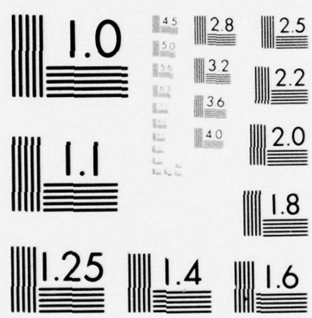
AD-A038 620 ARMY ELECTRONICS COMMAND FORT MONMOUTH N J F/G 9/2
DESIGN SYMBOLS LIBRARY, A COMBINED DESIGN AND DOCUMENTATION TOO--ETC(U)
DEC 76 T J WHEELER
UNCLASSIFIED ECOM-4452 NL

| OF |
ADA038620

END
DATE
FILMED
5 - 77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A038620

Research and Development Technical Report

ECOM -4452

DESIGN SYMBOLS LIBRARY, A COMBINED DESIGN AND DOCUMENTATION TOOL

Thomas J. Wheeler

R & D Technical Support Activity

December 1976

DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

ECOM

US ARMY ELECTRONICS COMMAND FORT MONMOUTH, NEW JERSEY 07703

NOTICES

Disclaimers

Disposition

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ECOM-4452
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): Design Symbols Library, a Combined Design and Documentation Tool
5. TYPE OF REPORT & PERIOD COVERED:
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Thomas J. Wheeler
8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: US Army Electronics Command, DRSEL-GG-C, Fort Monmouth, New Jersey 07703
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: Same
12. REPORT DATE: December 1976
13. NUMBER OF PAGES: 50
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): CAD, CAM, Design, Documentation, Printed Circuits
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

Current methods of design of electronic equipment produce documentation after the design, resulting in duplication of cost and inadequate documentation. A system is described which makes use of a Standard Library of design symbols to integrate the design/documentation process, thereby reducing cost and insuring adequacy of documentation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

037620

TABLE OF CONTENTS

Background..............................1

Introduction............................1

Symbols Library.........................1

PCB Design Using Symbols Library.........2

Implementation of the Design Symbols Library System........................5

Alternative Implementations.............9

Conclusions.............................9

Acknowledgement.........................10

APPENDICES

I Description of Aldraft...........11

II Symbols Definition...............12

III Initial Library..................14

IV Design Example...................35

V Listing of Design Assistance Software.......................42

BACKGROUND

The end results of a design effort for a piece of electronic equipment are two: A piece of hardware and a documentation package. The documentation is more valuable than the hardware since it is from the documentation that all production of the equipment is done. Currently, however, the design of the equipment and the production of the documentation are done separately with the effort applied in producing the documentation being a minimum, usually only enough to meet contractual obligations. The documentation supplied at the end of the development is usually done by a drafting department from sketches developed during the design process, with the result that some of the documentation does not describe adequately the equipment.

The production of documentation as an integral part of the design process will eliminate the problem of inadequate documentation since both the equipment and the documentation come concurrently out as two outputs from one process.

INTRODUCTION

The cost of designing electronic equipment can be reduced by integrating the documentation process into the design process. In order to effect this design cost savings and improve the quality of the documentation of electronic equipment, an approach was conceived which would make use of a standard symbols library and an automated drafting system. The use of the design symbols as design tools rather than just documentation tools results in the equipment being fabricated from the design documentation rather than the documentation being copied from the equipment. This assures that the documentation is adequate for production.

Since this project was for the purpose of development and demonstration of the design method, its scope was limited to the design of printed circuit boards (PCB) but the method is generally valid.

SYMBOLS LIBRARY

The Symbols Library is a design and documentation tool consisting of an open-ended catalog collection of named symbols. The symbols are manipulated by the designer to produce a piece of documentation. Each symbol is allowed multiple definitions so as to allow different types of documentation to be produced from the same input.

A symbol is a collection of line segments which can be logically grouped and used as a group more than once. All that is required is that each line segment maintain its position relative to the other line segments which make up the group, and that there be a potential re-use of the group.

A symbol may be as simple as a circle or an arrow or as complicated as an entire assembly. Symbols may be moved, rotated, scaled, repeated, etc., along with other symbols and lines to make up a final drawing.

A symbol may also be made up of other symbols, i.e., symbols may be 'nested'. Nesting is the normal method of producing all symbols above the primitive set, as shown in Appendix 2, which illustrates some sample symbols and their definitions.

The Symbols Library concept modifies the simple definition of a symbol to allow a given symbol to be multiply defined (see Appendix 2 for a sample) so that, for example, the assembly drawing and artwork for a printed circuit board are able to be produced from the same drafting program by simply changing the symbol set.

The result of allowing multiple definitions of symbols is that the designer can think of symbols representing logical items rather than a collection of lines. Thus, he would think of a symbol called 5420 as a 14 pin IC containing two four input Nand gates knowing that the system would draw the schematic symbol, package symbol or artwork symbol as required, but the designer would always refer to it as "5420 at location such and such".

The symbols are cataloged into a library which is open-ended allowing for growth. The printed circuit board (PCB) Library consists of symbols with four definitions; namely, the artwork symbol, the package outline, the internal schematic, and the alphanumeric label symbol. (An initial library is included in Appendix 3).

The Symbols Library is intended for use as an integral part of the design process, as will be seen in the next section, thus providing the dual benefits of simplifying the documentation of a design and guaranteeing the adequacy of the documentation. There is an additional benefit that occurs after the design is complete in that engineering changes are vastly more simple to document since only that portion of the drafting program which is affected by the change need be redone and new drawings can be immediately produced on the drafting system.

PCB Design Using Symbols Library

The starting point of the design process assumes that a valid schematic of the circuit to be designed is available. Note that design is the discipline which produces a PCB from a schematic as opposed to engineering which produces a prototype and schematic from a specification.

The design process (FIG 1) is begun by the designer reviewing the design symbol catalog and adding new symbols using the library maintenance software if necessary. When the library contains all of the symbols (usually there is one symbol for one component), the designer prepares a design input deck which contains all symbols (components) that he uses and their location and orientation.

The design input is run through the design assistance software. The designer gets out of this program a parts list and a magnetic tape for the drafting system with the component layout on it and information for the routing software.

If the PCB is a one or two-sided board, it is uneconomical to use a computer routing program to provide the interconnection. The designer usually routes the interconnection manually. This process follows the dotted line path on Figure 1 as follows:

- The designer receives a two or four times blowup of the artwork upon which he sketches or draws the interconnection pattern. This pattern is digitized by the drafting machine and run through the computer post-processing program to produce the interconnection pattern of the correct width.

- If the PCB is a multilayer board the interconnection data is sent to a routing program which routes the connections and returns a magnetic tape.

- The component layout tape and the routing tape are then processed by the symbol resolving system and drawn using an automated drafting system. The end results of this process are the artwork, a numerical control (NC) drilling tape, an assembly drawing and a schematic.

The artwork is used to fabricate a PCB which is drilled using the NC tape. The components are then inserted using the assembly drawing as a guide. If automatic assembly machines are used, the data for the program is already available since the location of all components is already in the Data Base. After soldering, the finished board is tested and after successful completion of the tests, the design is complete.

<u>Since the documentation has been used to fabricate the PCB, the fact that the testing proved that the PCB design is correct, also insures that the documentation is adequate</u>.

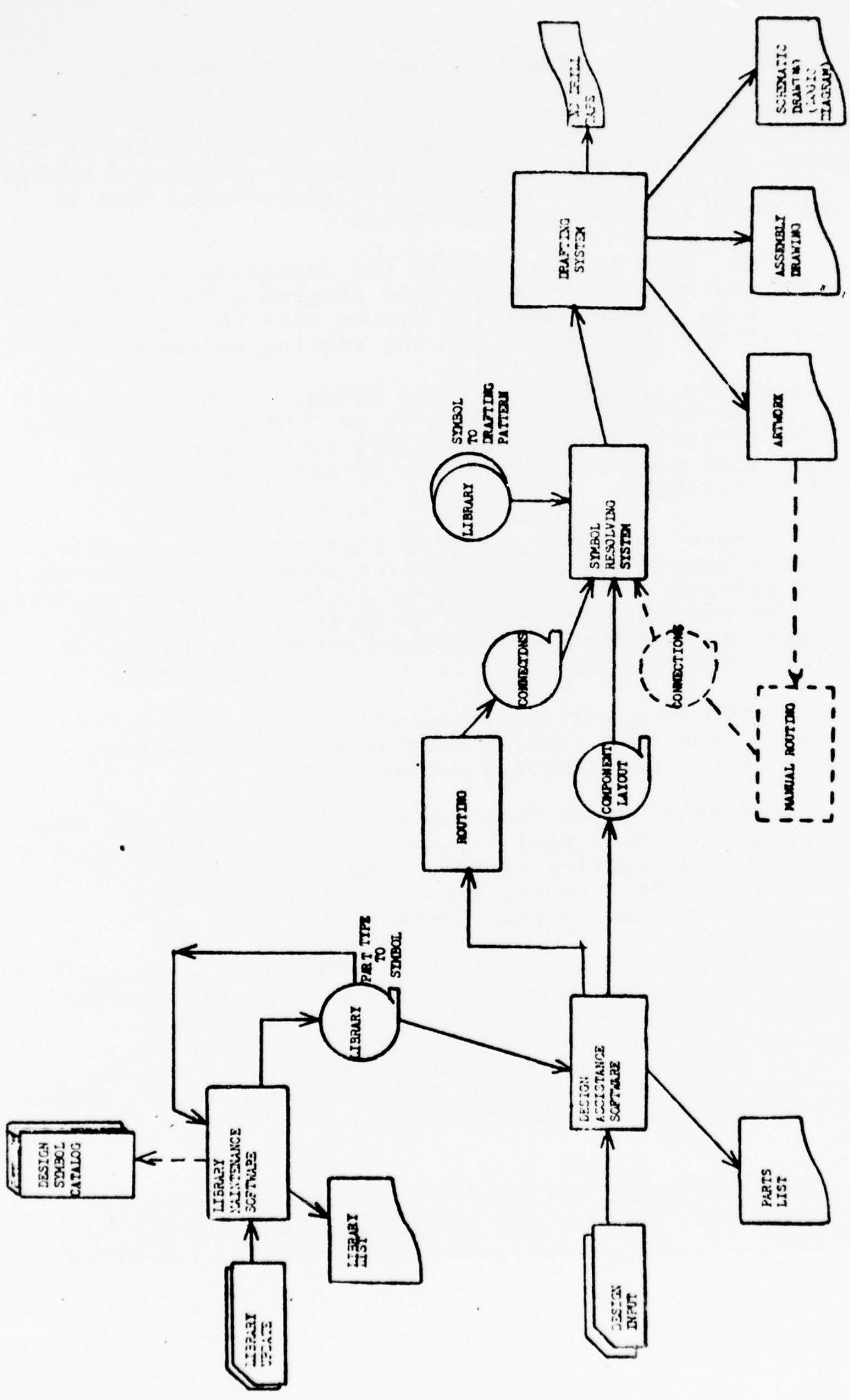


FIGURE 1 DESIGN SYMBOLS SYSTEM

IMPLEMENTATION OF THE DESIGN SYMBOL LIBRARY SYSTEM

The Design Symbol Library has been implemented at ECOM by the use of a combination of computer programs on a large computer (Burrough B5700) and a computer controlled automated drafting system (TRIDEA).

A description of the structure of the software and hardware components of the Design Symbols system follows.

LIBRARY

The Design Symbols Library is stored in two parts. The first part contains the data which allows translation from part number or component type to symbol name and the second part contains data which allows translation from symbol name to drafting commands for the drafting system. This division of the library into two parts is due to many different electronic components having the same symbol; e.g., the many components that are housed in 14 pin dual in line packages (DIP). The two-stage revolving of the component or part allows only one copy of each symbol to be stored.

The first part of the library is the part which interfaces the system to the designer. The data contained in this part for each component includes: component type, component description, schematic symbol name, package symbol name, power dissipation, component number, pad size, location of component center and component orientation. This part of the library is stored on magnetic tape located with the large computer system (B5700). In addition to providing the data for the component layout symbol tape this part of the library supplies the data for parts lists.

