>
<td>*K 33 ADDR</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>ISUBC</td>
<td>8 01 ADDR</td>
<td>34</td>
<td>31</td>
</tr>
<tr>
<td>ITAPE</td>
<td>K 00 0021</td>
<td>28</td>
<td>27</td>
</tr>
<tr>
<td>ITYPE</td>
<td>0 00 0013</td>
<td>25A</td>
<td>26</td>
</tr>
<tr>
<td>LDANR</td>
<td>*K 54 ADDR</td>
<td>18</td>
<td>23</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</td>
<td>*K 25 ADDR</td>
<td>09</td>
<td>20</td>
</tr>
<tr>
<td>MYPY</td>
<td>5 01 ADDR</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>MRGANR</td>
<td>*K 56 ADDR</td>
<td>19</td>
<td>23</td>
</tr>
<tr>
<td>NAME</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>NOP</td>
<td>0 00 0000</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>OALPHA</td>
<td>K 53 ADDR</td>
<td>17</td>
<td>23</td>
</tr>
<tr>
<td>OCMND</td>
<td>K 45 ADDR</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>OCBRTB</td>
<td>K 43 ADDR</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>OFLCR</td>
<td>K 37 ADDR</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>OFLEX</td>
<td>0 00 0012</td>
<td>25A</td>
<td>26</td>
</tr>
<tr>
<td>OFLTB</td>
<td>K 36 ADDR</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>OFORM</td>
<td>K 42 ADDR</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>OFXCR</td>
<td>K 41 ADDR</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>OFXTB</td>
<td>K 40 ADDR</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>OLOC</td>
<td>0 00 0075</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>OOCNTAL</td>
<td>K 00 0027</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>OPRINT</td>
<td>0 00 0016</td>
<td>25A</td>
<td>26</td>
</tr>
<tr>
<td>ORG</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>OOSPEC</td>
<td>K 00 0028</td>
<td>31</td>
<td>28</td>
</tr>
<tr>
<td>OTAPE</td>
<td>K 00 0020</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>OTBNO</td>
<td>K 44 ADDR</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>OTYPE</td>
<td>0 00 0014</td>
<td>25A</td>
<td>26</td>
</tr>
<tr>
<td>PAXIB</td>
<td>K 00 0223</td>
<td>29</td>
<td>32</td>
</tr>
<tr>
<td>PAXID</td>
<td>K 00 0323</td>
<td>29</td>
<td>32</td>
</tr>
<tr>
<td>PAXIL</td>
<td>K 00 0423</td>
<td>29</td>
<td>32</td>
</tr>
<tr>
<td>PAXJB</td>
<td>K 00 1223</td>
<td>29</td>
<td>32</td>
</tr>
<tr>
<td>PAXJD</td>
<td>K 00 1323</td>
<td>29</td>
<td>32</td>
</tr>
<tr>
<td>PAXJL</td>
<td>K 00 1423</td>
<td>29</td>
<td>32</td>
</tr>
<tr>
<td>PUSTOP</td>
<td>0 00 0079</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>PXAIB</td>
<td>K 00 0222</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td>PXAID</td>
<td>K 00 0322</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td>PXAIL</td>
<td>K 00 0422</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td>PXAJB</td>
<td>K 00 1222</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td>PXAJD</td>
<td>K 00 1322</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td>PXAJL</td>
<td>K 00 1422</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td>RAD</td>
<td>*K 34 ADDR</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>REL</td>
<td>0 00 0006</td>
<td>24</td>
<td>25</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>SJJ</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 70 ADDR</td>
<td>21</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>TZE *K 60 ADDR</td>
<td>20</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>WRFILE K 46 ADDR</td>
<td>12</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>XEQ X 01 ADDR</td>
<td>34</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>ZEROIR 0 00 0010</td>
<td>25</td>
<td>26</td>
<td></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>Y±a</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACJ</td>
<td>Y±a</td>
<td>Set direct, indirect, and relative bank control and jump</td>
</tr>
<tr>
<td>ADB</td>
<td>Y±a</td>
<td>Add backward</td>
</tr>
<tr>
<td>ADC</td>
<td>Y±a</td>
<td>Add constant</td>
</tr>
<tr>
<td>ADD</td>
<td>Y±a</td>
<td>Add direct</td>
</tr>
<tr>
<td>ADF</td>
<td>Y±a</td>
<td>Add forward</td>
</tr>
<tr>
<td>ADI</td>
<td>Y±a</td>
<td>Add indirect</td>
</tr>
<tr>
<td>ADM</td>
<td>Y±a</td>
<td>Add memory</td>
</tr>
<tr>
<td>ADN</td>
<td>Y±a</td>
<td>Add no address</td>
</tr>
<tr>
<td>ADR</td>
<td>Y±a</td>
<td>Add relative</td>
</tr>
<tr>
<td>ADS</td>
<td></td>
<td>Add specific</td>
</tr>
<tr>
<td>AOB</td>
<td>Y±a</td>
<td>Replace add one backward</td>
</tr>
<tr>
<td>AOC</td>
<td>Y±a</td>
<td>Replace add one constant</td>
</tr>
<tr>
<td>AOD</td>
<td>Y±a</td>
<td>Replace add one direct</td>
</tr>
<tr>
<td>AOF</td>
<td>Y±a</td>
<td>Replace add one forward</td>
</tr>
<tr>
<td>AOI</td>
<td>Y±a</td>
<td>Replace add one indirect</td>
</tr>
<tr>
<td>AOM</td>
<td>Y±a</td>
<td>Replace add one memory</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</td>
</tr>
</tbody>
</table>

