Fourth Annual Telesis User Group Meeting

Technical Proceedings

Making Technology Work for You
October 1986

FOURTH ANNUAL
INTERNATIONAL
TELESIS USER GROUP MEETING

PROCEEDINGS

CONTENTS **PAGE**

SYSTEM MANAGEMENT

COMPUTER-AIDED ENGINEERING

DESIGN APPLICATIONS

SPECIAL APPLICATIONS & TECHNOLOGIES

FILE MANAGEMENT AND NETWORKING

POSTPROCESSING

GENERAL INFORMATION

REGISTRATION & INFORMATION CENTER

HOURS:
Tuesday, October 14 - Noon to 6 pm
Wednesday, October 15 - 7:00 am to 7:00 pm
Thursday, October 16 - 7:00 am to 7:00 pm
Friday, October 17 - 7:00 am to 2:00 pm

MESSAGES: A message board will be located near the Registration area. All calls, messages, and last minute schedule changes will be posted. Please check the board on a regular basis.

BADGES: Attendees should wear their badges to all scheduled events.

TRANSPORTATION: There is a limo service available to and from the airport. There is also a bus that makes a loop through Disney World from the hotel -- you will need your hotel pass for this free service.

CREDITS AND DISCLAIMERS: Permission to reproduce these papers and proceedings in part, or in full, with credit given to the author and references may be obtained by contacting Telesis Systems Corporation at 617/256-2300, Attention: Mr. Charlie Powderly.

USER CAUCUSES

SESSION CHAIRPERSONS

EASTERN CHAPTER

- Arn Buck, Transcom Electronics
- Paul Gingras, Telesis Systems Corporation

MIDWEST CHAPTER

- Mike Gilman, Research, Inc.
- Charles Powderly, Telesis Systems Corporation

WESTERN CHAPTER

- Real Beaudette, Larse Corporation
- Garrett Cavanaugh, Telesis Systems Corporation

INTERNATIONAL USERS

- Daihachiro Kusunoki, Keihin Artwork & Company, Ltd.
- Paul Gingras, Telesis Systems Corporation

MONITORING CAD SYSTEM PERFORMANCE

Michael A. Hayden

Manager Engineering Services

Broadcast Electronics, Inc.

Quincy, Illinois

ABSTRACT

Cad system managers must assume many responsibilities to insure that their computer-aided design systems perform efficiently and productively. One of these responsibilities is monitoring system performance. The following paper provides the system manager with a method for accumulating , storing and retrieving data in a format that is use- ful for system performance evaluation.

INTRODUCTION

Communication between the system manager and the workstation operators is key in monitoring system performance. When a workstation crashes, the system manager needs to be informed of the sequence of events occurring prior to the crash. If too much time elapses between when the crash occurs, and when the system manager is informed, important details may be lost and go unreported.

The system manager must also have a means of tracking system performance problems. While some problems happen frequently enough to indicate that a trend is developing, other performance trends are not so obvious. This is where it becomes especially important for the system manager to have a tracking method in place.

The customer copies of the Telesis System Performance Reports are useful for tracking system performance, however, they are distributed monthly which hinders the ability to determine a possible trend developing.

In addition, unless the system manager has a way to pull system performance data together, the ability to evaluate overall system performance is hampered as well.

TWO-STEP APPROACH

In the two-step approach, the operator first records performance data as problems occur. Next, the system manager collects the data and formats it so that it's easily analyzed. This method allows the system manager to determine possible trends, and provide feedback to the system operators.

The first step is to provide the operator with a form to record all system performance information. Figure 1 illustrates a sample in-house performance report that may be customized to suit the particular needs of the user.

The upper left hand corner of the form contains information about each work- station. The information is itemized by workstation and includes the serial number, hardware revision, software revision, and software version. (This information is useful to have handy when contacting Telesis Customer Support). When filling out the report, the operator circles the applicable workstation number.

The report should be filled out as completely as possible and include the following items:

1. System operator

2. Date and time

3. Active software - (Mechanical Design, Printed Circuit, or Computer-Aided Engineering)

4. Project name and revision

5. Drawing or file name and revision

6. Nature of the problem - (Software, Hardware, or Documentation). Note that the nature of the problem may not be known at the time a problem occurs. Once a solution has been found, it's important to remember to record the nature of the problem

7. Operation being performed - (Draw Schematic, Draw Symbol, Design Board, or Other)

8. Last command used

9. Error message - List any error messages displayed on the system or monitor

10. Description of the problem

Once a solution or workaround has been found, the operator should describe this in the appropriate location on the form. Additionally, the operator should also indicate whether or not Telesis Customer Support was contacted and if so, record the event number from the system performance report.

Finally, there is an area near the bottom of the report for any notes that might be applicable. Once the report has been completed it is turned into the system manager.

In the second step, the system manager takes the individual reports and assembles them into a common database using a personal computer and a database management software package. There are several quality database management packages on the market, however, the one used to create the report summaries shown in this paper is REFLEX from Borland/Anayltica, Inc.

Figure 2 shows a summary of in-house reports over a six-month period. The type and amount of information entered into the database may vary depending on the needs of the individual user. In this case the following information was extracted from the in-house reports.

1. Date and time
2. Workstation serial number
3. System operator
4. Active software
5. Nature of the problem
6. Project name
7. Drawing or file name
8. Last command used
9. Error message or description of the problem
10. Whether or not Telesis Customer Support was contacted
11. Event number from the Telesis System Performance Report

The REFLEX database management software used to generate this report can accomodate up to 128 field names and each field name can have a maximum of 7Ø characters. The form is designed to fit the user's needs and may be stored away as a "blank" form for future data input.

When the system manager receives the completed report from the operator, the appropriate details are input into the database management software using the "blank" form. Each completed form is then stored away in the software as a record. The records may be displayed individually using "form" view, or in their entirety using the software's "list", "graph", "crosstab", or "report" views.

Figures 2, 3A, 3B, 4A and 4B are all displayed using the "list" view. The data in the summary in Figure 2 was sorted by date and time. This is simply a chronological listing of reports received over a six-month period.

Figure 3A shows the same data sorted by workstation number. Here the number of reports per workstation during the six-month period are evenly split. It can also be noted that workstation number two had four problems in one month, all on the same project.

In Figure 3B, the data was sorted by the nature of the problem. In this instance, the greatest number of problems were software in nature. Further sorting by error message and problem description was done in Figure 4A. This information indicates that there was a three-time occurrence of the same error message during the six-month period. In addition, the error message occurred on both systems.

A sort by project name and drawing or file name was performed in Figure 4B. As shown in Figure 3A, one project had four problems in one month.

CONCLUSION

While filling out the in-house performance report and logging the information into a common database doesn't require a great deal of time, the benefits derived from this process are endless. One such example would be the occurence of the error message "SEND ERROR -2 TO TASK" three times in a six-month period. The reoccurence of the same error message, on different workstations, in more than one command sequence would indicate a trend. The system manager should call attention to this by contacting the Telesis Customer Support personnel.

The ideal situation would be to have a system that would always perform perfectly. However, this isn't the case with any computer-aided design systems in ther real world. There will always be hardware and software problems, but system managers owe it to themselves and to those working for them to be knowledgeable of how well their systems are performing. Using an in-house performance report and database management software is one way of accomplishing this goal.

TELESIS SYSTEM IN-HOUSE PERFORMANCE REPORT

WORKSTATION NO. (CIRCLE)	1	2
SERIAL NO.	1447	1349
HARDWARE REV	2.1	2.1
SOFTWARE REV	G	I
VERSION: EDA 3000	2.1	2.1
EDA 3100	2.1	2.1
EDA 4000	1.1	—

SYSTEM OPERATOR

DATE

TIME A.M. P.M.

ACTIVE SOFTWARE

MD □ PC □ CAE □

PROJECT NAME / REV

/

NATURE OF PROBLEM

SOFTWARE □ HARDWARE □ DOCUMENTATION □

DWG OR FILE NAME / REV

/

DRAW SYM □ DRAW SCH □ DESIGN BD □

OTHER □ ______________________________

LAST COMMANDS USED

LAST:

NEXT:

NEXT:

NEXT:

ERROR MESSAGE

DESCRIPTION OF PROBLEM

SOLUTION/WORK AROUND

TELESIS CUSTOMER SUPPORT CONTACTED? YES □ NO □

TELESIS SYSTEM PERFORMANCE REPORT EVENT NO.______

NOTES

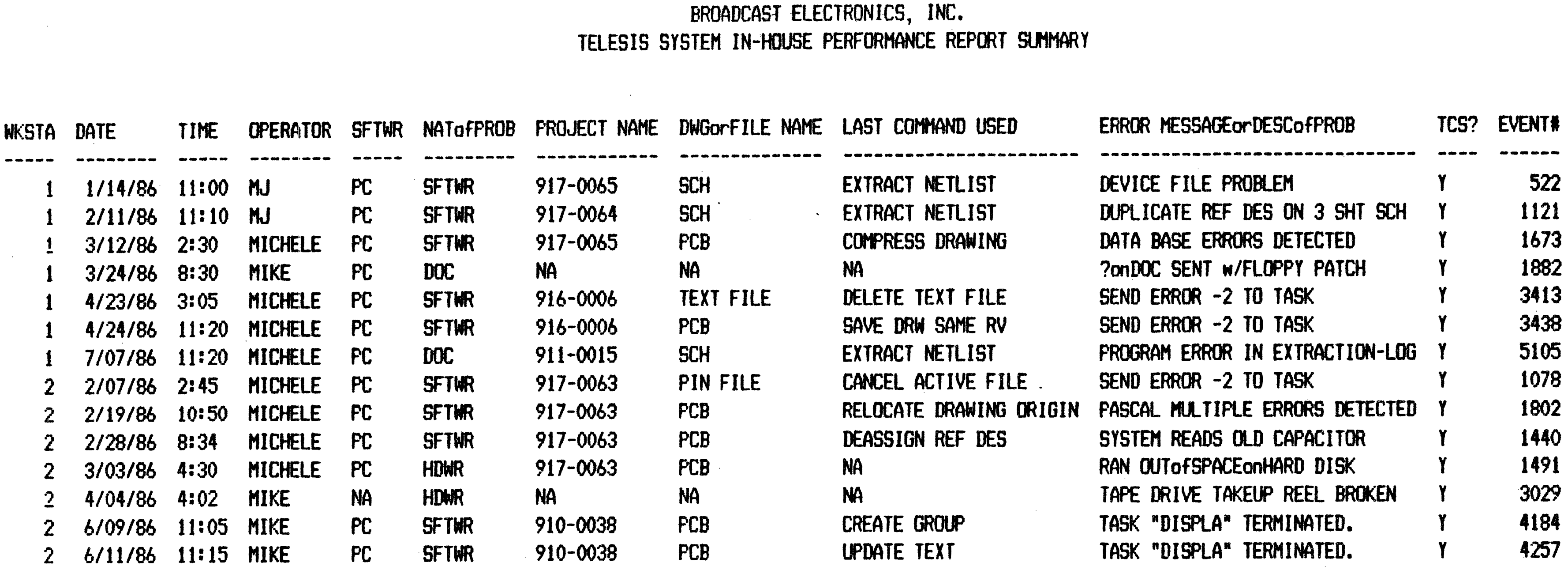
FIGURE 1

BROADCAST ELECTRONICS, INC.

TELESIS SYSTEM IN-HOUSE PERFORMANCE REPORT SUMMARY

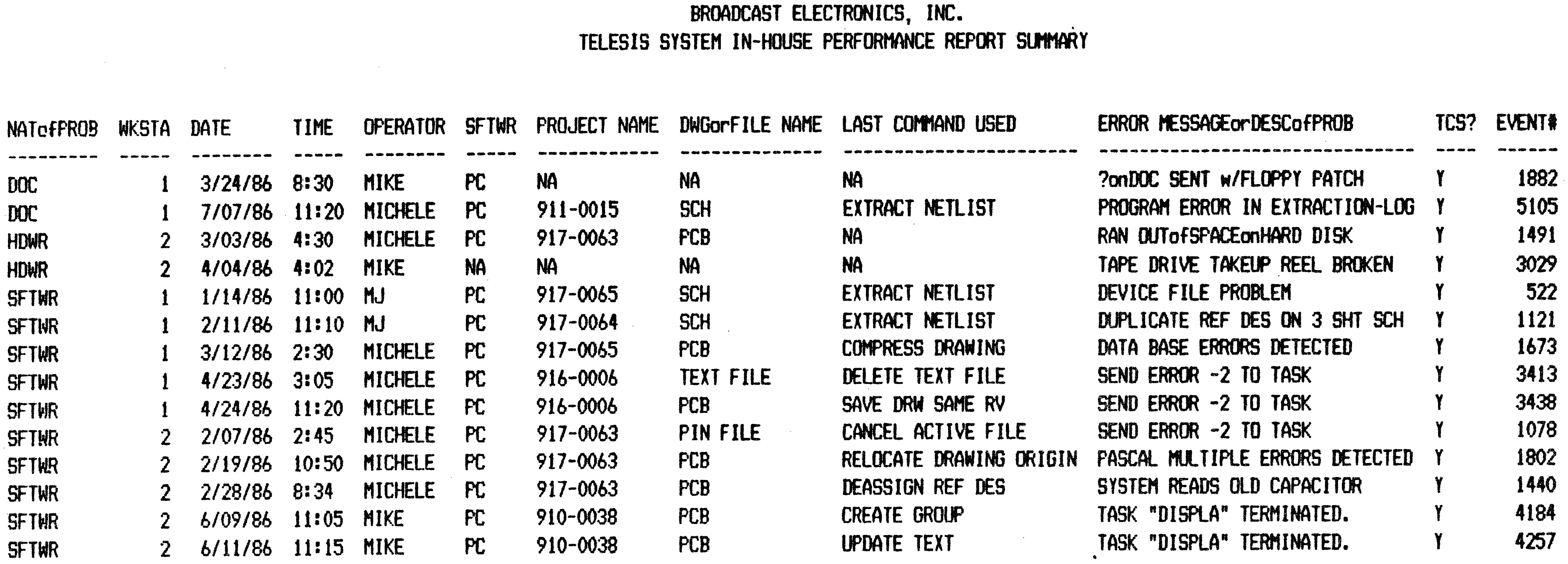
DATE	TIME	WKSTA	OPERATOR	SFTWR	NATofPROB	PROJECT NAME	DWGorFILE NAME	LAST COMMAND USED	ERROR MESSAGEorDESCofPROB	TCS?	EVENT#
1/14/86	11:00	1	MJ	PC	SFTWR	917-0065	SCH	EXTRACT NETLIST	DEVICE FILE PROBLEM	Y	522
2/07/86	2:45	2	MICHELE	PC	SFTWR	917-0063	PIN FILE	CANCEL ACTIVE FILE	SEND ERROR -2 TO TASK	Y	1078
2/11/86	11:10	1	MJ	PC	SFTWR	917-0064	SCH	EXTRACT NETLIST	DUPLICATE REF DES ON 3 SHT SCH	Y	1121
2/19/86	10:50	2	MICHELE	PC	SFTWR	917-0063	PCB	RELOCATE DRAWING ORIGIN	PASCAL MULTIPLE ERRORS DETECTED	Y	1802
2/28/86	8:34	2	MICHELE	PC	SFTWR	917-0063	PCB	DEASSIGN REF DES	SYSTEM READS OLD CAPACITOR	Y	1440
3/03/86	4:30	2	MICHELE	PC	HDWR	917-0063	PCB	NA	RAN OUTofSPACEonHARD DISK	Y	1491
3/12/86	2:30	1	MICHELE	PC	SFTWR	917-0065	PCB	COMPRESS DRAWING	DATA BASE ERRORS DETECTED	Y	1673
3/24/86	8:30	1	MIKE	PC	DOC	NA	NA	NA	?onDOC SENT w/FLOPPY PATCH	Y	1882
4/04/86	4:02	2	MIKE	NA	HDWR	NA	NA	NA	TAPE DRIVE TAKEUP REEL BROKEN	Y	3029
4/23/86	3:05	1	MICHELE	PC	SFTWR	916-0006	TEXT FILE	DELETE TEXT FILE	SEND ERROR -2 TO TASK	Y	3413
4/24/86	11:20	1	MICHELE	PC	SFTWR	916-0006	PCB	SAVE DRW SAME RV	SEND ERROR -2 TO TASK	Y	3438
6/09/86	11:05	2	MIKE	PC	SFTWR	910-0038	PCB	CREATE GROUP	TASK "DISPLA" TERMINATED.	Y	4184
6/11/86	11:15	2	MIKE	PC	SFTWR	910-0038	PCB	UPDATE TEXT	TASK "DISPLA" TERMINATED.	Y	4257
7/07/86	11:20	1	MICHELE	PC	DOC	911-0015	SCH	EXTRACT NETLIST	PROGRAM ERROR IN EXTRACTION-LOG	Y	5105

FIGURE 2. IN-HOUSE PERFORMANCE REPORT SUMMARY SORTED BY DATE & TIME



```
                                        BROADCAST ELECTRONICS, INC.
                               TELESIS SYSTEM IN-HOUSE PERFORMANCE REPORT SUMMARY


WKSTA DATE     TIME  OPERATOR  SFTWR NATofPROB PROJECT NAME DWGorFILE NAME LAST COMMAND USED      ERROR MESSAGEorDESCofPROB      TCS? EVENT#
----- -------- ----- --------- ----- --------- ------------ -------------- ---------------------- ------------------------------ ---- ------
    1  1/14/86 11:00 MJ        PC    SFTWR     917-0065     SCH            EXTRACT NETLIST        DEVICE FILE PROBLEM            Y       522
    1  2/11/86 11:10 MJ        PC    SFTWR     917-0064     SCH            EXTRACT NETLIST        DUPLICATE REF DES ON 3 SHT SCH Y      1121
    1  3/12/86 2:30  MICHELE   PC    SFTWR     917-0065     PCB            COMPRESS DRAWING       DATA BASE ERRORS DETECTED      Y      1673
    1  3/24/86 8:30  MIKE      PC    DOC       NA           NA             NA                     ?onDOC SENT w/FLOPPY PATCH     Y      1882
    1  4/23/86 3:05  MICHELE   PC    SFTWR     916-0006     TEXT FILE      DELETE TEXT FILE       SEND ERROR -2 TO TASK          Y      3413
    1  4/24/86 11:20 MICHELE   PC    SFTWR     916-0006     PCB            SAVE DRW SAME RV       SEND ERROR -2 TO TASK          Y      3438
    1  7/07/86 11:20 MICHELE   PC    DOC       911-0015     SCH            EXTRACT NETLIST        PROGRAM ERROR IN EXTRACTION-LOG Y     5105
    2  2/07/86 2:45  MICHELE   PC    SFTWR     917-0063     PIN FILE       CANCEL ACTIVE FILE     SEND ERROR -2 TO TASK          Y      1078
    2  2/19/86 10:50 MICHELE   PC    SFTWR     917-0063     PCB            RELOCATE DRAWING ORIGIN PASCAL MULTIPLE ERRORS DETECTED Y    1802
    2  2/28/86 8:34  MICHELE   PC    SFTWR     917-0063     PCB            DEASSIGN REF DES       SYSTEM READS OLD CAPACITOR     Y      1440
    2  3/03/86 4:30  MICHELE   PC    HDWR      917-0063     PCB            NA                     RAN OUTofSPACEonHARD DISK      Y      1491
    2  4/04/86 4:02  MIKE      NA    HDWR      NA           NA             NA                     TAPE DRIVE TAKEUP REEL BROKEN  Y      3029
    2  6/09/86 11:05 MIKE      PC    SFTWR     910-0038     PCB            CREATE GROUP           TASK "DISPLA" TERMINATED.      Y      4184
    2  6/11/86 11:15 MIKE      PC    SFTWR     910-0038     PCB            UPDATE TEXT            TASK "DISPLA" TERMINATED.      Y      4257
```

FIGURE 3A. IN-HOUSE PERFORMANCE REPORT SUMMARY SORTED BY WORKSTATION NUMBER



```
                                        BROADCAST ELECTRONICS, INC.
                               TELESIS SYSTEM IN-HOUSE PERFORMANCE REPORT SUMMARY


NATofPROB WKSTA DATE     TIME  OPERATOR SFTWR PROJECT NAME DWGorFILE NAME LAST COMMAND USED      ERROR MESSAGEorDESCofPROB      TCS? EVENT#
--------- ----- -------- ----- -------- ----- ------------ -------------- ---------------------- ------------------------------ ---- ------
DOC           1  3/24/86 8:30  MIKE     PC    NA           NA             NA                     ?onDOC SENT w/FLOPPY PATCH     Y      1882
DOC           1  7/07/86 11:20 MICHELE  PC    911-0015     SCH            EXTRACT NETLIST        PROGRAM ERROR IN EXTRACTION-LOG Y     5105
HDWR          2  3/03/86 4:30  MICHELE  PC    917-0063     PCB            NA                     RAN OUTofSPACEonHARD DISK      Y      1491
HDWR          2  4/04/86 4:02  MIKE     NA    NA           NA             NA                     TAPE DRIVE TAKEUP REEL BROKEN  Y      3029
SFTWR         1  1/14/86 11:00 MJ       PC    917-0065     SCH            EXTRACT NETLIST        DEVICE FILE PROBLEM            Y       522
SFTWR         1  2/11/86 11:10 MJ       PC    917-0064     SCH            EXTRACT NETLIST        DUPLICATE REF DES ON 3 SHT SCH Y      1121
SFTWR         1  3/12/86 2:30  MICHELE  PC    917-0065     PCB            COMPRESS DRAWING       DATA BASE ERRORS DETECTED      Y      1673
SFTWR         1  4/23/86 3:05  MICHELE  PC    916-0006     TEXT FILE      DELETE TEXT FILE       SEND ERROR -2 TO TASK          Y      3413
SFTWR         1  4/24/86 11:20 MICHELE  PC    916-0006     PCB            SAVE DRW SAME RV       SEND ERROR -2 TO TASK          Y      3438
SFTWR         2  2/07/86 2:45  MICHELE  PC    917-0063     PIN FILE       CANCEL ACTIVE FILE     SEND ERROR -2 TO TASK          Y      1078
SFTWR         2  2/19/86 10:50 MICHELE  PC    917-0063     PCB            RELOCATE DRAWING ORIGIN PASCAL MULTIPLE ERRORS DETECTED Y    1802
SFTWR         2  2/28/86 8:34  MICHELE  PC    917-0063     PCB            DEASSIGN REF DES       SYSTEM READS OLD CAPACITOR     Y      1440
SFTWR         2  6/09/86 11:05 MIKE     PC    910-0038     PCB            CREATE GROUP           TASK "DISPLA" TERMINATED.      Y      4184
SFTWR         2  6/11/86 11:15 MIKE     PC    910-0038     PCB            UPDATE TEXT            TASK "DISPLA" TERMINATED.      Y      4257
```

FIGURE 3B. IN-HOUSE PERFORMANCE REPORT SUMMARY SORTED BY NATURE OF PROBLEM

BROADCAST ELECTRONICS, INC.
TELESIS SYSTEM IN-HOUSE PERFORMANCE REPORT SUMMARY

ERROR MESSAGEorDESCofPROB	WKSTA	DATE	TIME	OPERATOR	SFTWR	NATofPROB	PROJECT NAME	DWGorFILE NAME	LAST COMMAND USED	TCS?	EVENT#
?onDOC SENT w/FLOPPY PATCH	1	3/24/86	8:30	MIKE	PC	DOC	NA	NA	NA	Y	1882
DATA BASE ERRORS DETECTED	1	3/12/86	2:30	MICHELE	PC	SFTWR	917-0065	PCB	COMPRESS DRAWING	Y	1673
DEVICE FILE PROBLEM	1	1/14/86	11:00	MJ	PC	SFTWR	917-0065	SCH	EXTRACT NETLIST	Y	522
DUPLICATE REF DES ON 3 SHT SCH	1	2/11/86	11:10	MJ	PC	SFTWR	917-0064	SCH	EXTRACT NETLIST	Y	1121
PASCAL MULTIPLE ERRORS DETECTED	2	2/19/86	10:50	MICHELE	PC	SFTWR	917-0063	PCB	RELOCATE DRAWING ORIGIN	Y	1802
PROGRAM ERROR IN EXTRACTION-LOG	1	7/07/86	11:20	MICHELE	PC	DOC	911-0015	SCH	EXTRACT NETLIST	Y	5105
RAN OUTofSPACEonHARD DISK	2	3/03/86	4:30	MICHELE	PC	HDWR	917-0063	PCB	NA	Y	1491
SEND ERROR -2 TO TASK	1	4/23/86	3:05	MICHELE	PC	SFTWR	916-0006	TEXT FILE	DELETE TEXT FILE	Y	3413
SEND ERROR -2 TO TASK	1	4/24/86	11:20	MICHELE	PC	SFTWR	916-0006	PCB	SAVE DRW SAME RV	Y	3438
SEND ERROR -2 TO TASK	2	2/07/86	2:45	MICHELE	PC	SFTWR	917-0063	PIN FILE	CANCEL ACTIVE FILE	Y	1078
SYSTEM READS OLD CAPACITOR	2	2/28/86	8:34	MICHELE	PC	SFTWR	917-0063	PCB	DEASSIGN REF DES	Y	1440
TAPE DRIVE TAKEUP REEL BROKEN	2	4/04/86	4:02	MIKE	NA	HDWR	NA	NA	NA	Y	3029
TASK "DISPLA" TERMINATED.	2	6/09/86	11:05	MIKE	PC	SFTWR	910-0038	PCB	CREATE GROUP	Y	4184
TASK "DISPLA" TERMINATED.	2	6/11/86	11:15	MIKE	PC	SFTWR	910-0038	PCB	UPDATE TEXT	Y	4257

FIGURE 4A. IN-HOUSE PERFORMANCE REPORT SUMMARY SORTED BY ERROR MESSAGE

BROADCAST ELECTRONICS, INC.
TELESIS SYSTEM IN-HOUSE PERFORMANCE REPORT SUMMARY

PROJECT NAME	DWGorFILE NAME	ERROR MESSAGEorDESCofPROB	WKSTA	DATE	TIME	OPERATOR	SFTWR	NATofPROB	LAST COMMAND USED	TCS?	EVENT#
910-0038	PCB	TASK "DISPLA" TERMINATED.	2	6/09/86	11:05	MIKE	PC	SFTWR	CREATE GROUP	Y	4184
910-0038	PCB	TASK "DISPLA" TERMINATED.	2	6/11/86	11:15	MIKE	PC	SFTWR	UPDATE TEXT	Y	4257
911-0015	SCH	PROGRAM ERROR IN EXTRACTION-LOG	1	7/07/86	11:20	MICHELE	PC	DOC	EXTRACT NETLIST	Y	5105
916-0006	PCB	SEND ERROR -2 TO TASK	1	4/24/86	11:20	MICHELE	PC	SFTWR	SAVE DRW SAME RV	Y	3438
916-0006	TEXT FILE	SEND ERROR -2 TO TASK	1	4/23/86	3:05	MICHELE	PC	SFTWR	DELETE TEXT FILE	Y	3413
917-0063	PCB	PASCAL MULTIPLE ERRORS DETECTED	2	2/19/86	10:50	MICHELE	PC	SFTWR	RELOCATE DRAWING ORIGIN	Y	1802
917-0063	PCB	RAN OUTofSPACEonHARD DISK	2	3/03/86	4:30	MICHELE	PC	HDWR	NA	Y	1491
917-0063	PCB	SYSTEM READS OLD CAPACITOR	2	2/28/86	8:34	MICHELE	PC	SFTWR	DEASSIGN REF DES	Y	1440
917-0063	PIN FILE	SEND ERROR -2 TO TASK	2	2/07/86	2:45	MICHELE	PC	SFTWR	CANCEL ACTIVE FILE	Y	1078
917-0064	SCH	DUPLICATE REF DES ON 3 SHT SCH	1	2/11/86	11:10	MJ	PC	SFTWR	EXTRACT NETLIST	Y	1121
917-0065	PCB	DATA BASE ERRORS DETECTED	1	3/12/86	2:30	MICHELE	PC	SFTWR	COMPRESS DRAWING	Y	1673
917-0065	SCH	DEVICE FILE PROBLEM	1	1/14/86	11:00	MJ	PC	SFTWR	EXTRACT NETLIST	Y	522
NA	NA	?onDOC SENT w/FLOPPY PATCH	1	3/24/86	8:30	MIKE	PC	DOC	NA	Y	1882
NA	NA	TAPE DRIVE TAKEUP REEL BROKEN	2	4/04/86	4:02	MIKE	NA	HDWR	NA	Y	3029

FIGURE 4B. IN-HOUSE PERFORMANCE REPORT SUMMARY SORTED BY PROJECT

JUSTIFYING

CAPITAL

EQUIPMENT

EXPENDITURES

HOW TO SPEND THE BOSSES MONEY

THIS PAPER WAS WRITTEN TO PROVIDE A PROCEDURAL FRAMEWORK FOR THOSE INTERESTED IN PROCURING AN EDA-620 CO-ROUTER.