The second part of the library is the part which interfaces to the drafting system. It is located on the disc storage system which is part of the TRIDEA automated drafting/digitizing system. The library is maintained in two sections; one section contains the fixed disc which has the unchangeable symbols, and the other contains the removable disc which has the changeable symbols, i.e, the assembly drawing symbols and the artwork symbols where the same symbol call produces a different configuration on line segments depending upon the symbol set that is on the removable disc at the time. This part of the library is comprised of files whose names are equivalent to the symbol names which are output from the design assistance program. Each file contains a combination of drafting commands and calls to subfiles, also drafting commands, which produce the required symbol when called by the drafting program.

DESIGN ASSISTANCE SOFTWARE

The Design Assistance Software is a set of computer programs which allows the designer to layout a printed circuit board using the design symbols library. These programs are written in FORTRAN to run on the large computer system (B5700) in conjunction with the data in the first part of the Design Symbols Library. The operation of the Design Assistance Software is as follows:

- The designer prepares and submits a deck of computer cards to the program. Each card of this deck contains a component identification, the component number, a component type, the position of pin 1, the orientation of the component and a part description.

- The software reads this card deck and sorts the data into order by component type.

- The software starts at the beginning of the sorted cards and searches the first part of the data base for a matching component type. It extracts from this library the data it needs. It then writes the symbol for that component onto the Artwork/Assembly drawing tape and, if requested, writes the schematic symbol to the schematic tape.

- The software next repeats that process for all cards in the input deck, and when it has finished the input deck it writes out the part list, power description list and pin/voltage list as requested and terminates.

- After all of the processing is completed, the program terminates the output of data to the tapes and writes and end of file on the tapes. The component layout tapes are sent to the automated drafting system and the lists are given to the designer completing the operation of the Design Assistance Software.

LIBRARY MAINTENANCE SOFTWARE

The Library Maintenance Software is a set of computer programs which allows the first part of the symbol library to be initially built and then updated as needed. The programs are written in FORTRAN and run on the B5700 Computer.

- The inputs to the program are the current symbols library Part 1 tape and an input library update card deck.

- The outputs are a new symbols library Part 1 tape, a listing of the changes, and if requested, a complete listing of the new symbols library Part 1.

The program operates in three phases: First the input data is read and sorted by component type; second, the input data is merged with the old symbols library and a new symbol library type is generated; and third, a listing of the new symbols library is generated.

Library records can be added, replaced or deleted. An input record whose component type is not in the old program library is added to the new program library. Input records whose component type is already defined will replace the old program library reference on the new program library. If, however, the input record contains the characters "*DEL" after the component name, the library entry with that component name is deleted from the new symbols library tape.

After the completion of the processing of all input cards, a listing is made of all the changes which were made in the symbols library tape. After the new symbols library tape is completed, a listing is made of the entire symbols library if that option was requested and the program terminates.

DESIGN SYMBOLS CATALOG

The Catalog is composed of the latest listing of the contents of the Part 1 tape in the symbols library which is produced by the library maintenance software along with a drawing of the symbol(s) produced by the automated drafting system which corresponds to that component.

The Design Symbols Library catalog is the document that allows the designer to know what components are presently available in the symbols library. The designer uses the components which are in the catalog in designing the circuit. If a particular component does not exist in the catalog, the designer or someone else can add it to the library by using the library maintenance software to update the catalog and library. If the drafting symbol that corresponds to that component does not exist, it is prepared by the operator of the automated drafting system. Thus the library and catalog grow as the library is used.

AUTOMATED DRAFTING SYSTEM

The drafting system is an ACTRON/TRIDEA Automated Drafting/Digitizing System. It is a flat bet computerized system with the following major components:

Table - The drafting surface is a 4 foot by 5 foot flatbed table. The drafting/digitizing head, which has a 6 position turret and a television camera, is positioned to within .001 inches by feedback controlled XY gantry. The drafting system is controlled by a Varian 620I Minicomputer which reads the drafting data from an input device and positions the drafting head using feedback data from the coordinate measurement system. In the digitizing mode, data from the coordinate measurement system is written to an output device.

Input/Output Devices - The TRIDEA System accepts input data from paper tape, 7 track magnetic tape or from the operator's console and outputs data to either paper tape or 7 track magnetic tape. Data is also kept on a magnetic disc storage device.

Drafting Software - The drafting software, which is known as Al-draft, is a computer program which reads the drafting data, which is in a special purpose drafting language, translates this data into movement commands for the table and controls the movement of the drafting head from feedback from the coordinate measurement unit.

Drafting Language (Appendix 1) - The syntax in which drafting programs are written is simple. Each input record describes one movement of the drafting head. One record tells it where the head is to move to and from its current position, whether the pen is down or up, or whether the movement is in a straight line, an arc, or part of a parabola. There are also auxiliary commands to tell which pen to use, whether the lines are dashed, whether to draw alphanumeric symbols, etc. The language also allows geometry commands such as mirroring, rotating, etc.

Digitizing Software - The digitized software accepts positional information from the coordinate measurement system and outputs data in the drafting language to an output device. The lines on the drawing to be digitized are followed by the television camera under control of either the operator or the automatic line following hardware and output record describing each line or are written on the output device.

SYMBOL RESOLVING SYSTEM

The Symbols Resolving System is implemented by a combination of software and hardware enhancements to the Automated Drafting System. In order to provide storage for the symbols, a disc storage system was added to the hardware. In order to provide access to the symbols on the disc system, a disc operating system and a revision of the input section of Aldraft were added to the software.

The disc provides space for five million characters to be used for symbol definitions and application drafting program storage. The disc operating system provides the means to build and access named data files. These data files are treated as symbol names of Aldraft and which starts reading instructions from a data file rather than the input program whenever a data file's (symbols) name is encountered in the input drafting program.

ALTERNATIVE IMPLEMENTATION

The symbols library concept can be implemented in a number of ways. The system described above is the one which was implemented at ECOM. Its design took advantage of the availability of an on-site large computer (B5700) and the availability of a disc operating system for the automated drafting system.

An alternative is to implement the symbol resolving system as well as the design assistance software on a general purpose computer. In this case the symbol resolving system would be a language pre-processor which would read the input drafting program, look for symbol calls and insert the proper set of drafting commands to produce that symbol on the destination drafting machine. This implementation would allow a designer to implement the symbols library without owning a drafting machine by renting time on a remote drafting machine and using a software symbols resolver to produce the drafting programs for that machine.

Other feasible implementations exist which would provide the symbol library function equally as well. The essence of the function is not in the implementation but rather is providing the designer with a catalog of standard symbols and requiring that he use these symbols in the combined design/documentation effort and that the end product be made from the documentation produced by this effort.

CONCLUSIONS

The Design Symbol Library has been designed as a solution to both the problem of the high cost of design and documentation and the problem of adequacy of documentation for production.

As a test of this system, an interface board was designed using the symbol library (Appendix 4). During the design process the following advantages over the standard design methods were noted:

- The current revision of the design was always available to the designer.
- No sketches were used in the design process.
- The accuracy of the final artwork was .001 inch independent of minor deviation in the designers input since the Drafting System gridded the final artwork.
- The artwork to produce the board was immediately available at the end of the design effort.
- The assembly drawing and the logic diagram were produced by merely changing symbol set and re-running the same drawing tape on the drafting system.
- The parts list was a computer listing from the Design Assistance Program and was, therefore, available without any extra effort.

The Design Symbols Library thus demonstrated that it does indeed solve the problem of inadequate documentation while reducing the cost of the design documentation process. The combining of design and documentation into an integral process results in a more efficient process through elimination of steps as well as insuring the adequacy of documentation.

ACKNOWLEDGEMENT

The Design Assistance Software which is an integral part of the symbols library was designed and implemented by Mr. T. Rorro and Mr. G. Panagos (US Army Electronics Command, CAD-E/CAM Division).

APPENDIX I

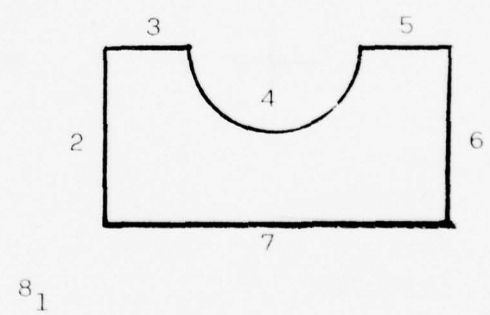
DESCRIPTION OF A SUBSET OF THE ALDRAFT LANGUAGE AND AN EXAMPLE

Aldraft is a special purpose computer language whose purpose is to allow the description of drawings. Each input record in this language is limited to 56 alphanumeric characters and describes either a setup command or one pen motion command. The language in which the commands are described consists of mnemonic operation codes and alphanumeric operands. A description of a subset of these commands is as follows:

OPERATION	MNEMONIC	OPERAND
Move to an X position	X number	X 50;
Move to a Y position	Y number	Y 10;
Move to X,Y position		X 50 Y 10;
Draw an Arc	Arc AnXnYn	Arc A45 X10 Y10;
Lift Pen	D2	D2;
Pen Down	D1	D1;
Write Text	D Number Text	D10 Example;
Number Program Line	N Number	N1
Incremental or absolute positioning	PO A or PO I	

AN EXAMPLE PROGRAM IS:

```
N01X 10 Y 10 D2;
N02 Y 20 D1;
N03 X 15;
N04ArcA180 X 20 Y 20
N05 X 30;
N06 Y 10;
N07 X 10;
N08 X 0 Y 0 D2;
```



The number beside each line on the figure does not appear in the drawing but are added here to aid in understanding the example.

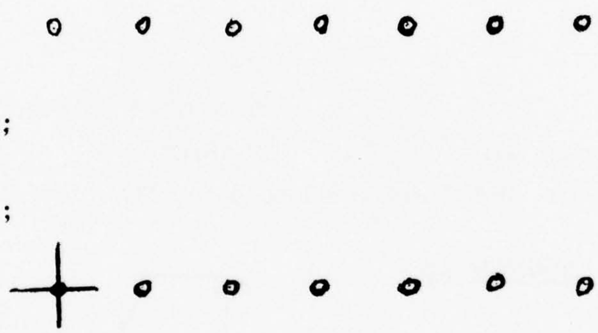
APPENDIX II

SYMBOL DEFINITION AND EXAMPLES

Symbols are files on the Disc Storage unit which contain descriptions in Aldraft of the desired figure. The description is allowed to refer to other symbols in order to define that figure, and, in fact, that is the normal mode of definition. Thus, if PAD075 is defined as:

```
POI;
X.075;
Arc A360 XOYO;
X-.075;
;
```

Then a 14 pin DIP (DIP14) would be defined as:



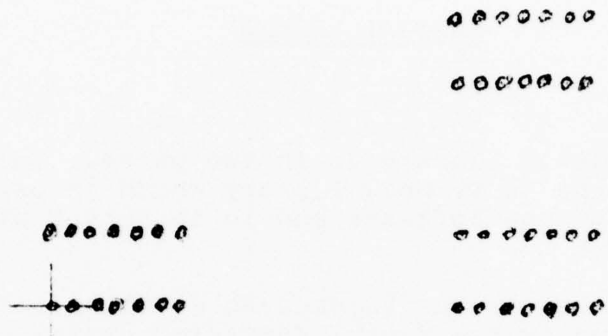
```
        POI;
        XOYO;
(1)     !PAD075;
        X.1
(2)     !PAD075;
        .
        .
        .
        Y.3
(8)     !PAD075;
        X-.1
(9)     !PAD075;
        X-.1;
(14)    !PAD075;
        Y-.3;
        ;
```

Note that symbol calls are denoted by ! symbol and the end of a symbol definition by a record containing only an EOB(=;).