MANUAL #145 PAGE
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATE Y ± a</td>
<td>A to buffer entrance register.</td>
<td>25</td>
</tr>
<tr>
<td>ATX Y ± a</td>
<td>A to buffer exit register.</td>
<td>25</td>
</tr>
<tr>
<td>BLS Y ± a</td>
<td>Block store.</td>
<td>24</td>
</tr>
<tr>
<td>BSS</td>
<td>Block starting with symbol.</td>
<td>+</td>
</tr>
<tr>
<td>CBC</td>
<td>Clear buffer controls</td>
<td>39</td>
</tr>
<tr>
<td>CIL</td>
<td>Clear interrupt lockout.</td>
<td>40</td>
</tr>
<tr>
<td>CTA</td>
<td>Bank controls to A.</td>
<td>26</td>
</tr>
<tr>
<td>DRJ Y ± a</td>
<td>Set direct and relative bank control and jump.</td>
<td>36</td>
</tr>
<tr>
<td>END</td>
<td>End compilation</td>
<td>+</td>
</tr>
<tr>
<td>EQU</td>
<td>Equality</td>
<td>+</td>
</tr>
<tr>
<td>ETA</td>
<td>Buffer entrance register to A</td>
<td>25</td>
</tr>
<tr>
<td>ERR</td>
<td>Error stop.</td>
<td>45</td>
</tr>
<tr>
<td>EXC Y ± a</td>
<td>External function constant.</td>
<td>44</td>
</tr>
<tr>
<td>EXF Y ± a</td>
<td>External function forward.</td>
<td>44</td>
</tr>
<tr>
<td>HLT</td>
<td>Halt</td>
<td>45</td>
</tr>
<tr>
<td>HWI Y ± a</td>
<td>Half write indirect.</td>
<td>29</td>
</tr>
<tr>
<td>IBI Y ± a</td>
<td>Initiate buffer input.</td>
<td>40</td>
</tr>
<tr>
<td>IBO Y ± a</td>
<td>Initiate buffer output.</td>
<td>41</td>
</tr>
<tr>
<td>INA</td>
<td>Input to A.</td>
<td>43</td>
</tr>
<tr>
<td>IRJ Y ± a</td>
<td>Set indirect and relative bank control and jump.</td>
<td>36</td>
</tr>
<tr>
<td>JFI Y ± a</td>
<td>Jump forward indirect.</td>
<td>39</td>
</tr>
<tr>
<td>JPI Y ± a</td>
<td>Jump indirect.</td>
<td>38</td>
</tr>
<tr>
<td>JPR Y ± a</td>
<td>Return jump.</td>
<td>38</td>
</tr>
<tr>
<td>LCB Y ± a</td>
<td>Load complement backward.</td>
<td>28</td>
</tr>
<tr>
<td>LCC Y ± a</td>
<td>Load complement constant.</td>
<td>28</td>
</tr>
<tr>
<td>LCD Y ± a</td>
<td>Load complement direct.</td>
<td>28</td>
</tr>
<tr>
<td>LCF Y ± a</td>
<td>Load complement forward.</td>
<td>28</td>
</tr>
<tr>
<td>LCI Y ± a</td>
<td>Load complement indirect.</td>
<td>28</td>
</tr>
<tr>
<td>LCM Y ± a</td>
<td>Load complement memory.</td>
<td>28</td>
</tr>
<tr>
<td>LCN Y ± a</td>
<td>Load complement no address.</td>
<td>28</td>
</tr>
<tr>
<td>LCR Y ± a</td>
<td>Load complement relative.</td>
<td>*</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>Load backward.</td>
<td>27</td>
</tr>
<tr>
<td>LDC</td>
<td>Load constant.</td>
<td>27</td>
</tr>
<tr>
<td>LDD</td>
<td>Load direct.</td>
<td>27</td>
</tr>
<tr>
<td>LDF</td>
<td>Load forward.</td>
<td>27</td>
</tr>
<tr>
<td>LDI</td>
<td>Load indirect.</td>
<td>27</td>
</tr>
<tr>
<td>LDM</td>
<td>Load memory.</td>
<td>27</td>
</tr>
<tr>
<td>LDN</td>
<td>Load no address.</td>
<td>27</td>
</tr>
<tr>
<td>LDR</td>
<td>Load relative.</td>
<td>27</td>
</tr>
<tr>
<td>LDS</td>
<td>Load specific.</td>
<td>27</td>
</tr>
<tr>
<td>LPB</td>
<td>Logical product backward.</td>
<td>34</td>
</tr>
<tr>
<td>LPC</td>
<td>Logical product constant.</td>
<td>34</td>
</tr>
<tr>
<td>LPD</td>
<td>Logical product direct.</td>
<td>34</td>
</tr>
<tr>
<td>LPF</td>
<td>Logical product forward.</td>
<td>34</td>
</tr>
<tr>
<td>LPI</td>
<td>Logical product indirect.</td>
<td>34</td>
</tr>
<tr>
<td>LPM</td>
<td>Logical product memory.</td>
<td>34</td>
</tr>
<tr>
<td>LPN</td>
<td>Logical product no address.</td>
<td>34</td>
</tr>
<tr>
<td>LPR</td>
<td>Logical product relative</td>
<td>27</td>
</tr>
<tr>
<td>LPS</td>
<td>Logical product specific.</td>
<td>34</td>
</tr>
<tr>
<td>LSB</td>
<td>Logical sum backward.</td>
<td>35</td>
</tr>
<tr>
<td>LSD</td>
<td>Logical sum direct.</td>
<td>35</td>
</tr>
<tr>
<td>LSF</td>
<td>Logical sum forward.</td>
<td>35</td>
</tr>
<tr>
<td>LSI</td>
<td>Logical sum indirect.</td>
<td>35</td>
</tr>
<tr>
<td>LSN</td>
<td>Logical sum no address.</td>
<td>35</td>
</tr>
<tr>
<td>LSR</td>
<td>Logical sum relative.</td>
<td>27</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>Negative jump backward.</td>
<td>37</td>
</tr>
<tr>
<td>Instruction</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>37</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>Page</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>SBS</td>
<td>Subtract specific.</td>
<td>30</td>
</tr>
<tr>
<td>SBU</td>
<td>$Y \pm a$ Set buffer bank control.</td>
<td>36</td>
</tr>
<tr>
<td>SCB</td>
<td>$Y \pm a$  (LSB) Selective complement backward.</td>
<td>35</td>
</tr>
<tr>
<td>SCC</td>
<td>$Y \pm a$ Selective complement constant.</td>
<td>35</td>
</tr>
<tr>
<td>SCD</td>
<td>$Y \pm a$  (LSD) Selective complement direct.</td>
<td>35</td>
</tr>
<tr>
<td>SCF</td>
<td>$Y \pm a$  (LSF) Selective complement forward.</td>
<td>35</td>
</tr>
<tr>
<td>SCI</td>
<td>$Y \pm a$  (LSI) Selective complement indirect.</td>
<td>35</td>
</tr>
<tr>
<td>SCM</td>
<td>$Y \pm a$ Selective complement memory.</td>
<td>35</td>
</tr>
<tr>
<td>SCN</td>
<td>$Y \pm a$  (LSN) Selective complement no address.</td>
<td>35</td>
</tr>
<tr>
<td>SCR</td>
<td>$Y \pm a$  (LSR) Selective complement relative.</td>
<td>*</td>
</tr>
<tr>
<td>SCS</td>
<td>Selective complement specific.</td>
<td>35</td>
</tr>
<tr>
<td>SDC</td>
<td>$Y \pm a$ Set direct bank control.</td>
<td>36</td>
</tr>
<tr>
<td>SIC</td>
<td>$Y \pm a$ Set indirect bank control.</td>
<td>36</td>
</tr>
<tr>
<td>SID</td>
<td>$Y \pm a$ Set indirect and direct bank control.</td>
<td>36</td>
</tr>
<tr>
<td>SLJ</td>
<td>$Y \pm a$ Selective jump.</td>
<td>45</td>
</tr>
<tr>
<td>SLS</td>
<td>$Y \pm a$ Selective stop.</td>
<td>45</td>
</tr>
<tr>
<td>SJS</td>
<td>$Y \pm a$ Selective stop and jump.</td>
<td>45</td>
</tr>
<tr>
<td>SRB</td>
<td>$Y \pm a$ Shift replace backward.</td>
<td>33</td>
</tr>
<tr>
<td>SRC</td>
<td>$Y \pm a$ Shift replace constant.</td>
<td>33</td>
</tr>
<tr>
<td>SRD</td>
<td>$Y \pm a$ Shift replace direct.</td>
<td>33</td>
</tr>
<tr>
<td>SRF</td>
<td>$Y \pm a$ Shift replace forward.</td>
<td>33</td>
</tr>
<tr>
<td>SRI</td>
<td>$Y \pm a$ Shift replace indirect.</td>
<td>33</td>
</tr>
<tr>
<td>SRJ</td>
<td>$Y \pm a$ Set relative bank control and jump.</td>
<td>36</td>
</tr>
<tr>
<td>SRM</td>
<td>$Y \pm a$ Shift replace memory.</td>
<td>33</td>
</tr>
<tr>
<td>SRR</td>
<td>$Y \pm a$ Shift replace relative.</td>
<td>*</td>
</tr>
<tr>
<td>SRS</td>
<td>Shift replace specific.</td>
<td>33</td>
</tr>
<tr>
<td>STB</td>
<td>$Y \pm a$ Store backward.</td>
<td>28</td>
</tr>
<tr>
<td>STC</td>
<td>$Y \pm a$ Store constant.</td>
<td>28</td>
</tr>
<tr>
<td>STD</td>
<td>$Y \pm a$ Store direct.</td>
<td>28</td>
</tr>
<tr>
<td>STE</td>
<td>$Y \pm a$ Store buffer entrance register at Location 6X and transfer A to buffer entrance register.</td>
<td>27</td>
</tr>
<tr>
<td>STF</td>
<td>$Y \pm a$ Store forward.</td>
<td>28</td>
</tr>
<tr>
<td>STI</td>
<td>$Y \pm a$ Store indirect.</td>
<td>28</td>
</tr>
<tr>
<td>STM</td>
<td>$Y \pm a$ Store memory.</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>STP</strong></td>
<td>Y ± a</td>
<td>Store P at location 5X.</td>
</tr>
<tr>
<td><strong>STR</strong></td>
<td>Y ± a</td>
<td>Store relative.</td>
</tr>
<tr>
<td><strong>STS</strong></td>
<td></td>
<td>Store specific.</td>
</tr>
<tr>
<td><strong>ZJB</strong></td>
<td>Y ± a</td>
<td>Zero jump backward.</td>
</tr>
<tr>
<td><strong>ZJF</strong></td>
<td>Y ± a</td>
<td>Zero jump forward.</td>
</tr>
<tr>
<td><strong>ZJR</strong></td>
<td>Y ± a</td>
<td>Zero jump relative.</td>
</tr>
<tr>
<td><em>(Col. 1)</em></td>
<td></td>
<td>REMARKS Card.</td>
</tr>
</tbody>
</table>
APPENDIX D

THIS IS A 7090 TRANSFORMATION OF A SICOM PROGRAM WRITTEN FOR THE 160-A COMPUTER BY J.O. DRIYAN

MLOC SILOC K OP ADDR SYMBOL OPCODE VARIABLE FIELD COMMENTARY PAGE

0000 0000
0000 0000 0 02 0010
0002 0001 1 03 0004
0004 0002 1 04 0008
0006 0003 1 13 0100
0008 0004 1 14 0100
0010 0005 2 03 0002
0012 0006 2 04 0004
0014 0007 3 03 0002
0016 0008

ORG 0

START ZEROIR

ZERO ALL COMPONENTS OF ALL INDEX REGISTERS

SID 4,1
SET I.R. 1 TO PICK UP X AND Y VALUES

SIL 8,1

SJF 100,1

SFL 100,1

SID 2,2
SET I.R. 2 TO PICK UP Z VALUES (DIVISORS)

SID 2,3
SET I.R. 3 FOR STORING RESULTS

NOTE THAT IT IS NOT NECESSARY TO SET THE 1 LIMIT OF INDEX REGISTER 3

BECAUSE THE NEXT COMMAND IN SEQUENCE IS TO BE EXECUTED REGARDLESS OF

THE INCREMNTING COMPARISON

0020 0018 2 64 1502

HALT3

0046 0019 2 64 1502

TSM4 NXTJOB DUMMY OJT

4540 1220 ORG 1200 DATA IS STORED HERE

4540 1200

4544 1202

4550 1204

4554 1206

4560 1208

4564 1210

5050 1300 ORG 1300 SECOND PART OF DATA STORED HERE

DATA DEC 7.9E+20 X4

DEC 8.8E-19 Y4

DEC 9.7E+18 X5

DEC 10.6E+17 Y5

DEC 11.5E+16 X6

DEC 12.6E+15 Y6

5360 1400 ORG 1400

5364 1402

5368 1402

5372 1404

5672 1500 ORG 1500 RESULTS TO BE STORED HERE

5672 1500

5674 1502 0 02 0004 NXTJOB HALT4 END START
APPENDIX E

THIS IS A 7090 TRANSFORMATION OF A SICOM PROGRAM WRITTEN FOR THE 160-A COMPUTER BY ANYONE U. WIS IN DEC. 20, 1962
MLOC SILOC KOP ADDR SYMBOL Opcode VARIABLE FIELD COMMENTARY PAGE 1