IN TODAYS HIGH TECH WORLD, BUILDING A QUALITY PRODUCT PRESENTS MANY CHALLENGES:

> THE CONSTANT TIME PRESSURE FOR NEW DESIGNS,
>
> THE SCRUTINY OF UPPER MANAGEMENT OF THE GROUP'S OVERALL PRODUCTIVITY,
>
> AND THE TEAM'S OWN DESIRE TO PRODUCE "PERFECT" DESIGNS THAT WILL FLOW EASILY INTO FABRICATION AND MANUFACTURING.

OVERHEAD

1. DETERMINE WHETHER YOU THINK YOU NEED A CO-ROUTER.
2. CALL YOUR SALES REPRESENTATIVE WHO WILL CONVINCE YOU THAT YOU NEED A CO-ROUTER.
3. COST COMPARISONS:

   OUTSIDE DESIGN SERVICE COSTS

   VS

   IN-HOUSE DESIGN COSTS

   VS

   CAPITAL EQUIPMENT CO-ROUTER COSTS
4. LIST YOUR PAST P.C. BOARD ACCOMPLISHMENTS

   P.C. BOARD DESIGNS (BOARD SIZE AND DENSITY)

   VS

   ACTUAL TIMES
5. PROJECTED WORK FLOW
6. CAPITAL EQUIPMENT PROPOSAL PACKAGE

DO YOU NEED A CO-ROUTER?

DETERMINE HOW MANY NEW PRINTED CIRCUIT BOARDS ARE DESIGNED IN A YEAR.

WHAT TYPE OF BOARDS NEED TO BE DESIGNED:

- ANALOG
- DIGITAL
- HIGH SPEED
- SIZE
- DENSITY
- SURFACE MOUNT, ETC.

CALCULATE MANHOURS THAT ARE SPENT EACH YEAR "REWORKING" THE ELECTRICAL ENGINEERS FIX TO HIS "PERFECT" DESIGN OF LAST YEAR.

IF YOU ONLY DESIGN SINGLE-SIDED, 2" X 3" PCB'S WITH TWO DIP IC'S, 25 DISCRETES AND ONE RELAY, WITH HIGH PRODUCTION QUANTITIES AND LOW COSTS, THEN A CO-ROUTER MAY NOT BE FOR YOU.

OVERHEAD

SALES INQUIRIES

CALL YOUR TELESIS SALES REPRESENTATIVE (HE'S THE GUY WHO ALWAYS CALLS WHEN YOUR SYSTEM IS DOWN AND ASKS "HOW'S IT GOING?"). HE WILL ASSIST YOU IN DETERMINING YOUR PARTICULAR NEEDS.

OBTAIN THE FOLLOWING:

- QUOTES
- TRAINING SCHEDULES
- DELIVERY
- MAINTENANCE AGREEMENTS

OVERHEAD

COST COMPARISONS

OVERHEAD

CO-ROUTER COMPARISONS

REQUEST A DEMONSTRATION ON AN EDA-620 VS YOUR PRESENT SYSTEM WITH A BOARD THAT YOU HAVE ROUTED TO OBTAIN A "ROUTER COMPARISON."

NOW THAT YOU ARE CONVINCED, AND TELESIS HAS YOUR CO-ROUTER ON THEIR SHIPPING DOCK IT'S TIME TO DEMONSTRATE TO MANAGEMENT THE BENEFITS OF THE EDA-620.

OVERHEAD

PAST P.C. DESIGN ACCOMPLISHMENTS

OVERHEAD

PROJECTED WORK FLOW

OVERHEAD

COST SAVINGS

THE COST ANALYSIS IS ACCOMPLISHED BY:

ASSIGNING AVERAGE TIME ESTIMATES TO THE TASKS

PRODUCTIVITY GAINS TO DETERMINE POTENTIAL TIME

TRANSLATE THESE PRODUCTIVITY GAINS INTO DOLLARS AND TIME SAVINGS

OVERHEAD

CAPITAL EQUIPMENT PROPOSAL PACKAGE

CONCLUSION:

I LIKE TO POINT OUT AT THIS TIME THE PRICES STATED ARE TELESIS LIST PRICES.

THE BOTTOM LINE IN EVALUATING ANY ADDITIONAL PIECE OF EQUIPMENT IS THE EFFICIENCY OF THE TOOL AND ITS APPLICABILITY TO THE DESIGN PROBLEM.

EFFORTS EXPENDED EARLY IN YOUR JUSTIFICATION FOR THE EDA-620 WILL PAY OFF WITH INCREASED PRODUCTIVITY, COST SAVINGS AND YOUR CHIEF EXECUTIVE OFFICERS' SIGNATURE ON YOUR PURCHASE REQUESTION.

1 DO YOU NEED A CO-ROUTER?

2 CALL YOUR SALES REPRESENTATIVE

3 COST COMPARISONS

4 PAST P.C. DESIGN ACCOMPLISHMENTS

5 PROJECTED WORK FLOW

6 CAPITAL EQUIPMENT PROPOSAL PACKAGE

EXAMPLE QUOTATION

Prepared For:	Prepared By:
Scan Optics, Inc. 22 Prestige Circle East Hartford, CT	Telesis Systems Corporation Two Omni Way Chelmsford, MA 01824
Attn: Ms. Linda Stegall Manager, Engineering Services	Sales Representative: Heath Wheeler

THANK YOU FOR YOUR INQUIRY. WE ARE PLEASED TO QUOTE YOU AS FOLLOWS:

ITEM	QTY	PART NUMBER	DESCRIPTION	NET AMOUNT
1	1	7010298-001	EDA-620 Routing Accelerator * Motorola 68020 CPU * 2 Mb Memory * 85 Mb Disk * EDA-620 Ethernet Drop * EDA-700 Ethernet Drop * Wyse-75 Terminal * EDA-620 Co-Route Software	$ 53,200.00

This quotation shall remain firm for 30 days from the data hereof, unless modified in writing by TELESIS Corporation prior to our acceptance of your contract offer. This quotation is subject to credit approval and is governed by standard terms and conditions of sale.

Any contract resulting from the quotation must be accepted by TELESIS' corporate office by a duly authorized representative.

SUBTOTAL	$ 53,200.00
INSTALLATION	$ 532.00
MAINTENANCE	$ 5,320.00
NET TOTAL	$ 59,052.00

ESTIMATED DELIVERY SCHEDULE (Subject to Modification by Telesis) ___________

DATE OF ISSUE: PAGE 1 of 1

DELIVERY DATES WILL BE CONFIRMED UPON ACCEPTANCE OF YOUR CONTRACT OFFER

PCB NAME:	TERMINAL CONTROLLER I/F
DWG. NO.	600ML577-1
DESIGNER.	SHERRY RIO

OUTSIDE DESIGN SERVICE COSTS

DESIGN	$ 16, 607.00
PHOTOPLOT	
DRILL TAPES	
DOCUMENTATION	

IN-HOUSE COSTS

LABOR

ENTRY	152 HOURS	$ 3720.36
DESIGN + DOCUMENTATION	106 HOURS	$ 2976.29
PHOTOPLOTTING + DRILL TAPE		$ 2000.00
	TOTAL COST	$ 8696.65

SUPPLIES + MAINTENANCE

$$\frac{\$\ 11,559 \times 258 \text{ HOURS}}{2080} = \$\ 1434.48$$

TOTAL IN-HOUSE COSTS $ 10131.13

SAVINGS

DESIGN SERVICE COSTS	$ 16, 607.00
IN-HOUSE COSTS	$ 10, 131.13
TOTAL SAVINGS	$ 6, 475.37

ROUTER COMPARISON

EDA-620 (co-router) vs EDA-300

Master Controller

EIC = 111
density = .71
conns = 538

	EDA-620		EDA-300	
	minutes	conn. comp.	minutes	conn. comp.
set-up	5	-	20	-
pass 1&2	.5	238	49	221
pass 5 (4 executes)	36	279	163	280

EDA-620 routed to 96.10% in 41.5 minutes

EDA-300 routed to 93.87% in 232 minutes

Mother Board

EIC = 619
density = .41
conns = 2788

	EDA-620		EDA-300	
	minutes	conn. comp.	minutes	conn. comp.
set-up	33	-	68	-
pass 1&2	6	2104	128	2258
pass 5 (10 executes)	128	631	**	

**Pass 5 not run

NEW PCB'S

PCB	Designer/ Engineer	Status	Board Size, No. of Holes, No. of Comp.	Schematic Start Date	Schematic Completion Date	Design Complete	Photoplot Complete	CAD Lib. Number
1. Page Memory Controller #1 P/N600ML534	Fred B.		Euro Size 225 IC's					129
2. Page Memory Controller #2 P/N600ML535-1	Fred B./ Sherri		Euro Size 150 IC's					126
3. Stacker Controller Vertical Replaces 701648-2	Bruce/		7 x 11.5 75 equiv. IC's					130
4. Stacker Controller Horizontal w/w Replaces 701649-2	Bruce/ Wayne		7 x 11.5 75 equiv. IC's					131
5. AC Distribution	Joe							134
6. Address & Handshake Goes in MacRack (Modified) Kebus	Joe		3 x 4 PCB					135
7. Video Processor	Earl		Euro Card 100 IC's					136
8. HP SCSI P/N600563-1	Dave G./ Mike		4 layers 1,800 holes					137
9. Transport Interface	Dave S.		VME 15 IC's					138
10. VME B/Bus Interface	Dave S./ Mike		VME 6 x 9 60 IC's					139
11. Video Memory Interface	Dave S.		VME 6 x 9 25 IC's					140
12. Paddle Board To VME P/N600582	Dave S. Mike							141
13. PC & Tachometer Signal Conditioning	Dave S./ Wayne		VME 18 I/C					142
14. EM & Control Panel I/F & Power Monitor	Dave S./ Wayne		VME 10 I/C					143
15. EM K-Row Adaptor	Dave S./ Wayne		5 x 5 4 I/C					144
16. Reset Control & Configuration	Dave S./ Wayne		VME 5 I/C					145

CAD UPDATES

PCB	Designer/ Engineer	Status	Board Size, No. of Holes, No. of Comp.	Est. Density	Est. Time Schematic	Est. Time Design	Actual Time	CAD Lib. Number	PCB's Required In House By:
1. B-Bus Driver P/N701677-1 (600ML578) REV. A	Brian/ Mike	5 x 8 ECR						125	
2. B-Bus Converter P/N701676-1 (600ML578-1) REV. A	Brian/ Sherri							113	
3. Stacker Controller Horizontal P/N701649-2	Bruce/ Wayne	7 x 11.5 75 Equiv. IC's						131	
4. HP Video Interface P/N600433-1 701701-1 801195	Joe F. Sherri	13 IC's						148	
5. DMA COMMS	Dave S.								
6. Convert 11/73 PCB to be Universal Using 11/84 PCB	Mike J.								
7. Z80 Controller MF and Ink Jet (wirewrap replacement)		NEW 7 x 16 Augat Size							
8. Peripheral Control P/N701617-1 REV. A	Art/	ECR Wirelist Entry Update						118	
9. Control Panel Adapter P/N701669-1 REV. A	Art/ Mike	ECR 22 wires						112	
10. Relay Interface P/N701524-1 REV. B	Art/ Wayne	NEW ECR 29 wires 2 x 4 2 IC							
11. Photocell Gate Jam Detail P/N701727-1		NEW							
12. Interface P/N701618-1 REV. A	Art/	ECR 18 wires			Sch. Entry Only	NOTE:	This PCB Was Not Designed On CAD But A W/L Was Previously Entered.		
13. Stacker Controller Vertical P/N701648-2	Bruce/ Mike	7 x 11.5 75 Equiv. IC's						130	
14. VME B-Bus/PM Bus Extender	Sherri/ Giordanella							150	
15. B-Bus/PM Bus Extender	Sherri/ Giordanella							151	

COMPLETED PCB's

PCB	Designer/ Engineer	Status	Board Size, No. of Holes, No. of Comp.	Est. Density	CAD-CAM Library Number	Schematic and Design Actual Time	Released to Production
1. * Gate Pulse Driver P/N 600565-1	Mike/ B. Millar	Done	5 x 6 (2 layer) 484 holes 41 comp.	1.03 ic/in	104	96 hrs. 2.4 wks.	12-06-85
2. Relay Driver Test Circuit P/N 600559-1	Mike/ B. Millar	Done	5 x 6 (2 layer) 322 holes	1.03 ic/in	103	84 hrs. 2.1 wks.	12-06-85
3. Mag Tape Controller CCI P/N 600529-1	Jonathan/ Mike J.	Done	10 x 12 (8 layer) 3,252 holes	1.97 ic/in	115	413.5 hrs. 10.3 wks.	12-11-85
4. * Stacker Clutch Driver P/N 600551-1	Mike F./ Bruce	Done	5 x 5		102	171 hrs. 4.275 wks..	12-06-85
5. Grant Continuity P/N 600536-1	D. Fincher/ Stone	Done	10 x 12	N/A	116	51.3 hrs. 1.3 wks.	10-11-85
6. * Kebus Prototype Bd. P/N 600537	Jonathan/ Stone	Done	10 x 12	N/A	117	142.9 hrs. 3.6 wks.	05-10-85
7. Peripheral Controller P/N 600523-1	Jonathan/ Art	Done	7 x 11.5	N/A	118	660.5 hrs. 16.5 wks.	05-30-85
8. * Jam Logic P/N 600521 (Wire List Entry)	Jonathan/ Art	Done	7 x 11.5			561.5 hrs. 14.0 wks.	10-23-85
9. SP370-2 Repl P/N 600550-1	Mike	Done	1.8 x 2.6	N/A	110	99 hrs. 2.5 wks.	
10. 11/73 Interface P/N 600541-1	Jonathan Mike J.	Done	10.5 x 12.5	1.25 ic/in		240 hrs. 6 wks.	02-21-85
11. SMCA/DMA P/N 600525-1	Jonathan/ Stone	Done	10 x 12	.87 ic/in	119	260 hrs. 6.5 wks.	10-11-85
12. Kebus Extender P/N 600543-1	Jonathan/ Stone	Done	10 x 12	N/A	120	53.9 hrs. 1.3 wks.	
13. Edit Terminal I/O Panel P/N 600567-1 REV. A	Mike/ Joe	Done	16.5 x 5.5 2 layers 1,358 holes	.87 ic/in	106	154 hrs.	Ordered 1-23-86
14. VFD Protocol Converter P/N 600574-1	Mike/ Dave G.	Done			111	20 hrs. .5 wks.	

15.	Mack Rack Addr. Xlator P/N 600575-1	Sherri/ Joe	Done	1.4 x 2.0	---	109	5.5 hrs. .2 wks.	Protos Inhouse
16.	Strobe Board P/N 600572-1	Sherri/ Paul B.	Done	2.5 x 3.6	---	114	9 hrs. .2 wks.	Protos Inhouse
17.	Dual Channel Interface Printer Adapter P/N 600558-1	Mike/ Stone	Done	(2 layers) 627 holes	1.15 ic/in	105	84 hrs. 2.1 wks.	Protos Inhouse
18.	Master Controller P/N 600ML522-1	Mike/ Art	Done	6.5 x 11.5 2,300 holes 153 Comp.	.71 in/ic	101	+Telesis 190 hrs. Total 4.8 wks.	
19.	Dual Channel Interface P/N 600ML554-1	Jonathan/ Stone	Done	12.5 x 13.5 (8 layers) 4,585 holes	2.06 ic/in	122	862 hrs Total 21.5 wks.	Protos ordered 1/31/86 Ship Date 2/28/86
20.**	Terminal Controller I/F P/N 600ML577-1	Sherri/ Joe	Done	12.5 x 13.5 (10 layers) 5,115 holes	1.84 ic/in	108	342 hrs. Total 8.5 wks.	Ordered 1/23/86 Ship Date 2/21/86
21.	Control Panel Adaptor PCB P/N 600549-1 530/540 Appearance Upgrade	Mike/ Paul	Done	6.5 x 11.5 2 layers 1,218 holes	1.09 ic/in	112	170 hrs. Total 4.3 wks.	Protos Inhouse
22.	B-Bus Convertor (Classifier Project) P/N 600ML576-1	Sherri/ Brian T.	Done	6 layers LRL Size 150 Comp. Digital	.61 in/ic	113	244 hrs. Total 6.1 wks.	Ordered 2/21/86 Ship Date 3/7/86
23.	B-Bus Driver (Classifier Project) P/N 600ML578	Mike/ Sherri/ Brian T.	Done	5 X 8 20 IC's Digital (8 layers)	2.1 ic/in .41 in/ic	125	119 hrs. Total 2.975 hrs.	Protos Ordered 1/31/86 Ship Date 2/28/86
24.	Terminal Controller Remote Panel P/N 600ML568-1	Mike/ Joe	Done	4 layers		107	120 hrs. Total 3 wks.	Ordered 1/23/86 Ship Date 2/21/86
25.	Euro Cards Mother Board P/N 600ML580	Fred B./ Wayne	Done	W/L Entry		124	Telesis/ Sherri	Ordered 2/13/86
26.	Double Document Detector Update 700186-1 P/N 600071	Art/ Wayne	Done	5 X 5 15 IC's		128	94 hrs. Total 2.4 wks.	Ordered 2/18/86 Ship Date 3/17/86
27.	PC 9 Overlength Detector New	Art/ Wayne	Done	5 X 5 12 IC's		127	72 hrs. Total	Ordered 2/18/86 Ship Date 3/17/86

ASSIGNING AVERAGE TIME

ESTIMATES TO THE TASKS

PRODUCTIVITY GAINS TO

DETERMINE POTENTIAL TIME

TRANSLATE THESE PRODUCTIVITY

GAINS INTO DOLLARS

AND TIME SAVINGS

PROPOSAL

TO: J. Belhumeur

FROM: L. Stegall

DATE: February 24, 1986

SUBJECT: Purchase of an EDA-620 Routing Station

In March of 1985, the second CAD/CAM System was purchased at a cost of $69,900.00. With the acquisition of the second system and a second shift that began in September 1985, a total of 15 printed circuit boards were designed and our outside design service costs were reduced to $28,943.00 compared to 1984 design services of $83,976.00 a savings of $55,033.00. Since then, I've been greatly concerned about our system efficiencies, and have taken a look at where things might be improved. One general conclusion is we can improve our throughput and cost efficiencies by moving our routing of printed circuit boards "off" to a co-router which would free our present systems to be used as a design station only.

The third system would consist of an EDA-620 Routing Station. The EDA-620 is used as an add-on to Telesis workstations which shortens project design time, improves individual user productivity and speeds overall system throughput. Based on Sun Microsystems' Sun-3 hardware architecture the accelerator speeds routing tasks by four to eleven times over execution rates on the Telesis EDA-300 workstation.

Attached is a brief synopsis justifying these purchases. This is broken down as follows:

Exhibit A - Quotation and EDA-620 Routing Brochure

Exhibit B - PC Board accomplishments for 1985

Exhibit C - Router comparison on the EDA-620 vs. EDA-300.
(Note: Scan-Optics was invited to Telesis and use the EDA-620 to get this comparison.)

Exhibit D - Capital Equipment Request and Purchase Requisition

Exhibit E - 1986 known new PC designs

Conclusion: Twelve (12) new printed circuit boards must be designed and fabricated by April 31, 1986 for the High Speed Image Capture Module and to meet this schedule the EDA-620 Routing Station is needed.

LS/tld

cc: J. Aucoin
G. Dwyer
R. Gocht
D. Newton

SCAN-OPTICS, INC.

CAPITAL EQUIPMENT REQUEST

CER #

MER #

Originator: Linda Stegall Dept. # 622 User Dept. # 622

Capital Asset Account #: 622-1790

Capital Equipment Description - To Be Completed By The Originator

Purpose of the Expenditure: To purchase an EDA-620 Router to improve our throughput and cost efficiencies by moving our routing of printed circuit boards "off" to a co-router.

Recommended Vendors: List 3 and Price Quotations:

1. Telesis - $59,052.00

2.

3.

Vendor Selected: Telesis

Purchase Requisition # Purchase Order #

Comments, Additions, or Deletions as Approved: (Initial)

Financial Evaluation - To Be Completed By Accounting

Expected Life Years. Internal Rate of Return % Over Years

Payback Period Years Assuming % Interest Rate

Recommendation

Approvals - In The Below Order

L Stegall 2/24/86

(1) Department Manager Date

2/24/86

(2) Functional Vice President/Director Date

(3) Accounting Date

(4) Capital Project Committee Date

HOW TO BUILD A BRIDGE FROM OPPOSITE SIDES OF A RIVER

OR

EFFECTIVE COMMUNICATIONS FOR SYSTEM IMPLEMENTATION

ARN D. BUCK

TRANSCOM ELECTRONICS, INC.
PORTSMOUTH, RHODE ISLAND 02871

ABSTRACT

Something very predictable would occur if you placed two master bridge builders on opposite sides of a river. Then after doing such, asked each of them to construct one half of a bridge with no communication allowed between them... You would end up with two half spans that do not meet. At best, the bridge would be worth no more than a tourist attraction or a candidate for a "Super Blooper" television show. Likewise, the inability of a system manager to communicate adequately can cause hundreds of hours to be lost during the implementation of a CAD system. It could also be a major contributing factor to the failure of a system.

The intent of this paper is to review the mechanics of communications, provide guidelines for establishing and administering effective communications for CAD systems and to stress the importance of doing such.

INTRODUCTION

Communication will be defined as the process used to achieve understanding between people. At least two individuals are necessary for communication to take place. One of these is the sender. The sender is the person who desires to transmit an idea or information to someone else. The target of the sender is the receiver. The receiver must interpret the message. The accuracy of the interpretation will be affected by the background, experiences, values, beliefs and prejudices of both the sender and receiver. Some other potential causes of misinterpretation include: vocabulary, preoccupation, semantics, communication overload, lethargy and filtering. Extra effort is occasionally required to overcome such barriers.

The author experienced a situation in which ten hours of wasted effort could have been saved by a simple question. In this case, an engineer had explained a special component placement requirement to a second shift designer during shift change. Before leaving for home, the author asked the designer if he understood the engineer's request. The designer said that he did. Upon returning the next day, the author discovered that the special requirement had not been incorporated into the board layout. The incident could have been prevented, if the designer had been asked to explain his interpretation of the engineer's request. It is not being suggested that such questions should be asked on a routine basis because the person being questioned would eventually deduce that he or she is not trusted. But such questions should be asked, if there is reason to believe that a receiver had misinterpreted a message.

FORMS OF COMMUNICATION

Two forms of communication that are frequently utilized are verbal and written. Knowing how and when to use each can increase the effectivity of a manager.

Verbal communication is used during conversations and meetings. It is most effective for short messages or for transmitting information that would not have a likely requirement for future reference.

Written communication should be used if the information to be sent is complex or should be made available for future reference. Typical applications are: policies, procedures, standards, specifications, guidelines, memorandums and notes.

Written communication can be used to minimize the affects of communication overload. This is a state in which a receiver is given more information than they can absorb in a given time period. Once this saturation point is reached, communication breakdown occurs. A common symptom of communication overload is when trustworthy workers claim that they were not told information that you distinctly remember telling them. It is not unusual for this condition to occur during system implementation.

LISTENING

Historically, it has been rightfully stressed that a manager must be a good sender. However, increased emphasis is being placed on the importance of a manager being a good receiver. Facts acquired through attentive listening can assist in making better decisions.

There are basically three types of listening: marginal, evaluative and projective. Marginal listing occurs when the receiver is preoccupied and giving little attention to the message being sent. As a result, little, if any, information is absorbed by the receiver. The small amount of information received may be acutely distorted due to its incompleteness.

WARNING: THE AUTHOR HAS DETERMINED THAT
FREQUENT USAGE OF MARGINAL
LISTENING MAY BE HAZARDOUS TO
THE HEALTH OF YOUR SYSTEM

Studies have shown that people can listen approximately four times faster than the average speaking speed. If this extra time is spent approving or disproving what is said, it is classified as evaluative listening. Even though this type of listening requires the receiver's full attention, it is not considered very effective. This is because the receiver is allocating too much time evaluating the sender and planning responses. In other words, evaluative listening is listening with a prejudice.

Active listening, like evaluative listening, requires the receiver's full attention. However, unlike the previous type, it is empathic. The receiver makes a concentrated effort to place them self in the "other person's shoes." Although this takes longer than other types of listening, the time spent is considerably more productive.

The previous section described some of the skills that can contribute significantly to the achievement of accurate communications in a general sense. The upcoming section will discuss specific communication channels that should be established for effective system implementation.

ROLE OF THE SYSTEM MANAGER

A CAD system manager should assume a centralized role (see figure 1) when dealing with information that is associated with the system. Because of his or her direct authority over the system, they are in a good position to coordinate communications. By doing this they can ensure that the system will function as expected.

This does not mean that the system manager should always be a middle man between two groups. There are times that it is advantageous for two groups to discuss topics relative to their specialties. The system manager should become involved in these specialized discussions if specific problems, questions, or suggestions relating to the system are brought up.

There are times that a system manager can make more effective use of his time by assigning an investigation to an individual or committee. The manger can then use the findings to evaluate the impact they might have on the system.

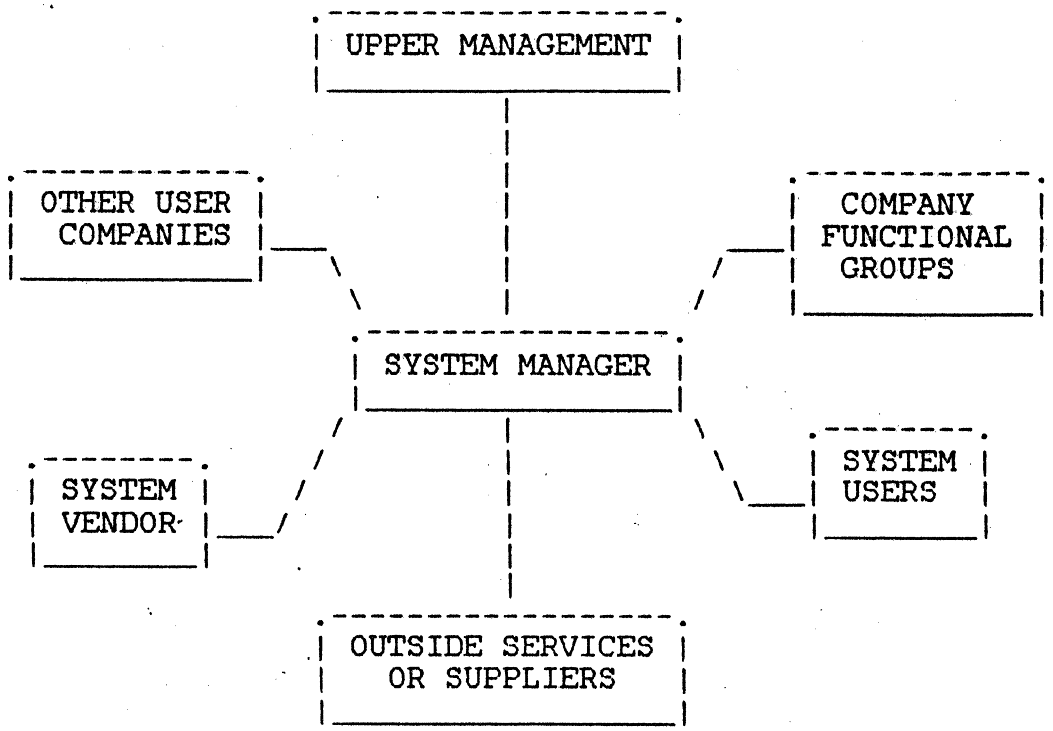


Figure 1.

COMMUNICATIONS WITH UPPER MANAGEMENT

System implementation requires direction. Like any other project, this direction can be attained by establishing goals and objectives. The goals for a capital investment such as a CAD system are often defined, or at least approved by upper management. It is extremely important that a system manager makes himself aware of and clearly understands these. It is his or her responsibility to transmit this information to any of the various groups that may have an impact on accomplishing these goals.