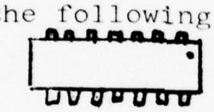
SYMBOL DEFINITION AND EXAMPLES

A layout of 14 pin IC pads would be described as follows:

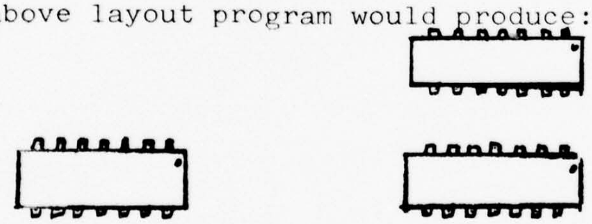
```
    POA
    XOYO;
(1) !DIP14;
    X 2;
(2) !DIP14;
    Y 1;
    !DIP14;
```



If a new set of symbols is loaded onto the Disc where DIP14 produces the following symbol:



Then the above layout program would produce:



This is how assembly drawings are produced from the same program as the artwork.

APPENDIX III

INITIAL LIBRARY

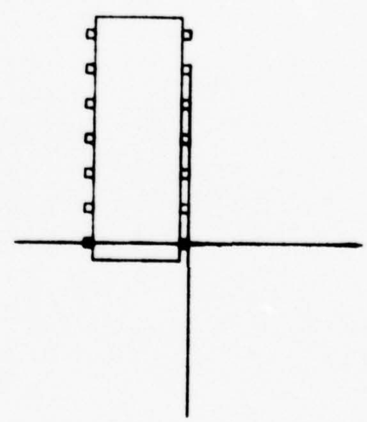
The initial library is in two parts. The first is the (part type to symbol) library which is used by the Design Assistance Software and is in Part 1 of Appendix III.

The second is the (symbol to drafting pattern) library which is used by the Symbol Resolving System and is in Part 2 of Appendix III.

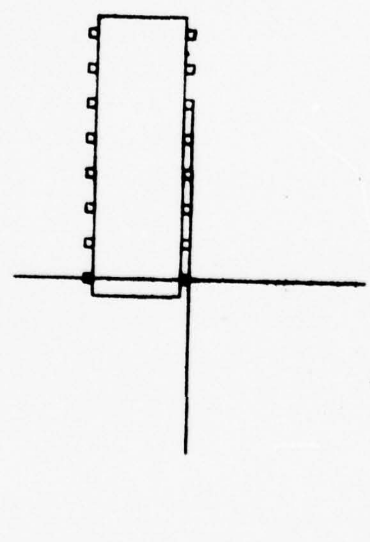
```
1    AY51013        UAR/T UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER  GIANT DIP51013
     DIP40      .2580-.3000     .950          1   5.0   2 -12.0   3   0.0

2    C10U15V        CAPACITOR 10 MICRO-FARADS AT 15 VOLTS                        CAPACITR
     CM15         0.0   0.0         .40

3    C.01U15V       CAPACITOR .01 MICRO FARADS AT 15 VOLTS                       CAPACITR
     RC20         0.0     0.0       .35

4    C.1U15V        CAPACITOR .1 MICRO-FARADS AT 15 VOLTS                        CAPACITR
     RC20         0.0     0.0       .35

5    CON18E.156     PC EDGE CONNECTOR 18 CONTACT POSITIONS-2 CONTACTS PER POSNONE
     CON18E.156.0000-.1990     1.326

6    CON18E.2       PC EDGE CONNECTOR 18 CONTACT POSITIONS .2 INCH SPACEING     NONE
     CON18E.2   .0000  -.20        1.9

7    CON25PCRS232   PC CONNECTOR  25 PIN - CINCH DBM - 25PB                      NONE
     CON25PC232.0000 -0.056    .654

8    CORNER         EDGE MARK FOR P.C. BOARD                                     NONE
     CORNER       0.0   -.2        .25

9    CTRLAB16P      16 POLE DOUBLE THROW SWITCH  CENTRAL LAB                     SW16PCLB
     SW16PC     .0000 -0.120    1.95

10   DE500-107      DUNCAN 7 POLE SINGLE THROW                                   SW07PDUN
     DIP14      .0000-0.150    +0.300

11   MC1488L        QUAD MDTL LINE DRIVER RS-232C  MOTOROLA                      DIP1488L
     DIP14      .5760-0.150    +0.300        7   0.0  14   15.   1  -15.

12   MC1489AL       QUAD MDTL LINE RECEIVERS RS-232C  MOTOROLA                   DIP1489L
     DIP14      .1000-0.150    +0.300        7  0.0   14  5.0

13   MFE5405        PLUG IN CRYSTAL OSCILLATOR - MF ELECTRONICS                  SP5405
     SP14DIP    .400  -.4000     .3000       7   0.0  14   5.0
```

```
14    PR10K5W        POTENTIOMETER 10K OHMS AT 5 WATTS ROUND CASE              VARRESTR
      POTRND.5   5.0   -.075      .0866

15    R1K.5W         RESISTOR 1K OHMS AT .5 WATTS                               RESISTOR
      RC20       0.5   .0         .35

16    S8233F         2-INPUT 4-BIT DIGITAL MULTIPLEXER                          DIP8233

      DIP16      .2000 -.1500    .3500       8  0.0  16  5.0

17    S8H16F         DUAL 4 INPUT NAND GATES ULTRA-HIGH SPEED  SIGNETICS        DIP8H16
      DIP14      .0925-0.150    +0.300       7  0.0  14  5.0

18    S8H70F         TRIPLE 3-INPUT NAND GATES - ULTRA HIGH SPEED  SIGNETICS  DIP8H70
      DIP14      .1386-0.150    +0.300       7  0.0  14  5.0

19    S8H80F         QUAD 2-INPUT NAND GATES - ULTRA HIGH SPEED  SIGNETICS     DIP8H80
      DIP14      .1848-0.150    +0.300       7  0.0  14  5.0

20    SN54121        MONOSTABLE MULTIVIBRATOR                                   DIP121
      DIP14      .0900-0.150    +0.300       7  0.0  14  5.0

21    SN54193        SYNCHRONOUS 4-BIT BINARY UP/DOWN COUNTER - PRESET INPUTS DIP193
      DIP16      .3250 -.1500    .3500       8  0.0  16  5.0

22    SN54H73        DUAL J-K MASTER-SLAVE FLIP FLOPS - HIGH SPEED              DIP73
      DIP14      .3680-0.150    +0.300       4 +5.0  11  0.0

23    SN54L30        8-INPUT POSITIVE NAND GATE - LOW POWER DISSIPATION         DIP30
      DIP14      .0010-0.150    +0.300       7  0.0  14  5.0

24    SN54L54        4-WIDE 3-2-2-3-INPUT AND-OR-INVERT GATE -LOW POWER         DIP54
      DIP14      .0050-0.150    +0.300       7  0.0  14  5.0

25    SN54S04        HEX INVERTERS - LOW POWER DISSIPATION                      DIP04
      DIP14      .0060-0.150    +0.300       7  0.0  14  5.0

26    SN74180        8-BIT ODD/EVEN PARITY GENERATORS/CHECKERS                  DIP180
      DIP14      .1700-0.150    +0.300       7  0.0  14  5.0
```

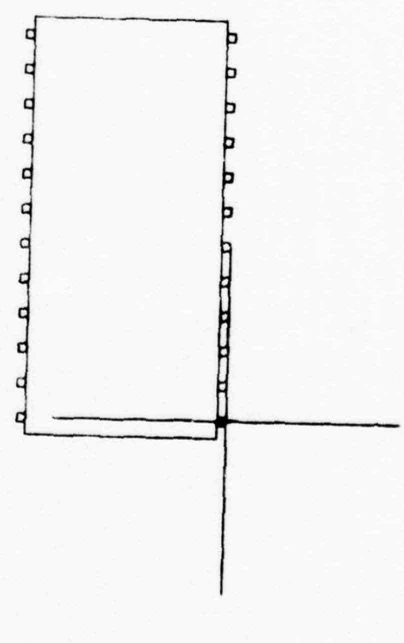
FILE NAME- DIP14
SYMBOL SET-
SCALE- 2 X
DATE- AUG 25, 1975



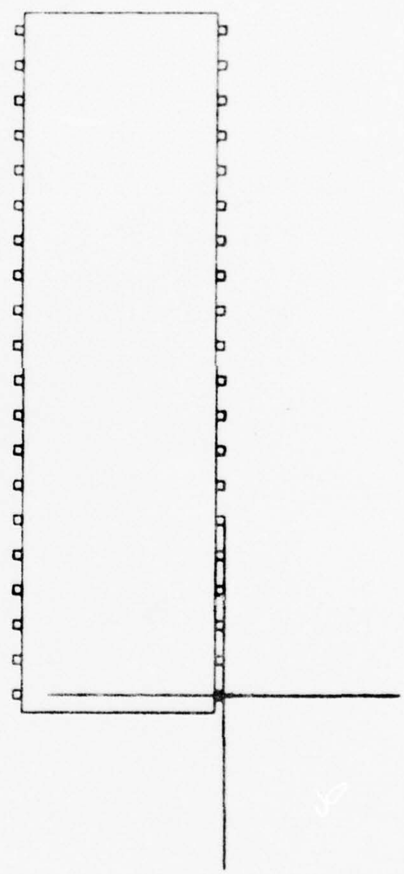
FILE NAME- DIP16
SYMBOL SET-
SCALE- 2X
DATE- AUG 25, 1975



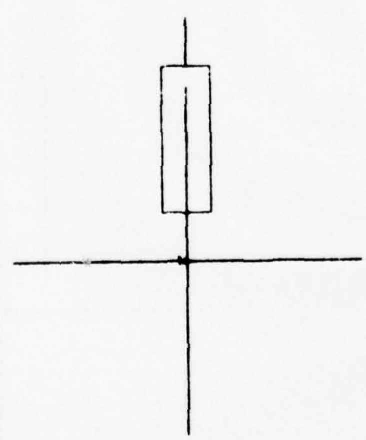
FILE NAME- DIP24
SYMBOL SET-
SCALE- 2X
DATE- AUG 25, 1975



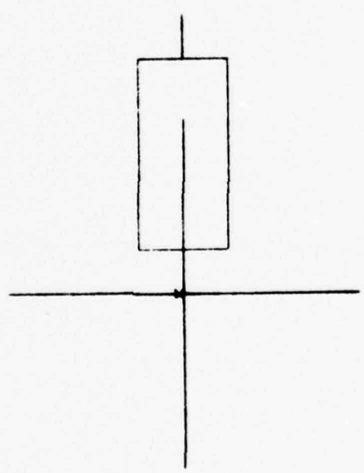
FILE NAME- DIP 40
SYMBOL SET-
SCALE- 2X
DATE- AUG 25, 1975



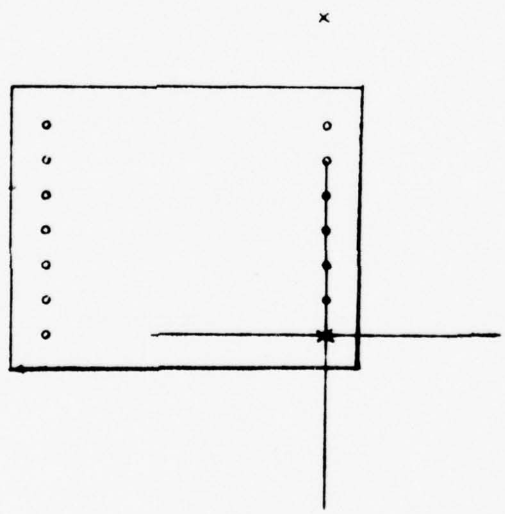
FILE NAME- RC20
SYMBOL SET-
SCALE- 2X
DATE- AUG 25, 1975