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

0310 0102 ORG 170
0310 0100 3 64 0122 BEGIN TSM3 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 RHO USING A, E AND V
0316 0103 0 74 0214 CALL OUTPUT MACHINE LANGUAGE OUTPUT SUBROUTINE
0320 0104 3 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 0208 MPY A SE\-MAJ\-
0332 0109 0 77 0110 STO DIVDN A(1-EX2)
0334 0110 0 22 0212 CLA V TRUE ANOMOLY
0336 0111 0 72 0126 TSR COSXV FIND COSINE V
0340 0112 0 25 0210 MPY E E COS(V)
0342 0113 6 01 0001 ADDC 1 1*E COS(V)
0344 0114 0 23 0118 DIVP DIVND DIVIDE INTO A(1-EX2)
0346 0115 0 77 0120 STO RHO STORE ANSWER
0350 0116 0 02 0065 RSM5 RETURN TO MAIN PROGRAM
0354 0118 DIVND DEC $.
0360 0120 RHO DEC $.
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 COSXV SIN SIN-COS SUBROUTINE ENTRY +D =COSINE
0640 0206 A DEC $.
0644 0210 E DEC $.
0650 0212 V DEC $.
MLAP
0654 7500 OUTPUT EXC 2402 EXTERNAL DEVICE
0655 4542
0656 7301 OUT 1 OUTPUT LOCATION RHO+3
0657 0363 RHO+3
0660 RETURN BACK TO MAIN PROGRAM
* 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...

```
ORG 15200
START  HPR
  OTPEND OTPEND
+1  STL  22
+4  SNT  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
  IIIMAGE 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  TIX  *-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
+9 STL 22
CSCRIB MEDTAP
+13 STL 22
IBCC TEMP,,1,1 WHAT KIND OF CARD

SKIP STL 43
+4 CLA TEMP TEST FOR COMMENT
+5 SUB STAR
+6 TZE READ2 IS COMMON IGNORE IT
IBCW CODE,,8,1 LOOK AT OP CODE

+7 STL 43
+11 CLA CCDE
+12 CAS SSCR
+13 TRA **+2
+14 TRA FCGRQR
+15 CAS SLOG
+16 TRA **+2
+17 TRA FCRLCG
+18 CAS SEXP
+19 TRA **+2
+20 TRA FCREXP
+21 CAS SSIN
+22 TRA **+2
+23 TRA FCRSIN
+24 CAS SATN
+25 TRA **+2
+26 TRA FCRATN
+27 CAS SICOM
+28 TRA CASG0B IS SICOM CARD
+29 TRA **+2
+30 TRA CASG0B
+31 STZ MCDE
+32 TRA READ2 CASGCB CAS GCBACK CHECK RETURN CARD
+1 TRA CAS160 IS IGNORE FIRST
+2 TRA **+2
+3 TRA CAS160
+4 CLA ICNTR
+5 ADD TWO
+6 STO ICNTR
+7 TRA ISOP CAS160 CAS 160A MACHINE MODE
+1 TRA CASNAM
+2 TRA **+2
+3 TRA CASNAM
+4 STL MCDE
+5 TRA READ2 CASNAM
+1 TRA CASCRG
+2 TRA **+2
+3 TRA CASCRG
IBCW HEAD1+14,,16,6
+4 STL 43
+8 TRA READ2 CASCRG CAS ORIGIN

46
+1 TRA ++2
+2 TRA ISORG
+3 CAS DECDAT
+4 TRA ++2
+5 TRA ISDEC
+6 CAS EXTMSK
+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 MOD2
+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 MCDE
+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 TIX ++1,,1,1
+7 SXA SYMCNT,1
+8 CLA ICNTR
AGN ADD TWO
+1 ZET MCDE
+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
+6  TXI  *+1,1,-1
+7  SXA  SYMCNT,1
    IINT  TEMP+3,16
+8  STL  43
+12  CLA  TEMP+3
+13  ZET  MCDE
+14  TRA  *+2
+15  ALS  1
+16  ADD  ICNTR
+17  STO  ICNTR
+18  TRA  READ2
    IINT  ICNTR,16

ISORG
+4  CLA  ICNTR
+5  ALS  1
+6  STO  ICNTR
+7  TRA  READ2

ISDEC
+1  TRA  ISOP
+2  SLN  1
+3  TSX  EVENUP,4
+4  TRA  ISOP
    IBCW  DYNAMO,,16,1

ISEND
+4  REWB  7
+5  TSX  FNDSYM,4
+6  TNZ  *+2
+7  CLA  LCCTAB
+8  ARS  1
+9  STA  G2G
+10  CLA  ATE
+11  STO  ICNTR
    CBLANK  120,,1
+12  STL  22
    CREADY
+15  STL  22
    CBLANK  120,,1
+17  STL  22
    CEOR  CUMP,,132
+20  STL  22
    IFILE  NMG1
+24  STL  43
+27  AXT  4096,4
+28  STZ  SIMAP+4096,4
+29  TIX  *-1,4,1
+30  AXT  56,4
+31  CLA  BLANK
+32  STO  INPUT+56,4
+33  TIX  *-1,4,1
+34  WPDA
+35  SPR  1
    CHEAD  CASE1,,1,,1
+36  STL  22
+41  STZ  PAGENC
+42 STZ MCDE
IMAGE1 EQU CUTPUT+21
IMAGE2 EQU IMAGE1+28
-4 STL 43
ICHAR 0
ISCRIB MDTAP
NOWGO STL 43

IREADY
+4 STL 43
IBCW TEMP,,8,1
+6 STL 43
+10 CLA TEMP
+11 SUB BLANK
+12 TZE NOWGO

IREADY
+13 STL 43
IFILE CYCLE
+15 STL 43
+18 CLA ONE
+19 STO NUMBER
ISCRIB MDTAP

CYCLE STL 43
IBCW WORK,,1,12
+4 STL 43
OBCW WORK,,28,12
+8 STL 22
OINT NUMBER,,115,4
+12 STL 22
+16 CLA NUMBER
+17 ADD ONE
+18 STO NUMBER
+19 STZ TEMP
IBCC TEMP,,1,1

TEST FOR COMMENT CARD
+20 STL 43
+24 CLA TEMP
+26 TZE CMTCRD

FOUND ONE
IBCW CODE,,8,1

TEST NOW FOR
+27 STL 43
+31 CLA CODE
+32 CAS SSQR
+33 TRA **2
+34 TRA SCRFOR
+35 CAS SLOG
+36 TRA **2
+37 TRA LCGFOR
+38 CAS SEXP
+39 TRA **2
+40 TRA EXPFOR
+41 CAS SSIN
+42 TRA **2
+43 TRA SINFOR
+44 CAS SATN
+45 TRA **2
FOUND A RETURN CARD

ORIGIN CARD

DEC DATA CARD

ORIGIN CARD

DEC DATA CARD
+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 GGOLOOK,1,1
+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
+6 PZE 6TYPE
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 ENDKEY,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
ICCHAR YESSYM,,8
TYPEØ STL 43
+4 STL 43
ICCHAR 0
+8 STL 43
+12 TSX VARFRM,4
EQU CARD
END CARD
IS MACHINE CODE
GOLOOK CAS LSTINS,1
E G BLS 0100 YYYY
E G PTA 0101
E G STP 015X
E G LDN 04XX
E G SLJ 77XX
RELATIVE
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
KOA2D2R PXA PAX
KOP=OP EG CLAC GO ETC
ICHAR YESSYM,,8
IINT SIADDR,,16,6
+4 STL 43
ICCHAR 0
+8 STL 43
+12 TSX VARFRM,4
+13 NOP
+14 TRA NCPE
ICHAR 0

YESSYM STL 43
+4 TSX FINDSYM,4 WAS IT AC SYMBOL
+5 ARS 1 NO
+6 STO SIADDR $00000
+7 SLT 2 FOR BINARY ROUTINE
+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
ALMST0 TSX CNVSAO,4
ASIF0 TSX OKTODR,4
+1 TSX LCCMCH,4 MOVE K OP ADDR TO OUTRAN OUTPUT REGI0
+2 TSX LCCSIC,4 PUT MACHINE LOCATION IN OUTRAN REGION
+3 TSX WRLIST,4 WRITE ONE LINE ON LISTING TAPE
+4 SLT 3
+5 TRA GCAHEG
+6 CLA RETURN
+7 ADD ONE
+8 STA *+1
+9 TRA *+0
GOAHEO 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 ZERO K O P A D FOR TYPE 1 INSTR
+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 MOVE 00000DR TO OUTPUT REGION
+12 TSX LCCMCH,4
+13 TSX LCCSIC,4
+14 TSX WRLIST,4
+15 LDQ D2
+16 MPY DECIMAL+2
+17 XCA
DONE THIS CARD
SEL MAC TAPE

TYPE2
STZ C
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
STG TEMP
LDQ C2
MPY DECIMAL+2
XCA
ADD R
ADD TEMP
STO SIADDR
TRA ASIF0

TYPE3
TSX WIPE+1
TSX VARFRM+4
NOP
LDQ RITEKY
CLA ZERC
RGL 6
LGL 6
STC D2
CLA ZERC
LGL 6
STO R
LDQ D2
MPY DECIMAL+2
XCA
ADD R
STC SIADDR
TRA ASIF0

TYPE4
TSX WIPE+1
CLA ZERC
STO D2
CLA NINE
STO R

INT AD+,16
STL 43
LDQ AC
CLA ZERC
CVH DECIMAL+2
STQ A
STO D1
+14  XCA
+15  ALS  3
+16  STO  K
+17  LDQ  D1
+18  MPY  DECMAL+1
+19  XCA
+20  ADD  D19
+21  STO  SIADDR
+22  TRA  ASIF0

TYPES  TSX  WIPE,4
+1  TSX  VARFRM,4
+2  NOP
+3  CLA  ZERO
+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  DECMAL+3,4
+2  STQ  TEMP+3,4
+3  TIX  LDQA,4,1
+4  CLA  R
+5  ADD  TEMP
+6  ADD  TEMP+1
+7  ADD  TEMP+2
+8  STO  SIADDR
+9  TRA  ASIF0

TYPE6  TSX  WIPE,4
+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
+4  STL  43
  IINT  SIADDR,,16,6
+8  STL  43
  ICHAR  0
+12  STL  43
+16  TRA  SEEIF

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

KOP = OP  ADDR
ADDR MAY BE SYMBOLIC OR A CONSTANT
ADDR IS SYMBOLIC
SUBROUTINE LINKAGE

SET UP HOLLERITH K,O,P

SUBROUTINE LINKAGE

CONVER BINARY SICOM ADDR TO BCD
SUBROUTINE LINKAGE

CONVER LOCATION COUTER IN OUTPUT REGIO
OCTAL 10000

SUBROUTINE LINKAGE

DECIMAL 10000

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

SUBROUTINE LINKAGE
+4 STO NOCODE
  OREADY
+5 STL 22
    OBCW NOCODE,,1,20
+7 STL 22
+11 TSX ERROR,4
+12 AXT 0,1
+13 ZET MCDE
+14 TRA THANX
+15 TRA FNDOP