Occasionally the original goals are not feasible. They may have been derived from inaccurate or incomplete information. Sometimes all the tasks that are associated with completing a design project are not accounted for during a benchmark. Sometimes a sales representative may provide misleading information (hopefully unintentionally). If this type of error is detected, it is advantageous for the system manager to tactfully explain the problem to upper management. Hopefully, they will allow the goals to be adjusted to be more realistic. Before taking such a step, it is prudent for a system manager to make certain that his observations are supported by facts.

It is not unusual to encounter unforeseeable obstacles during system implementation. Such obstacles may have a negative impact on the achievement of the system goals. Other factors may cause the implementation to be ahead of established target dates. Upper management should be made aware of such changes and be in a position to suggest, approve, or disapprove the corrective action to be taken. This can be accomplished by making periodic status reports to them. The system manager should come to an agreement with upper management about how often these reports should be given. They may be given at specific time intervals (weekly, bi-monthly, monthly) or at the accomplishment of specific milestones. These status reports should be made in a written format so that past performance can be reviewed to identify trends which could be helpful in making adjustments to the system expectations.

Figure 2 shows typical communications between a system manager and upper manager.

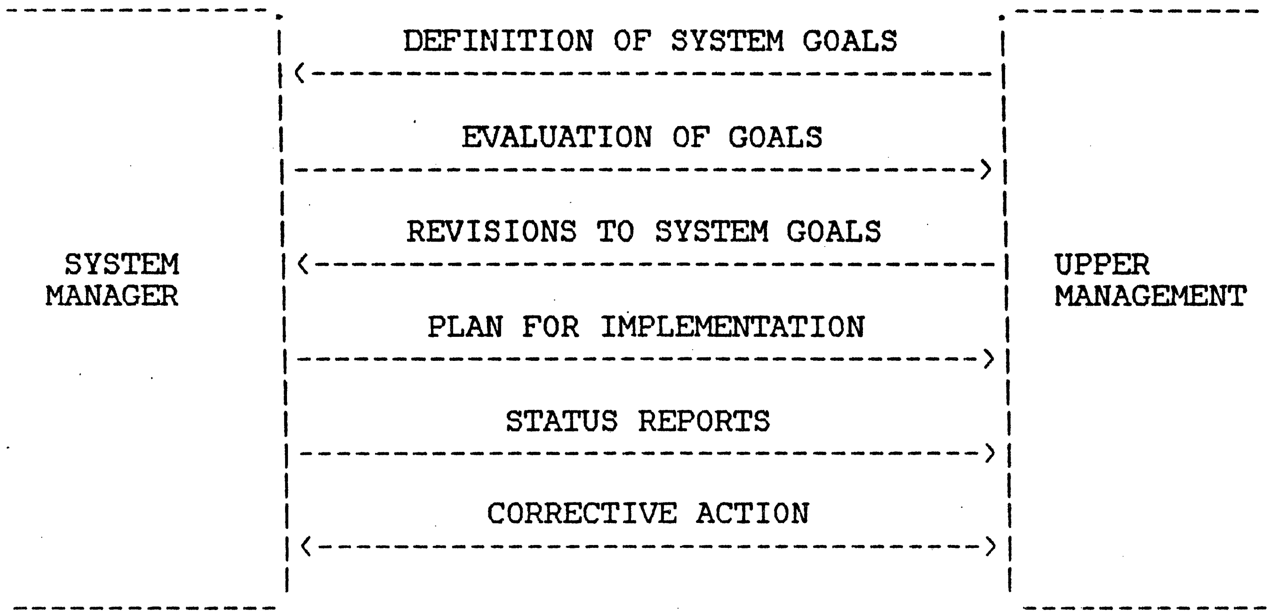


Figure 2.

COMMUNICATIONS TO COMPANY FUNCTIONAL GROUPS

Various functional groups within the user company will either be inputing information to the system or using the outputs from the system. The system manager can save unnecessary rework by identifying and establishing communications with these groups (see figure 3).

Manufacturing, for example, might require pin numbers to be shown on certain components to aid assemblers in inserting the parts. If this feature was not discovered until after the library was created, there would be a considerable amount of rework required to add this feature.

The system manager should inform these groups of the system's capabilities and short comings. They, in turn, should make certain that the system manager understands their requirements. This would enable the groups to work together to develop objectives that would not only ensure an effective system implementation, but could also improve the efficiency of the whole operation.

```
--------------.                                      .-------------
              |                                      |
              |    REQUIREMENTS FOR THEIR FUNCTION   |
   SYSTEM     |<-------------------------------------|  COMPANY
   MANAGER    |                                      |  FUNCTION
              |       CAPABILITIES OF THE SYSTEM     |  GROUPS
              |------------------------------------->|
              |                                      |
--------------                                        -------------
```

Figure 3.

COMMUNICATION WITH SYSTEM USERS

The highest communication demands on a system manager come from the system users. This is because the users require a lot of information in order to take full advantage of the system features. To help meet this need, the supplier of the system typically provides user training and detailed operational manuals. This information, however, is only a portion of what is required. Although CAD systems as supplied by a vendor have the functions to do the job built in, they must be adapted to meet the specific requirements of the user's application. The steps taken to adapt the system must be documented and made available to all users. To clarify their purpose and to facilitate their retrieval, these documents can be distributed among four categories: procedures, standards, guidelines, and reference.

Procedures define a step by step approach for the accomplishment of routine tasks. As a result procedures allow system users to concentrate their technical skills on more demanding non-routine tasks. Thus making more efficient use of their abilities. An additional benefit of doing such is that it usually increases job satisfaction.

Examples of procedures associated with CAD systems are shown below:

a. Archiving procedures
b. Procedures for symbol creation and control
c. System maintenance procedures
d. Data preparation procedures
e. Post processing procedures

Standards provide a basis of comparison between established requirements and some action to be taken. The use of standards ensure a uniform quality to projects done on the system. Some typical standards include:

a. Graphic standards
b. Text standards
c. Company defined data base layer standards
d. Symbol naming conventions
e. File naming conventions

The purpose of guidelines is to present examples. These examples can be used to assist users in making decisions while performing complex tasks. A chart that shows historically successful routing parameter set ups would be considered a guideline.

The most general category of the four is reference documentation. This classification provides a source of information that users can utilize to refresh or expand their knowledge of the system. Examples of reference documentation include:

a. Drawings of library symbols
b. Copies of standard text files
c. An index of standard execute files
d. An index of standard libraries
e. Tables of geometric constructions
f. System training documents
g. Instruction manuals for the system and its peripheral devices.
h. System application notes
i. Papers from user group meetings.
j. Software release notices
k. Books and periodicals

A considerable effort must be expended to generate this amount of documentation. It is very likely that only the most important documents will be created during system implementation. However, it is an important project. Its' completion should be planned and scheduled. A manager who does not make adequate system documentation available will be constantly badgered by questions from users. This may make him or her feel more valuable to the company. However, it is a tremendous waste of time for all parties involved.

In the course of performing daily tasks, users discover better ways of operating the system. A manager should develop methods of transmitting this information to all users. There are various ways to accomplish this. One method is to make a bulletin board available for users to post their ideas. The short coming of this approach is that the space on a bulletin board is limited. Older but still pertinent suggestions would have to be removed to make room for new ones. This problem could be overcome by providing users with a standard form on which to write their suggestions. This would allow the sheets to be placed into a binder after they have been removed from the bulletin board.

Another way that managers can encourage users to share operating tips is by holding internal user group meetings. These meetings will provide a forum for users to present their ideas to each other. Such meetings can also be used by the manager to introduce and train users in the use of new system features and operating procedures. The loss of productive system time during these meetings should be offset by increased productivity and operator satisfaction.

The following diagram summarizes the communications between a system manager and system users.

```
-------------.                                       .------------
             |                                       |
             |      SYSTEM OPERATING PROCEDURES      |
             |-------------------------------------->|
             |                                       |
             |            SYSTEM STANDARDS           |
             |-------------------------------------->|
             |                                       |
             |         APPLICATION GUIDELINES        |
             |-------------------------------------->|
   SYSTEM    |                                       |   SYSTEM
   MANAGER   |          NEW SYSTEM FEATURES          |   USERS
             |-------------------------------------->|
             |                                       |
             |       SOFTWARE/HARDWARE PROBLEMS      |
             |<------------------------------------->|
             |                                       |
             |        ENHANCEMENT SUGGESTIONS        |
             |<--------------------------------------|
             |                                       |
             |      OPERATING TIPS AND SHORT CUTS    |
             |<--------------------------------------|
             |                                       |
-------------                                         ------------
```

Figure 4.

COMMUNICATIONS WITH SYSTEM VENDOR

Communication with the supplier of a system is essential for successful system implementation and continued operation. The success of a company that markets systems is usually dependent on the success of their customers. As a result it is beneficial to the system supplier to make as much information available to a user site as possible. This information includes the following:

a. Training classes
b. Operational manuals
c. Software release notices
d. System performance reports
e. Application notes
f. Facility planning assistance

The system manager should ensure that any documentation received is made available for users. Updates to documentation should be posted as soon as possible. If the updates are substantial, the system manager should arrange training for users.

From time to time users may experience problems when operating the system. Such problems are usually resolved by telephone calls to customer support. The system manager should coordinate these calls to save time for both his people and customer support. This can be accomplished by the manager requiring that all problems be discussed with him before calling the support center. By doing this some of the redundant problems and operator errors can be weeded out. He should also insist that all problems be written in a log book along with their solutions or work arounds.

COMMUNICATIONS WITH OUTSIDE VENDORS AND SUPPLIERS

When dealing with outside services and suppliers, the system manager must make certain that communications are accurate. For this reason all significant information should be transmitted in writing and carefully checked.

The author has heard of several photoplotting tragedies that were caused by seemingly insignificant errors. In one case a designer anxiously opened a package containing films of the first design done on their system. To his surprise the only image on each film was a small black rectangle. It was enough to ruin his (and his boss's and his boss's) whole day. The cause was later found to be a decimal point being one place off. Unfortunately it was off in the line that describes the x-y coordinate format to the photoplotter. This resulted in film images that were ten times smaller than they were supposed to be. To compound the problem, the plotter controller followed instructions and used the same size apertures as if the images were the correct size. The end result... garbage. If the system manager had obtained a copy of the format required by the photoplotting service, prior to sending out the job, this story would be different.

When buying from suppliers it is important that they understand the requirements of the system. They will ship whatever is ordered. It is usually difficult for a system manager to explain why the one year supply of pens he just ordered do not fit the penplotter.

Figure 5 shows the typical communications between a photoplotting service and a system manager. The communications for other services would be of a similar format.

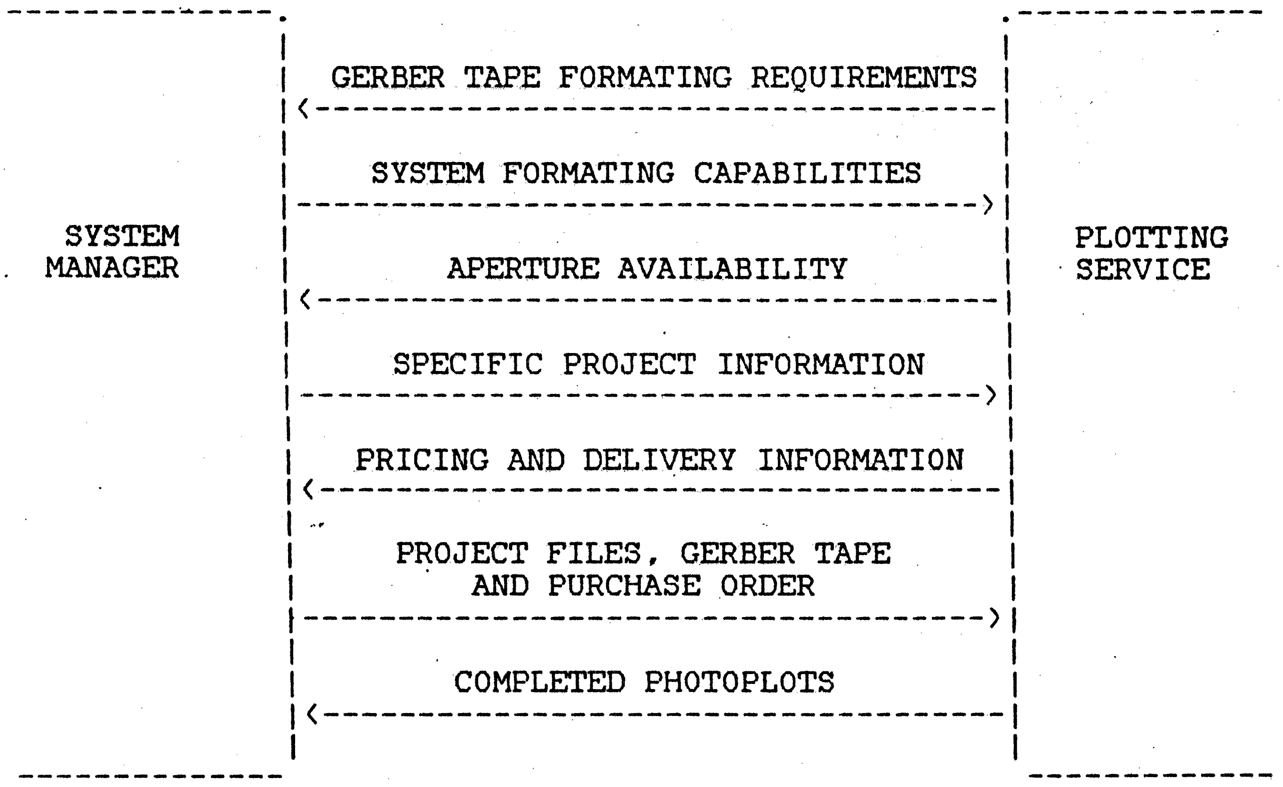

SYSTEM MANAGER
PLOTTING SERVICE
GERBER TAPE FORMATING REQUIREMENTS
SYSTEM FORMATING CAPABILITIES
APERTURE AVAILABILITY
SPECIFIC PROJECT INFORMATION
PRICING AND DELIVERY INFORMATION
PROJECT FILES, GERBER TAPE AND PURCHASE ORDER
COMPLETED PHOTOPLOTS

Figure 5.

CONCLUSION

The establishment of the proper communication channels can be an important factor in the success or failure of the implementation of a system. It is the responsibility of the system manager to work with all those involved with the system to ensure that such channels are established and utilized.

SPACE REQUIREMENTS FOR HARD DISK, FLOPPY DISK, AND MAGNETIC TAPE MEDIA STORAGE

Bob Rough, Jr.
Owens-Illinois

DETERMINING FILE SPACE REQUIREMENTS
FOR ON AND OFF-LINE STORAGE

Effective utilization of disk space should be one of the primary areas to be considered during the development phase of any data-base management scheme. Regardless of the number of systems you have, it is important at the initiation of any new project, to be able to predetermine the types of files which will be used within a project, and the types of storage media which will most efficiently effect the transfer and storage of those particular project files.

Project allocation of disk space can and will improve your overall system performance.

1. Types of files that belong within a project:
   - Symbol files for unique* devices or components
   - Device files for unique* symbols
   - Pin files for unique* components
   - Processing/Control files
     - Layer std.
     - Photoplot - CON
     - NC Drill - FIG
     - Aperture - TAB
   - Schematic Drawings
   - Net-Data-Base
   - Reports, Logs
   - PCB Drawing

2. File Transfer from hard disk to various other storage media.

8" Double-Sided Dual Density Floppy

- Although floppies are limited in size, they can be an effective means of storage for smaller projects during and after the design phase provided that the data has been subdivided into content groupings (e.g., text files, schematic drawings, PCB drawing).

Listed below is a representation of the types of files and their respective sizes in disk blocks that might constitute a "typical" project.

FLOPPY SIZE = 2000 BLOCKS

		APPROXIMATE
FLOPPY NO.	FILES TO BE STORED	TOTAL SIZE OF FILES
1.	30 Text Files	30 Blocks
	2 D-Size Schematics	1800 Blocks
	Reports, Logs, Net-Data-Base	120 Blocks
	Total	1950 Blocks
2.	PCB Drawing File	1500 Blocks
	(50 EIC's - 40 sq. in.)	
	Net-Data-Base	120 Blocks
	30 Text Files	30 Blocks
	8 Symbols	200 Blocks
	Total	1850 Blocks
3.	Photoplot artwork files (8 plot files)	1700 Blocks
	Total Project Size	5500 Blocks

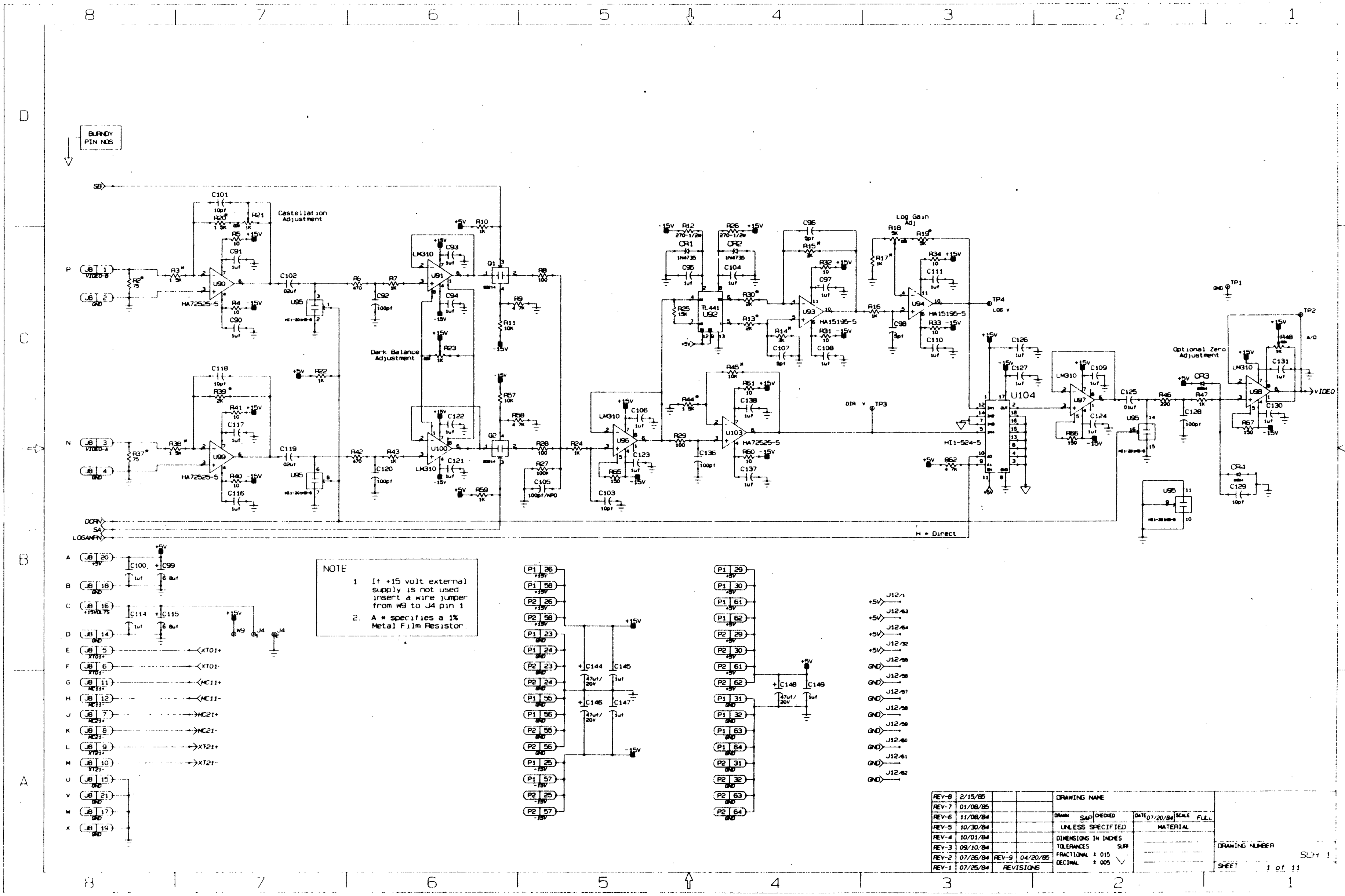
MAGNETIC TAPE 16ØØ BPI 24ØØ' REEL

- Magnetic Tape is, without question, the most common form of storage media in use today. Because of its larger storage capacity, 5ØK - 6ØK Blocks, several projects can be stored on one tape.

Even though there are different characteristics in the way data is written to floppies as opposed to magnetic tape, the overall capacity and ease of use makes magnetic tape the preferred media.

(1) 24ØØ' 16ØØ BPI Magnetic Tape = 25 floppies.

PLEASE NOTE that the numbers presented in the preceding section are approximations. The data in the tables and figures that follow are derived from actual schematic & PCB drawings and could be used as a basis for estimating your own drawing file size approximations.



NOTE

1 If +15 volt external supply is not used insert a wire jumper from W9 to J4 pin 1

2. A * specifies a 1% Metal Film Resistor.

REV-8	2/15/85		
REV-7	01/08/85		
REV-6	11/08/84		
REV-5	10/30/84		
REV-4	10/01/84		
REV-3	09/10/84		
REV-2	07/26/84	REV-9	04/20/85
REV-1	07/25/84	REVISIONS	

DRAWING NAME

DRAWN SAP CHECKED DATE 07/20/84 SCALE FULL

UNLESS SPECIFIED

MATERIAL

DIMENSIONS IN INCHES
TOLERANCES SURF
FRACTIONAL ± 015
DECIMAL ± 005

DRAWING NUMBER SCH 1

SHEET 1 of 11

FIGURE 1

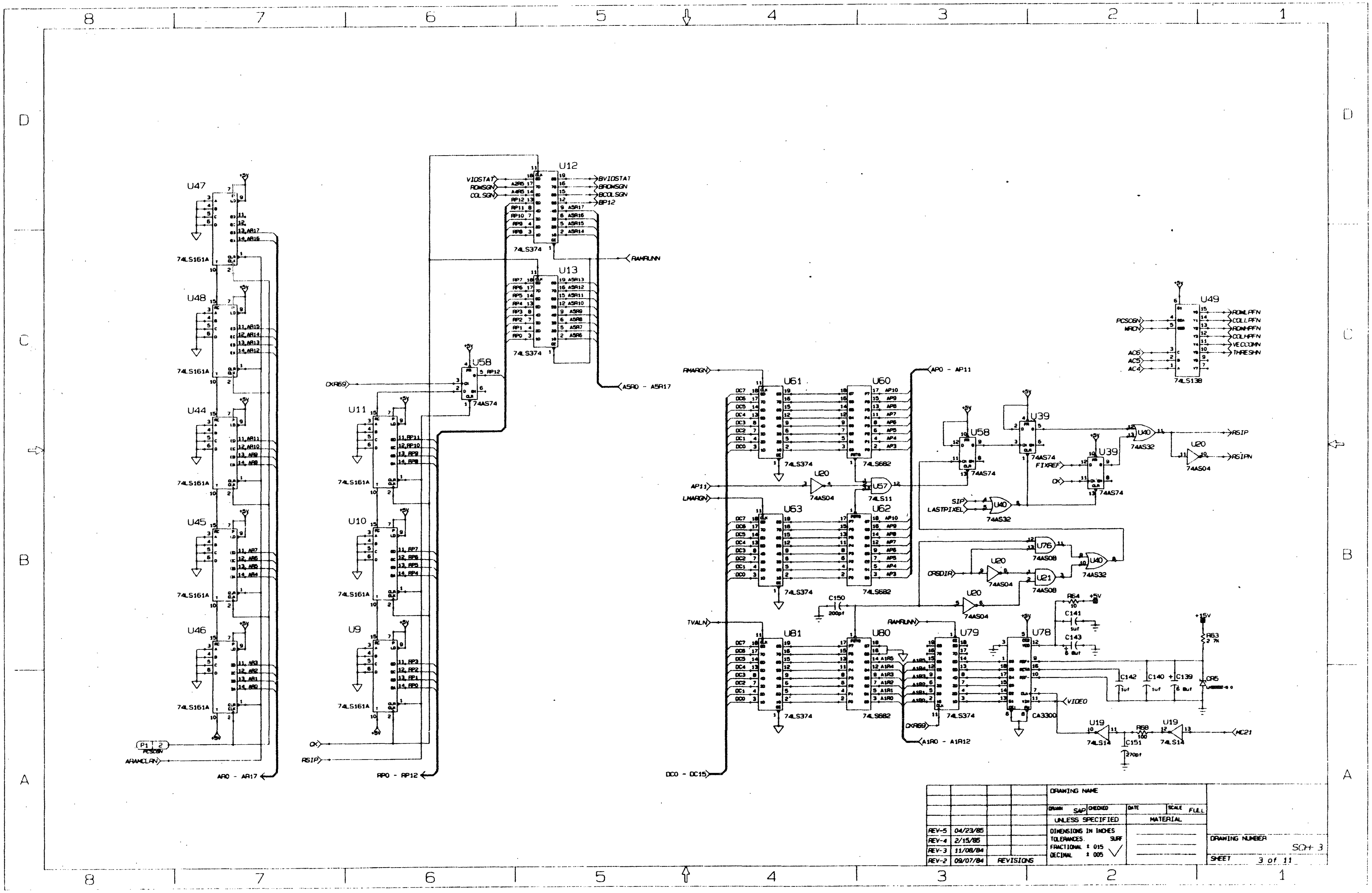

8
7
6
5
4
3
2
1
D
C
B
A
U47
74LS161A
AR17
AR16
U48
AR15
AR14
AR13
AR12
U44
AR11
AR10
AR9
AR8
U45
AR7
AR6
AR5
AR4
U46
AR3
AR2
AR1
AR0
P1 2
PCSGN
ARAMCLRN
AR0 - AR17
U12
VIDSTAT
ROWSGN
COLSGN
BVIDSTAT
BROWSGN
BCOLSGN
BP12
A5R17
A5R16
A5R15
A5R14
74LS374
RAMRUNN
U13
A5R13
A5R12
A5R11
A5R10
A5R9
A5R8
A5R7
A5R6
A5R0 - A5R17
U58
CKR69
RP12
74AS74
U11
RP11
RP10
RP9
RP8
U10
RP7
RP6
RP5
RP4
U9
RP3
RP2
RP1
RP0
CK
RSIP
RP0 - RP12
RMARGN
U61
U60
74LS682
AP0 - AP11
AP10
AP9
AP8
AP7
AP6
AP5
AP4
AP3
DC7
DC6
DC5
DC4
DC3
DC2
DC1
DC0
AP11
U20
74AS04
U57
74LS11
LMARGN
U63
U62
C150
200pf
TVALN
U81
U80
A1R5
A1R4
A1R3
A1R2
A1R1
A1R0
U79
U78
CA3300
VIDEO
A1R0 - A1R12
DC0 - DC15
U39
FIXREF
SIP
LASTPIXEL
U40
74AS32
RSIP
RSIPN
U76
74AS08
CRSDIR
U21
R64
10
+5V
C141
1uf
C143
6.8uf
+15V
R63
2.7K
C142
C140
C139
CR5
U19
74LS14
R68
100
C151
270pf
MC21
U49
74LS138
PCSGN
WRCN
AC6
AC5
AC4
ROWLPFN
COLLPFN
ROWHPFN
COLHPFN
VECCOMN
THRESHN
DRAWING NAME
DRAWN SAP
CHECKED
DATE
SCALE FULL
UNLESS SPECIFIED
MATERIAL
DIMENSIONS IN INCHES
TOLERANCES.
SURF
FRACTIONAL ± 015
DECIMAL ± 005
REV-5 04/23/85
REV-4 2/15/85
REV-3 11/08/84
REV-2 09/07/84
REVISIONS
DRAWING NUMBER
SCH- 3
SHEET 3 of 11