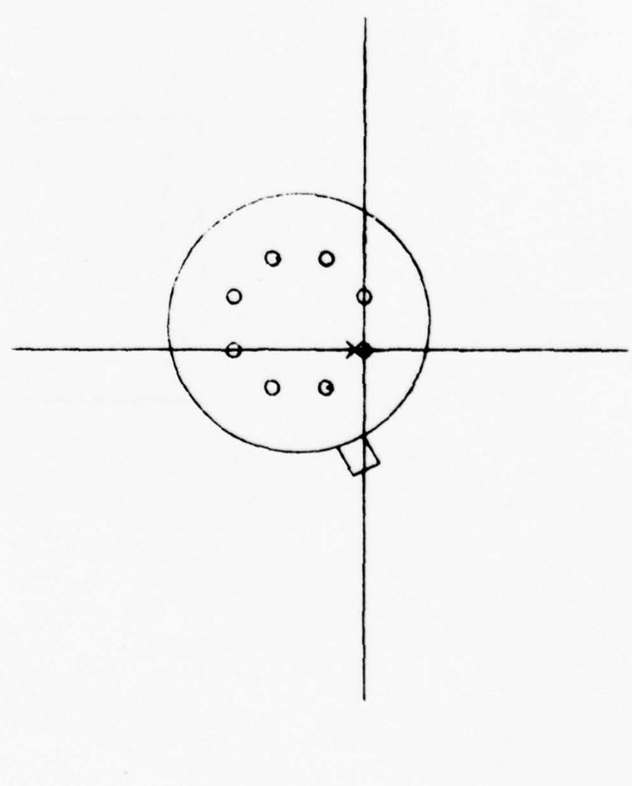
FILE NAME- CM15
SYMBOL SET-
SCALE- 2X
DATE- AUG 25, 1975



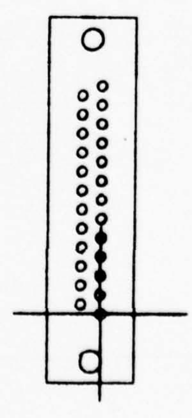
FILE NAME- SP14DI
SYMBOL SET-
SCALE- 2X
DATE- 20 NOV 75



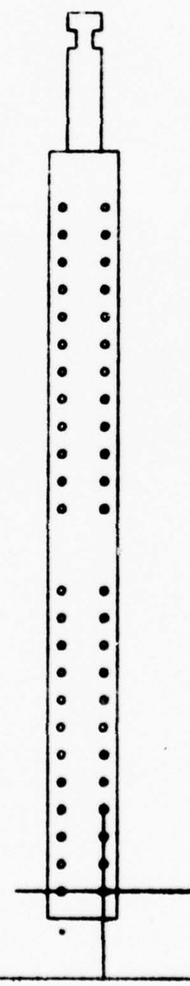
FILE NAME- T099
SYMBOL SET-
SCALE- 4X
DATE- NOV 20, 1975



FILE NAME- CON25PC
SYMBOL SET-
SCALE- 1X
DATE- AUG 25, 1975

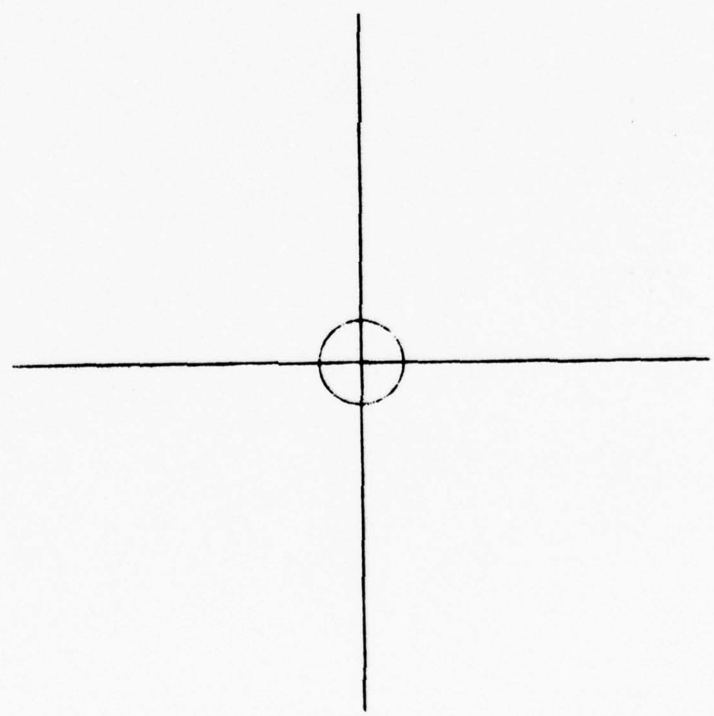


FILE NAME- SW16PC
SYMBOL SET-
SCALE- 1X
DATE- Aug 25, 1975

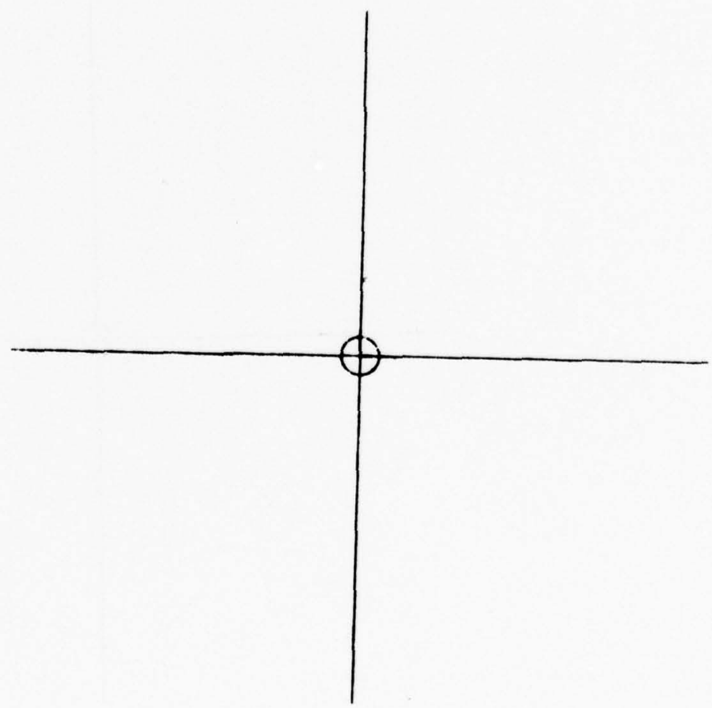


FILE NAME- CON18E
SYMBOL SET-
SCALE- 1X
DATE- AUG 25, 1975

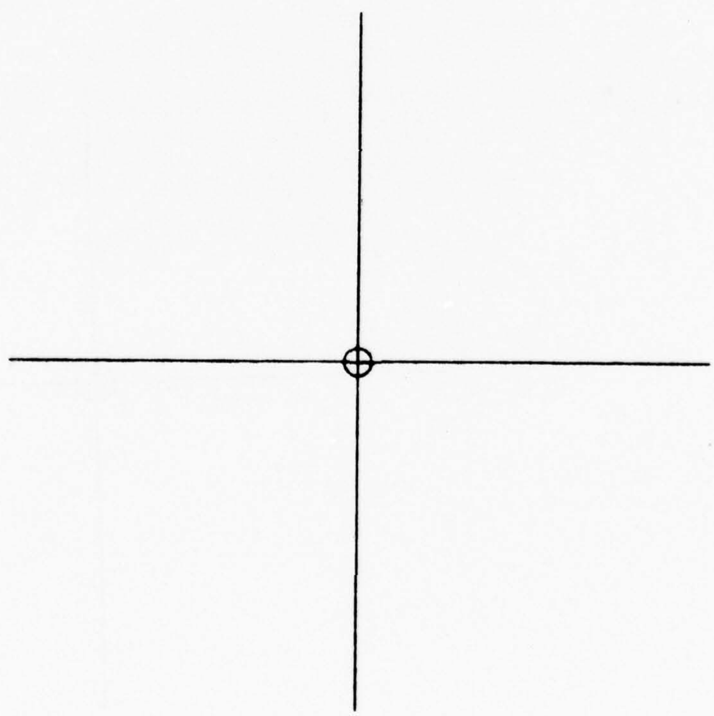
FILE NAME- CIR12C
SYMBOL SET-
SCALE- 4X
DATE- 25AUG 75



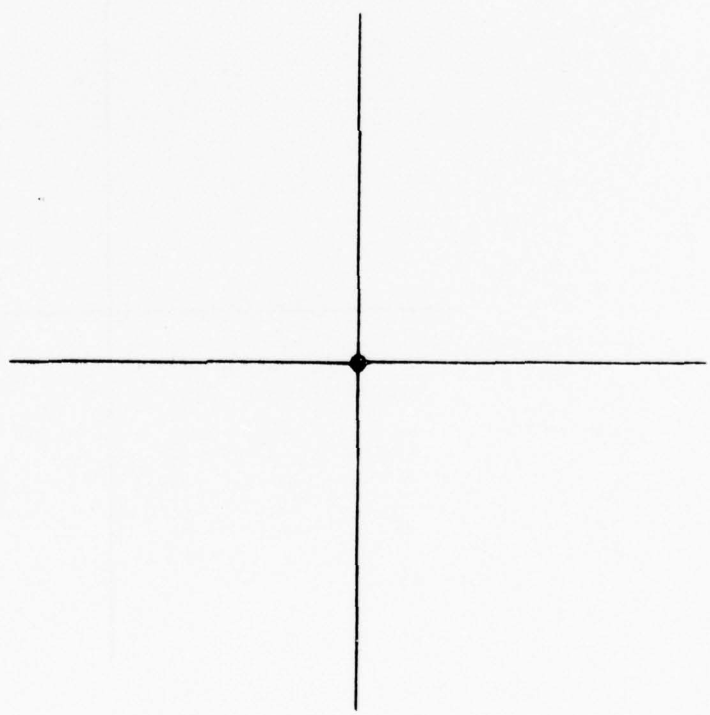
FILE NAME- CIR055
SYMBOL SET-
SCALE- 4X
DATE- 25 AUG 75



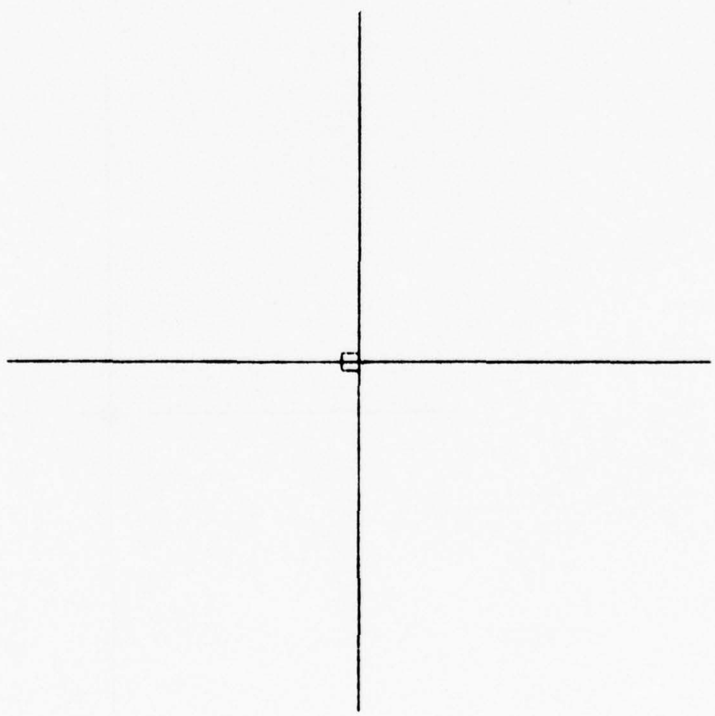
FILE NAME- CIRC40
SYMBOL SET-
SCALE- 4X
DATE- 25 AUG 75