* THIS ROUTINE WILL SET UP TWO REGISTERS COMPRISING A SICOM COMMAND
  BEGIN 1,7
STOCMD TXL **5,0,0
  SUBROUTINE LINKAGE
  EVEN LOCATION = SIADDR
+8 CLA WHRMAP
+9 ADD ICNTR
+10 STA STOREV
  ODD LOCATION = KOP
+11 ADD ONE
+12 STA STOROD
+13 CLA SIADDR
STOREV STO **0
+1 SLT 2
  IS AC
+2 TRA NOTAC
  NO
+3 CLA ABLNK
  YES 00000B
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
  IS XEQ,SETFWA,SETLWA
+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
  SUBROUTINE LINKAGE
  SPECIAL RETURN
+8 TSX VARGRM,4
+9 TRA **2
+10 TRA NCRMAL
+11 CLA MYOWN
+12 ALS 1
RETURN FNDSYM
+13 TRA FNDSYM+1
NORMAL AXT 0,4
+1 LAC SYMCTN,1
  NUMBER SYMBOLS STORED THUS FAR
+2 CLA AFIELD
+3 CAS ACCODE
+4 TRA **2
+5 TRA ISAC
LCCKAT CAS SYMTAB,4
+1 TRA **+2
+2 TRA FCUND
+3 TXI **+1,4,-1
+4 TIX LCOKAT,1,1
+5 STO UNDEF
SHORT
+6 STL 22
OBCW UNDEF,1,20
+8 STL 22
+12 TSX ERRCR,4
+13 CLA ZERO
+14 TRA FOUNDX
ISAC SLN 2
+1 CLA ACNUM
RETURN FNDSYM
+2 TRA FNDSYM+1
FOUND CLA LOCTAB,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 EUDSYM
FOUNDX TRA FNDSYM+1
STUPID STO DCOUBLE
SHORT
+1 STL 22
OBCW DCOUBLE,1,20
+3 STL 22
+7 TSX ERROR,4
+8 TRA FINE
BEGIN 1,4
ERROR TXL **+3,0,0
OSCRIB CUTAPE,,20,1
+4 STL 22
+8 SWT 3
RETURN ERROR
+9 TRA ERROR+1
SHORT
+10 STL 22
OSCRIB PRINTR,,20
+12 STL 22
RETURN ERROR
+16 TRA ERROR+1
BEGIN 2,7
VARFRM TXL **+5,0,0
IMASK ,,31
SUBROUTINE LINKAGE
+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  TIX  ++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  FNDAAD
    +4  CLA  ISNEG
    +5  CAS*  REFADD
    +6  TRA  ++2
    +7  TRA  FNDAAD
    +8  TIX  ++1,1,1
    +9  TIX  LCG2,4,1
    +10  AXT  0,1
FNDAAD  SXA  ADDCOL,1
    +1  LXA  BNKCOL,4
    +2  AXT  0,1
    +3  CLA  ISCOM
LOG3  CAS*  REFADD
    +1  TRA  ++2
    +2  TRA  FNDCOM
    +3  TIX  ++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
+1 TRA YESADD
  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 ADDCOL
  ISOLATE SYMBOL FROM + OR -
INTPL TPL **2
  +1 CLA ZERC ERROR
  +2 XCA
  +3 MPY SIXX
  +4 XCA
  +5 STA SHFTR (6-C)X6 = NO BITS TO RIGHT OF SYMOL
  +6 LXA ADDCOL,1 NO CHARACTERS IN SYMBOL
  +7 AXT 0,2
  +8 CLA ZERC
  +9 CLA ZERC
  +10 ZET CCMCOL
  +11 TRA **2
  +12 CLA BNKCOL
  +13 SUB ADDCOL
  +14 ALS 18
  +15 STD IFDEC+3
  +16 STD IFOCT+3
  +17 ZET IMODE ZERO = DEC MODE
  +18 TRA ACKOCT
  ICHAR NONOCT,,8
+19 STL 43
  IINT ADDIT,,***,**
IFDEC STL 43
  +4 ZET MODE
  +5 TRA MOVEON
  +6 CLA ADDIT
  +7 ALS 1
  +8 STD ADDIT
  +9 MOVEON NZT CCMCOL IS THERE A TAG FIELD
  RETURN VARFRM SYMBOL SET IN AFIELD +OR- IN ADDIT
+1 TRA VARFRM+1
+2 TRA ISTAG
  ICHAR NONOCT,,8
ACKCCT  STL  43
          IOCTAL  ADDIT,,**,**
IFCCT  STL  43
          RETURN  VARFRM
+4  TRA  VARFRM+1
ICHAR  0
NONCCT  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
YSOCMA  ZET  ADDCOL
+1  TRA  YESADD
+2  TRA  HARD1
ISTAG  CLA  COMCOL
+1  ADD  COLDF
+2  ADD  ONE
+3  STA  PROTAG+3
          IINT  K,,**,1
PROTAG  STL  43
+4  CLA  K
+5  ALS  30
+6  SLW  INDEXR
+7  STL  1
          RETURN  VARFRM
+8  TRA  VARFRM+1
          RETURN  VARFRM,1
+9  AXT  1,,4
          ICHAR  0
          THIS CASE HAS NO ADD BUT TAG + MAYBE N
HARD1  STL  43
          ICHAR  ISALPH,,8
+4  STL  43
+8  CLA  COMCOL
+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  COMCOL
+6  STO  ADDCOL
+7  CLA  SIXX
+8  SUB  COMCOL
+9  TRA  INTPL
COMCOL  PZE
BNKCOL  PZE
ADDCOL  PZE

63
SUBROUTINE LINKAGE

ASSEMBLE + STORE DEC CARD COLS 16+

SUBROUTINE

ICOLR 16

UNPACK 15 COLS STARTING COL 16

STZ DATA+15,4

IBCC DATA+15,4,1

TIX UNPACK,4,1

IMASK 0

STL 43

CLA EXP0

STO EXPCNT

AXT 10,4

STZ DIGIT+10,4

TIX +1,4,1

STZ MANSGN

STZ EPART

STZ ISTNZ

STZ DPTIND

AXT 15,1

LOOK AT 15 COLUMNS

AXT 0,2

COLUMN COUNTER

AXT 0,4

DIGIT STORER

STZ DPTIND

CLA DATA+15,1

LOOK AT A COLUMN

CAS ISBLNK

TRA ++2

TRA GETCUT

FOUND A BLANK

CAS TEN

TRA GR10

ERROR

TNZ MARKIT

GR0 IS BETWEEN 1 AND 9

ZET 1STNZ

IS THIS A LEADING 0

TRA ++2

NO

TRA MCRCLM

YES IGNORE IT

MARKIT NZT 1STNZ

TRA 1STONE

64
STODIG STO DIGIT,4
+1 TXI **+1,4,-1
MORCLM TXI **+1,2,1
+1 TIX LOCKC,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 ISTNZ
+16 TRA STODIG
+17 TRA ISTONE
YESPOS STZ MANSGN
+1 TRA MORCLM
YESNEG STL MANSGN
+1 TRA MORCLM
YESDPT SXA DECCLM,2
+1 STL DPTIND
+2 TRA MORCLM
ISTONE SXA NZCOLM,2
+1 STL ISTNZ
+2 TRA STODIG
YESE SXA ECOLMN,2 E FOUND WILL TERMINATE SEARCH
+1 STL EPART
+2 LAC ECOLMN,4 ICOLR 17,4
+3 STL 43
ICCHAR ILEGAL,,8
+6 STL 43
IINT EEXP,,,3
+10 STL 43
GETOUT CLA DECCLM DECP T COLUMN IS LARGER
+1 CAS NZCOLM
+2 TRA DECGR OH REALLY
+3 HPR NON ZERO COLUMN IS LARGER
+4 CLA NZCOLM
+5 SUB DECCLM
+6 SUB ONE
+7 SSM
TESTE NZT EPART WAS THERE AN EPART
+1 TRA **+2 NO
+2 ADD EEXP YES ADD IN
+3 ADD EXPONT OCT 100
+4 STO EXPCNT
+5 TPL ASSMBL
+6 CLA EXP0
+7 SUB EXPCNT
+8 STO EXPCNT
+9 TRA ASSMBL
DECGR CLA DECELM
+1 SUB NZCOLM
+2 TRA TESTE
ASSMBL CLA ZERO
+1 ZET MANSGN
+2 CLA ONE
+3 ALS 7
+4 CRA EXPCNT
+5 ALS 4
+6 ORA DIGIT
EXPD1 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 CRA 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
ICHAR 0
ILEGAL STL 43
READY
+4 STL 22
CB CW BADDEC,,1,20
+6 STL 22
+10 TSX ERROR,4
+11 STZ MASKER
RETURN SCANIT
+12 TRA SCANIT+1
EVENUP CAL ICNTR
+1 ANA 2BITS
+2 SSM
+3 ADD D4
+4 ANA 2BITS
+5 ADD ICNTR

ASSEMBLE SIGN EXP + D1
NEG MANTISSA
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 TCOB *
+11 TRCB REWB
+12 REWB 8
  RETURN RITOUT
+13 TRA RITCUT+1

@TYPE CAL RITEKY
  +1 ARS 21
  +2 STO FOROUT
  +3 ANA M7777
  +4 STC FCMAP
  +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
ICHAR 0
BNKCRD STL 43
  ICHAR DILLY,8
  +4 STL 43
  +8 NZT IMODE
  +9 TRA QDECM
  IOCTAL TEMP,,16,4
+10 STL 43
+14 TRA PICKUP
  IINT TEMP,,16,4
QDECM STL 43
PICKUP CLA TEMP
  +1 ANA M7777

67
+2 TRA STOFOR
  ICHAR 0

CILLY STL 43
  +4 TSX FNDSYM,4
  +5 ANA M7777
  +6 TRA STOFCR

1TYPE CAL RITEKY
  +1 ARS 21

STFCR STO FCRCUT
  +1 ANA M7777
  +2 STO FCRMAP
  +3 TSX OLTCOD,4
  +4 TSX LCCMCH,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
    INT TEMP,,16,6
  +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 FNDSYM,4
  +5 STO TEMP
  +6 TRA CLAM7
    ICHAR 0

EG STD 015X

3TYPE STL 43
    ICHAR NILLY,,8
  +4 STL 43
  +8 KZT IMODE
  +9 TRA GCDEC
    ICOTAL TEMP,,16,6
  +10 STL 43
  +14 TRA CLAM77
    INT TEMP,,16,6

GCDEC STL 43

CLAM77 CLA M77
  +1 CAS TEMP
  +2 TRA CALRIT

WITHIN RANGE

STZTEM STZ TEMP
  +1 TSX OCRNG,4
  +2 TRA CALRIT
    ICHAR 0

NILLY STL 43
  +4 TSX FNDSYM,4
+2 INZ JCD
+6 STZ TEMP
+7 TRA CALRIT

JCD STO TEMP
+1 CAL RITEKY
+2 ANA TGMASK
+3 TZE CLAM77
+4 SUB 1TAG
+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 RITEKY
+4 ALS 12
+5 ACL RITEKY
+6 SLW RITEKY
+7 TRA CLAM77

BEGIN 1, 4

CUTCCD TXL *+3, 0, 0
OCTAL FOROUT, 20, 4
+4 STL 22
RETURN CUTCCD
+8 TRA OUTCCD+1
BEGIN 1, 4
MAPIT TXL *+3,0,0
+4 CLA WHRMAP
+5 ADD ICNTR
+6 STA *+2
+7 CLA FCRMAP
+8 STO **0
RETURN MAPIT
+9 TRA MAPIT+1
BEGIN 1,4
OCRNG TXL *+3,0,0
OREADY
+4 STL 22
OBCW RANGER,,1,20
+6 STL 22
+10 TSX ERRCR,4
RETURN CORNG
+11 TRA OCRNG+1
IANCHAR GOOD,,8
OCICRO 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 AXI 3,1