FIGURE 2

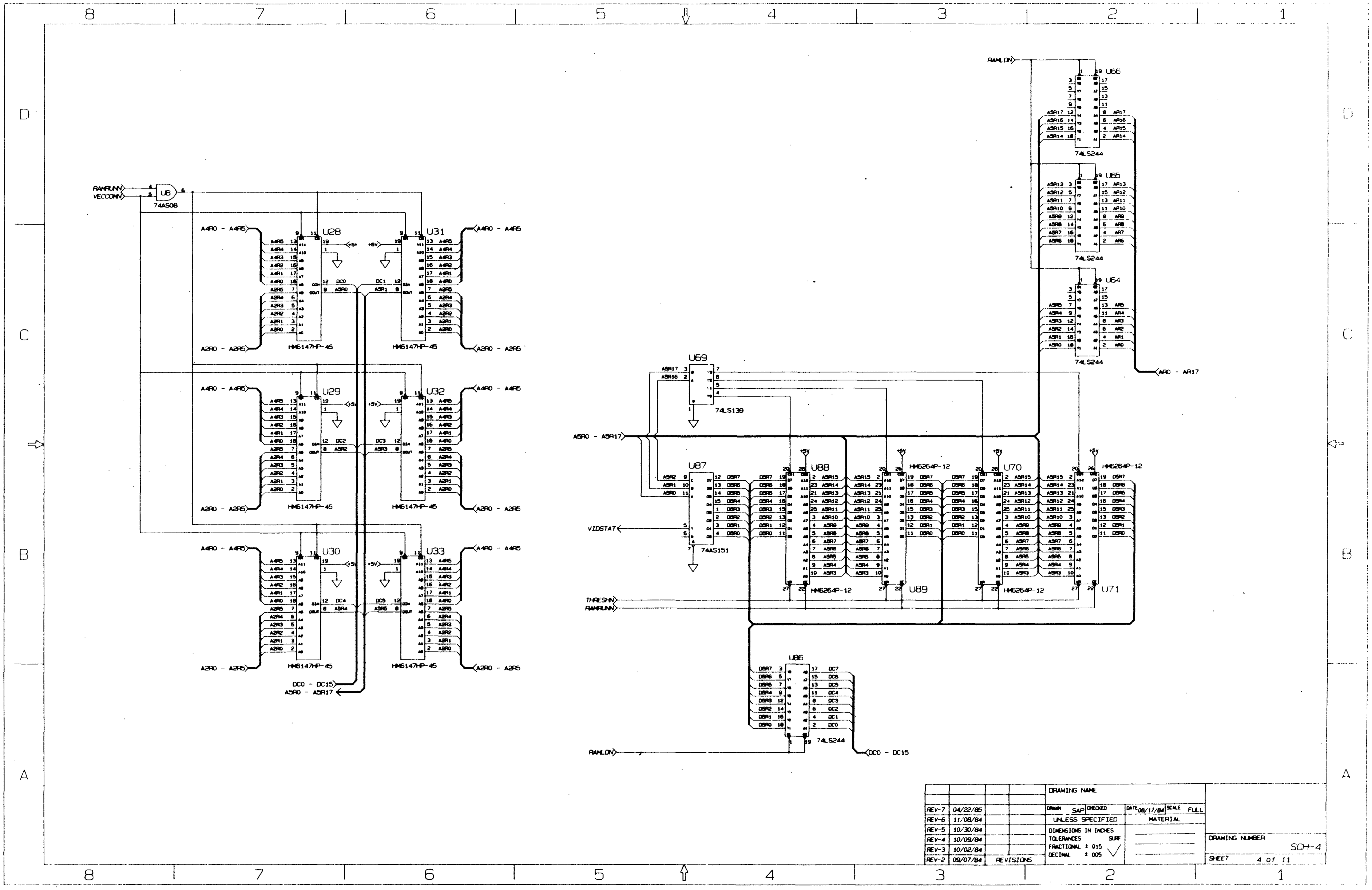

RAMRUNN
VECCOMN
U8
74AS08
A4R0 - A4R5
A2R0 - A2R5
U28
U29
U30
U31
U32
U33
HM6147HP-45
+5V
DC0 - DC15
A5R0 - A5R17
RAMLON
U66
U65
U64
74LS244
AR0 - AR17
U69
74LS139
A5R17
A5R16
U87
74AS151
VIDSTAT
U88
U89
U70
U71
HM6264P-12
THRESHN
U86
DSR0 - DSR7
DRAWING NAME
DRAWN SAP
CHECKED
DATE 08/17/84
SCALE FULL
UNLESS SPECIFIED
MATERIAL
DIMENSIONS IN INCHES
TOLERANCES
SURF
FRACTIONAL ± 015
DECIMAL ± 005
REV-7 04/22/85
REV-6 11/08/84
REV-5 10/30/84
REV-4 10/09/84
REV-3 10/02/84
REV-2 09/07/84
REVISIONS
DRAWING NUMBER
SCH-4
SHEET 4 of 11
8
7
6
5
4
3
2
1
D
C
B
A

FIGURE 3

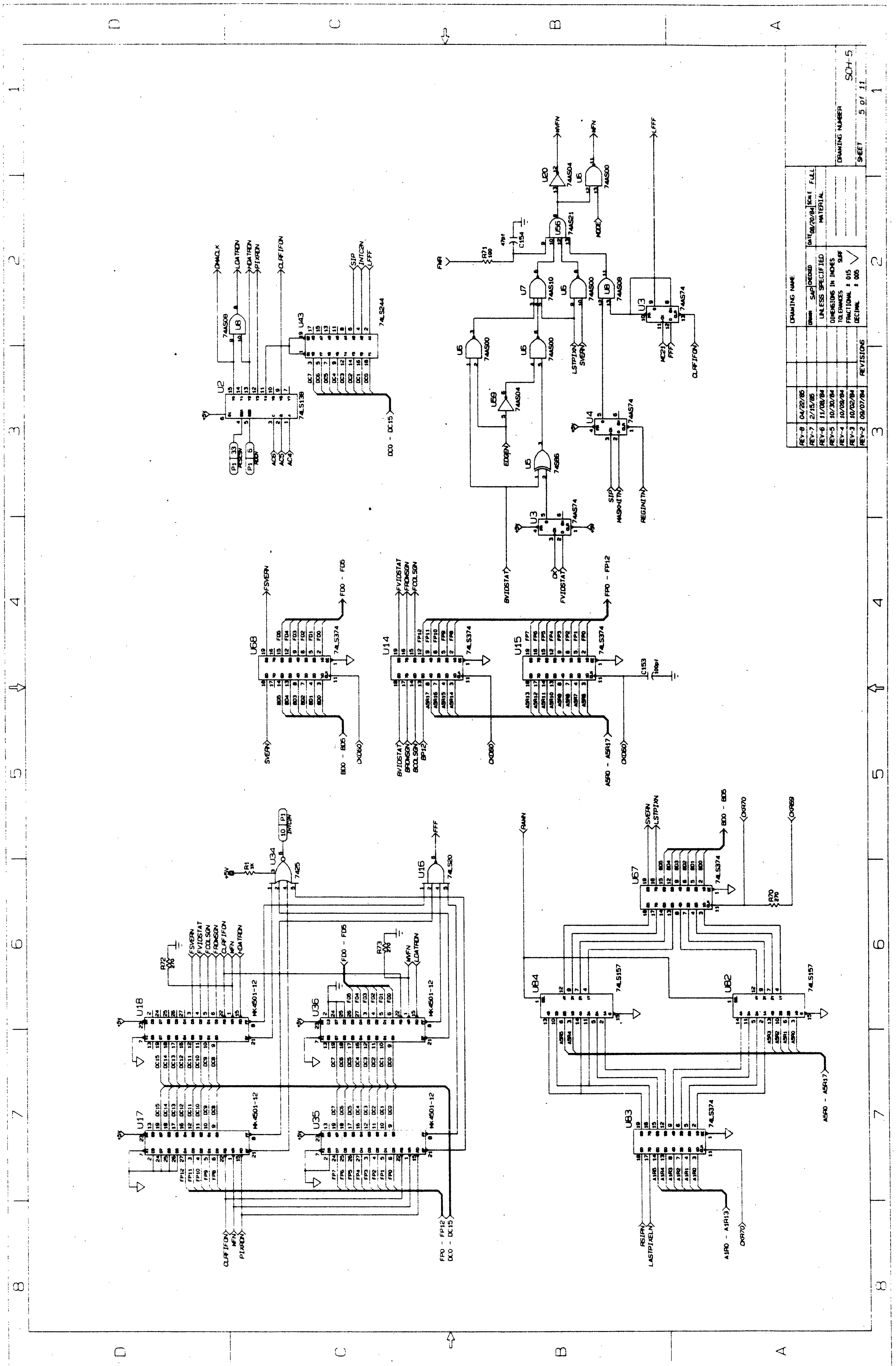

U2 74LS138
U8 74AS08
U43 74LS244
U5 74AS86
U59 74AS04
U6 74AS00
U7 74AS10
U56 74AS21
U20 74AS04
U3 74AS74
U4 74AS74
U68 74LS374
U14 74LS374
U15 74LS374
U34 7425
U16 74LS20
U18 MK4501-12
U17 MK4501-12
U36 MK4501-12
U35 MK4501-12
U67 74LS374
U84 74LS157
U82 74LS157
U83 74LS374
DC0 - DC15
FD0 - FD5
FP0 - FP12
BD0 - BD5
ASR0 - ASR17
A1R0 - A1R13
CLRFIFON
FFF
LFFF
SIP
INTC2N
DMACLK
LDATRDN
HDATRDN
PIXRDN
BVIDSTAT
FVIDSTAT
SVERN
FSVERN
REV-8 04/22/85
REV-7 2/15/85
REV-6 11/08/84
REV-5 10/30/84
REV-4 10/09/84
REV-3 10/02/84
REV-2 09/07/84
REVISIONS
DRAWING NAME
DRAWN SAP
CHECKED
DATE 08/20/84
SCALE FULL
UNLESS SPECIFIED
DIMENSIONS IN INCHES
TOLERANCES
FRACTIONAL ± .015
DECIMAL ± .005
MATERIAL
DRAWING NUMBER SCH-5
SHEET 5 of 11

FIGURE 4

POWER/GROUND CONNECTIONS							BYPASS-CAP			
REF-DES	DEVICE TYPE	+5V	GND				REF-DES	VALUE	PWR	GND
U26	74LS161A	16	8				C26	1uf	+5V/1	GND/2
U27	74LS161A	16	8				C27	1uf	+5V/1	GND/2
U28	HM6147HP-45	20	10				C28	1uf	+5V/1	GND/2
U29	HM6147HP-45	20	10				C29	1uf	+5V/1	GND/2
U30	HM6147HP-45	20	10				C30	1uf	+5V/1	GND/2
U31	HM6147HP-45	20	10				C31	1uf	+5V/1	GND/2
U32	HM6147HP-45	20	10				C32	1uf	+5V/1	GND/2
U33	HM6147HP-45	20	10				C33	1uf	+5V/1	GND/2
U34	7425	14	7				C34	1uf	+5V/1	GND/2
U35	MK4501-12	28	14				C35	1uf	+5V/1	GND/2
U36	MK4501-12	28	14				C36	1uf	+5V/1	GND/2
U37	74AS74	14	7				C37	1uf	+5V/1	GND/2
U38	74LS153	16	8				C38	1uf	+5V/1	GND/2
U39	74AS74	14	7				C39	1uf	+5V/1	GND/2
U40	74AS32	14	7				C40	1uf	+5V/1	GND/2
U41	74AS74	14	7				C41	1uf	+5V/1	GND/2
U42	74AS74	14	7				C42	1uf	+5V/1	GND/2
U43	74LS244	20	10				C43	1uf	+5V/1	GND/2
U44	74LS161A	16	8				C44	1uf	+5V/1	GND/2
U45	74LS161A	16	8				C45	1uf	+5V/1	GND/2
U46	74LS161A	16	8				C46	1uf	+5V/1	GND/2
U47	74LS161A	16	8				C47	1uf	+5V/1	GND/2
U48	74LS161A	16	8				C48	1uf	+5V/1	GND/2
U49	74LS153	16	8				C49	1uf	+5V/1	GND/2
U50	74LS153	16	8				C50	1uf	+5V/1	GND/2

COMMENTS....