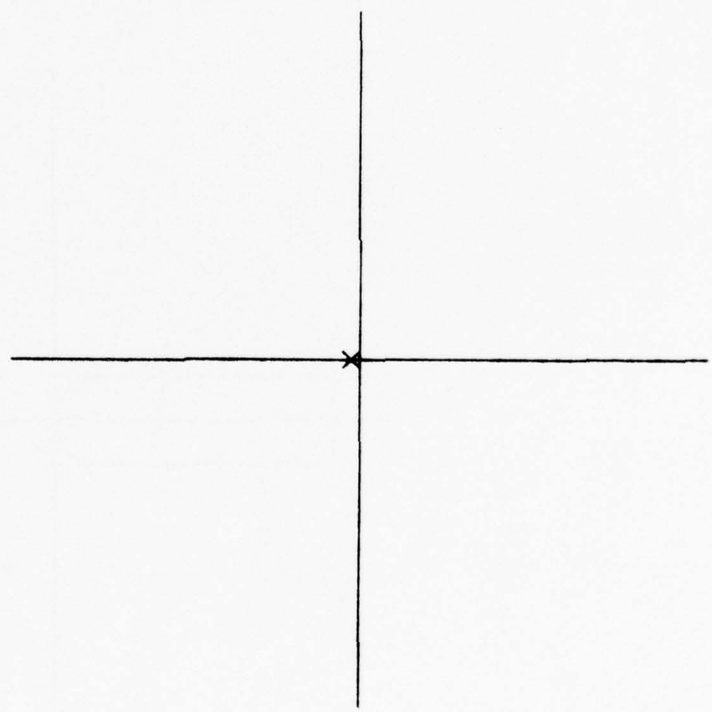
FILE NAME- CIR020
SYMBOL SET-
SCALE- 4X
DATE- 25 AUG 75



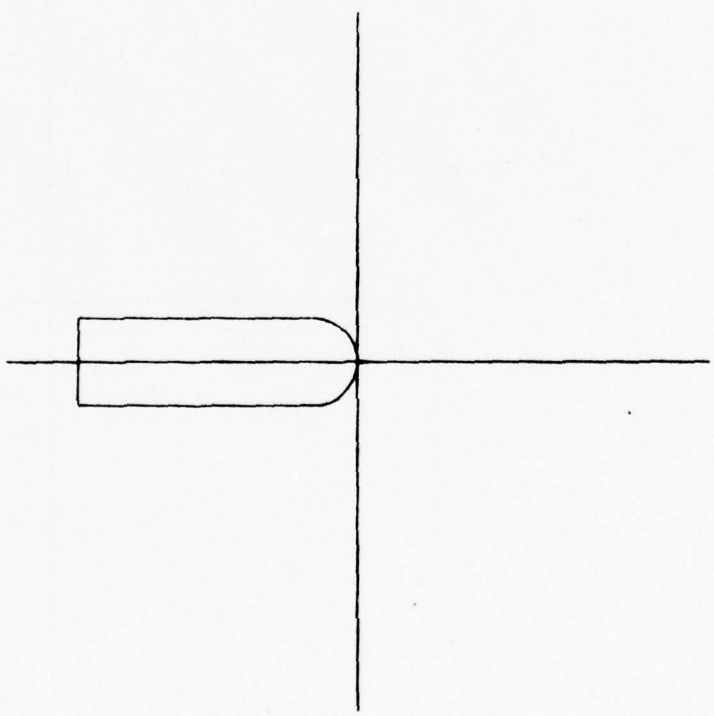
FILE NAME- PIN
SYMBOL SET-
SCALE- 4X
DATE- 25 AUGUST 75



FILE NAME- MARK1
SYMBOL SET-
SCALE- 4X
DATE- AUG 25, 1975



FILE NAME- EPAD
SYMBOL SET-
SCALE- 4X
DATE- 25 AUG 75



APPENDIX IV

DESIGN EXAMPLE

An example of a printed circuit board designed using the Design Symbols Library system is included here:

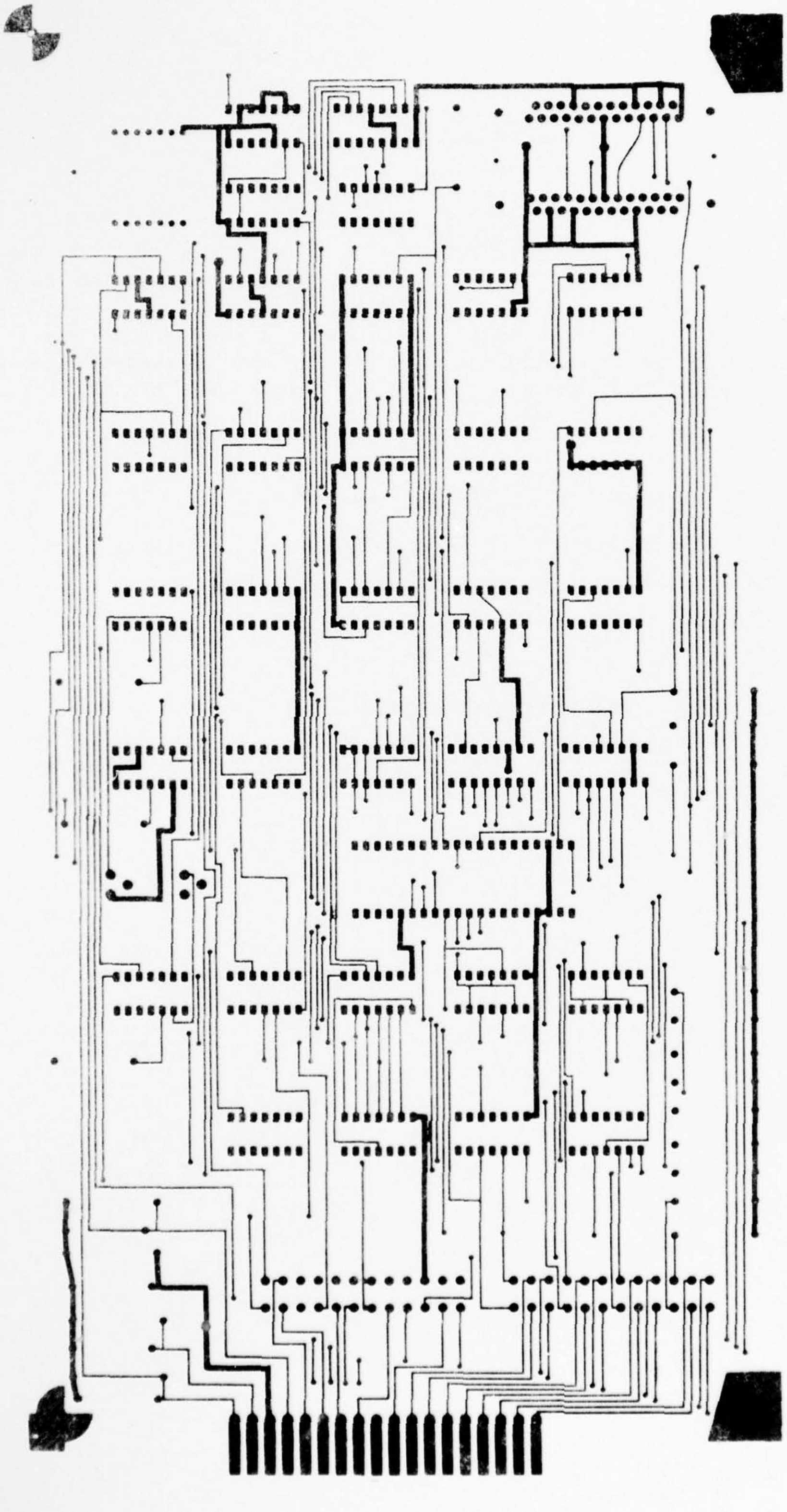
Fig 4.1 is the front of the artwork

Fig 4.2 is the rear of the artwork

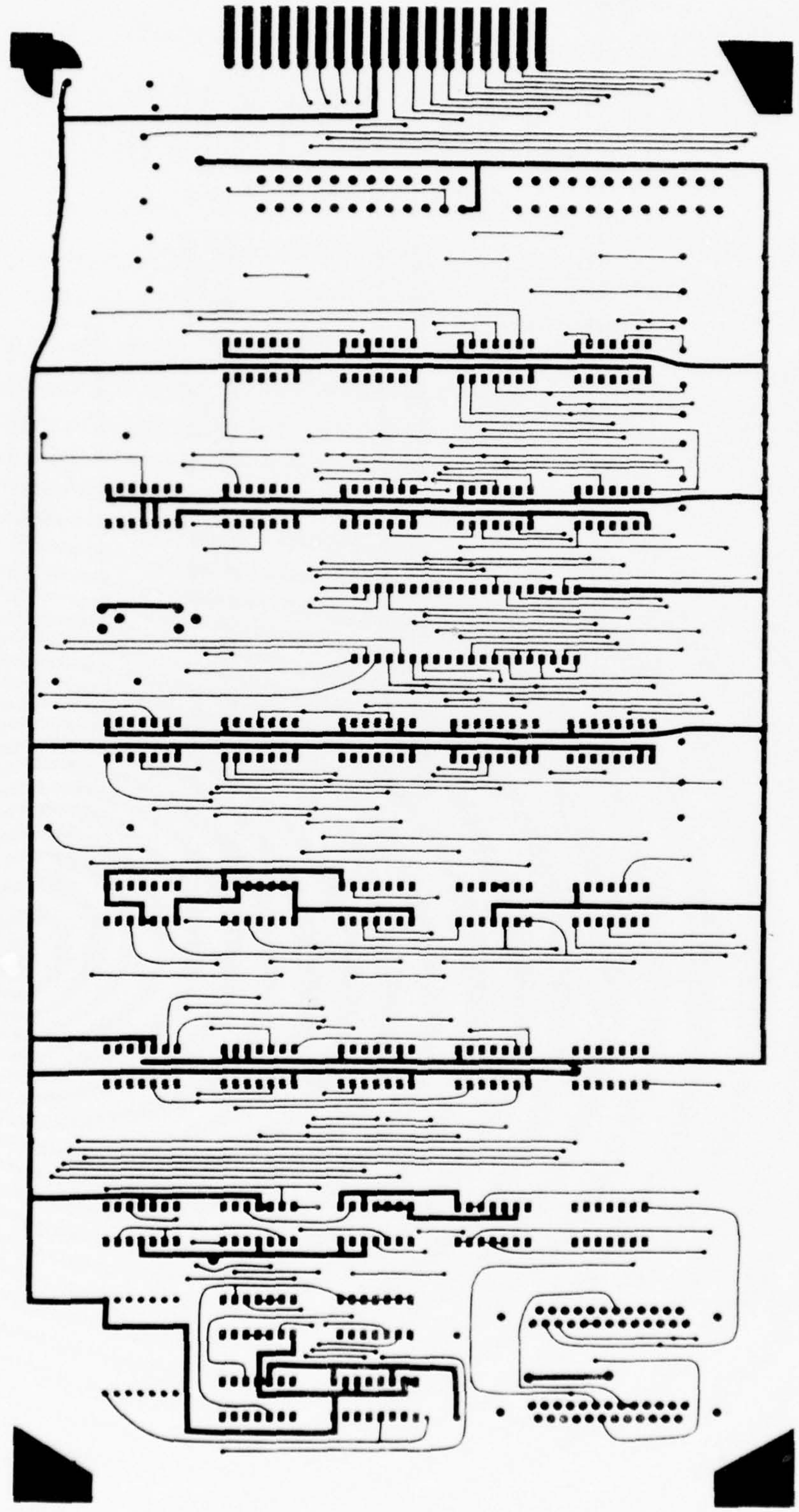
Fig 4.3 and 4.4 are an assembly drawing of the board