+1 TSX LCCMCH,4
+2 TSX WRLIST,4
CLABAC CLA BACKTO+3,1
+1 STO FCRMAP
+2 TSX MAPIT,4
+3 CLA ICNTR
+4 ADD ONE
+5 STO ICNTR
+6 TIX CLABAC,1,1
+7 TRA CYCLE
INSTR EQU ICNTR
DETAIL DETAIL
QSUBRA SSCR,BBB1,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 SXA SYMCNT,1
IANCHAR BBB1,,8
+7 STL 43
+8 TXL 44,0,12
+9 HTR BBB1,0
+10 PZE 8
  I1NT TEMP,1,6
+11 STL 43
+12 TXL 44,0,4
+13 STO TEMP,0
+14 PZE 1,6
CD1  CLA ENDSQR
   ADD INSTR
   STO INSTR
   TRA READ2
  IBCW SYMTAB,1,1,1
BBB1  STL 43
   +1 TXL 44,0,2
   +2 HTR SYMTAB,1
   +3 PZE 1,1
  ICHAR ALLBNK,2
   +4 STL 43
   +5 TXL 44,0,12
   +6 HTR ALLBNK,0
   +7 PZE 2
  ICHAR ISSYM,8
   +8 STL 43
   +9 TXL 44,0,12
   +10 HTR ISSYM,0
   +11 PZE 8
   +12 CLA ICNTR
   +13 STO LCCTAB,1
   +14 TXI *+1,1,-1
   +15 SXA SYMCNT,1
   +16 TRA DD1
LIST
QSUBRA SLOG,BBB2,ENDLOG,DD2
FORLOG TXE EVENUP,4
QSUBRA SEXP,BBB3,ENEXP,DD3
FOREXP TXE EVENUP,4
QSUBRA SSIN,BBB4,ENDSIN,DD4
FORSIN TXE EVENUP,4
QSUBRA SATN,BBB5,ENDATN,DD5
FORATN TXE EVENUP,4
DETAIL
QSUBRB ENDSQR
SQRFOR TXE EVENUP,4
   +1 CLA WHRMAP
   +2 ADD ICNTR
   +3 ADD ENDSQR
   +4 STA */+3
   +5 LXA ENDSQR,4
   +6 CLA ENDSQR,4
   +7 STO */0,4
   +8 TIX *-2,4,1
   +9 TXE LCOPCH,4
   +10 TXE LCSCIC,4

71
+11  TSX  WRLIST,4
+12  CLA  ENDSQR
+13  ADD  INSTR
+14  STO  INSTR
+15  TRA  CYCLE
LIST
QSUBRB  ENDLOG
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 EEEX FORMAT HAS BEEN REPLACED BY ZERO...****
+18  BCI  2,**********
M77  OCT  77
GCCN  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,BLS
         TYPE A TWO REGISTERS
<table>
<thead>
<tr>
<th>BTYPE</th>
<th>BCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>BCI</td>
</tr>
<tr>
<td>+2</td>
<td>BCI</td>
</tr>
<tr>
<td>+3</td>
<td>BCI</td>
</tr>
<tr>
<td>+4</td>
<td>BCI</td>
</tr>
<tr>
<td>+5</td>
<td>BCI</td>
</tr>
<tr>
<td>+6</td>
<td>BCI</td>
</tr>
<tr>
<td>+7</td>
<td>BCI</td>
</tr>
<tr>
<td>+8</td>
<td>BCI</td>
</tr>
<tr>
<td>+9</td>
<td>BCI</td>
</tr>
<tr>
<td>+10</td>
<td>BCI</td>
</tr>
<tr>
<td>+11</td>
<td>BCI</td>
</tr>
<tr>
<td>+12</td>
<td>BCI</td>
</tr>
<tr>
<td>+13</td>
<td>BCI</td>
</tr>
<tr>
<td>+14</td>
<td>BCI</td>
</tr>
<tr>
<td>+15</td>
<td>BCI</td>
</tr>
<tr>
<td>+16</td>
<td>BCI</td>
</tr>
<tr>
<td>+17</td>
<td>BCI</td>
</tr>
<tr>
<td>+18</td>
<td>BCI</td>
</tr>
<tr>
<td>+19</td>
<td>BCI</td>
</tr>
<tr>
<td>+20</td>
<td>BCI</td>
</tr>
<tr>
<td>+21</td>
<td>BCI</td>
</tr>
<tr>
<td>+22</td>
<td>BCI</td>
</tr>
<tr>
<td>+23</td>
<td>BCI</td>
</tr>
<tr>
<td>+24</td>
<td>BCI</td>
</tr>
<tr>
<td>+25</td>
<td>BCI</td>
</tr>
<tr>
<td>+26</td>
<td>BCI</td>
</tr>
<tr>
<td>+27</td>
<td>BCI</td>
</tr>
</tbody>
</table>

B TYPE (B) HAS NO VARIABLE F1
+28 BCI 1,STP
+29 BCI 1,STE
+30 BCI 1,SRJ
+31 BCI 1,SJC
+32 BCI 1,IRJ
+33 BCI 1,SDC
+34 BCI 1,DRJ
+35 BCI 1,SID
+36 BCI 1,ACJ
+37 BCI 1,SBU
+38 BCI 1,NOP
+39 BCI 1,SLS
+40 BCI 1,LDN
+41 BCI 1,LDI
+42 BCI 1,LDI
+43 BCI 1,LDF
+44 BCI 1,LDB
+45 BCI 1,LCN
+46 BCI 1,LCD
+47 BCI 1,LCI
+48 BCI 1,LCF
+49 BCI 1,LCB
+50 BCI 1,STD
+51 BCI 1,STI
+52 BCI 1,STF
+53 BCI 1,STB
+54 BCI 1,HWI
+55 BCI 1,ADN
+56 BCI 1,ADD
+57 BCI 1,ADI
+58 BCI 1,ADF
+59 BCI 1,ADB
+60 BCI 1,SBN
+61 BCI 1,SBW
+62 BCI 1,SBF
+63 BCI 1,SBB
+64 BCI 1,RAD
+65 BCI 1,RAI
+66 BCI 1,RAF
+67 BCI 1,RAB
+68 BCI 1,RAO
+69 BCI 1,ADO
+70 BCI 1,ACI
+71 BCI 1,ACF
+72 BCI 1,ACB
+73 BCI 1,SRD
+74 BCI 1,SRI
+75 BCI 1,SRF
+76 BCI 1,SRB
+77 BCI 1,LPN
+78 BCI 1,LPD
+79 BCI 1,LPF
+80 BCI 1,LPB
+81 BCI 1,LPB

TYPE C FILL IN LO ORDER OCTAL DIGIT

TYPE D FILL IN TWO LO ORDER OCTAL DIGIT
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VFD</td>
<td>015/2600,21/0</td>
<td></td>
</tr>
<tr>
<td>+6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/4100,21/0</td>
<td></td>
</tr>
<tr>
<td>+7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/4200,21/0</td>
<td></td>
</tr>
<tr>
<td>+8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/3100,21/0</td>
<td></td>
</tr>
<tr>
<td>+9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/3200,21/0</td>
<td></td>
</tr>
<tr>
<td>+10</td>
<td>VFD</td>
<td>015/3500,21/0</td>
</tr>
<tr>
<td>+11</td>
<td>VFD</td>
<td>015/3600,21/0</td>
</tr>
<tr>
<td>+12</td>
<td>VFD</td>
<td>015/5100,21/0</td>
</tr>
<tr>
<td>+13</td>
<td>VFD</td>
<td>015/5200,21/0</td>
</tr>
<tr>
<td>+14</td>
<td>VFD</td>
<td>015/5500,21/0</td>
</tr>
<tr>
<td>+15</td>
<td>VFD</td>
<td>015/5600,21/0</td>
</tr>
<tr>
<td>+16</td>
<td>VFD</td>
<td>015/4500,21/0</td>
</tr>
<tr>
<td>+17</td>
<td>VFD</td>
<td>015/4600,21/0</td>
</tr>
<tr>
<td>+18</td>
<td>VFD</td>
<td>015/1100,21/0</td>
</tr>
<tr>
<td>+19</td>
<td>VFD</td>
<td>015/1200,21/0</td>
</tr>
<tr>
<td>+20</td>
<td>VFD</td>
<td>015/1500,21/0</td>
</tr>
<tr>
<td>+21</td>
<td>VFD</td>
<td>015/1600,21/0</td>
</tr>
<tr>
<td>+22</td>
<td>VFD</td>
<td>015/7100,21/0</td>
</tr>
<tr>
<td>+23</td>
<td>VFD</td>
<td>015/7200,21/0</td>
</tr>
<tr>
<td>+24</td>
<td>VFD</td>
<td>015/7300,21/0</td>
</tr>
<tr>
<td>+25</td>
<td>VFD</td>
<td>015/7500,21/0</td>
</tr>
<tr>
<td>+26</td>
<td>VFD</td>
<td>015/101,21/1</td>
</tr>
<tr>
<td>+27</td>
<td>VFD</td>
<td>015/107,21/1</td>
</tr>
<tr>
<td>+28</td>
<td>VFD</td>
<td>015/130,21/1</td>
</tr>
<tr>
<td>+29</td>
<td>VFD</td>
<td>015/2300,21/1</td>
</tr>
<tr>
<td>+30</td>
<td>VFD</td>
<td>015/2700,21/1</td>
</tr>
<tr>
<td>+31</td>
<td>VFD</td>
<td>015/4300,21/1</td>
</tr>
<tr>
<td>+32</td>
<td>VFD</td>
<td>015/4300,21/1</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>015/40, 21/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>+60</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/50, 21/2</td>
</tr>
<tr>
<td>+61</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/60, 21/2</td>
</tr>
<tr>
<td>+62</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/70, 21/2</td>
</tr>
<tr>
<td>+63</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/140, 21/2</td>
</tr>
<tr>
<td>+64</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1, 21/2</td>
</tr>
<tr>
<td>+65</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/7700, 21/2</td>
</tr>
<tr>
<td>+66</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/400, 21/3</td>
</tr>
<tr>
<td>+67</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/2000, 21/3</td>
</tr>
<tr>
<td>+68</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/2100, 21/3</td>
</tr>
<tr>
<td>+69</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/220001, 15/3</td>
</tr>
<tr>
<td>+70</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/230002, 15/3</td>
</tr>
<tr>
<td>+71</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/500, 21/3</td>
</tr>
<tr>
<td>+72</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/2400, 21/3</td>
</tr>
<tr>
<td>+73</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/2500, 21/3</td>
</tr>
<tr>
<td>+74</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/260001, 15/3</td>
</tr>
<tr>
<td>+75</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/270002, 15/3</td>
</tr>
<tr>
<td>+76</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/4000, 21/3</td>
</tr>
<tr>
<td>+77</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/4100, 21/3</td>