DRAWING NAME			
DRAWN PGD	CHECKED	DATE 11/13/84	SCALE FULL
		REVISIONS	

DRAWING NUMBER SCH-8

SHEET 8 of 11

FIGURE 5

COMPONENT-ASSY

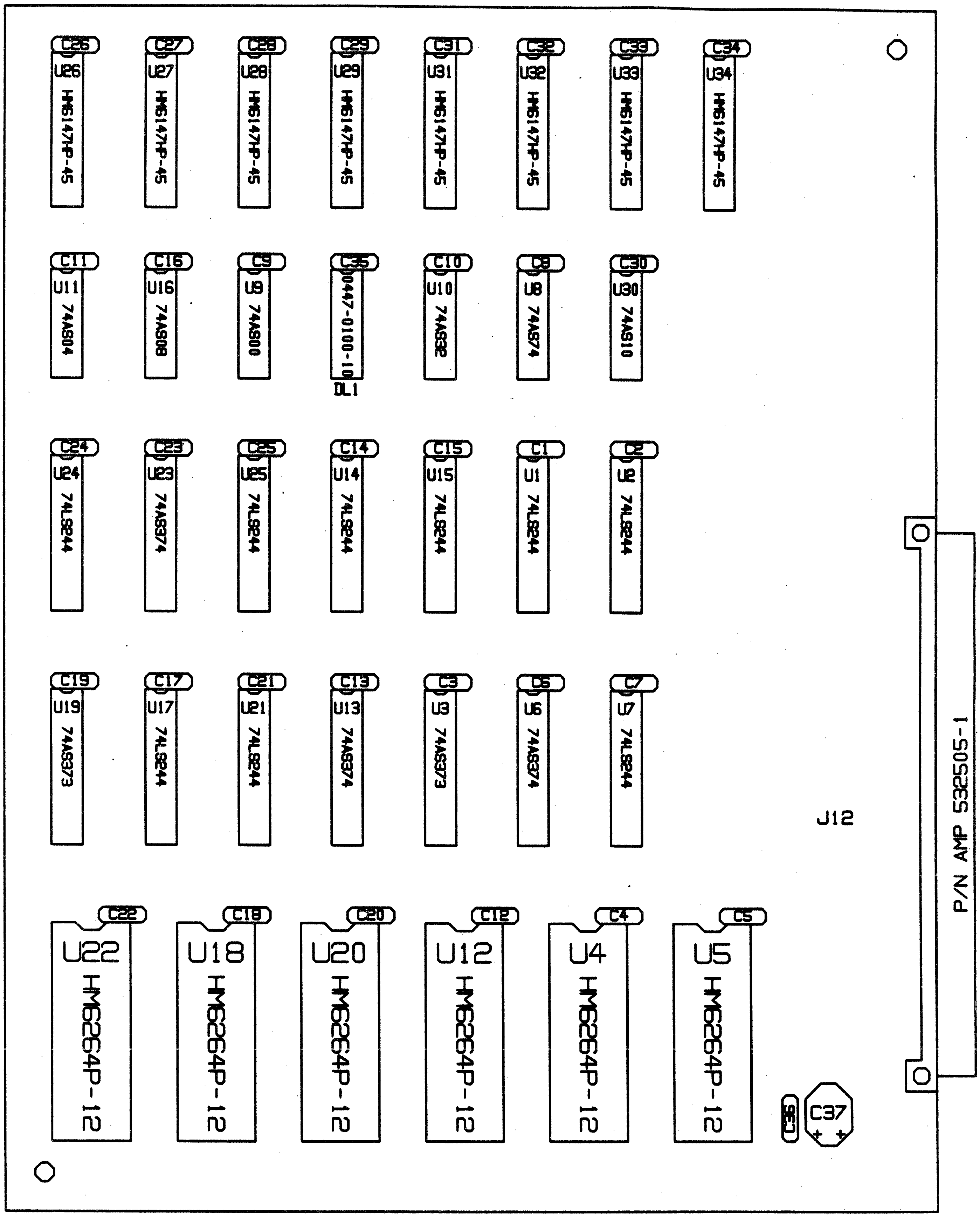


FIGURE 6

SOLDER-SIDE ART

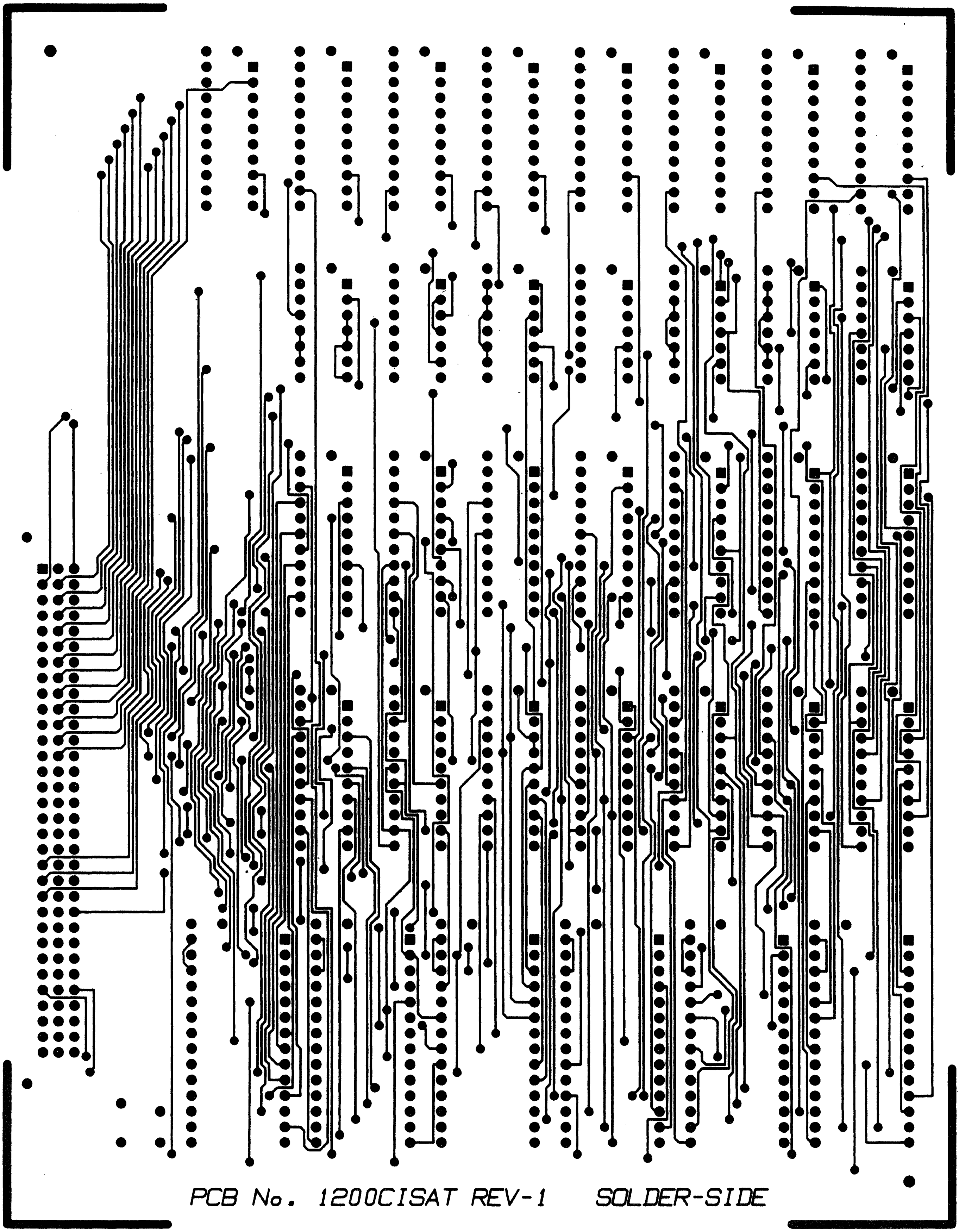


FIGURE 7

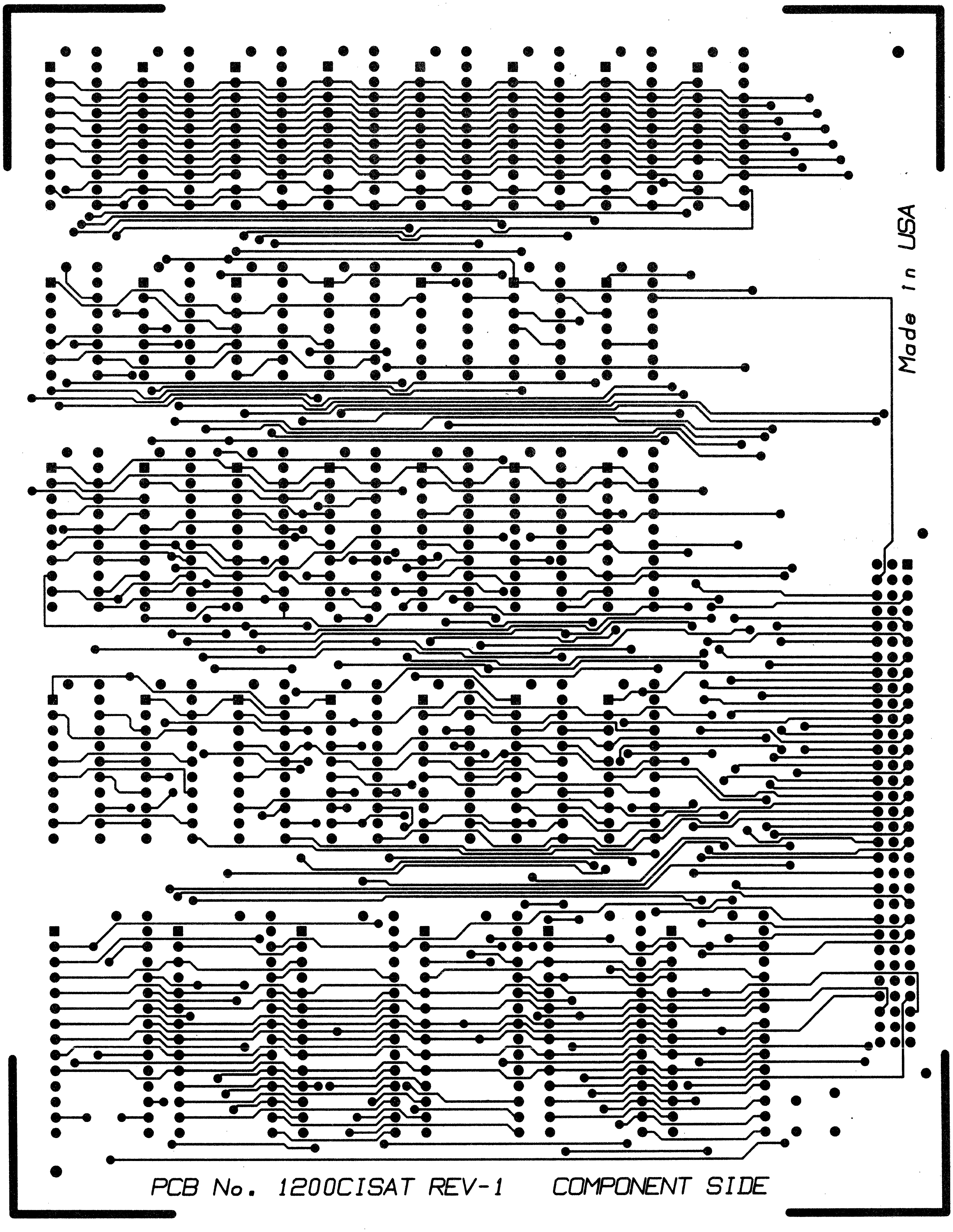


FIGURE 8

PWR-PLANE ART

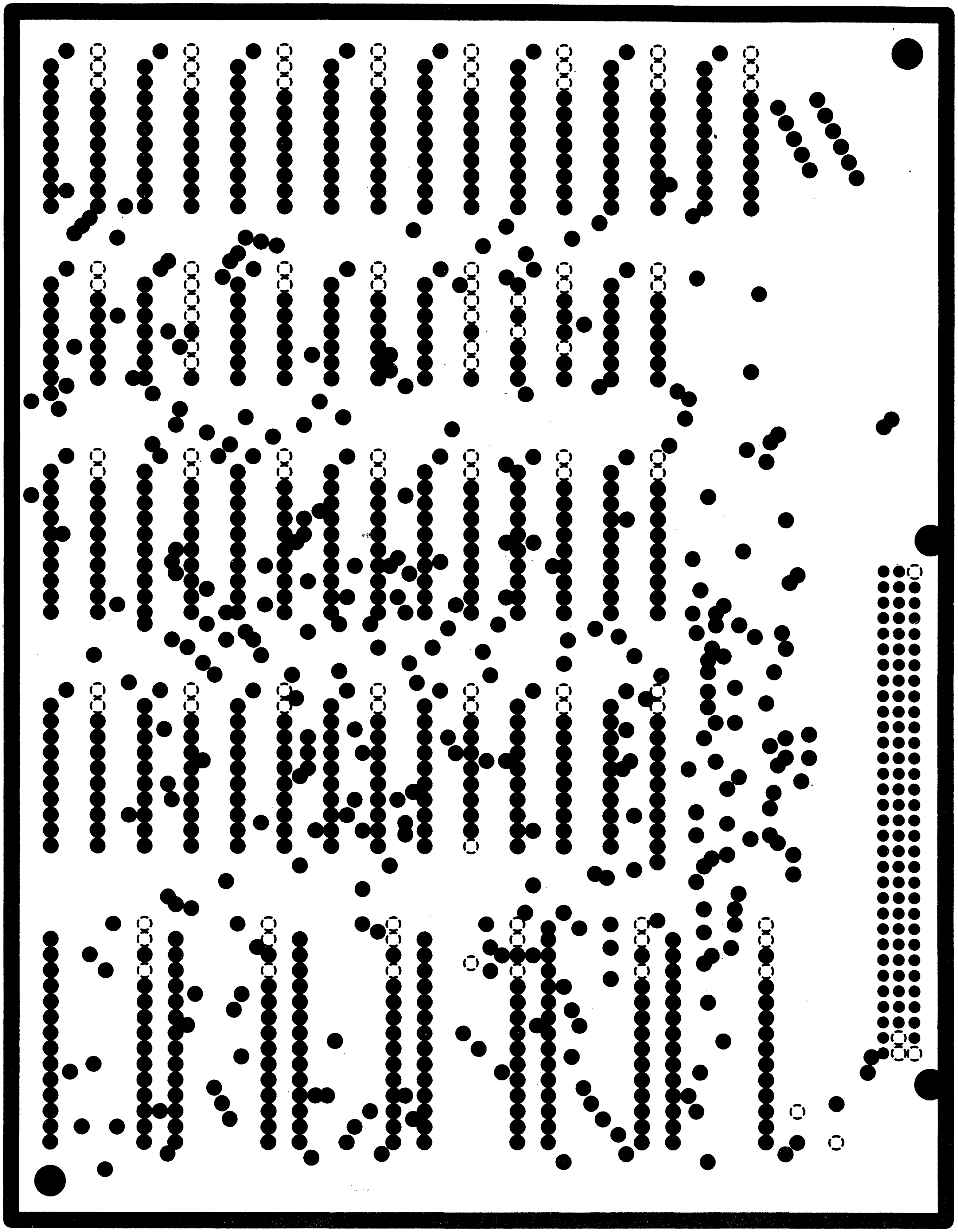


FIGURE 9

GND-PLANE ART

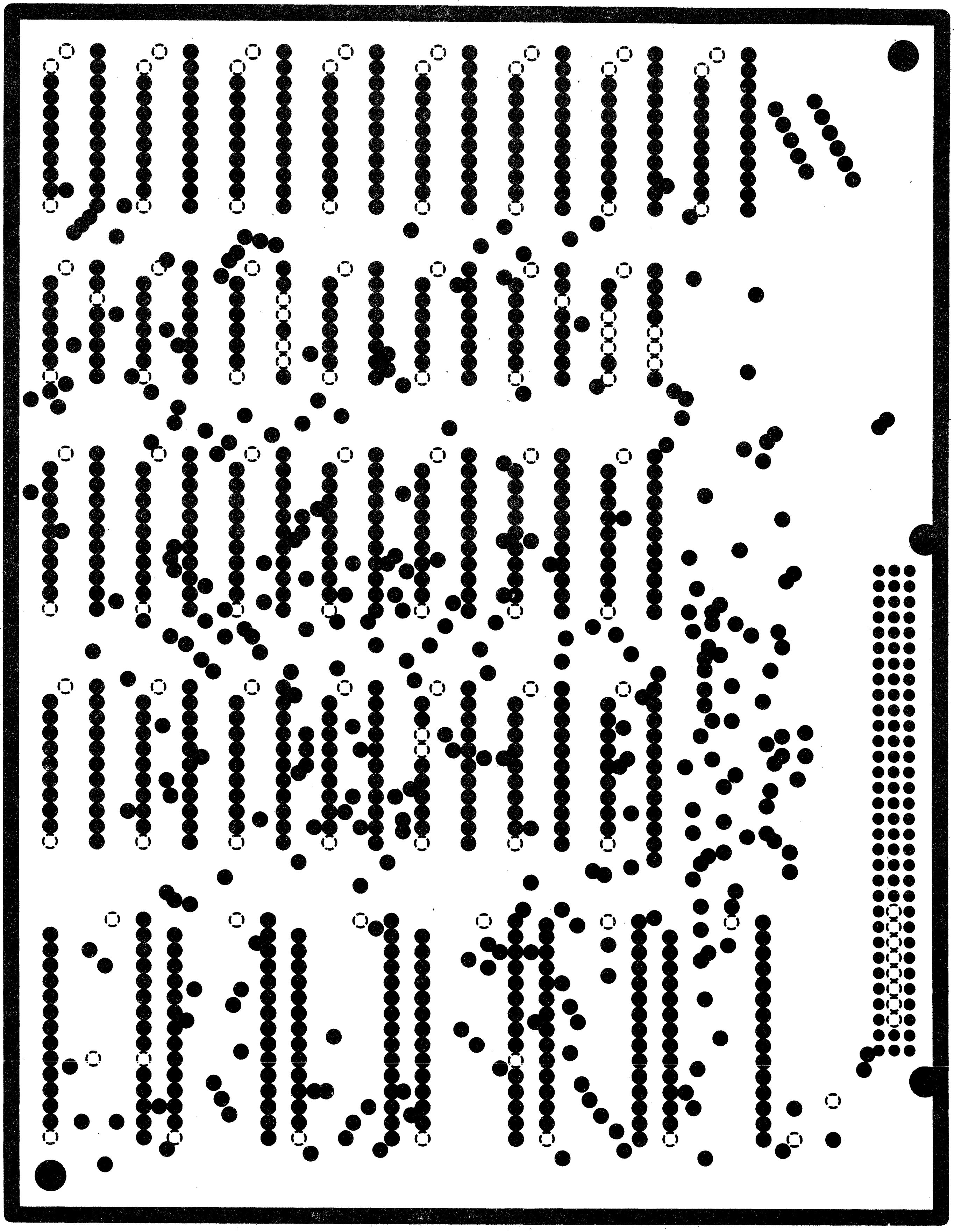


FIGURE 10

SOLDER-MASK/SS ART

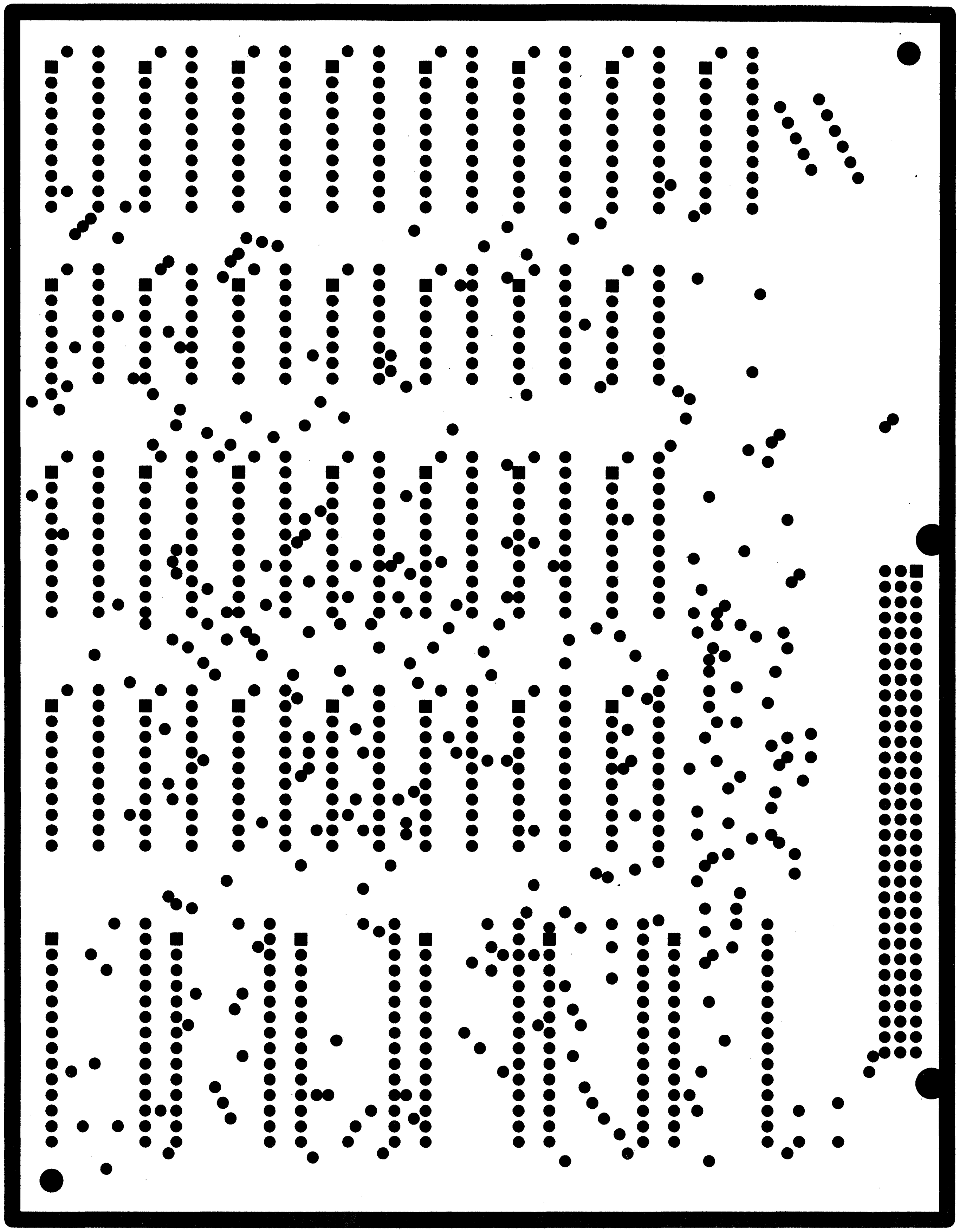


FIGURE 11

SILKSCREEN ART

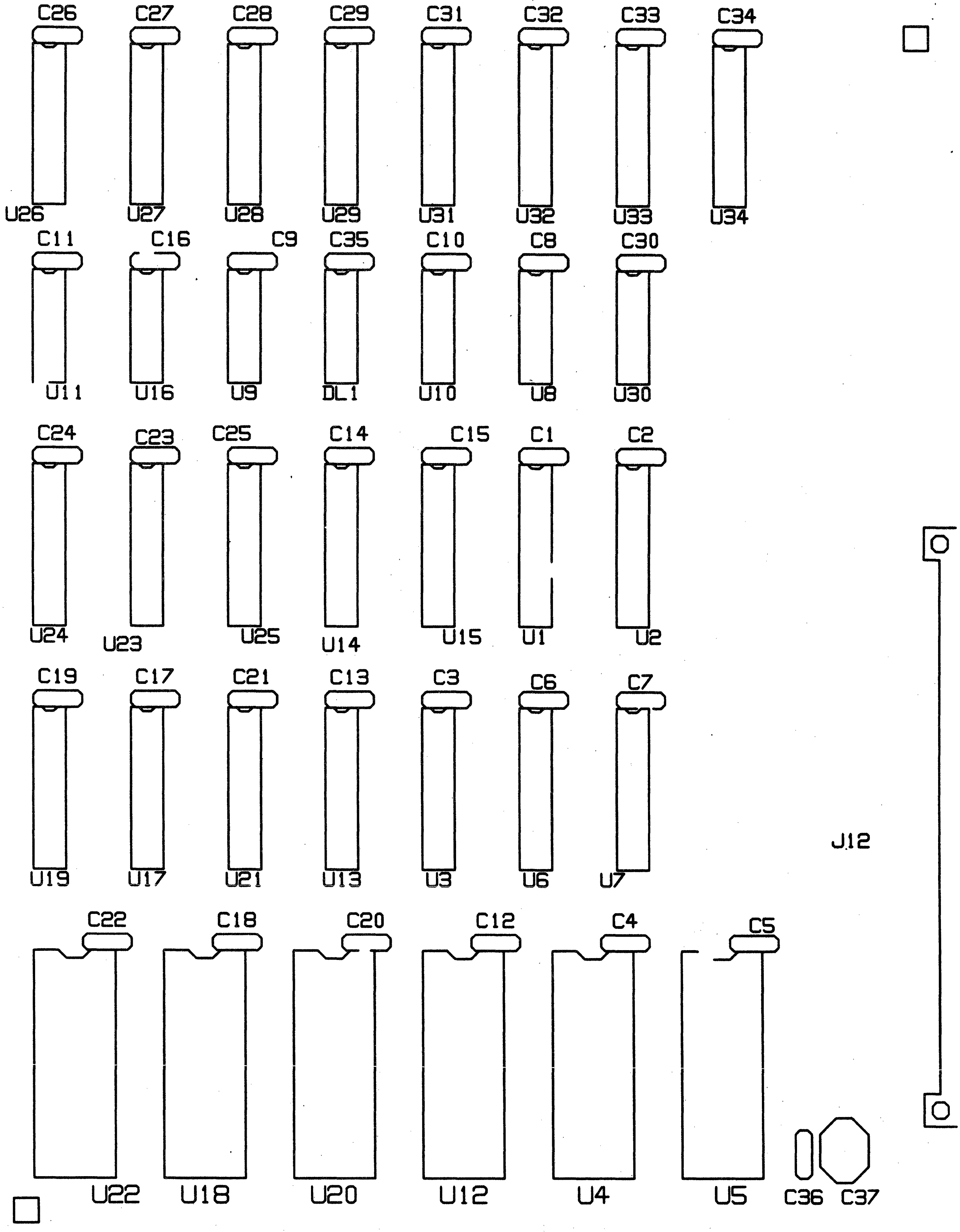
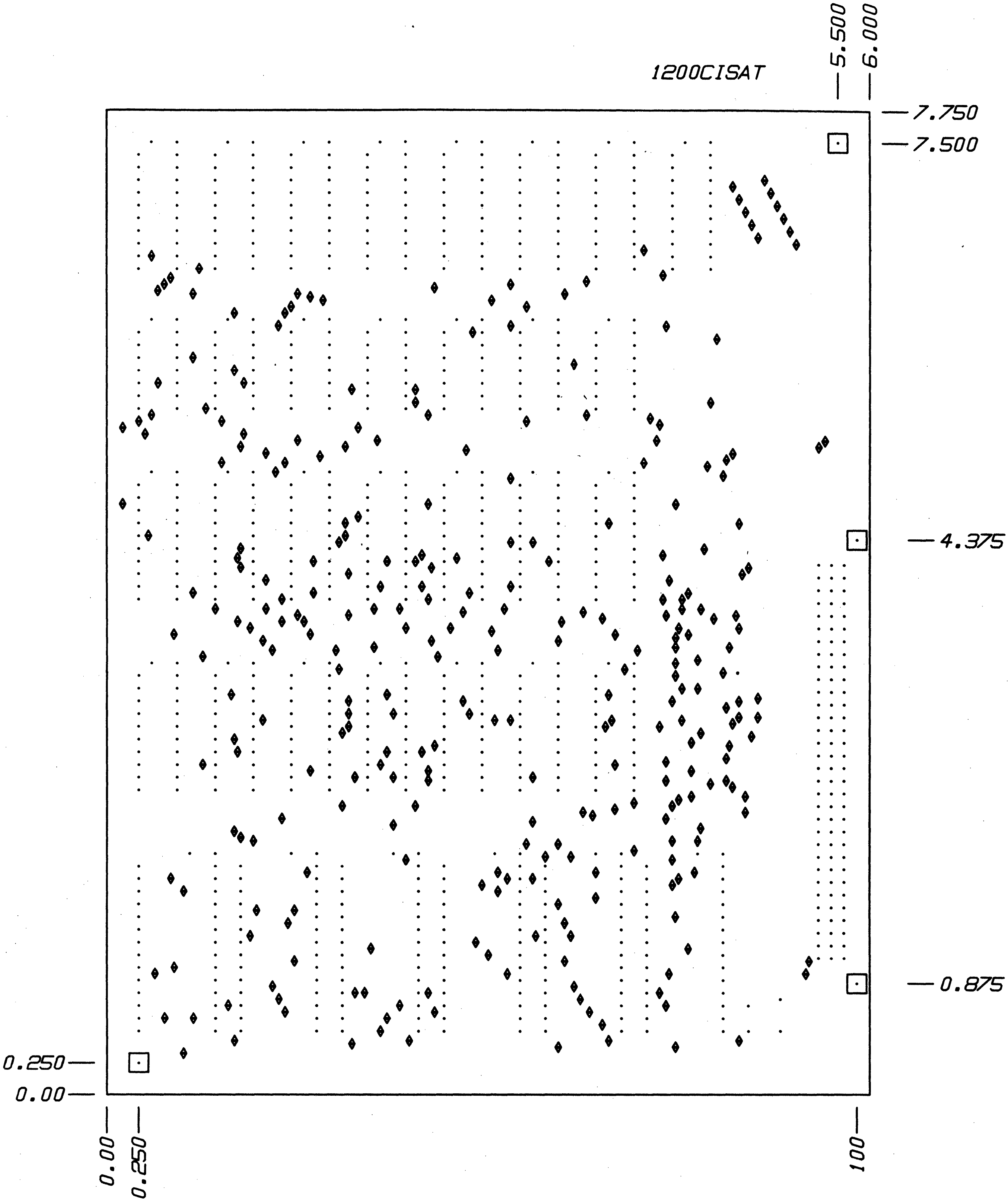
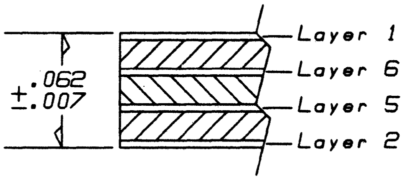


FIGURE 12





/FIGURE/	HOLESIZE/	QTY/
□	.125-N	4
◊	.031-P	305
unmarked	.037-P	876

FIGURE 13

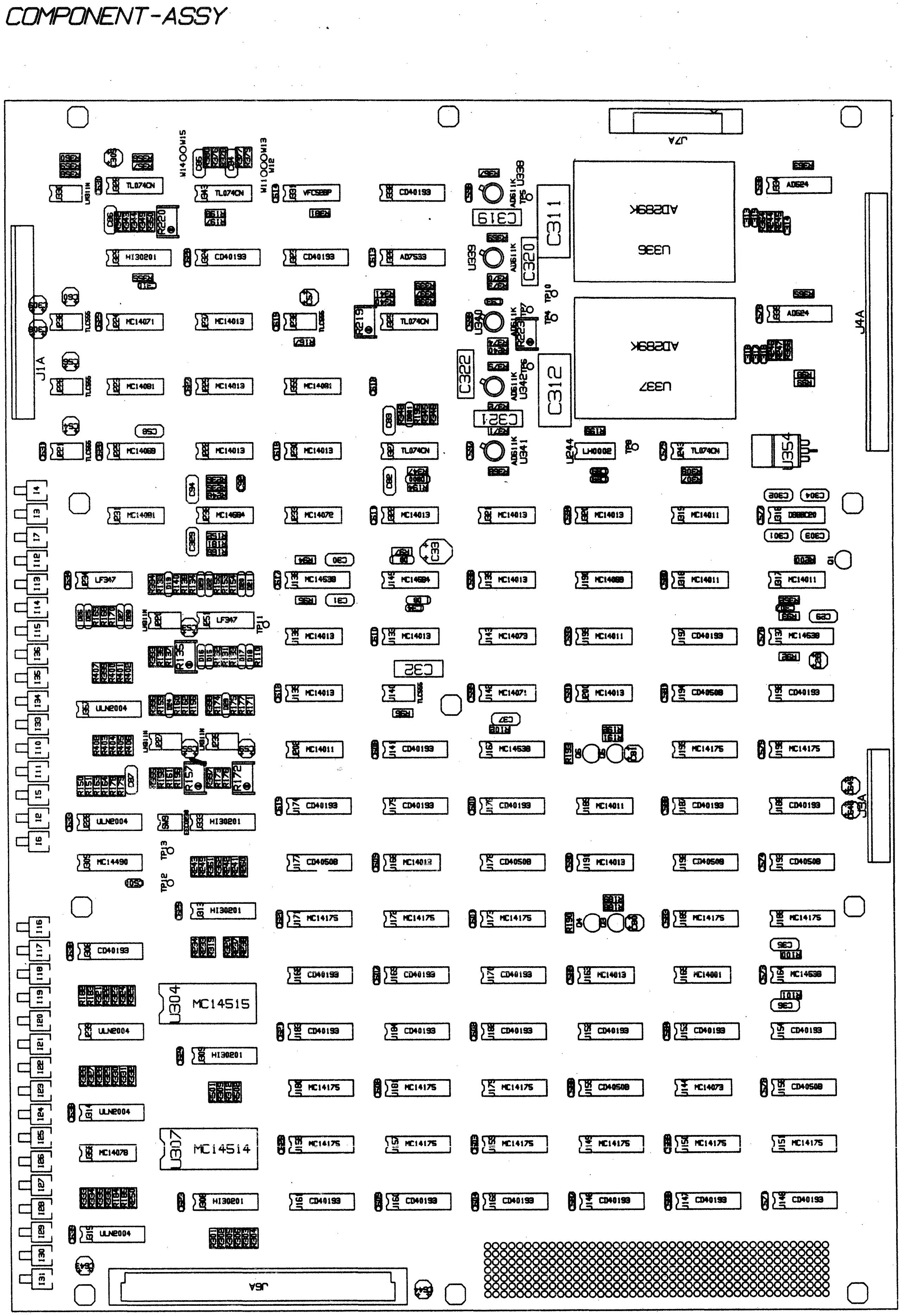

COMPONENT-ASSY
J1A
J3A
J4A
J6A
J7A
U304
MC14515
U307
MC14514
U336
AD289K
U337
AD289K
U354
C311
C312
C319
C320
C321
C322
C32
C33
U338
U339
U340
U341
U342
U244
AD611K
LH0002
R135
R157
R172
R219
R220
R223
TP5
TP6
TP7
TP8
TP9
TP10
TP11
TP12
TP13
W11
W12
W13
W14
W15
Q1
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
I12
I13
I14
I15
I16
I17
I18
I19
I20
I21
I22
I23
I24
I25
I26
I27
I28
I29
I30
I31
I32
I33
I34
I35
I36