Fig 4.5 and 4.6 are a logic diagram of the board

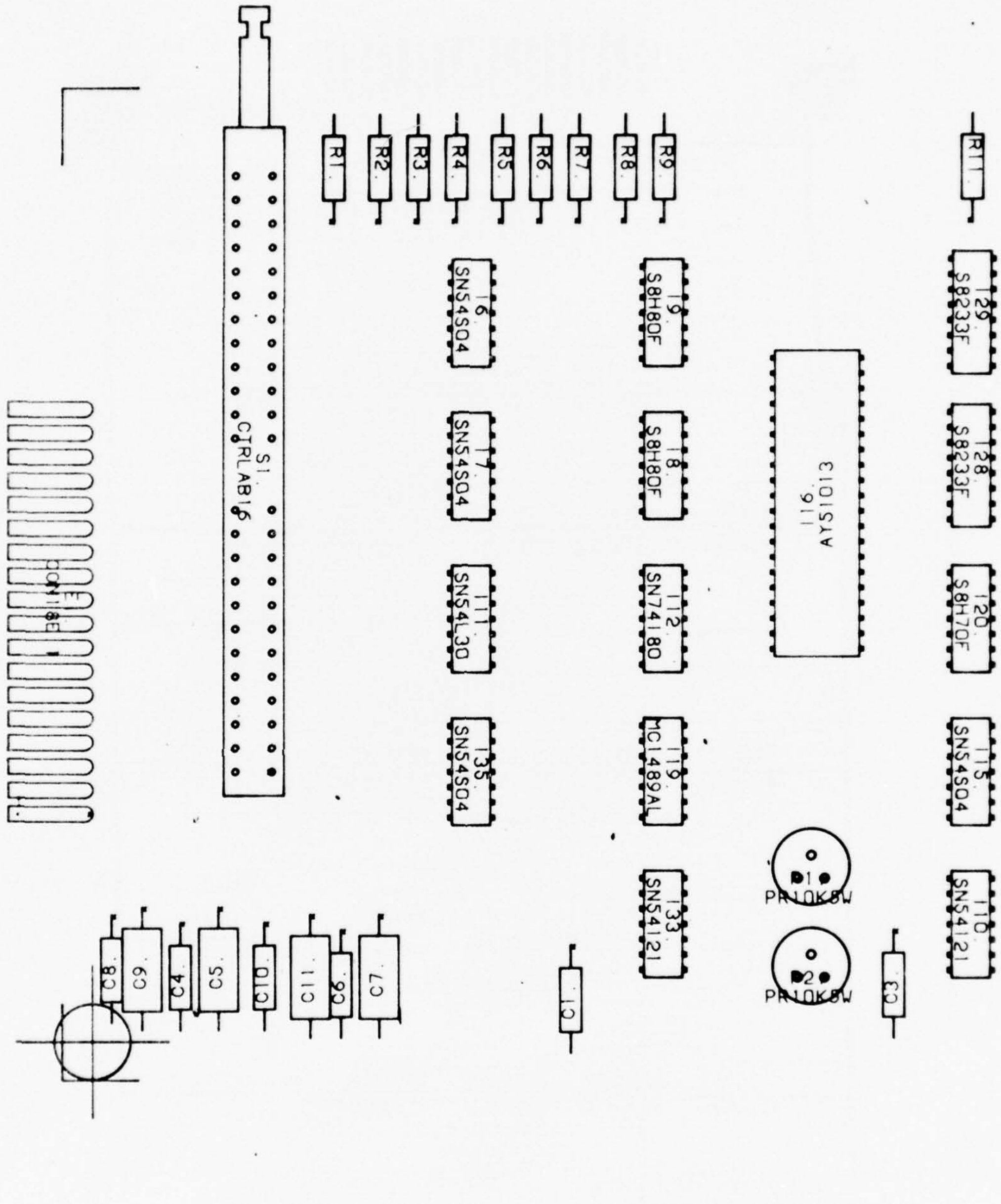
All of these figures were produced from the same lay-out tape by using different symbol set.Routing information was added to produce the artwork and logic diagram and lettering was added to the assembly and logic drawings.



FRONT

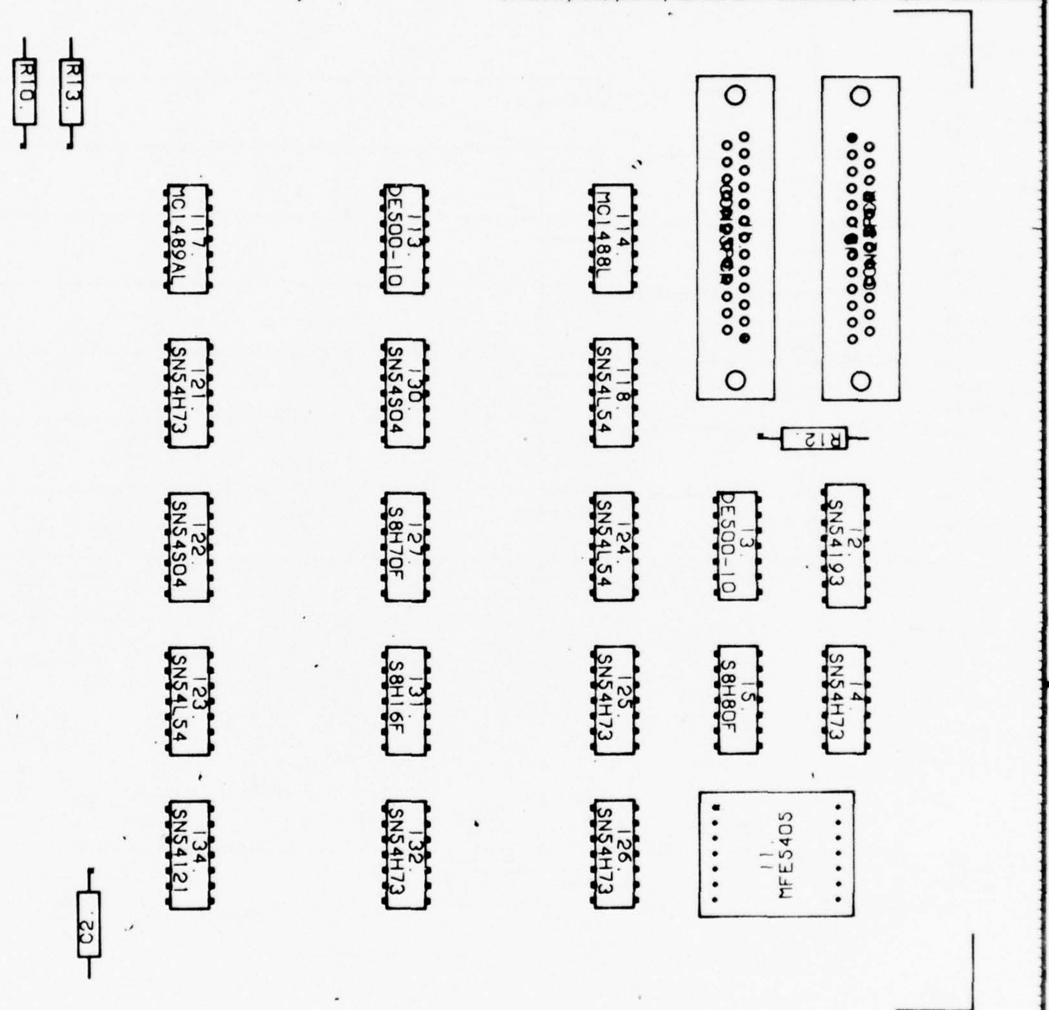


REAR



NOTES:
1. THIS DESIGN IS FOR ALLEN-BRADLEY OR CINCINATTI CONTROLLERS.

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
A	MODIFICATION FOR G.E. CONTROLLER TIE S1 JUMPER TO +5 VOLTS & REMOVE IC I7, I35 AND SHORT PIN PAIRS 1-2, 3-4, 5-6, 8-9, 10-11, 12-13, RESPECTIVELY		



ITEM NO	QTY REQD	PART OR IDENTIFYING NUMBER	NOMENCLATURE OR DESCRIPTION	SPECIFICATION	NOTE

PARTS LIST

NEXT ASSY	USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ON FRACTIONS DECIMALS ANGLES