</tr>
<tr>
<td>+78</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/420001, 15/3</td>
</tr>
<tr>
<td>+79</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/430002, 15/3</td>
</tr>
<tr>
<td>+80</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/7600, 21/3</td>
</tr>
<tr>
<td>+81</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/600, 21/3</td>
</tr>
<tr>
<td>+82</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/3000, 21/3</td>
</tr>
<tr>
<td>+83</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/3100, 21/3</td>
</tr>
<tr>
<td>+84</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/320001, 15/3</td>
</tr>
<tr>
<td>+85</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/330002, 15/3</td>
</tr>
<tr>
<td>+86</td>
<td></td>
</tr>
</tbody>
</table>

78
<table>
<thead>
<tr>
<th>VFD</th>
<th>015/700,21/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>+87</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/3400,21/3</td>
</tr>
<tr>
<td>+88</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/3500,21/3</td>
</tr>
<tr>
<td>+89</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/360001,15/3</td>
</tr>
<tr>
<td>+90</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/370002,15/3</td>
</tr>
<tr>
<td>+91</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/5000,21/3</td>
</tr>
<tr>
<td>+92</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/5100,21/3</td>
</tr>
<tr>
<td>+93</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/520001,15/3</td>
</tr>
<tr>
<td>+94</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/530002,15/3</td>
</tr>
<tr>
<td>+95</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/5400,21/3</td>
</tr>
<tr>
<td>+96</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/5500,21/3</td>
</tr>
<tr>
<td>+97</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/560001,15/3</td>
</tr>
<tr>
<td>+98</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/570002,15/3</td>
</tr>
<tr>
<td>+99</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/4400,21/3</td>
</tr>
<tr>
<td>+100</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/4500,21/3</td>
</tr>
<tr>
<td>+101</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/460001,15/3</td>
</tr>
<tr>
<td>+102</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/470002,15/3</td>
</tr>
<tr>
<td>+103</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/200,21/3</td>
</tr>
<tr>
<td>+104</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1000,21/3</td>
</tr>
<tr>
<td>+105</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1100,21/3</td>
</tr>
<tr>
<td>+106</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/120001,15/3</td>
</tr>
<tr>
<td>+107</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/130002,15/3</td>
</tr>
<tr>
<td>+108</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/300,21/3</td>
</tr>
<tr>
<td>+109</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1400,21/3</td>
</tr>
<tr>
<td>+110</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>015/1500,21/3</td>
</tr>
<tr>
<td>+111</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/160001,15/3</td>
</tr>
<tr>
<td>+112</td>
<td></td>
</tr>
<tr>
<td>VFD</td>
<td>021/170002,15/3</td>
</tr>
<tr>
<td>+113</td>
<td></td>
</tr>
</tbody>
</table>

79
<table>
<thead>
<tr>
<th>VFD</th>
<th>021/600001,15/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>+114</td>
<td>VFD 021/610001,15/3</td>
</tr>
<tr>
<td>+115</td>
<td>VFD 021/620001,15/3</td>
</tr>
<tr>
<td>+116</td>
<td>VFD 021/630001,15/3</td>
</tr>
<tr>
<td>+117</td>
<td>VFD 021/640002,15/3</td>
</tr>
<tr>
<td>+118</td>
<td>VFD 021/650002,15/3</td>
</tr>
<tr>
<td>+119</td>
<td>VFD 021/660002,15/3</td>
</tr>
<tr>
<td>+120</td>
<td>VFD 021/670002,15/3</td>
</tr>
<tr>
<td>+121</td>
<td>VFD 015/7000,21/3</td>
</tr>
<tr>
<td>+122</td>
<td>VFD 021/710001,15/3</td>
</tr>
<tr>
<td>+123</td>
<td>VFD 021/720001,15/3</td>
</tr>
<tr>
<td>+124</td>
<td>VFD 021/730001,15/3</td>
</tr>
<tr>
<td>+125</td>
<td>VFD 015/7400,21/3</td>
</tr>
<tr>
<td>+126</td>
<td>VFD 015/7500,21/3</td>
</tr>
<tr>
<td>+127</td>
<td>VFD 015/7600,21/3</td>
</tr>
<tr>
<td>+128</td>
<td>VFD 015/7700,21/3</td>
</tr>
<tr>
<td>+129</td>
<td>VFD 015/7700,21/4</td>
</tr>
<tr>
<td>+130</td>
<td>VFD 021/320001,15/5</td>
</tr>
<tr>
<td>+131</td>
<td>VFD 021/310001,15/5</td>
</tr>
<tr>
<td>+132</td>
<td>VFD 021/260001,15/5</td>
</tr>
<tr>
<td>+133</td>
<td>VFD 021/220001,15/5</td>
</tr>
<tr>
<td>+134</td>
<td>VFD 021/120001,15/5</td>
</tr>
<tr>
<td>+135</td>
<td>VFD 021/160001,15/5</td>
</tr>
<tr>
<td>+136</td>
<td>VFD 021/630004,15/5</td>
</tr>
<tr>
<td>+137</td>
<td>VFD 021/610004,15/5</td>
</tr>
<tr>
<td>+138</td>
<td>VFD 021/620004,15/5</td>
</tr>
<tr>
<td>+139</td>
<td>VFD 021/520001,15/5</td>
</tr>
<tr>
<td>+140</td>
<td>VFD 021/360001,15/5</td>
</tr>
</tbody>
</table>

80
ENDATN PZE **-ATANXX
CREGIN BSS 1400,C
SCRIBE IOCD OREGIN,,1365
RETURN SCANIT
+1 TRA SCANIT+1

UNDEF BCI 9, IS AN UNDEFINED SYMBOL. IT WILL BE EQUATED TO 0
+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, XXXXX
D19 DEC 19
RITEOP BCI 1,
RITEKY PZE 0
ACCODE BCI 1,$
AFIELD BCI 1,
DATA BSS 15,C
DIGIT BSS 12,C
MANSgn PZE
EXPONT PZE
EXP0 CCT 100
EXPSGN 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
DECClM PZE
NZCOLM PZE
TEN DEC 10
ECCLMN PZE
ISDCPT BCI 1.00000.
OPTIND PZE
C17 DEC 17
EEP 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,
VCHLCC 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
+18 BCI 2, BE UNKNOWN.
HEAD1 BSS 20,0
HEAD2 BCI 9, MLOC SILOC K OP ADDR SYMBOL OPCODE VARIABLE FIE
+9 BCI 9, LD COMMENTARY PAGE
PAGE BCI 2, 
NAME BCI 1, NAME
PAGENO PZE
CASE1 CLA 22
  +1 SUB ONE
  +2 STA SWTCH+1
  OHEAD CASE1,,1,55,1
  +3 STL 22
  OREADY
  +8 STL 22
  CBCW HEAD1,,1,20
  +10 STL 22
  OSCIRB 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
  OSCIRB CUTAPE,,20,1
  +31 STL 22
  OREADY
  +35 STL 22
  CLRANK 120,,1
  +37 STL 22
  OSCIRB 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
  OSCIRB PRINTR,,20
  +10 STL 22
  OREADY
  +14 STL 22
  OBCW HEAD2,,1,20
  +16 STL 22
  CINT PAGENO,,109,2
  +20 STL 22
  OSCIRB PRINTR,,20
  +24 STL 22
  OREADY

84
NORMAL SICOM COMMAND = 7 CHARACTERS

CARD READER
PRINTER
CARD OR TAPE

B7 FOR COPY CARD INPUT

ZERO = ON LINE CARD INPUT
CN 
SYNCR PZE 
VFD 012/12,24/0
LEADER 
ENABLE OCT 2000
CCTAL BCI 1, OCT
NUMBER PZE
SIMP BSS 4096,0
WHRMAP HTR SIMAP
WHRDAT HTR DATA
REFADD PZE 0,4
RIKAP DEC 18 READ IN PROGRAM 
+1 OCT 0
+2 DEC 21 READ TAE 0
+3 OCT 100
+4 DEC 4
+5 OCT 64
GO2 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, SJB
+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
86
<table>
<thead>
<tr>
<th>Address</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>+30 BCI</td>
<td>0TBNC</td>
</tr>
<tr>
<td>+31 BCI</td>
<td>0CMND</td>
</tr>
<tr>
<td>+32 BCI</td>
<td>WRFILE</td>
</tr>
<tr>
<td>+33 BCI</td>
<td>SRFILE</td>
</tr>
<tr>
<td>+34 BCI</td>
<td>GRPIN</td>
</tr>
<tr>
<td>+35 BCI</td>
<td>SNGIN</td>
</tr>
<tr>
<td>+36 BCI</td>
<td>IOCTAL</td>
</tr>
<tr>
<td>+37 BCI</td>