FIGURE 14

COMPONENT-SIDE ART

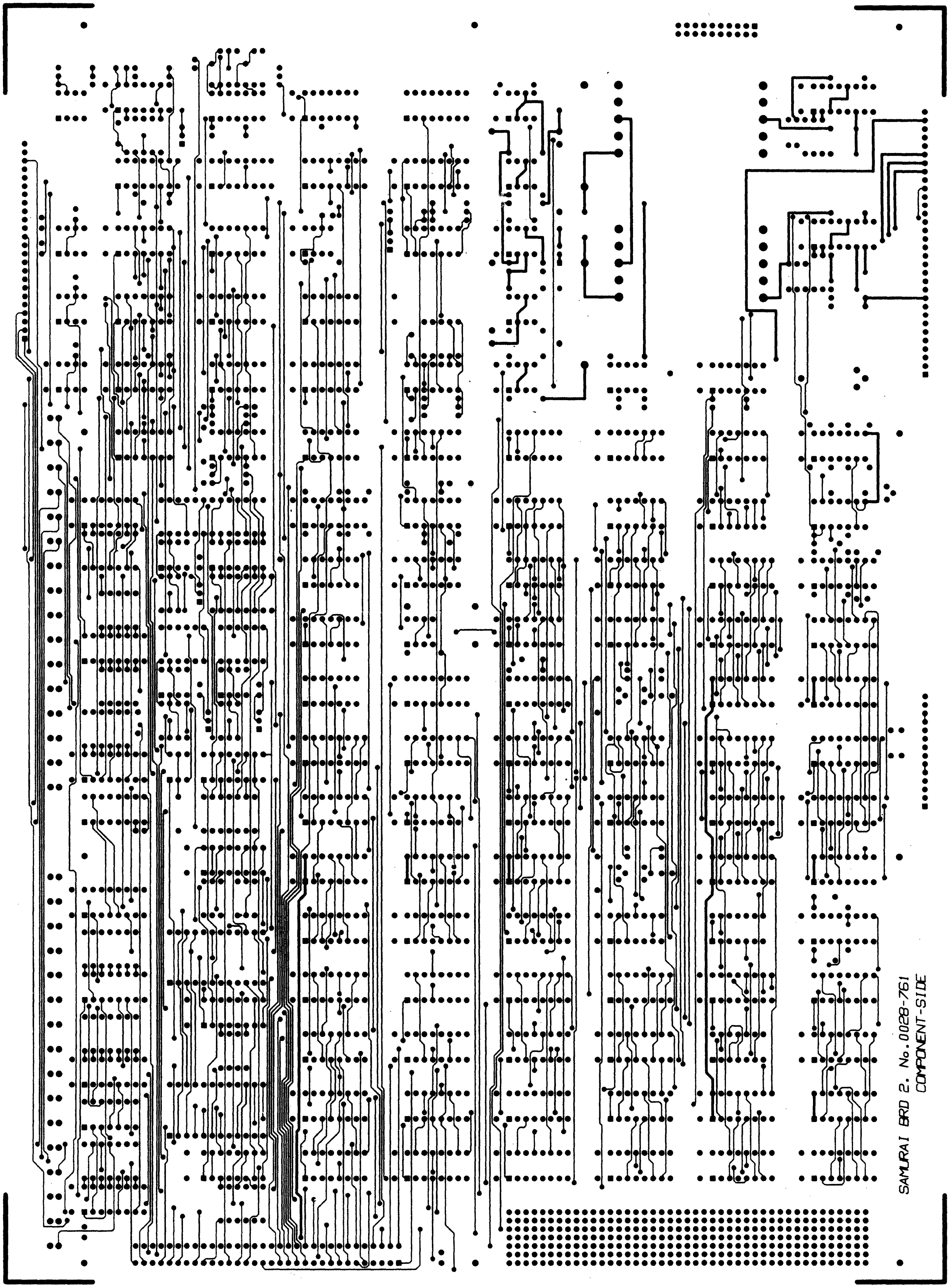


FIGURE 15

SOLDER-SIDE ART

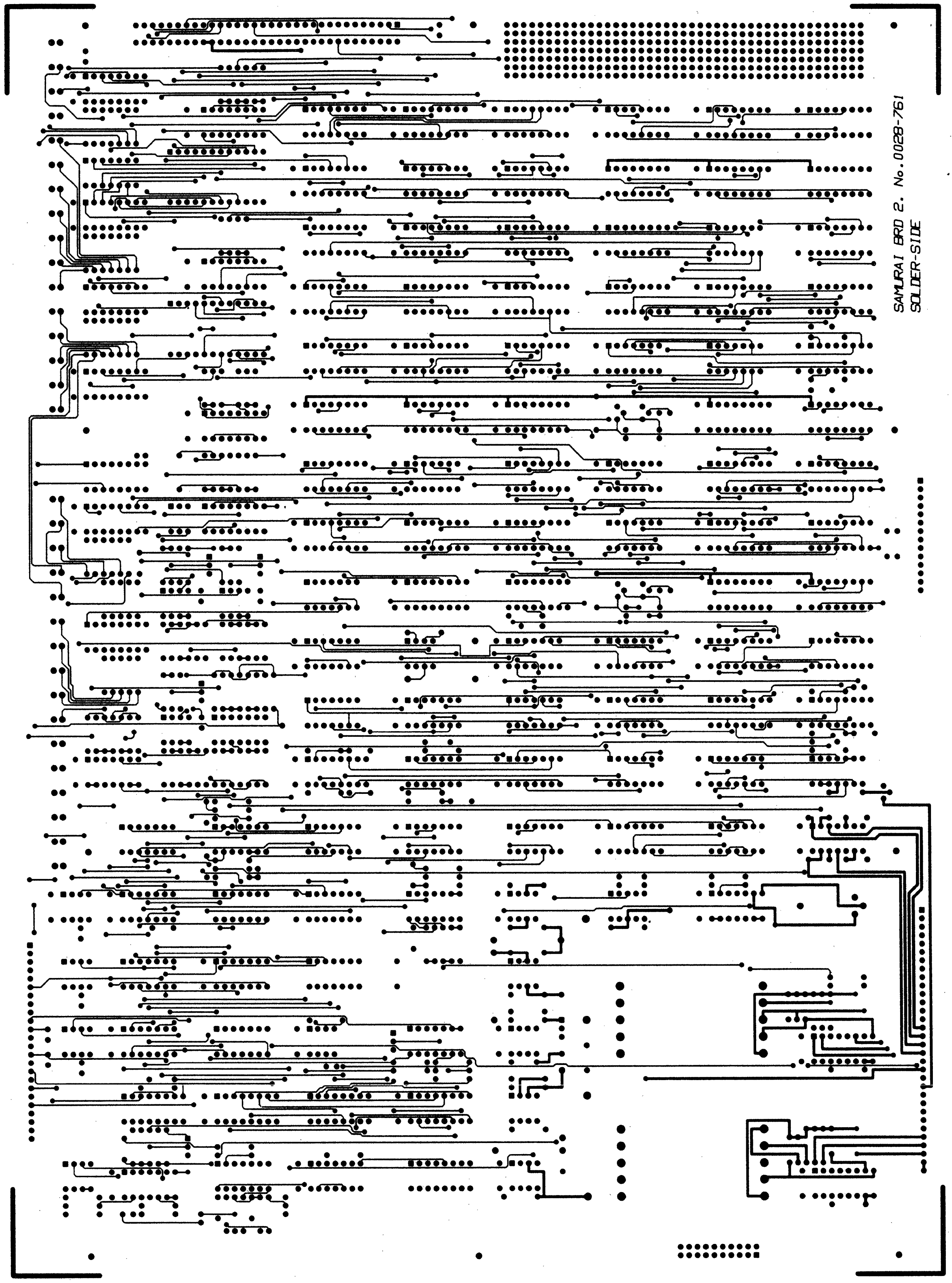


FIGURE 16

PWR-PLANE ART

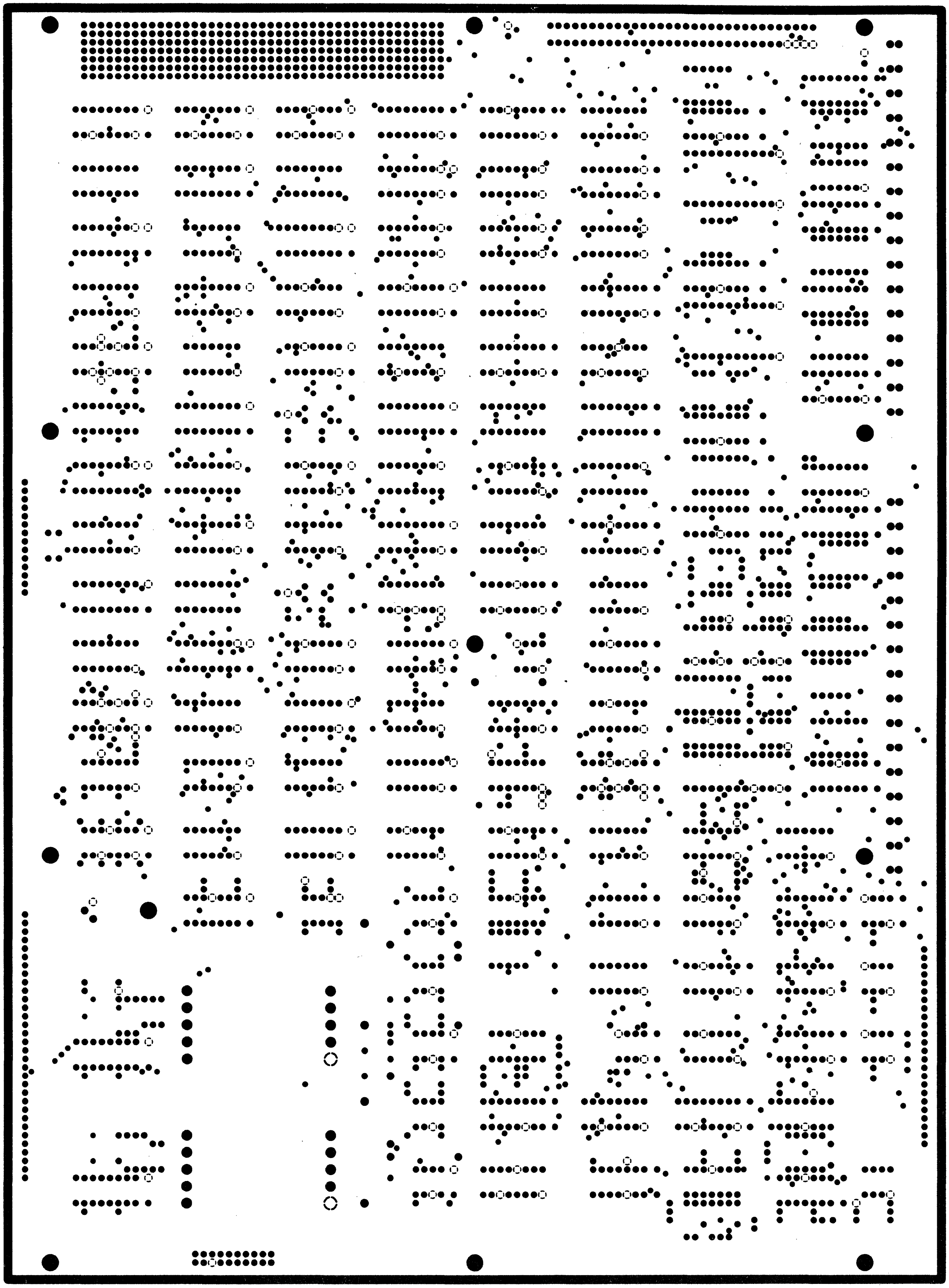


FIGURE 17

DRILL/TRIM CODE

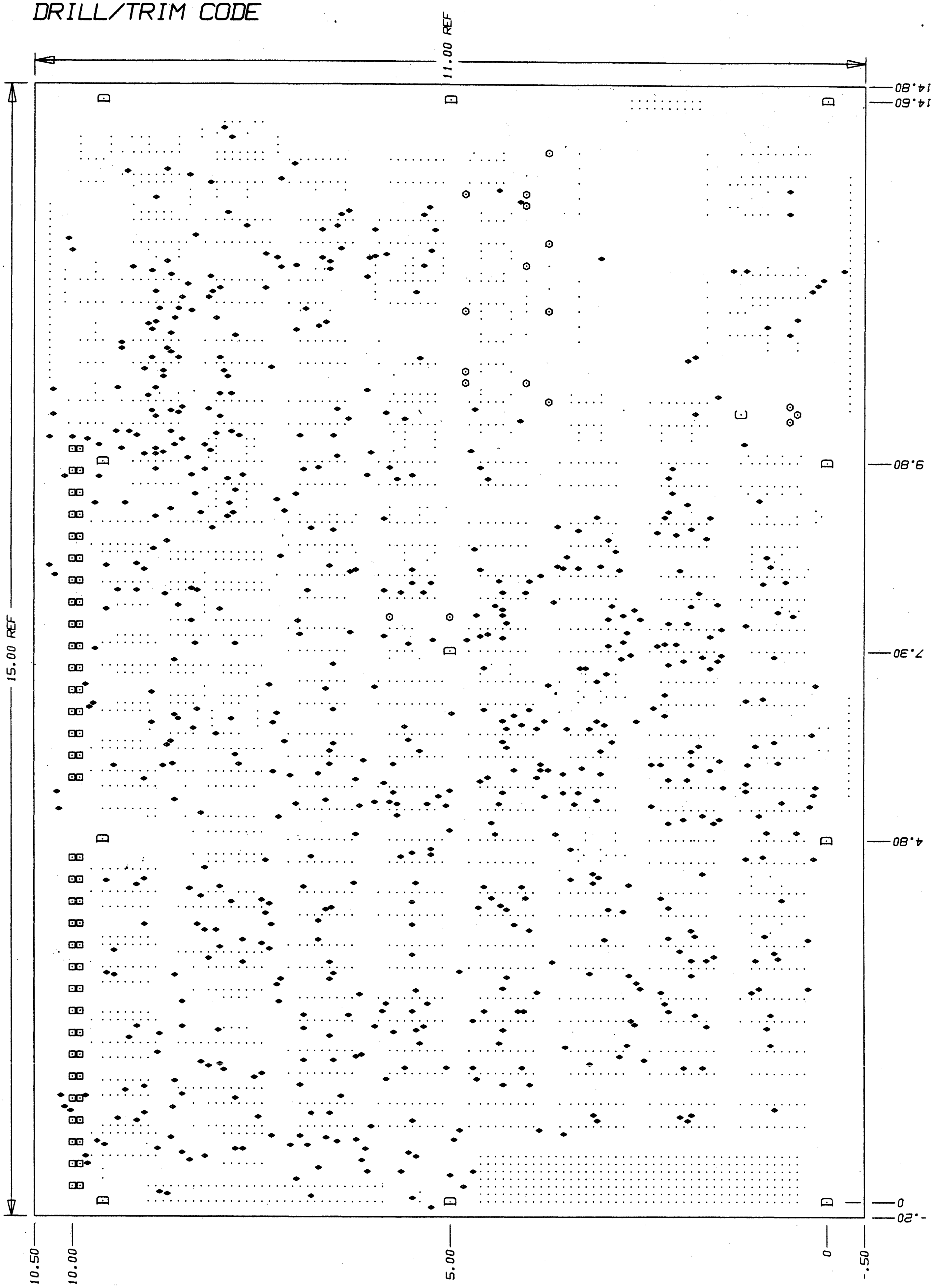


FIGURE 18

SIZE OF VARIOUS FILES

SCHEMATIC SYMBOL FILES	APPROXIMATE DISK SPACE REQUIRED
Resistor	20 Blocks
Nand Gate	20 Blocks
Amplifier	20 Blocks
Flip Flop	20 Blocks
Title Block	30 Blocks
D-Size Drawing Format	50 Blocks

PCB SYMBOL FILES	APPROXIMATE DISK SPACE REQUIRED
Edgecard Connector (100pos.)	60 Blocks
RS232 Connector (25pos.)	20 Blocks
DIP 40	20 Blocks
Disc Cap	20 Blocks
TO-3 Transistor	20 Blocks
TO-3 Transistor	20 Blocks

DETERMINATION OF DISK SPACE USAGE FOR DRAWING TYPES

SCHEMATIC DRAWINGS

		DISK BLOCKS USED *BEFORE* COMPRESS DRAWING	DISK BLOCKS USED *AFTER* COMPRESS DRAWING
Figure 1	D-Size	2020	1158
Figure 2	D-Size	1356	872
Figure 3	D-Size	1140	724
Figure 4	D-Size	1220	823
Figure 5	A-Size	452	319
	TOTALS	6188	3896

Reduction in the amount of disk space required after COMPRESS DRAWING = 2292 Blocks (37%)

PCB DRAWING "A"

Board Size = 6"x8"
Equivalent IC's = 50
No. of Layers = 4 (2-Signal, 1-Power, 1-Ground)

(PCB Drawing "A" is represented by figures 6 through 13)

	DISK BLOCKS USED BEFORE COMPRESS DRAWING	DISK BLOCKS USED AFTER COMPRESS DRAWING
4-LAYER PCB	2180*	1534

Reduction in the amount of disk space required after COMPRESS DRAWING = 646 (30%)

*Please note that NET-DATA-BASE is unaffected by COMPRESS DRAWING and is therefore not included in the total of PCB drawing block size.

ARTWORK PHOTOPLOT FILE SIZES (PCB "A")

Filename	Blocks Used
Component Side	281
Solder Side	241
Power Plane	143
Ground Plane	143
Solder Mask	146
Component Assy.	290
Silkscreen	213
Drill & Trim Detail	277
	1734

Total Project size including (5) schematics, text files, & PCB = 6623 blocks
Photoplot artwork files = 26% of total project size

PCB DRAWING "B"

Board Size = 11"x15"
Equivalent IC's = 209
No. of Layers = 6 (2-Signal, 3-power, 1-Ground)

(PCB Drawing "B" is represented by figures 14 through 18)

	DISK BLOCKS USED *BEFORE* COMPRESS DRAWING	DISK BLOCKS USED *AFTER* COMPRESS DRAWING
6-LAYER PCB	6215*	3488

Reduction in the amount of disk space required after COMPRESS DRAWING = 2727 (44%)

*Please note that NET-DATA-BASE is unaffected by COMPRESS DRAWING and is therefore not included in the total of PCB drawing block size.

ARTWORK PHOTOPLOT FILE SIZES (PCB "B")

Filename	Blocks Used
Component Side	526
Solder Side	495
+5V Power Plane	295
+12V Power Plane	295
-12V Power Plane	295
Ground Plane	295
Solder Mask	295
Component Assy.	2318
Silkscreen	2035
Drill & Trim Detail	563
	7412

Total Project Size = 13722 Blocks
Photoplot Artwork Files = 54% of total project size
(Data-base generated from TEXT NETLIST input)

TELESIS THIRD PARTY CAE INTERFACE

Frank Boyle

Product Marketing Specialist

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

The Telesis Third Party CAE Interface Program was developed to establish an organized method of generating interfaces between CAE systems and Telesis PCB CAD systems. The following paper outlines the Third Party Interface process.

INTRODUCTION

Why is Telesis committed to supporting CAE vendor interfaces to their PCB CAD systems? For a variety of reasons. Many Telesis customers already have CAE equipment installed that they would like to be capable of interfacing to. And, while Telesis has their own EDA-12ØØ CAE solutions to offer, they want to remain flexible to other solutions as well. With a formal process in place, Telesis can support the CAE vendor in the developement of a working interface. This will insure that the customer does not have to develop their own interface, or obtain an interface that is not supported.

THE INTERFACE PROCESS

All third party vendors desiring an interface should be referred to Telesis Corporate Headquarters, Chelmsford, MA. They will be supplied with the appropriate information to write a netlist and back annotation interface to the Telesis PCB CAD system, and the manpower for auditing all documentation. The information includes three Telesis Product Specifications; **Telesis Netlist Input File, Telesis Back Annotation File,** and **Preparing a Magnetic File for Telesis Input.** A brief description of these product specifications follows:

TELESIS NETLIST INPUT FILE: This describes the netlist file that must be created on the third party system, to be read into the Telesis system.

TELESIS BACK ANNOTATION FILE: This file is automatically generated on the Telesis system and is read by the third party system to update schematic drawings and reflect changes occuring due to component, gate, and pin swapping.

PREPARING A MAGNETIC FILE FOR TELESIS INPUT: This describes the format necessary for file transfer from a third party system to the Telesis system, using magnetic tape.

The third party vendor then creates a netlist and returns the media to Telesis for verification. Telesis evaluates the media and sends feedback to the vendor. Finally, a back annotation file is created and the vendor verifies that the information is readable.

If the vendor is IBM-based, they will be supplied with the **THIRD PARTY IBM NETLIST INTERFACE SOFTWARE (Part No. 8000158-001)**. If they are non-IBM-based, they will be supplied with the **THIRD PARTY NETLIST INTERFACE SOFTWARE (Part No. 80001555-001)**.

THIRD PARTY INTERFACE DATA FLOW PROCESS

The first step for both IBM-based, and non-IBM-based CAE vendors is the creation of a schematic on the third party system. Next, the third party translator is run to generate a Telesis netlist (refer to Telesis Function Specification:Telesis Netlist Input File).

If the vendor is IBM-based, the second step is to copy the netlist using a 5 1/4 inch floppy and standard (DOS) commands. Next, the file transfer is directed from the IBM/PC to the Telesis system using an RS232 file transfer.

If the vendor is non-IBM-based, the second step is to transfer the netlist file from the Telesis using magnetic tape in ASCII format (refer to Telesis Product Specification:Preparing Magnetic Tape for Telesis Input). Next, the netlist file is copied from the tape to the Telesis system using the [**ASCII FROM TAPE**] command.

On the Telesis system, a Net-Data-Base is created using the [**LOAD TEXT NETLIST**] command. Then the printed circuit board is designed. Once the design is completed, one of the following steps is performed.

1. If the third party CAE vendor reads the netlist file for back annotation, then a new netlist file is created using the [**CREATE TEXT NETLIST**] command.

2. If the third party CAE vendor reads Back-file for back annotation, then a back annotation file is created using the **[CREATE TEXT BACK ANNOTATION FILE]**.

Finally, the netlist file or back annotation file (back file) is transferred to the third party system using the process outlined earlier.

On the third party CAE system, the new netlist or back annotation file is read and the schematic is updated (refer to Telesis Product Specification: Telesis Back Annotation File).

THIRD PARTY CAE VENDORS

The following is a list of Third Party CAE vendors who currently have interfaces to the Telesis system.

FUTURENET - IBM/PC-based; DASH 2 and DASH 3 (Netlist In, Back Annotation); DASH 4 (Netlist In - DASH 3 format, Back Annotation currently in process). Telesis and FutureNet are working on a long-term strategy to support DASH 4 and additional new product offerings from FutureNet.

VALID LOGIC - non-IBM/PC-based; Netlist In, Back Annotation (Back Annotation File);

CASE TECHNOLOGY - IBM/PC-based; Netlist In (Back Annotation File);

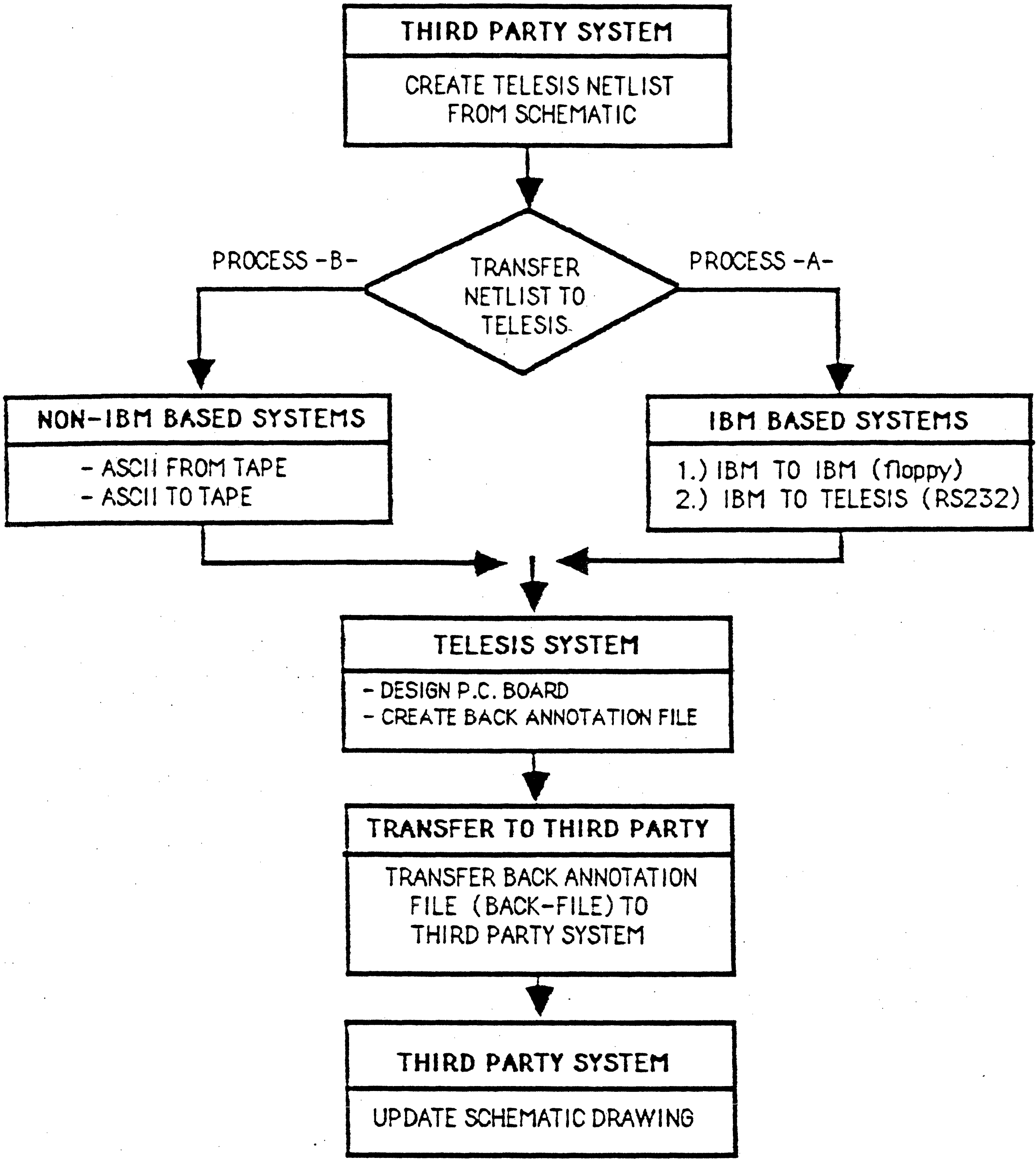
DAISY SYSTEMS - DNX, Netlist In, Back Annotation; non-IBM/PC-based system.

VIEW LOGIC - Netlist In, Back Annotation (Netlist File); IBM/PC-based system.

MENTOR GRAPHICS - Netlist In, No Back Annotation; non-IBM/PC-based system.

The following vendors have interfaces that are still in progress: **HEWLETT PACKARD, XEROX (VERSATEC), SILVAR LISCO, VISIONICS CORP., DOUBLE T SERVICES.**

THIRD PARTY CAE INTERFACE DATA FLOW



THIRD PARTY CAE INTERFACE

1986 Telesis

PRODUCT SPECIFICATION

TELESIS NETLIST INPUT FILE

CONTENTS

PAGE

INTRODUCTION

The Telesis Netlist Input File (NETLIST) is a text file used as an alternative to drawins schematics, for inputting logical data to the Telesis Printed Circuit Board layout system.

DEFINITIONS

Package Name (Package): This is the symbol used on the physical board layout. Exp: DIP14

Device Type (Gate): This identifies the logic function or gate on the schematic. Exp: 74LS0

Value: Describes electrical characteristics of some components a resistor may have. Exp: (10K 5% 1/4W)

Function Designator (Funcdes): This is an identifier that links

Reference Designator (Refdes): This is an identifier that links logic functions to the package. Exp: U2 = 7400
R1 = Resistor

Net-Data-Base: This is the data base created on the Telesis system from a netlist or schematic that is used to do the physical layout.

Device Description File (Device File): A text file that describes the logical functions in each package on the P.C. board.

Example:

```
(Device Description File: 7400)
PACKAGE DIP14
PINCOUNT 14
PINORDER 7400 A B C
PINUSE 7400 IN IN OUT
FUNCTION NAND1 7400 1 2 3
FUNCTION NAND2 7400 4 5 6
FUNCTION NAND3 7400 9 10 8
FUNCTION NAND4 7400 12 13 11
POWER +5V; 14
GROUND GND; 7
END
```

Pin Number: Identifies the pins on logic functions and cross reference to physical pins on each package.

Pin Name: This is the logical pin names on each device. An example would be 7400 has pin names A, B, Y, where A and B identify the two input pins and Y is the output pin for each 2-input NAND Gate.

Pin Description File (Pin File): A text file that describes the pad and hole sizes to be used for each symbol on the P.C. board.

Example:

```
(PIN FILE FOR PACKAGE SYMBOL DIP 14)
(NAME OF FILE DIP14-PIN)
(TELESIS STANDARD PACKAGE LIBRARY)
PINTYPE A
DRILL  .039-P
PAD SQUARE= .062 COMPONENT SIDE
PAD SQUARE= .062 SOLDER-SIDE
PAD CIRCLE= .050 INTERNAL-SIGNAL
THERMAL-RELIEF FLASH=AB00 IMBEDDED-PLANE
ANTI-PAD CIRCLE= .062 IMBEDDED-PLANE
PAD SQUARE= .080 COMPONENT-SOLDER-MASK
PAD SQUARE= .080 SOLDER-SOLDER-MASK
PINTYPE B
DRILL .039-P
PAD CIRCLE= .062 COMPONENT-SIDE
PAD CIRCLE= .062 SODLER-SIDE
PAD CIRCLE= .050 INTERNAL-SIGNAL
THERMAL-RELIEF FLASH= AB00 IMBEDDED-PLANE
ANTI-PAD CIRCLE= .062 IMBEDDED-PLANE
PAD CIRCLE= .080 COMPONENT-SOLDER-MASK
PAD CIRCLE= .080 SOLDER-SOLDER-MASK
PIN 1 A
PIN 2-14 B
END
```

FORMAT OF THE NETLIST

A netlist for Telesis input is a single text file logically divided into four parts:

- o PART 1 ($WASIS) is a single line that labels the file for back annotation. (Optional)

- o PART 2 ($PACKAGES) is a list of packages and reference designators. (Optional)

- o PART 3 ($FUNCTIONS) is a list of functions (FUNCTIONS = DEVICE TYPES). (Optional)

- o PART 4 ($NETS) is a list of all nets or connections. (Required)

NOTE: The netlist file must be formatted in one of the following ways:

- o OPTION 1: Parts 2 and 4; this is a fully assigned netlist that cannot be back annotated. (refer to examples on page)

- o OPTION 2: Parts 2, 3, and 4; this format allows unassigned, partially assigned, or fully assigned netlist that can be back annotated. (refer to examples on page)

- o OPTION 3: Any combination of Option 1 and Option 2.

Netlist Format Option 1:

This option extracts data from the $PACKAGES adn $NETS sections only. Al packages are assigned reference designator and pin numbers. There is no information on the logical device types to allow gate and pin swapping or back annotation to the schematic drawing.

```
        Example:    (PACKAGES SECTION)

                    $PACKAGES
                    DIP14!74LS04;U2
                    DIP16!74LS112;U3 U4
                    DIP14!74LS32;U1

                    (NETS SECTION)

                    $NETS
                    CLEAR;U3.14 U3.15 U4.14 U4.15
                    CLK;U3.13 U3.1 U4.13 U4.1
                    ONE;U4.11 U4.12
                    PRESET;U3.10 U3.4 U4.10 U4.4
                    ;U4.9 U4.3 U4.2
                    ;U3.11 U4.5
                    ;U1.6 U1.2 U2.1
                    ;U2.2 U3.12
                    ;U1.3 U2.3
                    ;U2.4 U3.3 U3.2
                    ;U1.1 U3.7
                    ;U1.4 U4.6
                    ;U1.5 U4.7
                    +5V;U1.14 U3.16 U4.16 U2.14
                    GND;U1.7 U3.8 U4.8 U2.7
                    $END
```

Netlist Format Option 2:

```
                    $WASIS
                    (FUNCTION SECTION)

                    $FUNCTIONS
                    74LS04;F1 F2
                    74LS112;F3 F4 F5 F6
                    74LS32;F7 F8

                    (NETS SECTION)

                    $NETS
                    CLEAR;U3:F5.CLR U3:F6.CLR U4:F3.CLR U4:F4.CLR
                    CLK  ;U3:F6.CK U3.F5.CK U4:F4.CK U4:F3.CK
                    RESET;U3:F6.PR U3:F5.PR U4:F4.PR U4:F3.PR
                    ;U4:F4.Q U4:F3.J U4:F3.K
                    ;U3:F6.J U4:F3.Q
                    ;U1:F8.Y U1:F7.B U2:F1.A
                    ;U2:F1.Y U3:F6.K
                    ;U1:F7.Y U2:F2.A
                    ;U2:F2.Y U3:F5.J U3:F5.K
                    ;U1:F7.A U3:F6.QN
                    ;U1:F8.A U4:F3.QN
                    ;U1:F8.B U4:F4.QN
                    $END
```

$WASIS

The $WASIS section consists of one line with the keyword $WASIS only.

KEYWORD: $WASIS

This optional keyword must be used in back annotable netlists, and must precede the $FUNCTIONS and $NETS sections.

$PACKAGES (Optional)

PURPOSE: The $PACKAGES section of the netlist file allows pre-assignment of reference designators to packages during creation of the net-data-base.

KEYWORD: $PACKAGES

o The $PACKAGES keyword is placed on the first line of the text file. A list of packages follows the keyword.

LIST OF PACKAGES

o The list includes packages to be used in the board design, with a reference designator for each instance of a package. Packages are listed by packages name and device type. The operator may omit the packages symbol name, however, the device type must be included if device description files are to be used to create the net-data-base. The packages name must be included if the system uses pin description files to create the net-data-base.

NOTES:

1. The value field is optional for symbol values.
2. Either package name or device type, or both, must always be present.

FORMAT: PACKAGE NAME!DEVICE TYPE!VALUE;REFERENCE DESIGNATOR

```
EXAMPLES: RES500!RLR05!10K 5% 1/4W;R1 R2 R3-- (Package, Device, Value Refdes)
          RLR05;R1 R2 R3                    -- (Device type refdes only)
          RES500;R1 R2 R3                   -- (Package name refdes only)
```

NOTES:

1. The value field is optional for symbol values.
2. Either package name or device type, or both, must always be present.

PACKAGE NAME: (Optional if Device Type used)

The package name must be precisely the same as the name of symbol to be placed on the board drawing.

A package name may contain up to 14 characters with no blank spaces allowed. (Allowed characters are: A-Z, 0-9, - + = % * / . ").

Each package name must be placed at the beginning of a new line.

Repeat the package name on a new line for each different device type.

DEVICE TYPE: (Optional if using package name)

Place the device type between the package name and the symbol value or list of reference designators. It must be preceded by an exclamation point.

The device type must be the same name of the device file to be used with the package.

A device type may contain up to 18 alphanumeric characters with no blank spaces. (Allowed characters are: A-Z, 0-9, - + = % * / . ").

VALUE: (Optional)

Place the value between the device type and the list of reference designators. It must be preceded by an exclamation point. If a value does not exist, the exclamation point must be omitted.

A value may contain up to 18 alphanumeric characters including blank spaces. (Allowed characters are: A-Z, 0-9, - + = % * / . ").

```
Example:     RES500!RLR05!10K 5% 1/4W;R1
             RES500!RLR06;R2
```

REFERENCE DESIGNATORS:

A semicolon (;) must be placed before the list of reference designators.

A reference designator may contain up to 8 characters with no blank spaces. (Allowed characters are: A-Z, 0-9, - + = % * / . ").

Reference designators must be separated from each other with a blank space, or with a comma (,).

```
Example:   RES500!RL05;R1, R2, R3 --
           RES500!RL05;R1, R2, R3 -- (Equal)
```

To continue the list of reference designators on a new line, place a comma at the end of the line and continue anywhere on the line below:

```
Example:   RES500!RC07;R1 R2 R3 R4 R5 R6 R7
```

To input a series of reference designators, brackets may be used:

```
Example:   U6 U[8-10] U12 U15   ---
           U6 U8 U9 U10 U12 U15 ---    --(equal)
```

- The repeating character(s) precedes the first bracket.
- A blank space before the first bracket is not allowed.
- The characters in the series must be numeric.
- The first bracket and the closing bracket must be on same line. (Blank spaces are allowed within the brackets.
- Use a hyphen (-) between the first and last numbers in the series.
- Leading zeros are allowed in reference designators (U01 U02 U03). If you input a series the number of digits in each reference designator must be the same. For example, if [U0008-U0100] is input all leading zeros are used; if [U008-U0100] is input the system ignores the leading zero in the last number and U0100 becomes U100.

TO OMIT A PACKAGE NAME:

Begin the new line with an exclamation point followed by the device type.

```
EXAMPLE:  ! 7400 ; U1  U4  U6
```

TO OMIT THE DEVICE TYPE:

Omit both the exclamation point and the device type name.

```
EXAMPLE:  DIP14 ; U1  U4  U6
```

$FUNCTIONS (Optional - Must follow the $PACKAGES section, if used)
- Devices specified under $PACKAGES cannot be re-specified under $FUNCTIONS, and the reverse.

PURPOSE: The optional $FUNCTIONS section of the text netlist file is useful if the operator is creating a netlist from an unannotated engineering sketch. For example, if the sketch shows eight NAND2 gates without pin number and reference designator annotation, the operator may input entries under $FUNCTIONS. An entry must contain a device type, an operator-defined FUNCTION DESIGNATOR, as well as the number of functions to be created for the device specified. Based on the information in the device file, the system creates the required number of packages needed to accommodate the number of functions.