AUTHENTICATION

DRAWN

CHECKED

VERIFIED

APPROVED

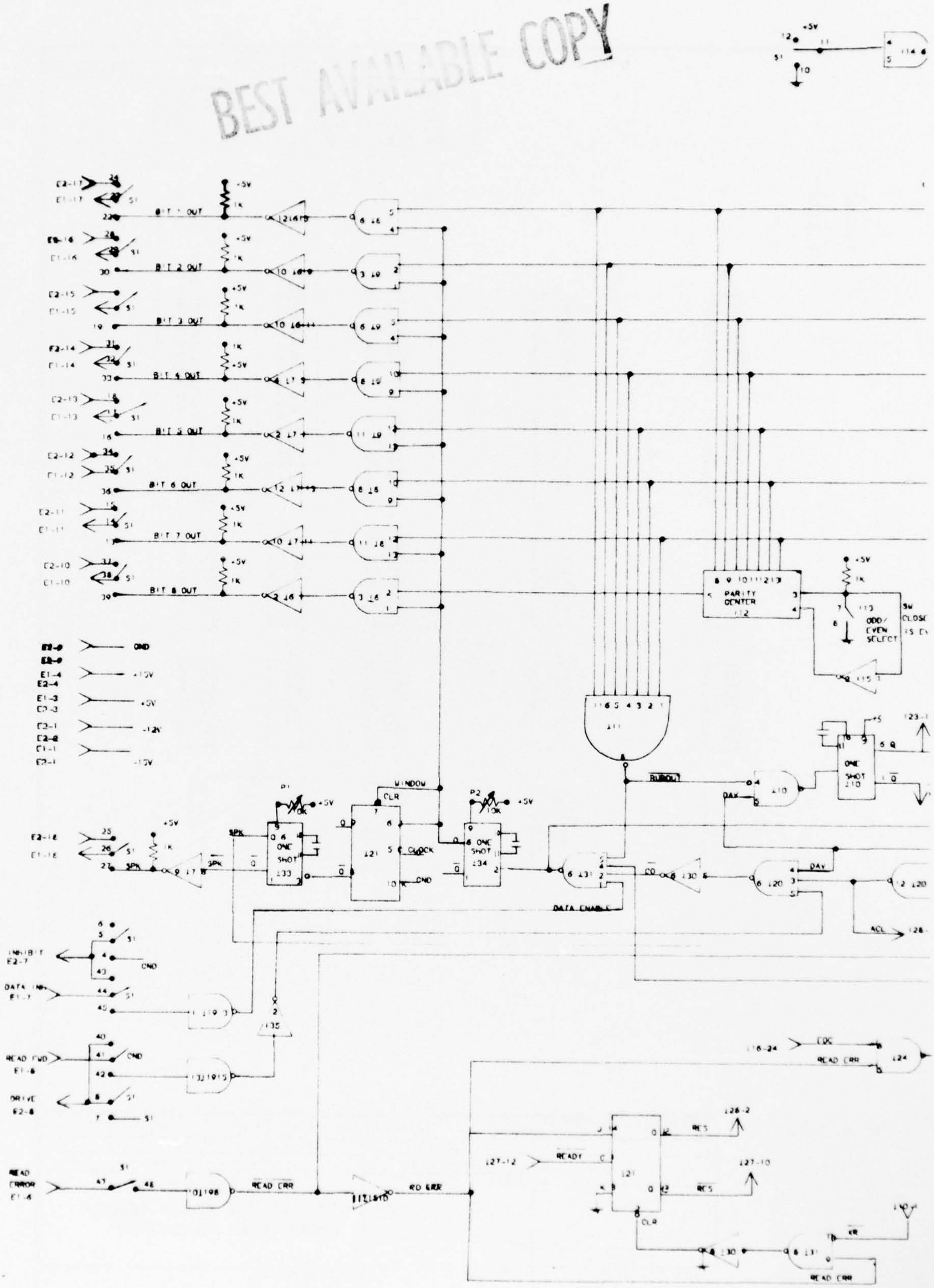
DATE

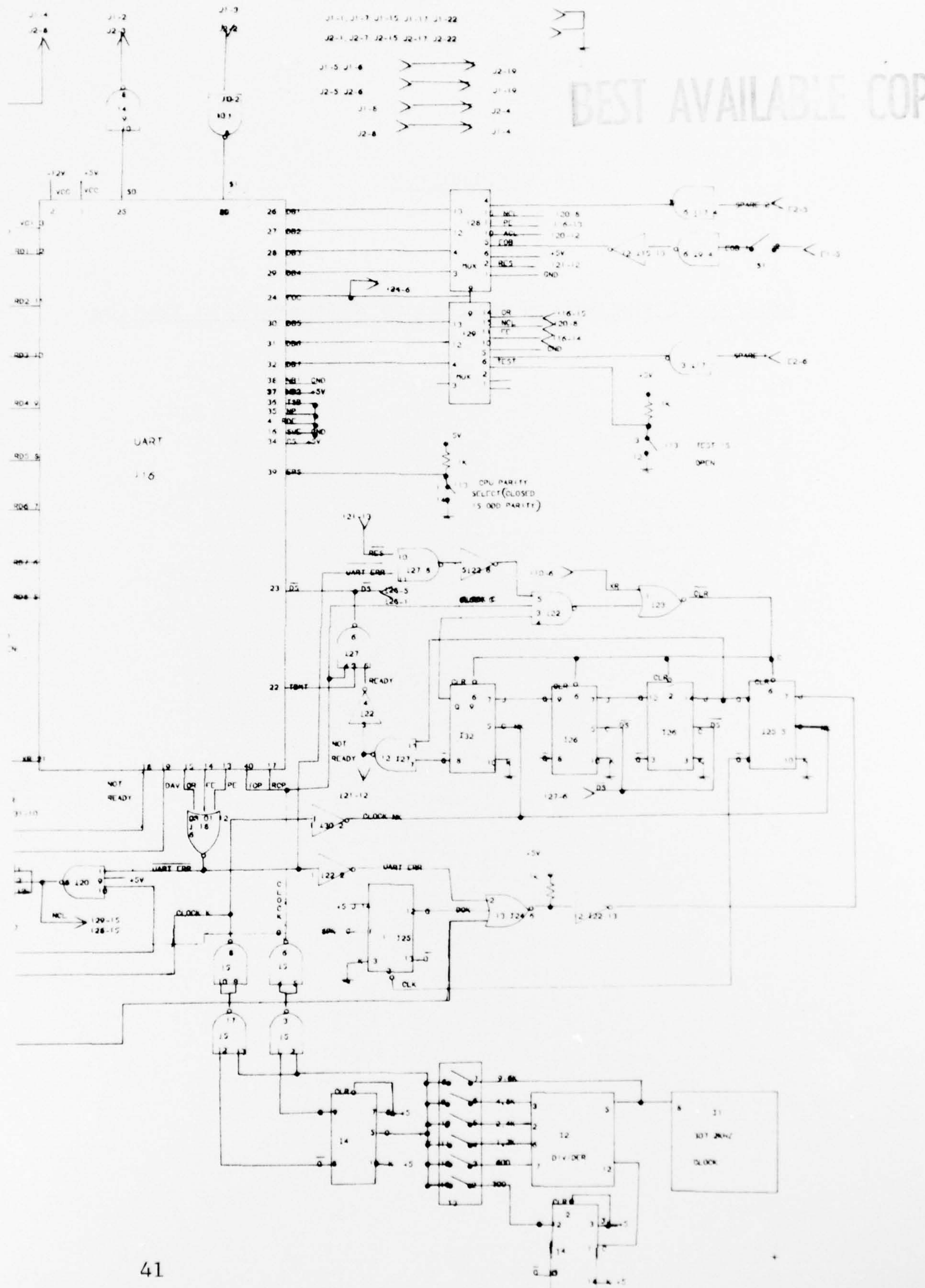
U.S. ARMY ELECTRONICS COMMAND
RESEARCH AND DEVELOPMENT LABORATORIES
FORT MONMOUTH, NEW JERSEY 07703

BEHIND THE TAPE READER INTERFACE ASSEMBLY

SIZE	CODE IDENT NO.	
D	20309	—

SCALE — SHEET —


BIT 1 OUT
BIT 2 OUT
BIT 3 OUT
BIT 4 OUT
BIT 5 OUT
BIT 6 OUT
BIT 7 OUT
BIT 8 OUT
+5V
1K
GND
+5V
-12V
PARITY CENTER
ODD/ EVEN SELECT
ONE SHOT
WINDOW
CLR
CLOCK
DATA ENABLE
RUBOUT
DAV
ACL
INHIBIT
DATA INH
READ FWD
DRIVE
READ ERROR
READ ERR
RD ERR
READY
RES
EOC