<td>QALPHA</td>
</tr>
<tr>
<td>+38 BCI</td>
<td>LDANR</td>
</tr>
<tr>
<td>+39 BCI</td>
<td>CMPANR</td>
</tr>
<tr>
<td>+40 BCI</td>
<td>MRGANR</td>
</tr>
<tr>
<td>+41 BCI</td>
<td>EXTRANR</td>
</tr>
<tr>
<td>+42 BCI</td>
<td>TZE</td>
</tr>
<tr>
<td>+43 BCI</td>
<td>TNZ</td>
</tr>
<tr>
<td>+44 BCI</td>
<td>TPL</td>
</tr>
<tr>
<td>+45 BCI</td>
<td>TMI</td>
</tr>
<tr>
<td>+46 BCI</td>
<td>TSMN</td>
</tr>
<tr>
<td>+47 BCI</td>
<td>TSM5</td>
</tr>
<tr>
<td>+48 BCI</td>
<td>TSM6</td>
</tr>
<tr>
<td>+49 BCI</td>
<td>TNR7</td>
</tr>
<tr>
<td>+50 BCI</td>
<td>TSR</td>
</tr>
<tr>
<td>+51 BCI</td>
<td>TSJ1</td>
</tr>
<tr>
<td>+52 BCI</td>
<td>TSJ2</td>
</tr>
<tr>
<td>+53 BCI</td>
<td>TSJ3</td>
</tr>
<tr>
<td>+54 BCI</td>
<td>CALL</td>
</tr>
<tr>
<td>+55 BCI</td>
<td>TBR</td>
</tr>
<tr>
<td>+56 BCI</td>
<td>STOANR</td>
</tr>
<tr>
<td>+57 BCI</td>
<td>STO</td>
</tr>
<tr>
<td>+58 BCI</td>
<td>NOP</td>
</tr>
<tr>
<td>+59 BCI</td>
<td>STOP1</td>
</tr>
<tr>
<td>+60 BCI</td>
<td>STOP2</td>
</tr>
<tr>
<td>+61 BCI</td>
<td>HALT3</td>
</tr>
<tr>
<td>+62 BCI</td>
<td>HALT4</td>
</tr>
<tr>
<td>+63 BCI</td>
<td>HPRIN</td>
</tr>
<tr>
<td>+64 BCI</td>
<td>REL</td>
</tr>
<tr>
<td>+65 BCI</td>
<td>ABS</td>
</tr>
<tr>
<td>+66 BCI</td>
<td>FAST</td>
</tr>
<tr>
<td>+67 BCI</td>
<td>TRACE</td>
</tr>
<tr>
<td>+68 BCI</td>
<td>ZEROIR</td>
</tr>
<tr>
<td>+69 BCI</td>
<td>IFLEX</td>
</tr>
<tr>
<td>+70 BCI</td>
<td>OFLEX</td>
</tr>
<tr>
<td>+71 BCI</td>
<td>IYPE</td>
</tr>
<tr>
<td>+72 BCI</td>
<td>OTYPE</td>
</tr>
<tr>
<td>+73 BCI</td>
<td>OPRINT</td>
</tr>
<tr>
<td>+74 BCI</td>
<td>SEL8CD</td>
</tr>
<tr>
<td>+75 BCI</td>
<td>SELBIN</td>
</tr>
<tr>
<td>+76 BCI</td>
<td>BSR</td>
</tr>
<tr>
<td>+77 BCI</td>
<td>REW</td>
</tr>
<tr>
<td>+78 BCI</td>
<td>OTAPE</td>
</tr>
<tr>
<td>+79 BCI</td>
<td>ITAPE</td>
</tr>
<tr>
<td>+80 BCI</td>
<td>BLCPY</td>
</tr>
<tr>
<td>+81 BCI</td>
<td>BLCLR</td>
</tr>
<tr>
<td>+82 BCI</td>
<td>IOCTAL</td>
</tr>
<tr>
<td>+83 BCI</td>
<td>OSPEC</td>
</tr>
</tbody>
</table>
+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,CEQ
+108 BCI 1,SETFWA
+109 BCI 1,SETLWA
+110 BCI 1,MPYC
+111 BCI 1,PXAIB
+112 BCI 1,PXAI D
+113 BCI 1,PXAIL
+114 BCI 1,PXAJB
+115 BCI 1,PXAJC
+116 BCI 1,PXAJL
+117 BCI 1,PXAJB
+118 BCI 1,PXIC
+119 BCI 1,PXIL
+120 BCI 1,PXJB
+121 BCI 1,PXJD
+122 BCI 1,PXJL
ENDOP PZE
CPKEY BCI 1,002000
SID BCI 1,003000
SIL BCI 1,004000
DIB BCI 1,005000
IID 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
CLB BCI 1,021000
CLA BCI 1,022000
JVP BCI 1,023000

THIS IS SIB 1ST OF TYPE 1 INSTRUCTIONS

88
LAP
Bci 1,024000
MPY BCI 1,025000
EXCH BCI 1,026000
EXTR BCI 1,027000
ACC BCI 1,030000
SUB BCI 1,031000
ADM BCI 1,032000
ISUB BCI 1,033000
RAD BCI 1,034000
CAS BCI 1,035000
CFLT BCI 1,036000
CFLCR BCI 1,037000
OFTB BCI 1,040000
OFXTB BCI 1,041000
CFORM BCI 1,042000
CGR TB BCI 1,043000
CTBNO BCI 1,044000
CCMND BCI 1,045000
wFNMT BCI 1,046000
SFNMT BCI 1,047000
GRPIN BCI 1,050000
SNGIN BCI 1,051000
ROCTT BCI 1,052000
ALNUG BCI 1,053000
CLAANR BCI 1,054000
CMERA BCI 1,055000
KRGANR BCI 1,056000
E),TANR BCI 1,057000
TZE BCI 1,060000
TNZ BCI 1,061000
TPL BCI 1,062000
TMI BCI 1,063000
TSM4 BCI 1,064000
TSM5 BCI 1,065000
TSM6 BCI 1,066000
TSM7 BCI 1,067000
TSR BCI 1,070000
TSW1 BCI 1,071000
TSW2 BCI 1,072000
TSW3 BCI 1,073000
TML BCI 1,074000
TBR BCI 1,075000
STOANR BCI 1,076000
STO BCI 1,077000
NOP BCI 1,000001
HLTSW1 BCI 1,001001
HLTSW2 BCI 1,002001
HALT3 BCI 1,003001
HALT4 BCI 1,004001
HLTGIN BCI 1,005001
SELREL BCI 1,006001
SELABS BCI 1,007001
SELNTR BCI 1,008001
RESTR BCI 1,009001

END OF TYPE 0 INSTRUCTIONS
TYPE 1 INSTR OP CODE =DR
TYPE 2 INSTR

CLIR  BCI  1,010001
RDFLW BCI  1,011001
WRFLW BCI  1,012001
RDTPW BCI  1,013001
WRTPW BCI  1,014001
WRPRN BCI  1,015001
SELTTP BCI  1,016001
SELMTD BCI  1,017002
ESR  BCI  1,018004
REW  BCI  1,019001
WRMT2 BCI  1,020003
RDMT2 BCI  1,021003
BLCPY BCI  1,022003
BLCLR BCI  1,023003
POT  BCI  1,024003
PST  BCI  1,025003
RDST  BCI  1,026003
RZERC BCI  1,027003
RNZERC BCI  1,028003
RPCS  BCI  1,029003
RNEG  BCI  1,030003
RMK4  BCI  1,031003
RMK5  BCI  1,032003
RMK6  BCI  1,033003
RMK7  BCI  1,034003
RSR  BCI  1,035003
RSH1  BCI  1,036003
RSH2  BCI  1,037003
RSH3  BCI  1,038003
CLCC  BCI  1,039003
PLSTCP BCI  1,040003
CLAC  BCI  1,041003
CLSC  BCI  1,042003
IDVPC BCI  1,043003
DVPC  BCI  1,044003
ADDC  BCI  1,045003
SUBC  BCI  1,046003
ISUBC BCI  1,047003
SHFTAC BCI  1,048003
+1  BCI  1,049003
SETTRC BCI  1,050003
STPTRC BCI  1,051003
MPYC  BCI  1,052003
PXIBA BCI  1,053003
PXIDA BCI  1,054003
PXILA BCI  1,055003
PXJBA BCI  1,056003
PXJDA BCI  1,057003
PXJLA BCI  1,058003
ATXIB BCI  1,059003
ATXID BCI  1,060003
ATXIL BCI  1,061003
ATXJB BCI  1,062003
ATXJD BCI  1,063003

TYPE 6

CLCC  BCI  1,010006
CLAC  BCI  1,011006
CLSC  BCI  1,012006
IDVPC BCI  1,013006
DVPC  BCI  1,014006
ADDC  BCI  1,015006
SUBC  BCI  1,016006
ISUBC BCI  1,017006
SHFTAC BCI  1,018006
+1  BCI  1,019006
SETTRC BCI  1,020006
STPTRC BCI  1,021006
MPYC  BCI  1,022006
PXIBA BCI  1,023006
PXIDA BCI  1,024006
PXILA BCI  1,025006
PXJBA BCI  1,026006
PXJDA BCI  1,027006
PXJLA BCI  1,028006
ATXIB BCI  1,029006
ATXID BCI  1,030006
ATXIL BCI  1,031006
ATXJB BCI  1,032006
ATXJD BCI  1,033006
<table>
<thead>
<tr>
<th>ADDR</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCI</td>
<td>PZE</td>
<td>1</td>
</tr>
<tr>
<td>BSS</td>
<td>1000,C</td>
<td></td>
</tr>
<tr>
<td>BSS</td>
<td>1000,+</td>
<td></td>
</tr>
<tr>
<td>CRC</td>
<td>32256</td>
<td></td>
</tr>
<tr>
<td>CORE</td>
<td>START,-1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADDR</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>STL</td>
<td>2169</td>
<td></td>
</tr>
<tr>
<td>CORE</td>
<td>43,43</td>
<td></td>
</tr>
<tr>
<td>STL</td>
<td>2169</td>
<td></td>
</tr>
<tr>
<td>CORE</td>
<td>22,22</td>
<td></td>
</tr>
<tr>
<td>STL</td>
<td>2169</td>
<td></td>
</tr>
<tr>
<td>TRA</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>START</td>
<td></td>
</tr>
<tr>
<td>halt</td>
<td>0045</td>
<td>myown</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>hard1</td>
<td>0019</td>
<td>name</td>
</tr>
<tr>
<td>head1</td>
<td>0040</td>
<td>namer</td>
</tr>
<tr>
<td>head2</td>
<td>0040</td>
<td>nilly</td>
</tr>
<tr>
<td>igctr</td>
<td>0041</td>
<td>nine</td>
</tr>
<tr>
<td>idvp</td>
<td>0044</td>
<td>nceo</td>
</tr>
<tr>
<td>idvpc</td>
<td>0046</td>
<td>ncp</td>
</tr>
<tr>
<td>ifdec</td>
<td>0018</td>
<td>nope</td>
</tr>
<tr>
<td>ifoct</td>
<td>0019</td>
<td>nct8</td>
</tr>
<tr>
<td>ib</td>
<td>0044</td>
<td>nctac</td>
</tr>
<tr>
<td>jb</td>
<td>0044</td>
<td>nctk</td>
</tr>
<tr>
<td>mode</td>
<td>0028</td>
<td>ncwgo</td>
</tr>
<tr>
<td>input</td>
<td>0041</td>
<td>ncin</td>
</tr>
<tr>
<td>instr</td>
<td>0026</td>
<td>c</td>
</tr>
<tr>
<td>intpl</td>
<td>0018</td>
<td>ccmnd</td>
</tr>
<tr>
<td>isac</td>
<td>0016</td>
<td>cqrtb</td>
</tr>
<tr>
<td>isbss</td>
<td>0003</td>
<td>cqt12</td>
</tr>
<tr>
<td>iscgm</td>
<td>0020</td>
<td>cqtal</td>
</tr>
<tr>
<td>isdsc</td>
<td>0004</td>
<td>oflcr</td>
</tr>
<tr>
<td>ise</td>
<td>0039</td>
<td>ofltb</td>
</tr>
<tr>