Entries made to the $FUNCTIONS section are normally unassigned. That is, the Telesis system assigns available reference designators automatically to the devices while creating Telesis net data base.

During the net data base creation , the Telesis system reads the entries within $FUNCTIONS, then reads the $NETS section of the file. The $NETS section must contain FUNCTION DESIGNATORs, and PIN NAMEs or pin numbers or both.

KEYWORD: $FUNCTIONS

o If used, the $FUNCTIONS keyword must follow the $PACKAGES section of the file. A list of device types follows the keyword. Devices specified under $PACKAGES cannot be re-stated in the $FUNCTIONS section of the netlist.

<u>LIST OF FUNCTIONS</u>

o The list includes a device type with an operator-defined FUNCTION DESIGNATOR. Based on the information in the device description file, the system creates the proper number of components for the number of logic functions specified. In addition an optional value field may be included

Package names are not required since device description files are used to support this optional section of the netlist file.

FORMAT: DEVICE TYPE ; FUNCTION DESIGNATOR

```
EXAMPLE:   $FUNCTIONS
           74LS00 ; AND[1-4]
           74L74 ; FLIP[1-2]
           RESPAK!100K;RES[1-8]
```

In the above example, the device 74LS00 is stated under $FUNCTIONS, with the operator-defined function designator AND, requiring four logic functions to be created in the net data base. The system creates a single component in the net data base, since the device 74LS00 only contains four logic functions.

The example below illustrates the typical entries to the $NETS section of the file, relating to the entries contained under $FUNCTIONS.

```
$NETS
; FLIP1.Q  AND2.A
; AND1.Y  FLIP2.D
```

The line ;FLIP1.Q AND2.A identifies available functions and pin names associated with that net. FLIP1.Q identifies an available function on the device 74L74, with pin name Q connected to pin name A of the function, AND2, of the device 74LS00. However, because function designators and pin names are specified, logic functions are free for swapping with other functions of the same device type on the placed board drawing. For example, the AND2 function assignment in the net data base is swappable with other available functions on the same device (AND1, AND3, AND4).

NOTE ON THE NUMBER OF FUNCTIONS

Under the $FUNCTIONS section of the netlist file, the operator may specify a device type, the function designator, and the total series specific, exceeding the number of functions available on one device. For example, the 74LS00 device contains four functions. The operator, however, may specify any number of functions. When the net data base is created, the system creates the appropriate number of components needed for the number of functions specified.

```
EXAMPLE:  $FUNCTIONS
          74LS00 ; AND[1-8]
```

In the above example, eight functions will be created in the net data base for the device 74LS00. Since only four functions can fit on one physical device, the system creates the net data base with two 74LS00 devices with four available functions on each device.

However, if eight functions are specified for a device under $FUNCTIONS, with only four functions reflected under the $NETS section, the system will still create two 74LS00 packages in the net data base. Five functions (AND[1-5]) reflected under $NETS will create two packages, since one 74LS00 contains only four logic functions.

If a device contains only one function, the brackets may be omitted. In the example below the operator-defined function designator, "NAND", contains only one function. Simple omit the brackets and place the number "1" immediately after the function designator. The second example shows how a discrete component is identified.

```
EXAMPLE:  $FUNCTIONS
          RESISTOR;RES1
```

NOTE ON FUNCTION TYPES

If a single device contains different logic functions, the function type specified in the device description file must be stated under $FUNCTIONS after the device name, preceding the operator-defined function designator. The system requires the additional function type entry to make the distinction between different functions on a single device. For example, the device 4000A contains two NOR3 gates and the inverter (INV) logic function. The 4000A device file is shown below:

```
(DEVICE DESCRIPTION FILE: 4000A)
PACKAGE DIP14
CLASS IC
PINCOUNT 14
PINORDER NOR3 A B C Y
PINUSE NOR3 IN IN IN OUT
PINSWAP NOR3 A B C
FUNCTION G1 NOR3 3 4 5 6
FUNCTION G3 NOR3 11 12 13 10
PINORDER INV A Y
PINUSE INV IN OUT
FUNCTION G2 INV 8 9
POWER +5V; 14
GROUND GND; 7
NC 1 2
END
```

The format under $FUNCTIONS is shown below for single device types containing different functions.

```
DEVICE NAME ! FUNCTION TYPE ; FUNCTION DESIGNATOR[SERIES SPECIFIED[n1-n2]
```

```
EXAMPLE:  $FUNCTIONS
          4000A! NOR3; XYZ[1-2]
          4000A! INV ; ABC1
```

- o Line entries must be repeated for the different function types.
- o In the above example, the NOR3 function type is specified by the operator-defined function designator XYZ, with two available functions. The function type INV is specified by the function designator ABC, with one available function.

DEVICE TYPE NAME:

The device type name must be the exact same name as its device description file

A device type may contain up to 18 alphanumeric characters with no blank spaces. (Allowed characters are: A-Z 0-9 - + = % * / . ")

The package name does not need to be included since $FUNCTIONS section of the netlist file requires device description files to create the net data base.

Simply input the device name at the beginning of each line, following the $FUNCTIONS statement.

```
EXAMPLE:  $FUNCTIONS
          74LS00 ;
```

FUNCTION DESIGNATORS and [SERIES SPECIFIED]

A semicolon (;) must follow the device name, preceding the function designator and the optional series specified.

An operator-defined function designator may contain up to eight characters with no blank spaces. (Allowed characters are A-Z 0-9 - + = % * / ")

Immediately following the function designator, input series specified , using brackets to specify a range. Do not input a blank space between the function designator and the series specified. Brackets may be omitted if the device contains only one function.

```
EXAMPLE:  $FUNCTIONS
          74LS00 ; AND[1-4]
          74L74  ;FLIP1
```

Refer to NOTE ON FUNCTION DESIGNATORS on the previous page for information on single devices containing different function types and an example of a typical entry under $FUNCTIONS.

$NETS

PURPOSE: The $NETS section of the netlist file specifies the net information for the entries made under the $PACKAGES and $FUNCTIONS sections of the file.

The net information specified must be entered in two different formats if the operator used the optional $FUNCTIONS section of the netlist. Nets specified for entries contained under $PACKAGES are entered by pre-assigned reference designators and pin numbers. Nets specified for entries under the $FUNCTIONS section are entered by function designator, function number, and pin name.

NOTE: The function number provides a unique name to to each function designator associated with a single logic function on a device. For example, AND1.A, AND1.B, AND1.Y are functions associated with a single function on the 74LS00, specified by the function designator AND listed under $FUNCTIONS.

The following example illustrates typical net entries. The first entry shown under $NETS uses reference designators and pin numbers, relating to $PACKAGES. The second net entry uses a function designator, function number, and pin name, relating to $FUNCTIONS. The third net entry under $NETS uses information pertaining to both the $PACKAGES and $FUNCTIONS sections.

```
(Example):$WASIS
          $PACKAGES
          RES400 ! RLR05 ; R2
          CAPRAD 300; C1
          $FUNCTIONS
          74LS00 ; AND[1-4]
          74L74 ; FLIP[1-2]
          $NETS
          ; R2.1 C1.2
          ; AND1.A FLIP2.QN
          ; R1.1 FLIP1.QN
          $END
```

NOTE: The function designator and function number, FLIP1 for example, under $NETS specifies an available function slot on the device. FLIP2 also specifies an available function on the device. The function number does not imply a selected function slot.

KEYWORD $NETS

o The keyword is placed on the line following the list of packages, and/or list of functions. The keyword $NETS precedes the list of nets.

LIST OF NETS

o This includes all nets that will be used in the board design, and provides a list of pins and/or function designator and pin names associated with each net.

FORMAT:

For nets relating to devices and reference designators specified under $PACKAGES:

```
FORMAT:  ;RefDes.PinNumber RefDes.PinNumber RefDes.PinNumber

EXAMPLE:  ;R2.1 C2.1 U4.3
```

For nets relating to devices, function designators and number of functions specified under $FUNCTIONS:

```
FORMAT: ;FunctionDesignatorFunctionNumber.PinName,
         FunctionDesignatorFunctionNumber.PinName

EXAMPLE:  ;AND1.A  FLIP2.QN
```

For nets relating to entries pertaining to both $PACKAGES and $FUNCTIONS sections.

```
FORMAT: ;RefDes.PinNumber FunctionDesignatorFunctionNum.PinName

EXAMPLE: ;R1.1  FLIP1.QN
```

INPUT SEQUENCE:

o Each net begins on a new line. A semicolon must precede the net information.

o <u>For nets specified by reference designator:</u>

-Show the reference designator and pin number for each pin. The format is --REFDES.PIN#--with no blank spaces allowed.

-Reference designators must correspond to the ones shown in the list of packages.

-A pin number may be any integer from 1 to 32767.

-Put at least one blank space between pins in the list.

-To continue the list of pins on a new line, place a comma at the end of the line and continue on the line below.

-As in the list of packages, brackets may be used to input a series of reference designators, a series of pin numbers, or both:

```
Z[1-3].7 is equivalent to Z1.7  Z2.7  Z3.7
Z3.[2-5] is equivalent to Z3.2  Z3.3  Z3.4 Z3.5
RP[1-3].[1-2] is equivalent to RP1.1  RP1.2  RP2.1
          RP2.1  RP2.2  RP3.1  RP3.2
```

There must be no blank spaces between the (.) and a bracket preceding it, or following it. Otherwise the the input requirements for a bracketed series are the same as in the $PACKAGES list.

```
EXAMPLE:  $NETS
          ;R2.2  C1.1
```

o For nets specified by function designator

-Show the function designator, the function number and the pin name. The format is --FunctionDesFunctionNum.PinName-- with no blank spaces allowed.

-Function designators must correspond to the ones shown in the list of functions.

-The function number (providing the unique name associated with a single logic function) immediately following the function designator must be within the range specified in brackets under the $FUNCTIONS section. Blank spaces are not allowed between the function designator and the function number.

-The pin name must be separated from the function number by by a (.). Blank spaces are not allowed.

-Put at least one blank space between pin name and the next entry on the line.

```
EXAMPLE:  $NETS
          ;FLIP1.Q  AND3.A
```

-To continue the net information on a new line, place a comma at the end of the line and continue on the line below.

NET NAME (OPTIONAL)

o The operator may optionally give each net a name.

FORMAT: NETNAME; RefDes.PinNum RefDes.PinNum

```
EXAMPLE:

$NETS
SERIAL CLOCK; J7.22  Z10.11
```

-The net name must be the same as the net name the operator uses on the board drawing.

-A net name may contain up to 18 alphanumeric characters with blank spaces allowed. (Allowed characters are; A-Z 0-9 - + = % * / . : ")

-Place the net name at the beginning of the line before the net information.

END STATEMENT (OPTIONAL)

o The END statement must be the last line of the text file if the operator chooses to use it.

FORMAT: $END

CONTINUATON LINES

If a list reference designators or pins is too long to fit on one line, then put a comma (,) after the last entry on one line, and continue the list on to next line. You may make as many continuation lines as needed, up to 256 entries.
Example:

```
$PACKAGES

DIP14  !  74LS04    ; Z03 Z04 Z05 Z06 Z07 Z23 Z14 Z01 Z34 Z56,
                      Z15 Z42 Z28 Z57 Z19
RES400 ! RES-PULLUP; R1  R3  R6  R2  R12

$NETS
GROUND ! Z01.7 Z03.7 Z04.7 Z05.7 Z07.7 Z06.7 Z14.7 R1.1 R3.1 R12.1
```

COMMENTS (Optional)

The Telesis System treats any characters inside parentheses as comments, and ignores them during processing. You can put comments anywhere in your input text file.

FORMAT: (COMMENT LINE)

- A comment may not extend beyond the length of one line.

Example: (The following section lists packages)

```
$PACKAGES

DIP14   !  74LLS10  ; Z05 Z03 Z11
        !  RESISTOR ; R3  R5  R2  R17
        !  CAP-BYP  ; C18 C03
DIP14   !  7400     ; Z14 Z02

(The following section lists the pins of each net)
$NETS

STROBE    ; Z14.11 Z05.2  R3.2  R17.2
```

THE USE OF BLANKS

Blank characters separate items in lists of reference designators and pins. Blanks inside net names are part of the net name. Otherwise, blanks are ignored. Therefore, the following two examples are considered the same by LOAD TEXT NETLIST:

```
$PACKAGES

DIP14   !  SN74181 ;  Z05   Z03   Z11
        !  RES5K   ;  R3    R5    R2  R17

$NETS

STROBE  ; Z14.11 Z05.2 R3.2 R17.2 Z14.5 C03.1 C18.1
```

Is the same as:

```
$PACKAGES

DIP14   !  SN74181 ; Z05   Z03   Z11
        !  RES5K   ; R3    R5    R2    R17

$NETS
STROBE  ; Z14.11 Z05.2 R3.2 R17.2 Z14.5 C03.1 C18.1
```

OPEN-SERIES REFERENCE DESIGNATOR:

The user can indicate that he wants to assign a function, NOT to a particular component, but to ANY of a reference designator series. For example, all resistors will be series Rn, where "n" represents a series of numbers, 1, 2, 3, ... Telesis currently offers this feature in the EDA-300/-700 schematic Extract Netlist assignment process: Any logic function with a reference designator ending in "*" will be assigned to the next available refdes in that series. For example, refdes R* is assigned to R1, R2, R3, or whatever is the next available.

Example:

U*:F5.A

will assign function F5 to the next available refdes in the series Un. If there is an available slot in an existing package of the correct device type, the function will be assigned there. Otherwise, we will create a new package of the correct type with the next refdes in series U*, and assign the function to that.

RULE FOR RESOLVING AMBIGUOUS FIELDS:

DESIGNATOR FIELD: (left of the period, containing either the refdes or funcdes, or both) If there is only one entry (that is, no colon in the field), compare the entry first with the list of reference-designators, then function-designators.

PIN FIELD: (right of the period, containing either the pin-name, pin-number, or both) If there is only one entry (that is, no colon in the field) and it is numeric, it is a pin-number otherwise it is a pin-name. We currently do not support alpha-numeric pin-numbers.

PRODUCT SPECIFICATION

TELESIS BACK ANNOTATION FILE

CONTENTS

PRODUCT SPECIFICATION

TELESIS BACK ANNOTATION FILE

INTRODUCTION

This specification describes the file format output by the Telesis workstation, for backannotation to third party schematic capture systems. Section 1 defines important terms. Section 2 describes the back annotation process and how Telesis and third party supplied programs automate the process. Section 3 defines the back annotation file format that the third party software will use to update drawings. Section 4 explains the effects editing on a Telesis workstation has on the back annotation process. Section 5 and 6 specify the function the third party back annotator should perform and suggests messages that should be generated when problems are detected. Section 7 is an example of a Telesis back annotatable netlist.

Definitions

Back Annotation

In Telesis, "back annotation" refers to the ability to update the reference designators and pin numbers on the original schematics to reflect package name changes, changes due to pin swapping and to gate swapping packages.

Function Designator

A "function designator" is the unique name each function (gate) has on a schematic. Note that a reference designator (circuit designator) does not uniquely identify the functions it contains since each function has the same refdes. For example, a 7400 has four gates on a schematic with a single refdes like U6. Reference designators provide unique identification of physical package instances on a board.

Back Annotation Process

1. Create a schematic using a third party package and generate a Telesis netlist.

2. Transfer the netlist to the Telesis PCB Design System and design the board.

3. On the Telesis workstation, automatically generate a back annotation file.

4. Transfer the back annotation file to the third party schematic capture system and update schematic.

Back Annotation File Format

The back annotation file is a free format file that relies on the concept of groups. Groups are collections of related records and are delimited by left and right parentheses. A sample extended BNF description of the file follows:

<back file> ::= <back header> <back record>+

<back header> ::= 'BACKANNOTATION' ',' <version #>

<version #> ::= '2'

<back record> ::= '(' <func des record>

[<part record>]

[<value record>]

<pin name record>+ ')'

<func des record> ::= 'FDES' ',' 'func des' ',' 'ref des' ','

<part record> ::= 'PART','part number'

<value record> ::= 'VAL', 'component value'

<pin name record> ::= 'PIN'',''pin name'',''pin number'

The files follow these conventions:

1. The first record in the file tells what the file contains and the version number of the file. In the example, the file contains back annotation information and is version 2.

2. The rest of the file is made of groups of records. A group exists for each function designator. A left parenthesis immediately precedes (on the same line) the function designator information. Then pin information follows - one record for each pin. There are two other records that may appear anywhere and are optional. These records contain part number and component value information. A right parenthesis closes out the group, but it appears on ITS OWN LINE - not immediately following. This inconsistency exist because right parenthesis might be part of the last field and hence must be on their own line to terminate the group.

3. Each record starts with a type field that indicates not only the contents of the record, but also the number of field in the record. Commas separate field. These are logical records, not physical records. A record may extend over multiple lines. Each field may be up to 255 bytes long.

The meaning of the Extended BNF is as follows:

- The first record says that this is a back annotation file that follows version 2 formatting conventions.

- The rest of the file consists of one or more groups of back annotation records. A back annotation record consists of a function designator record, followed by optional device type and value records, followed by one or more pin records.

- The 'FDES' type indicates that a function designator record follows. The field order is: function designator then reference designator. This record will always be the first record in a group.

  The function designator has meaning only to the schematic capture system and must be in the format specified in the enclosed document on the back annotation netlist format.

  The function designator is at most 8 characters long. This is due to constraints in the database. The back annotation file processor should be able to read fields of any length. When it comes from a Telesis system, it will be at most 8 characters long. Examples of legal function designators include 1-1-10, F1 and SHEET 1-1.

- The 'PART' type indicates that a part number (device type) follows. All symbols are required to have a part type when used in conjunction with a Telesis system. All physical CAD systems require such information to characterize the packaging considerations. The user might want his part number to read 74LS04 on his schematic, but the physical CAD system just needs 7404 to get the packaging because all 7404 series components have the same packaging.

- The 'VAL' type indicates that a component value record follows. These are most often used for capacitors or resistors.

- The 'PIN' type indicates that a pin record follows. The field order is pin name, pin number. Where pin name is the old value of the field and pin number is the new value. FDES, VAL, PART and PIN appear on the same line as their corresponding fields.

<u>An example back annotation file might be:</u>

```
BACKANNOTATION,2
(FDES,1-1,U1
PART,74LS00
PIN,A,1 ----
PIN,B,2      |(Pin Names assigned to pin numbers)
PIN,Y,3 ----
)
(FDES,102,U2
PART,74LS04
PIN,1,5 ---┐ (Pin Numbers swapped from 1,2 to 5,6)
PIN,2,6 ---┘
)
(FDES,1-3,R1
PART,RES1K
PIN,1,1 ---┐ (Pins 1,2 no change)
PIN,2,2 ---┘
VAL,1K
)
```

This file will be sorted by function designator. This will be accomplished by doing a simple lexicographical sort on the function designator.

Other Considerations

Unusual cases happen if symbols are deleted or modified on the PC board side using the Telesis PCB CAD package. A PC designer might add a function to the board that does not appear on the schematic. In the simplest case, this will be recognized because the function designator field will be blank. In a more complex case, he may add his own function designator. In this case, the function designator will not be in the format that the third party schematic capture system expects and is recognized as an added symbol. In the most complex case, a user creates a function designator that looks like function designators the third party system is expecting. In this case, the PC designer might successfully fool everybody and that's his problem.

If a pin has been added to a function (i.e. the schematic does not have a load line tied to power), the pin name field will be blank. If a pin or function has been deleted from the board, this may only be determined by examining the drawings, seeing what symbols have been back annotated and then flagging the rest. There is no way to determine from this file whether the network description has been changed when pins and components haven't been added or deleted, but merely had the interconnections changed. An additional problem occurs if the length of the new value of a field to be back annotated is longer than the original contents. Illegal symbols might be formed if the values overlap symbol boundaries. These cases must be detected by the back annotation program.

Example Errors

Some simple messages might include:

"<func des> can not be located on drawing <drawing file>."

The user has probably added a function designator on a Telesis workstation that follows the format expected, but cannot be located on the current sheet. This is an error.

"...is not associated with any of the drawing files in the directory."

The user has probably added a function designator on a Telesis workstation that does not follow the format expected ... it could be blank. This is an error.

"<pin name> can not be located on symbol <symbol number> on drawing <drawing file>.

Somehow the pin number can not be matched to the symbol on the current sheet. This is an error.

"No circuit designator on symbol <symbol number> on sheet <drawing file>."

The circuit designator field has been deleted from a symbol or was never there. This is an error.

"No part number on symbol <symbol number> on sheet <drawing needs>."

The component value has been deleted from a symbol or was never there. This is an error.

"Symbol <symbol number> on sheet <drawing file> has previously

Somehow, duplicate copies of a symbol got into the back annotation file. This is an error.

"Pin name <pin name> on symbol <symbol number> on sheet <drawing file> has previously been back annotated."

Somehow, duplicate copies of a pin name got into the back annotation file. This is an error.

"Part number on symbol <symbol number> on sheet <drawing file> has previously been back annotated. Only one PART record is allows per FDES."

Somehow, duplicate copies of the PART record got into the back annotation file. This is an error.

"Component value on symbol <symbol number> on sheet <drawing file> has previously been back annotated. Only one VALUE record is allowed per REFDES."

Somehow, duplicate copies of the VAL record got into the back annotation file. This is an error.

Symbol messages are detected on a symbol by symbol basis. Sheet messages may only be detected when an entire sheet has been processed. Recall that there is no offending record, so no record will appear in the logfile. The messages are:

"The following pins on symbol <symbol number> on sheet <drawing file> were never back annotated. <pin1> ... <pinn>."

Either pins were always unconnected or perhaps deleted by the PC board designer are not in the final board description. This is a warning. Pins with an explicit no-connect signal should not generate this message. This is a warning.

"The following symbols on sheet <drawing file> were never back annotated. <symbol number1> ... <symbol number>."

These symbols were probably deleted by the PC board designer. This is a warning.

"The part number field <part>, on symbol <symbol number> on sheet <drawing file> was never back annotated."

There should be a part number field in the back file. Something is wrong.

"The component value field <value>, on symbol <symbol number> on sheet <drawing file> was never back annotated."

The PCB designer deleted the component value field from the Telesis side.

Exceptions to Message

The following cases should probably not generate messages:

1. If there are unused pins on a symbol, these should not generate the "not back annotated pins" messages.

2. If a symbol has no pins (and is therefore not back annotated), it should not generate a "not back annotated symbol" message. Examples include ground symbols.

3. If a symbol has a signal attached to it named N/C, it should be considered to be an unused pin.

PRODUCT SPECIFICATION

PREPARING MAGNETIC TAPE FOR TELESIS INPUT

CONTENTS

INTRODUCTION

It is possible to write, onto magnetic tape, a text netlist, or any ASCII text file, created by another computer system, so the netlist can be read by the Telesis system. If properly formatted, this text can transmit the device and netlist information required for the Telesis LOAD TEXT NETLIST command. You can then create the NET-DATA-BASE required for placement and routing of PC board drawings on Telesis.

A Telesis system equipped with a magnetic tape drive can read and write 1/2" mylar, nine-track, odd-parity tapes at 800 and 1600 BPI (bits per inch) density.

TELESIS MAGNETIC TAPE TEXT FILE FORMAT

If you write a text data file from another computer onto nine-track magnetic tape in the format given below, the text can be read into the Telesis system using the ASCII TAPE IN commands.

The Telesis system reads and writes text files to and from magnetic tape in 80-byte physical ASCII records. Each record represents one line of the text file, padded with blanks to 80 characters. A line of a Telesis text file is considered to be all characters up to, but not including the next carriage return <15 octal> in the file. The text file on tape ends with two end-of-file tape marks after the last 80-byte record. A sample (Sample -A) is given in the next section.

SAMPLE OF A TELESIS TEXT FILE ON MAGNETIC TAPE
ORIGINAL FILE STORED ON TELESIS (OR OTHER) SYTEM

SAMPLE -A

```
(SAMPLE TELESIS NETLIST INPUT FILE)

$PACKAGES
DIP14 ! 74LSØØ ; U4
DIP14 ! 74LSØ4 ; U3
DIP16 ! 74LS138 ; U2
DIP16 ! 74LS163 ; U1
CAPRAD3ØØ ! CAP ; C1 C2 C3
CON1 ! CON1 ; J1
$NETS
VCC; J1.3 C3.1 C2.1 C1.1
GND; U4.7 U3.7 U2.8 U1.8 U3.3 J1.6 J1.5 U2.5 U2.4 U1.6 U1.5 U1.3 C3.2 C2.2,
 C1.2
RESET; J1.2 U1.1
CLOCK; U4.3 U1.2
ONEGEN; U3.4 U2.6 U1.1Ø U1.7 U1.4
LOAD; U3.2 U1.9
CNT3; U2.1 U1.11
CNT2; U2.2 U1.12
CNT1; U2.3 U1.13
CNTØ; J1.7 U1.14
TC; U3.1 U1.15
DECØ; U2.7
DEC1; U2.9
DEC2; U2.11 U2.1Ø
DEC4; J1.8 U2.12
DEC5; J1.9 U2.13
DEC6; J1.1Ø U2.14
DEC7; J1.11 U2.15
;U4.2 J1.1
+5V; U4.14 U3.14 U2.16 U1.16
$END
```

PRINTOUT OF TELESIS ASCII TAPE
EACH LOGICAL BLOCK IS A SINGLE PHYSICAL RECORD ON THE TAPE

SAMPLE -B

```
Dump of MTØ:
                  Logical block Ø,ØØØØØØ - Size 8Ø. bytes


ØØØØØØ    (   S   A   M   P   L   E       T   E   L   E   S   I   S
ØØØØ2Ø    N   E   T   L   I   S   T       I   N   P   U   T       F   I
ØØØØ4Ø    L   E   )
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                  Logical block Ø,ØØØØØ1 - Size 8Ø. bytes


ØØØØØØ
ØØØØ2Ø
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                  Logical block Ø,ØØØØØ2 - Size 8Ø. bytes


ØØØØØØ    $   P   A   C   K   A   G   E   S
ØØØØ2Ø
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                  Logical block Ø,ØØØØØ3 - Size 8Ø. bytes


ØØØØØØ    D   I   P   1   4       !       7   4   L   S   Ø   Ø       ;
ØØØØ2Ø        U   4
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                  Logical block Ø,ØØØØØ4 - Size 8Ø. bytes


ØØØØØØ    D   I   P   1   4       !       7   4   L   S   Ø   4       ;
ØØØØ2Ø        U   3
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ
```

SAMPLE -B (CONT.)

```
Dump of MTØ:
                    Logical block Ø,ØØØØØ5 - Size 8Ø. bytes

ØØØØØØ     D   I   P   1   6       !       7   4   L   S   1   3   8
ØØØØ2Ø     ;       U   2
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                    Logical block Ø,ØØØØØ6 - Size 8Ø. bytes

ØØØØØØ     D   I   P   1   6       !       7   4   L   S   1   6   3
ØØØØ2Ø     ;       U   1
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                    Logical block Ø,ØØØØØ7 - Size 8Ø. bytes

ØØØØØØ     C   A   P   R   A   D   3   Ø   Ø       !       C   A   P
ØØØØ2Ø     ;       C   1       C   2       C   3
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                    Logical block Ø,ØØØØ1Ø - Size 8Ø. bytes

ØØØØØØ     C   O   N   1       !       C   O   N   1       ;       J   1
ØØØØ2Ø
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                    Logical block Ø,ØØØØ11 - Size 8Ø. bytes

ØØØØØØ     $   N   E   T   S
ØØØØ2Ø
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                    Logical block Ø,ØØØØ12 - Size 8Ø. bytes

ØØØØØØ     V   C   C   ;       J   1   .   3       C   3   .   1       C
ØØØØ2Ø     2   .   1       C   1   .   1
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ
```

SAMPLE -B (CONT.)

```
Dump of MTØ:
                        Logical block Ø,ØØØØ13 - Size 8Ø. bytes


ØØØØØØ    G   N   D   ;       U   4   .   7       U   3   .   7       U
ØØØØ2Ø    2   .   8       U   1   .   8       U   3   .   3       J   1
ØØØØ4Ø    .   6       J   1   .   5       U   2   .   5       U   2   .
ØØØØ6Ø    4       U   1   .   6       U   1   .   5       U   1   .   3
ØØØ1ØØ        C   3   .   2       C   2   .   2   ,

Dump of MTØ:
                        Logical block Ø,ØØØØ14 - Size 8Ø. bytes


ØØØØØØ        C   1   .   2
ØØØØ2Ø
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                        Logical block Ø,ØØØØ15 - Size 8Ø. bytes


ØØØØØØ    R   E   S   E   T   ;       J   1   .   2       U   1   .   1
ØØØØ2Ø
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                        Logical block Ø,ØØØØ16 - Size 8Ø. bytes


ØØØØØØ    C   L   O   C   K   ;       U   4   .   3       U   1   .   2
ØØØØ2Ø
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                        Logical block Ø,ØØØØ17 - Size 8Ø. bytes


ØØØØØØ    O   N   E   G   E   N   ;       U   3   .   4       U   2   .
ØØØØ2Ø    6       U   1   .   1   Ø       U   1   .   7       U   1   .
ØØØØ4Ø    4
ØØØØ6Ø
ØØØ1ØØ

Dump of MTØ:
                        Logical block Ø,ØØØØ2Ø - Size 8Ø. bytes


ØØØØØØ    L   O   A   D   ;       U   3   .   2       U   1   .   9
ØØØØ2Ø
ØØØØ4Ø
ØØØØ6Ø
ØØØ1ØØ
```

PRINTOUT OF TELESIS ASCII TAPE IN BYTE-OCTAL FORMAT

Sample -C is the same netlist text file used in Sample -A and -B. The data is output in byte-octal form to show exactly what each ASCII character position looks like. Each logical block is a single physical record on the tape.