UART
MUX
MUX
CPU PARITY
SELECT (CLOSED
IS ODD PARITY)
UART ERR
NOT READY
CLOCK
DIVIDER
CLOCK

APPENDIX V

Listing of the Design Assistance Software Main Routine

```
C*********************************************************************
C
C      THIS PROGRAM IS THE HEART OF THE DESIGNERS DATA BASE.
C
C      THE USER INPUTS HIS COMPONENT LIST WITH
C
C      COMPONENT IDENTIFIER (ALPHA) = COLUMN 1
C      COMPONENT NUMBER             = COLUMN 2 THRU 5
C      COMPONENT PART NUMBER        = COLUMN 9 THRU 20
C      X POSITION IN INCHES         = COLUMN 21 THRU 30       (PIN 1)
C      Y POSITION IN INCHES         = COLUMN 31 THRU 40
C      ORIENTATION IN DEGREES       = COLUMN 41 THRU 45
C      DESCRIPTION OR COMMENTS      = COLUMN 46 THRU 73
C      PAD SIZE FOR COMPONENT       = COLUMN 74 THRU 80
C
C      A MAXIMUM OF 200 COMPONENTS MAY BE ENTERED
C
C      THE PROGRAM SORTS THE DATA IN ASCENDING ORDER BY COMPONENT PART
C      NUMBER.
C
C      A SEARCH IS THEN MADE THROUGH THE LIBRARY TO LOCATE THE COMPONENT
C      PART NUMBER. IF THE PART IS FOUND THE SYMBOL NAME FOR THE TRIDEA
C      DRAWING SYMBOL LIBRARY IS WRITTEN ON THE OUTPUT TAPE.  IF THE
C      COMPONENT IS NOT FOUND A DEFALT SYMBOL IS DRAWN ANT THE APPROPRIATE
C      MESSAGE IS PRINTED.
C
C
C      LUN 18 IS THE PROGRAM LIBRARY TAPE.
C      LUN 19 IS THE TRIDEA TAPE FOR PACKAGE AND PAD DATA.
C      LUN 20 IS THE COMPONENT POWER FILE
C      LUN 21 IS THE COMPONENT PIN FILE
C      LUN 22 IS THE PARTS FILE BY TYPE.
C      LUN 23 IS THE TRIDEA TAPE FOR LETTERING AND SCHEMATIC DRAWINGS.   00015500
C
C *********************************************************************
```

```
C*****************************************************************************
C
CBEGIN - INITALIZATION
C
      LOGICAL KPINS,KPWER,NEW,LABEL
C
C                              KPINS IS FLAG FOR PIN TO VOLTAGE LIST
C                              KPWER IS FLAG FOR POWER LIST
C                              NEW IS A FLAG SET TRUE FOR EACH NEW LIBRARY
C                              RECORD WHICH IS READ
C                              LABEL IS A FLAG SET TO LABEL COMPONENTS
C                              WITH COMPONENT NO OR DRAW INTERNAL SCHMIC
C
      DIMENSION  RLIB(34),RCMP(200,17),TEMP(19)
C
C     RLIB IS THE LIBRARY RECORD BUFFER.
C
C     WORD NO              CONTENTS
C      1 TO  3       COMPONENT TYPE (3A4)
C      4 TO 18       COMPONENT DESCRIPTION (15A4)
C     19 TO 20       SCHEMATIC REFERENCE FOR TRIDEA LIBRARY (2A4)
C     21 TO 23       CASE SYMBOL FOR TRIDEA LIBRARY (2A4,A2)
C     24             POWER  (1F5  )
C     25             CENTER FOR LETTERING X (F10.5)
C     26             CENTER FOR LETTERING Y (F10.5)
C     27,29,31,33    PIN  (1X,F3.0,1X)
C     28,30,32,34    POWER ON PIN (F5.1)
C
C     RCMP IS THE COMPONENT RECORD
C
C     WORD                 CONTENTS
C      1             ALPHA COMPONENT ID. (1A1)
C      2             COMPONENT NUMBER(F4.0)
C      3 TO  5       COMPONENT TYPE (3A4)
C      6             X POSITION PIN 1 (F10.5)
C      7             Y POSITION PIN 1 (F10.5)
C      8             ORIENTATION IN DEGREES (F5.0)
C      9 TO 15       COMPONENT DESCRIPTION (7A4)
C     16             COMPONENT PAD SIZE
C     17             SPARE
C
C *****************************************************************************
```

```
**-******************************************************************

COMMON /TAPE/ IPACK, ILETER
COMMON/IO/ LUN, LUNER
COMMON/CMD/ X, Y

THESE COMMON AREAS PASS INFORMATION TO THE SUBROUTINES WHICH
WRITE ON TRIDEA TAPE


DATA LPWR/20/, LPIN/21/, LCMP/22/
DATA LPL/18/, LLP/6/, LCR/5/, EOF/4H99**/
CALL TIMEON(6)
IPACK = 19
ILETER = 23
LUNER=6

LPL = PROGRAM LIBRARY
LLP = LINE PRINTER
LCR = CARD READER
LUN = TRIDEA OUTPUT TAPE
LUNER= ERROR OUTPUT (LINE PRINTER)
 LPWR = COMPONENT POWER FILE
 LPIN = COMPONENT PIN/VOLT FILE
 LCMP = COMPONENT PARTS LIST FILE
 IPACK = TRIDEA TAPE FOR PACKAGE DRAWING AND PAD CONFIGURATION
 ILETER= TRIDEA TAPE FOR PACKAGE LETTERING OR SCHEMATIC DRAWING


CALL CARDTODISK
                    THIS SUBROUTINE READS THE DATA CARDS AND
                    PUTS THEM ON THE BURROUGHS DISK...........

'EWIND LPL
'EWIND IPACK
'EWIND ILETER
            SET UP TAPES

 UN = IPACK
 PEN = 1
            SELECT TRIDEA PACKAGE TAPE WITH BLACK PEN (PEN 1)
 ALL BEGIN (IPEN)
 UN = ILETER
 PEN = 2
         SELECT TRIDEA LETTERING TAPE WITH RED PEN (PEN 2)
 ALL BEGIN(IPEN)
                     INITALIZE BOTH TRIDEA TAPES.

 RITE(LLP, 140)
                   TOP OF FORM
 RITE(LUNER, 140)
                   TOP OF FORM

 ******************************************************************
```

```
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
CBEGIN INPUT OF PROGRAM CONTROL
C
      CALL INPUT (LCR, SF, BR, KPWER, KPINS, LUNER, LABEL)
C
C                                           SF= SCALE FACTOR
C                                           BR= BASE ROTATION
C                                        KPWER= LIST POWER REQUIREMENTS
C                                        KPINS= LIST VOLTAGE & GND PINS
CEND
C
CBEGIN DRAW ORAGIN
C
      LUN = IPACK                                                       00024450
      CALL ORG (SF)                                                     00024460
      LUN = ILETER                                                      00024470
      CALL ORG (SF)
C                     DRAW A SCALED ORAGIN ON BOTH TAPES.               00024600
CEND
CBEGIN  CARD INPUT SORT
C
      CALL SORTIT (RCMP, LCR, LUNER, LENGTH)
C
C          THIS SUBROUTINE WILL SORT THE INPUT CARD RECORD INTO ARRAY
C          RCMP AND STORE THE NUMBER OF COMPONENTS IN LENGTH.
C
CEND
C
CBEGIN TEST OF RESULT OF CALL TO SORTIT.
C
      IF(LENGTH.GT.0)     GO TO 10
C
C        ELSE
C
         WRITE (LUNER, 100)
C                           WRITE ERROR MESSAGE  *ABORT*
         CALL EXIT
C                           **ABORT***EXIT**
   10 CONTINUE
CEND
C *******************************************************************************
```

```
C*************************************************************************
C
CBEGIN SEARCH OF PROGRAM LIBRARY FOR THE RECORD COMPONENTS AND PROCESS
C       THEM.
C
      NEW=.TRUE.
      DO 80 I=1,LENGTH
C                        SEARCH FOR EACH RECORD IN RCMP ARRAY
C
            IF (.NOT.NEW) GO TO 30
C                                  DONT READ A NEW LIBRARY REFERENCE UNLES
C                                  COMPONENT TEST WANTS IT DONE.
C
   20       READ(LPL,110) (RLIB(J),J=1,34)
C                                            READ A RECORD FROM LIBRARY
            NEW=.TRUE.
C                        A NEW RECORD HAS BEEN READ FROM THE LIBRARY.
            IF(RLIB(1).NE.EOF) GO TO 30
C                                            IF NOT AT LIBRARY END
C                                            GO TO 30
C             ELSE
              WRITE (LUNER,120)
C                                 WRITE ERROR MESSAGE ON LUNER
              DO 25 J=I,LENGTH
C                                  LIST THE COMPONENTS NOT FOUND
                  WRITE(LUNER,130) (RCMP(J,K),K=1,5)
C
   25         CONTINUE
C
              GO TO 81
C                   COMPLETE THE JOB AND OUTPUT REQUIRED LISTS.
C
   30       CONTINUE
C                   NOT AT END OF LIBRARY.
C
C ************************************************************************
```

```
C*********************************************************************
C
C          BEGIN TEST OF COMPONENT VS LIBRARY
C
           DO 60 J=1,3
C                    TEST 3 WORDS OF EACH RECORD
                K=J+2
C                    TEST 1-3 OF LIBRARY AGAINST 3-5 OF COMPONENT
C
                IF(RLIB(J)-RCMP(I,K))      20,60,40
C                                         IF COMPONENT GT LIBRARY READ
C                                         ANOTHER LIBRARY RECORD (20)
C                                         IF COMPONENT EQ LIBRARY DRAW
C                                         THE COMPONENT AS SPECIFIED (60)
C                                         IF COMPONENT LT LIBRARY WRITE
C                                         MESSAGE ON LUNER AND TRY NEXT
C                                         RECORD-RCMP- (40)
C
  40              CONTINUE
C
C                 BEGIN ERROR MESSAGE FOR COMPONENT NOT FOUND IN LBRY.
C
                  WRITE(LUNER,150) (RCMP(I,KK),KK=1,5)
C
C                 END
C                 BEGIN SET UP TO CHECK NEXT COMPONENT AGAINST LIBRARY
                  I=I+1
C                       INCREMENT TO NEXT COMPONENT OF RCMP ARRAY
C
                  IF(I-LENGTH)  30,30,80
C                                          CHECK IF ALL COMPONENTS HAVE
C                                          BEEN PROCESSED. YES THEN 80
C                                          NO THEN CHECK NEXT COMPONENT
C                                          AGAINST LIBRARY(30)
C                 END
C
  60       CONTINUE
C                  A MATCH OF COMPONENT TO LIBRARY HAS BEEN FOUND
C          END
C
C *********************************************************************
```

```
C*******************************************************************
C
C          BEGIN WRITE APPROPRIATE DATA FOR PRINT OUT LATER
C
           CENTX=RLIB(25)+RCMP(I,6)
           CENTY=RLIB(26)+RCMP(I,7)
C                                COMPUTE ABSOLUTE CENTER FROM ORIGIN
C
           WRITE(LPWR,160) (RCMP(I,JJ),JJ=1,5),CENTX,CENTY,
     +                     RLIB(24)
C
C                          WRITE DATA TO THE POWER FILE.  NOTE FORMAT
C                          ICNO.,TYPE,CENTERX,CENTERY,POWER
C
           WRITE(LPIN,170) (RCMP(I,JJ),JJ=1,5),(RLIB(KK),KK=27,34)
C
C                          WRITE DATA TO COMPONENT PIN/VOLTAGE FILE.
C                          NOTE FORMAT.
C
           IF(.NOT.NEW) GO TO 70
C                                      CHECK IF LIBRARY RECORD WAS
C                                      ALREADY OUTPUT. YES THEN 70
C            ELSE
             WRITE(LCMP,310)
             WRITE(LCMP,310)
             WRITE(LCMP,310)
C                                WRITE 3 BLANK RECORDS
             WRITE(LCMP,190) (RLIB(KK),KK=1,18)
C
C                            WRITE LIBRARY DATA TO COMPONENT FILE.
             NEW=.FALSE.
C                        SET FLAG TO INDICATE LIBRARY REFERENCE
C                        HAS BEEN PROCESSED.
C
             WRITE(LLP,190) (RLIB(KK),KK=1,18)
C
C                        DISPLAY LIBRARY COMPONENT AND DESCRIPTION
C                        ON LINE PRINTER
C
   70      CONTINUE
C
           WRITE(LLP,200) (RCMP(I,JJ),JJ=3,5),(RCMP(I,KK),KK=9,15),
     +                    (RCMP(I,LL),LL=1,2),RCMP(I,16)
C
C                        DISPLAY COMPONENT DESCRIPTION ON LINE PRINTER
C
           WRITE(LCMP,210) (RCMP(I,JJ),JJ=6,7), RCMP(I,16),
     +                    (RCMP(I,KK),KK=1,2)
C
C                        WRITE COMPONENT ID ON COMPONENT FILE. AND
C                        PIN 1 POSITION
C          END
C
C *******************************************************************
```

```
C***************************************************************************
C
C           BEGIN TRIDEA SETUP
C
            DO 75 JJ=1,17                                                       00038510
            TEMP(JJ)=RCMP(I,JJ)                                                 00038520
C                                    PUT ACTIVE COMPONENT DATA INTO ARRAY FOR   00038530
C                                    TRANSFER TO TRIDEA SUBROUTINE.             00038535
   75       CONTINUE                                                            00038540
C
            CALL TRIDEA(TEMP,RLIB,SF,BR,LABEL)                                  00038700
C
C           DRAW IC ON THE TRIDEA WITH APPROPRIATE LABEL
C
C
   80 CONTINUE
   81 CONTINUE
C END PROCESS OF COMPONENTS AND LIBRARY
C
C **************************************************************************
C
C
C **************************************************************************
C
CBEGIN TERMINATION OF FILES
C
      WRITE(LPWR,220) EOF
      WRITE(LPIN,220) EOF
      WRITE(LCMP,230) EOF
C                            WRITE END OF FILE TAG ON EACH FILE
      LUN = IPACK                                                               00040250
      CALL THEEND                                                               00040260
      LUN = ILETER                                                              00040280
      CALL THEEND
C                      CLOSE THE FILES
CEND TERMINATION OF FILES
C
C **************************************************************************
```

```
C*********************************************************************
C
CBEGIN DATA LISTS
C
      REWIND LPWR
      REWIND LPIN
      REWIND LCMP
C                    REWIND FILES FOR LISTS
C
      IF(.NOT.KPWER) GO TO 85
C                                        CHECK IF POWER DISSIPATION LST REQ
C                                        NO - GO TO 85
C        ELSE
         WRITE(LLP,240)
C                          TOP OF FORM ON LINE PRINTER AND HEADER
C
   83    READ(LPWR,160) (TEMP(JJ),JJ=1,8)
C                                              READ A RECORD FROM POWER FILE
C
         IF(TEMP(3).EQ.EOF) GO TO 85
C                                        CHECK FOR END OF FILE MARK
C                                        YES THEN 85
C           ELSE
            WRITE(LLP,250) TEMP(3),TEMP(4),TEMP(5),TEMP(1),TEMP(2),       00042200
     1                     TEMP(6),TEMP(7),TEMP(8)
C
C                          PRINT TYPE,COMPONENTNO.,CENT-X,CENT-Y,POWER
            GO TO 83
C                    GET NEXT RECORD
C
   85 CONTINUE
C              FINISHED WITH POWER LIST
C
C *******************************************************************
```

```
C*****************************************************************************
C
      IF(.NOT.KPINS) GO TO 90
C                                   CHECK IF PIN VOLTAGE LIST IS REQ
C                                   NO THEN 90
C       ELSE
        WRITE(LLP,260)
C                    TOP OF FORM AND HEADING
C
   87   READ(LPIN,170) (TEMP(JJ),JJ=1,13)
C                                         READ VOLTAGE DATA
        IF(TEMP(3).EQ.EOF) GO TO 90
C                                   CHECK FOR END OF FILE
C                                   YES THEN 90
C         ELSE
C         BEGIN  COMPUTE THE INDEX OF THE LAST VOLT/PIN PAIR
C
          IEND=5
C                INITALIZE INDEX
C
          DO 89 IX=6,12,2
C                      CHECK EACH OF FOUR PIN NUMBERS TO FIND LAST
C
               IF(TEMP(IX).NE.0.0) IEND=IX+1
C                                              IF PIN NO OS NOT ZERO THEN
C                                              IT IS CONNECTED TO VOLTAGE
   89     CONTINUE
C         END
C
          WRITE(LLP,270)(TEMP(JJ),JJ=1,IEND)
C                                              PRINT  COMP NO,TYPE,PIN,
C                                              VOLT,PIN,VOLT....
          GO TO 87
C                  READ THE NEXT RECORD
C
   90 CONTINUE
C               FINISHED WITH VOLTAGE/GROUND LIST
C
C *****************************************************************************
```

```
C*********************************************************************
C
C      PARTS LIST ALWAYS GENERATED.
       WRITE(LLP,280)
C                     TOP OF FORM AND HEADING
C
   95 READ(LCMP,290)(TEMP(JJ),JJ=1,19)
C                                   READ A RECORD FROM COMPONENT LIST
C
       IF(TEMP(1).EQ.EOF) GO TO 99
C                                END OF FILE THEN 99
C        ELSE
         WRITE(LLP,300)(TEMP(JJ),JJ=1,19)
C                                    WRITE DATA ON LINE PRINTER
         GO TO 95
C                  GO TO READ NEXT RECORD
   99 CONTINUE
       REWIND IPACK
       REWIND ILETER
C                        REWIND BOTH TRIDEA TAPES...
C
       REWIND LPL
C                REWIND PROGRAM FILE
C
       CALL TIMEOFF(6)
       CALL EXIT.
C                    **EXIT**
C *********************************************************************
```

```
C*************************************************************************
C
  100 FORMAT(42H0ZERO RECORDS PROCESSED FROM INPUT DEVICE //
     1          26H **** PROGRAM ABORTED **** /// )
  110 FORMAT(3A4,15A4,2A4,2A4,1A2,F5.4,2F10.5,4(1X,F3.0,1X,F5.1))
  120 FORMAT(42H0END OF FILE REACHED ON COMPONENTS LIBRARY///
     1          36H THE FOLLOWING COMPONENTS NOT FOUND // )
  130 FORMAT(1H ,1A1,F5.0,3X,3A4/)
  140 FORMAT(1H1)
  150 FORMAT(1H0,1A1,F5.0,3X,3A4,5X,22H*NO LIBRARY REFERENCE*//)
  160 FORMAT(1A1,F4.0,3A4,2F12.4,F12.5 )
  170 FORMAT(1A1,F4.0,3A4,4(2F10.2))
  180 FORMAT(3A4,2X,15A4)
  190 FORMAT(1H0,//,1H ,3A4,5X,15A4/)                                  00051300
  200 FORMAT(1H ,3A4,6X,7A4,32X,1A1,F5.0,5X,5HPAD= ,F10.4)
  210 FORMAT( 4H X= ,F10.4, 5X,4H Y= ,F10.4,5X,5HPAD= ,F10.4,5X,
     +11HCOMPONENT= ,1A1,F4.0)
  220 FORMAT (5H     ,A4, 88X)
  230 FORMAT (1A4, 93X)
  240 FORMAT (1H1,25X,23HPOWER DISSIPATION LIST /
     156H     TYPE        COMPONENT    CENTER-X     CENTER-Y     POWER    /
     256H                               INCHES       INCHES       WATTS    //)
  250 FORMAT (1H0,3A4,4X,1A1,F4.0,5X,F10.4,1X,F10.4,1X,F7.3 )
  260 FORMAT (1H1,25X,24HVOLTAGE AND GROUND LIST  /
     110H COMPONENT,5X,4HTYPE,9X,4(3HPIN,2X,5HVOLTS,2X)//)             00052300
  270 FORMAT(1H0,1A1,F4.0,6X,3A4,4X,4(F3.0,2X,F5.1,2X))                00052500
  280 FORMAT(1H1,40X,11HPARTS LIST //)                                 00052600
  290 FORMAT(19A4)
  300 FORMAT(1H0,19A4)
  310 FORMAT( 74(1H ))                                                 00052810
C
      END
```