<td>isend</td>
<td>0004</td>
<td>oform</td>
</tr>
<tr>
<td>isneg</td>
<td>0039</td>
<td>ofxcr</td>
</tr>
<tr>
<td>isop</td>
<td>0003</td>
<td>ofxrb</td>
</tr>
<tr>
<td>isorg</td>
<td>0004</td>
<td>oloc</td>
</tr>
<tr>
<td>ispq</td>
<td>0039</td>
<td>one</td>
</tr>
<tr>
<td>isrev</td>
<td>0025</td>
<td>choff</td>
</tr>
<tr>
<td>issym</td>
<td>0003</td>
<td>cornc</td>
</tr>
<tr>
<td>istag</td>
<td>0019</td>
<td>ckey</td>
</tr>
<tr>
<td>isub</td>
<td>0045</td>
<td>obmgr</td>
</tr>
<tr>
<td>isubc</td>
<td>0046</td>
<td>obknm</td>
</tr>
<tr>
<td>jdd</td>
<td>0025</td>
<td>page</td>
</tr>
<tr>
<td>k</td>
<td>0041</td>
<td>pct</td>
</tr>
<tr>
<td>kop</td>
<td>0041</td>
<td>pst</td>
</tr>
<tr>
<td>ldqa</td>
<td>0010</td>
<td>pxiba</td>
</tr>
<tr>
<td>logic</td>
<td>0017</td>
<td>pxida</td>
</tr>
<tr>
<td>log2</td>
<td>0017</td>
<td>pxila</td>
</tr>
<tr>
<td>log3</td>
<td>0017</td>
<td>pxjba</td>
</tr>
<tr>
<td>lookc</td>
<td>0020</td>
<td>pxjda</td>
</tr>
<tr>
<td>loop</td>
<td>0010</td>
<td>pxjla</td>
</tr>
<tr>
<td>m7</td>
<td>0028</td>
<td>cdec</td>
</tr>
<tr>
<td>m77</td>
<td>0028</td>
<td>ccdec</td>
</tr>
<tr>
<td>m777</td>
<td>0028</td>
<td>c</td>
</tr>
<tr>
<td>mapit</td>
<td>0026</td>
<td>rad</td>
</tr>
<tr>
<td>mode</td>
<td>0031</td>
<td>ravel</td>
</tr>
<tr>
<td>mode</td>
<td>0028</td>
<td>rdflw</td>
</tr>
<tr>
<td>mpy</td>
<td>0045</td>
<td>rdmt2</td>
</tr>
<tr>
<td>mpyc</td>
<td>0046</td>
<td>rdst</td>
</tr>
</tbody>
</table>

12/20/62
<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. Ferguson</td>
<td>Col. L. S. Norman</td>
</tr>
<tr>
<td>S. H. Dodd</td>
<td>E. Harvey</td>
<td>P. Fleck</td>
<td>J. Frankovich</td>
<td>Capt. J. Kieper (1)</td>
</tr>
<tr>
<td>Group 21</td>
<td>E. Hofmann</td>
<td>T. Goblick</td>
<td>J. Heart</td>
<td>Lockheed</td>
</tr>
<tr>
<td>W. Crowther</td>
<td>B. Moriarty</td>
<td>S. McCarrall</td>
<td>S. Ornstein</td>
<td>H. Batten (5)</td>
</tr>
<tr>
<td>A. Curtiss (2)</td>
<td>W. Vecchia</td>
<td>R. Price</td>
<td>Group 52</td>
<td>Philco WDL</td>
</tr>
<tr>
<td>J. Drinan</td>
<td>D. Wiggert</td>
<td>W. Smith</td>
<td>G. Blustein</td>
<td>F. Harris (5)</td>
</tr>
<tr>
<td>O. Fortier</td>
<td></td>
<td></td>
<td>S. Larkin</td>
<td></td>
</tr>
<tr>
<td>H. Frachtman</td>
<td></td>
<td></td>
<td>F. Nagy</td>
<td></td>
</tr>
<tr>
<td>D. Hafford</td>
<td></td>
<td></td>
<td>M. Wendell</td>
<td></td>
</tr>
<tr>
<td>F. Heart</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. Mathiasen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R. Saliga</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P. Smith</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P. Stylos</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R. Teoste</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S. White</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group 22</td>
<td>Group 26</td>
<td>Group 35</td>
<td>Group 55</td>
<td></td>
</tr>
<tr>
<td>M. Andrews</td>
<td>H. Bostick</td>
<td>B. Fowler</td>
<td>F. Slagle</td>
<td></td>
</tr>
<tr>
<td>A. Armcnti</td>
<td>D. Underwood</td>
<td>P. Miller</td>
<td>J. Slagle</td>
<td></td>
</tr>
<tr>
<td>J. Avery</td>
<td>H. Zieman</td>
<td>L. Tahmoush</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V. Brady</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R. Carroll</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N. Ciampa</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D. Clapp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J. FitzGerald</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. Freed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R. Gauvreau (15)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. Grometstein</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J. Halberstein</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C. Hardy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J. Hartogensis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P. Heffernan</td>
<td></td>
<td></td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>D. Kramer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B. Lee</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R. Leonard</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D. Lovenvirth</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. MacGillivray</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F. Mahoney</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M. Malone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L. Meixsell</td>
<td></td>
<td></td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>J. O'Brien</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F. Oliver</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D. Reiner</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J. Rheinstein</td>
<td></td>
<td></td>
<td></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>
<tr>
<td>A. Smith</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H. Sussman</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J. Winett</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Group 28 | Group 36 | Group 58 | | |
| J. Arnow | D. Hamilton | E. Freed | | |
| D. Koniver | J. Max | M. Goodman | | |
| H. Meily | | L. Klem | | |
| H. Peterson | | C. McElwain | | |
| | | R. Mitchell | | |
| | | G. Mueser | | |
| | | J. Smith | | |
| | | M. Smith | | |
| | | A. Stowe | | |
| | | F. Trask | | |
| | | B. White | | |
| | | D. Yntema | | |

| Group 32 | Group 37 | Group 59 | | |
| B. Benn | N. Holway | E. Freed | | |
| H. Briscoe | Group 314 | M. Goodman | | |
| R. Brown | | L. Klem | | |
| J. Burdette | | C. McElwain | | |
| N. Donald | | R. Mitchell | | |
| R. Martinson | | G. Mueser | | |
| J. Morriello | | J. Smith | | |
| G. Price | | M. Smith | | |
| K. Ralston | | A. Stowe | | |
| H. Shimmin | | F. Trask | | |
| | | B. White | | |
| | | D. Yntema | | |

| Group 33 | Group 38 | Group 71 | | |
| L. Bird | R. Brockelman | E. Boyce | | |
| V. Casaceli | M. Meeks | W. Fanning | | |
| E. Crowley | | C. Pappas | | |
| J. Crowley | | D. Regillo | | |
| R. Ingalls | | F. Wan | | |
| L. Peterson | | | | | |
| D. Quinn | | | | | |
| G. Shannon | | | | | |
| R. Shaputnic | | | | | |
| M. Tausner | | | | | |
| S. Wang | | | | | |
| P. Willmann | | | | | |
| F. Wilson | | | | | |
| N. Reid | | | | | |

| Group 41 | Group 42 | Group 72 | | |
| N. Doucett | J. Cogdell | E. Mahoney | | |
| J. Fielding | J. Dinsmore | M. Durgin | | |
| | | V. Mason | | |
| | | N. Rawson | | |
| | | S. Sillers | | |
| | | C. Work | | |

| Group 43 | Group 44 | Group 73 | | |
| C. Freed | C. Guay | M. Dresselhaus | | |
| C. Muehe | | W. Mason | | |

| Group 45 | Group 46 | Group 81 | | |
| C. Berger | D. Moore | M. Durgin | | |
| S. Catalano | P. Stetson | V. Mason | | |
| H. Jones | | N. Rawson | | |
| R. Wishner | | S. Sillers | | |
| | | C. Work | | |

| Group 47 | Group 48 | Group 82 | | |
| C. Berger | C. Guay | M. Dresselhaus | | |
| S. Catalano | | W. Mason | | |
| H. Jones | | Group 22 Central Depository | | |
| R. Wishner | | | J. FitzGerald | |