SAMPLE -C

```
Dump of MT0:
                    Logical block 0,000000 - Size 80. bytes

000000   050 123 101 115 120 114 105 040 124 105 114 105 123 111 123 040
000020   116 105 124 114 111 123 124 040 111 116 120 125 124 040 106 111
000040   114 105 051 040 040 040 040 040 040 040 040 040 040 040 040 040
000060   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000001 - Size 80. bytes

000000   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000020   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000040   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000002 - Size 80. bytes

000000   044 120 101 103 113 101 107 105 123 040 040 040 040 040 040 040
000020   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000040   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000003 - Size 80. bytes

000000   104 111 120 061 064 040 041 040 067 064 114 123 060 060 040 073
000020   040 125 064 040 040 040 040 040 040 040 040 040 040 040 040 040
000040   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000004 - Size 80. bytes

000000   104 111 120 061 064 040 041 040 067 064 114 123 060 064 040 073
000020   040 125 063 040 040 040 040 040 040 040 040 040 040 040 040 040
000040   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100   040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
```

SAMPLE -C (CONT.)

```
Dump of MTØ:
                    Logical block Ø,ØØØØØ5 - Size 8Ø. bytes


ØØØØØØ   1Ø4 111 12Ø Ø61 Ø66 Ø4Ø Ø41 Ø4Ø Ø67 Ø64 114 123 Ø61 Ø63 Ø7Ø Ø4Ø
ØØØØ2Ø   Ø73 Ø4Ø 125 Ø62 Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ4Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ6Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØ1ØØ   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø

Dump of MTØ:
                    Logical block Ø,ØØØØØ6 - Size 8Ø. bytes


ØØØØØØ   1Ø4 111 12Ø Ø61 Ø66 Ø4Ø Ø41 Ø4Ø Ø67 Ø64 114 123 Ø61 Ø66 Ø63 Ø4Ø
ØØØØ2Ø   Ø73 Ø4Ø 125 Ø61 Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ4Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ6Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØ1ØØ   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø

Dump of MTØ:
                    Logical block Ø,ØØØØØ7 - Size 8Ø. bytes


ØØØØØØ   1Ø3 1Ø1 12Ø 122 1Ø1 1Ø4 Ø63 Ø6Ø Ø6Ø Ø4Ø Ø41 Ø4Ø 1Ø3 1Ø1 12Ø Ø4Ø
ØØØØ2Ø   Ø73 Ø4Ø 1Ø3 Ø61 Ø4Ø 1Ø3 Ø62 Ø4Ø 1Ø3 Ø63 Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ4Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ6Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØ1ØØ   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø

Dump of MTØ:
                    Logical block Ø,ØØØØ1Ø - Size 8Ø. bytes


ØØØØØØ   1Ø3 117 116 Ø61 Ø4Ø Ø41 Ø4Ø 1Ø3 117 116 Ø61 Ø4Ø Ø73 Ø4Ø 112 Ø61
ØØØØ2Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ4Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ6Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØ1ØØ   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø

Dump of MTØ:
                    Logical block Ø,ØØØØ11 - Size 8Ø. bytes


ØØØØØØ   Ø44 116 1Ø5 124 123 Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ2Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ4Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ6Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØ1ØØ   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø

Dump of MTØ:
                    Logical block Ø,ØØØØ12 - Size 8Ø. bytes


ØØØØØØ   126 1Ø3 1Ø3 Ø73 Ø4Ø 112 Ø61 Ø56 Ø63 Ø4Ø 1Ø3 Ø63 Ø56 Ø61 Ø4Ø 1Ø3
ØØØØ2Ø   Ø62 Ø56 Ø61 Ø4Ø 1Ø3 Ø61 Ø56 Ø61 Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ4Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØØ6Ø   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
ØØØ1ØØ   Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø Ø4Ø
```

SAMPLE -C (CONT)

```
Dump of MT0:
                    Logical block 0,000013 - Size 80. bytes


000000  107 116 104 073 040 125 064 056 067 040 125 063 056 067 040 125
000020  062 056 070 040 125 061 056 070 040 125 063 056 063 040 112 061
000040  056 066 040 112 061 056 065 040 125 062 056 065 040 125 062 056
000060  064 040 125 061 056 066 040 125 061 056 065 040 125 061 056 063
000100  040 103 063 056 062 040 103 062 056 062 054 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000014 - Size 80. bytes


000000  040 103 061 056 062 040 040 040 040 040 040 040 040 040 040 040
000020  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000040  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000015 - Size 80. bytes


000000  122 105 123 105 124 073 040 112 061 056 062 040 125 061 056 061
000020  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000040  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000016 - Size 80. bytes


000000  103 114 117 103 113 073 040 125 064 056 063 040 125 061 056 062
000020  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000040  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000017 - Size 80. bytes


000000  117 116 105 107 105 116 073 040 125 063 056 064 040 125 062 056
000020  066 040 125 061 056 061 060 040 125 061 056 067 040 125 061 056
000040  064 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040

Dump of MT0:
                    Logical block 0,000020 - Size 80. bytes


000000  114 117 101 104 073 040 125 063 056 062 040 125 061 056 071 040
000020  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000040  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000060  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
000100  040 040 040 040 040 040 040 040 040 040 040 040 040 040 040 040
```

2. If the third party CAE vendor reads Back-file for back annotation, then a back annotation file is created using the **[CREATE TEXT BACK ANNOTATION FILE]**.

Finally, the netlist file or back annotation file (back file) is transferred to the third party system using the process outlined earlier.

On the third party CAE system, the new netlist or back annotation file is read and the schematic is updated (refer to Telesis Product Specification: Telesis Back Annotation File).

THIRD PARTY CAE VENDORS

The following is a list of Third Party CAE vendors who currently have ~~completed~~ interfaces to the Telesis system.

FUTURENET - DASH 2 and DASH 3 (Netlist In, Back Annotation); IBM/PC-based system. DASH 4 (Netlist In - DASH 3 format, Back Annotation currently in process). Telesis and FutureNet are working on a long-term strategy to support DASH 4 and additional new product offerings from FutureNet.

VALID LOGIC - Netlist In, Back Annotation (Back Annotation File); non-IBM /PC-based system.

CASE TECHNOLOGY - Netlist In (Back Annotation File); IBM/PC-based system

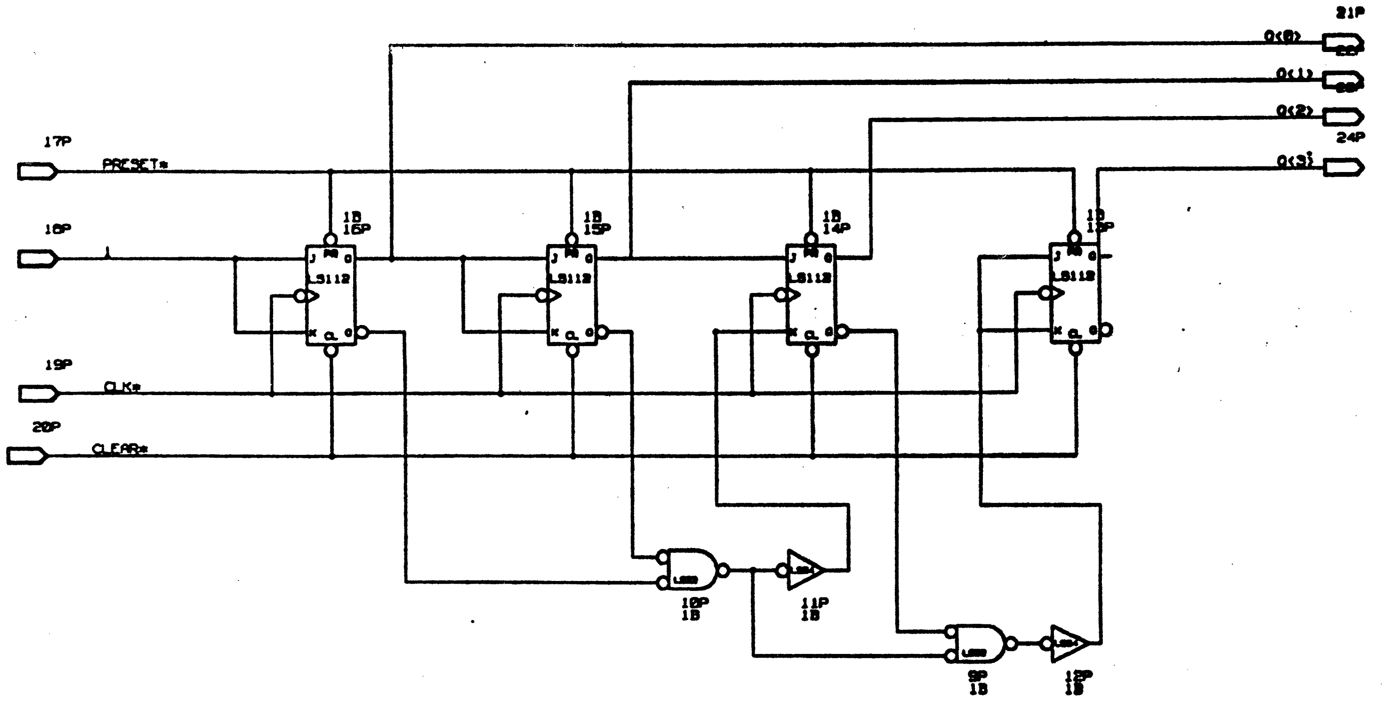
DAISY SYSTEMS - DNX, Netlist In, Back Annotation; non-IBM/PC-based system.

VIEW LOGIC - Netlist In, Back Annotation (Netlist File); IBM/PC-based system.

MENTOR GRAPHICS - Netlist In, No Back Annotation; non-IBM/PC-based system.

The following vendors have interfaces that are still in progress: **HEWLETT PACKARD, XEROX (VERSATEC), SILVAR LISCO, VISIONICS CORP., DOUBLE T SERVICES.**

```
**********************************************************
*                                                        *
*                      VALID LOGIC                       *
*                      -----------                       *
*                                                        *
*     TELESIS NETLIST CREATED FROM VALID SCHEMATIC       *
*                                                        *
**********************************************************


$WASIS

( FUNCTIONS SECTION )

$FUNCTIONS
74LS04 ; F1 F2
74LS112 ; F3 F4 F5 F6
74LS32 ; F7 F8


( NETS SECTION )

$NETS
CLEARL; U3:F5.14 U3:F6.15 U4:F3.14 U4:F4.15
CLKL: U3:F6.1 U3:F5.13 U4:F4.1 U4:F3.13
ONE; U4:F2.2 U4:F4.3
PRESETL; U3:F6.4 U3:F5.10 U4:F4.4 U4:F3.10
Q0; U4:F4.5 U4:F3.11 U4:F3.12
Q1; U3:F6.3 U4:F3.9
UN1LS0411PA0; U1:F8.8 U1:F7.13 U2:F1.13
UN1LS0411PY0: U2:F1.12 U3:F6.2
UN1LS0412PA0; U1:F7.11 U2:F2.11
UN1LS0412PY0: U2:F2.10 U3:F5.11 U3:F5.12
UN1LS11214PQ0; U1:F7.12 U3:F6.6
UN1LS11215PQ0; U1:F8.9 U4:F3.7
UN1LS11216PQ0: U1:F8.10 U4:F4.6
$END
```



DEFINE
X_FIRST=0
X_STEP=SIZE

DRAWING.
LAST_MODIFIED=Wed Aug 15 16:15:03 1984

DESIGN:
design1

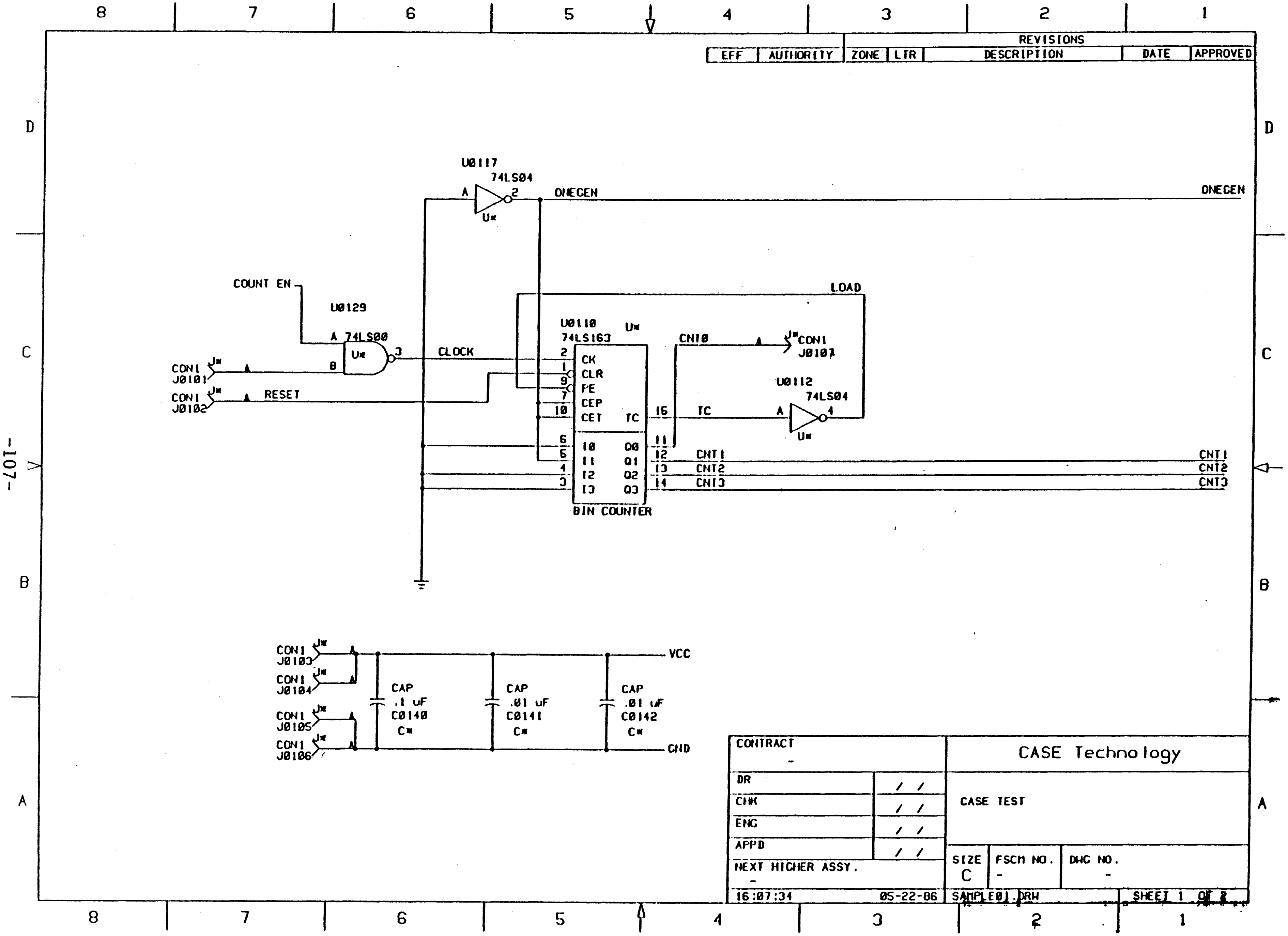
THIS DOCUMENT CONTAINS INFORMATION PROPRIETARY TO VALID LOGIC SYSTEMS INCORPORATED (VALID). USE OR DISCLOSURE WITHOUT THE WRITTEN PERMISSION OF AN OFFICER OF VALID IS EXPRESSLY FORBIDDEN.
COPYRIGHT (C) VALID 1982

VALID
LOGIC SYSTEMS INCORPORATED

TITLE: telesis test1
ENGINEER: swaraba nath
DATE: 08.15.84
PAGE: 1 OF 1

```
**********************************************************************
*                                                                    *
*                         CASE TECHNOLOGY                            *
*                         ---------------                            *
*                                                                    *
*           TELESIS NETLIST CREATED FROM CASE SCHEMATIC              *
*                                                                    *
*                       (2 sheet schematic)                          *
*                                                                    *
**********************************************************************


$WASIS
$FUNCTIONS
CAP:1-1-40 1-1-41 1-1-42
CON1;1-1-1 1-1-2 1-1-3 1-1-4 1-1-5 1-1-6 1-1-7 1-2-2 1-2-3 1-2-4 1-2-5
74LS163:1-1-10
74LS04;1-1-12 1-1-17
74LS00:1-1-29
74LS138:1-2-1
$NETS
CLOCK: U*:1-1-10.2 U*:1-1-29.Y
CNT0: J*:1-1-7.A U*:1-1-10.14
CNT1: U*:1-1-10.13 U*:1-2-1.3
CNT2: U*:1-1-10.12 U*:1-2-1.2
CNT3: U*:1-1-10.11 U*:1-2-1.1
COUNTEN; U*:1-1-29.A
DEC0; U*:1-2-1.7
DEC1: U*:1-2-1.9
DEC2; U*:1-2-1.10 U*:1-2-1.11
DEC4; J*:1-2-2.A U*:1-2-1.12
DEC5; J*:1-2-3.A U*:1-2-1.13
DEC6; J*:1-2-4.A U*:1-2-1.14
DEC7: J*:1-2-5.A U*:1-2-1.15
GND; C*:1-1-40.2 C*:1-1-41.2 C*:1-1-42.2 J*:1-1-5.A J*:1-1-6.A U*:1-1-10.3,
U*:1-1-10.5 U*:1-1-10.6 U*:1-1-17.A U*:1-2-1.4 U*:1-2-1.5
J0101%A/M; J*:1-1-1.A U*:1-1-29.B
LOAD: U*:1-1-10.9 U*:1-1-12.Y
ONEGEN; U*:1-1-10.10 U*:1-1-10.4 U*:1-1-10.7 U*:1-1-17.Y U*:1-2-1.6
RESET: J*:1-1-2.A U*:1-1-10.1
TC: U*:1-1-10.15 U*:1-1-12.A
VCC: C*:1-1-40.1 C*:1-1-41.1 C*:1-1-42.1 J*:1-1-3.A C*:1-1-4.A
$END
```

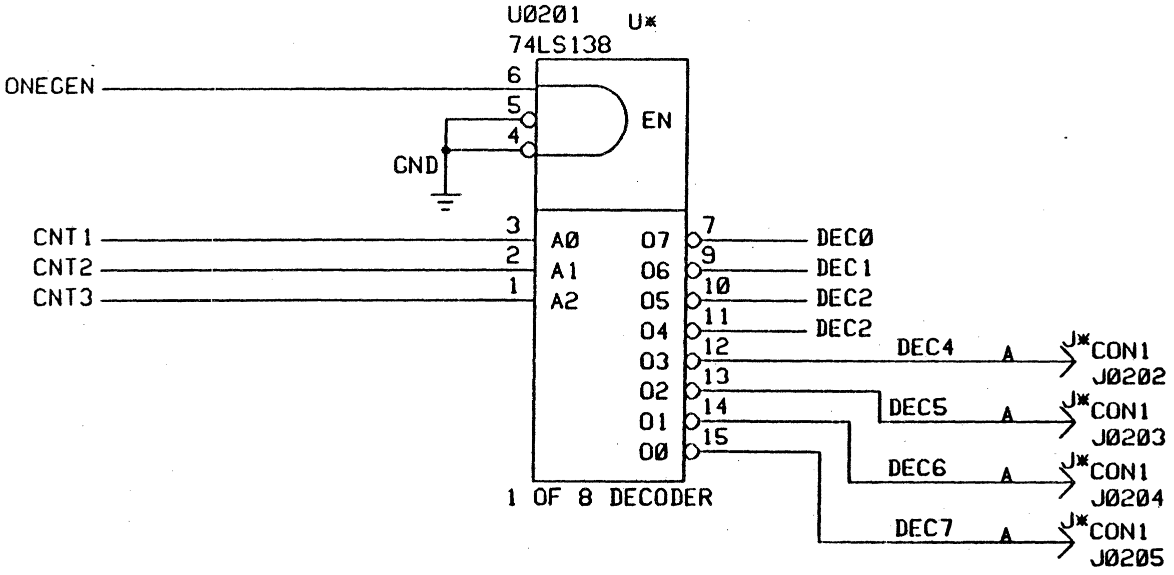
Sample Schematic
Sheet 1 of 2



CONTRACT -		CASE Technology
DR	/ /	CASE TEST
CHK	/ /	
ENG	/ /	
APPD	/ /	
NEXT HIGHER ASSY. -		SIZE C · FSCM NO. - · DWG NO. -
16:07:34	05-22-86	SAMPLE01.DRW · SHEET 1 OF 2

Sample Schematic
Sheet 2 of 2

				REVISIONS		
EFF	AUTHORITY	ZONE	LTR	DESCRIPTION	DATE	APPROVED



CONTRACT -	FIRM NAME
DR / /	CASE TEST
CHK / /	
ENG / /	
APPD / /	SIZE B · FSCM NO. - · DWG NO. -
NEXT HIGHER ASSY.	
16:16:58 05-22-86	SAMPLE02.DRW · SHEET 2 OF 2

LOADING INSTRUCTIONS

AND

TELESIS THIRD PARTY NETLIST INTERFACE

P/N 7600035-001 REV A

JULY, 1986

CONTENTS:

I. MEDIA

1. Two floppy diskettes (8-inch) containing the upgrade to the EDA-3000/3100 software. These will be loaded with the "SELF LOAD" feature. See the comments in Section V to determine whether you wish to load the software immediately. The part numbers are 8000074-001 and 8000131-001.

2. One 5-1/4 inch floppy diskette containing IBM software.

II. LOADING INSTRUCTIONS - GENERAL

NOTE: PLEASE READ ALL OF THESE NOTES BEFORE LOADING YOUR SOFTWARE! YOU MIGHT NOT WANT TO LOAD YOUR SOFTWARE IMMEDIATELY.

To load this software, the following loading steps must be performed:

1. Load the 8-inch floppy diskettes onto the Telesis workstation following the instructions in Section III. The Telesis station must be REBOOTED.

2. Do not load previous revisions of the software. This release supercedes all other revisions.

3. Load the 5-1/4 inch floppy diskette onto the IBM Personal Computer following the instructions in Section IV.

III. LOADING INSTRUCTIONS - EDA-3000/3100

CRITERIA

1. The EDA-3000 system should be at the OLD PROJECT/NEW PROJECT menu page — with no user activity being performed.

2. TELESIS software EDA-3000/3100 must exist on the system.

STEP 1. Insert Volume 1, 8-inch floppy in the Telesis floppy drive.

STEP 2. From the menu, pick the menubox <SELF LOAD>.

STEP 3. You will then be prompted on the function screen to:

INSERT THE FLOPPY AND HIT PAGE -> TO CONTINUE

<PLEASE ENSURE THAT THE FLOPPY IS INSERTED IN THE DRIVE.>

STEP 4. After PAGE -> is hit, the floppy will continue to load the appropriate software onto the system without further operator assistance.

STEP 5. A message will be issued at the bottom of the function screen to indicate the completion of loading. Remove the floppy from the drive.

STEP 6. Repeat Steps 1-5 for second volume.

STEP 7. IT WILL BE NECESSARY TO REBOOT THE SYSTEM.

IV. IBM-PC LOADING INSTRUCTIONS

STEP 1. Insert one 5-inch floppy into the IBM Personal Computer floppy drive.

STEP 2. Make sure that drive "C" is your default drive by typing: C:

STEP 3. Position yourself in whatever directory you wish to contain the software. We typically make a single directory called DCS. It is up to you. See your DOS manual on how to make directories and change your default directories. An example is:

```
C> MKDIR DCS
C> CD   DCS
```

STEP 4. Copy the files from the floppy to the appropriate directory on harddisk by typing:

```
C> COPY A:*.*
```

STEP 5. Remove the floppy from the drive.

Introduction to Simulation

Kyle Krauss
Product Marketing
Support Specialist
Telesis, Inc

User's Group

1986 Telesis

Introduction to Simulation

What is Simulation?

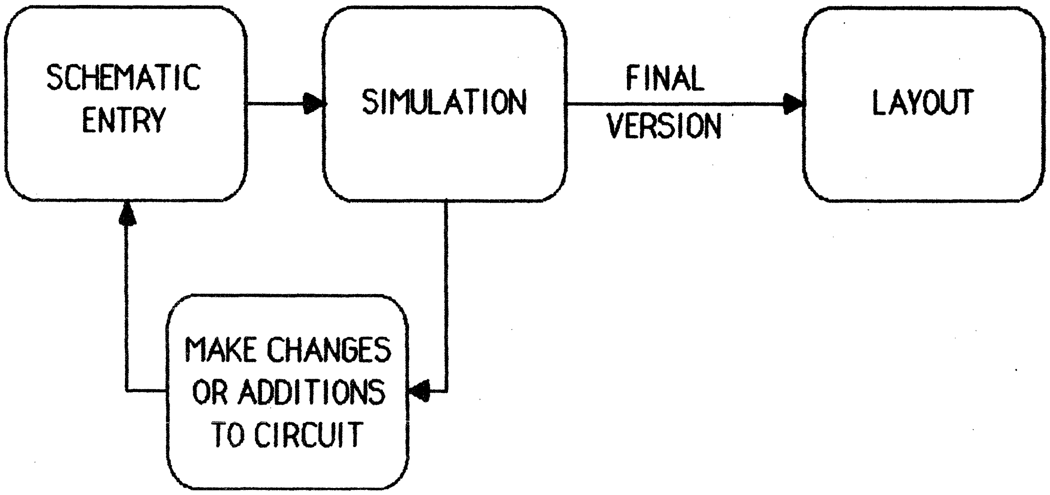
Simulation is a software tool that is used to verify proper logical and timing operation of circuit board designs. Simulation can also be used to determine how the circuit will respond to typical faulty conditions on the board.

When is Simulation Used?

Logic simulators are used exclusively for Integrated Circuit and board designs using digital logic. Analog circuit simulation requires the use of a totally different simulator. That simulator is usually a form of SPICE, a public domain circuit analysis program.

Where does Simulation Happen In The Board Design Process?

The simulation tools will be used by the development engineer prior to submitting the board for layout. The following diagram illustrates the process:



The engineer will typically make several passes through the simulator because he finds errors in his design or he adds another section to the design. Once he feels he has verified the operation of the board, it will go to layout. At this time, he may or may not build a hardware prototype.

Why Use Simulation?

-By using simulation, the engineer can verify that the circuit is correct logically before any time is spent on building a prototype. It also eliminates the unknowns of bad devices and wiring errors in the prototype. Considerable time can be saved in the prototype stage by using simulation.

-Simulation allows the engineer to verify the circuit's performance over the entire timing window of each component on the board and their combined effects. A hardware prototype only verifies the circuit for the given conditions on that board. Prototype verification alone can allow timing design errors to pass through to manufacturing where they are difficult and time consuming to diagnose. Timing verification during the design phase is becoming increasingly important with the faster speeds of new board designs.

-The data that is generated during simulation in the design phase can be used for testing of the board in manufacturing. Use of this data will greatly reduce the time spent generating a test program in manufacturing.

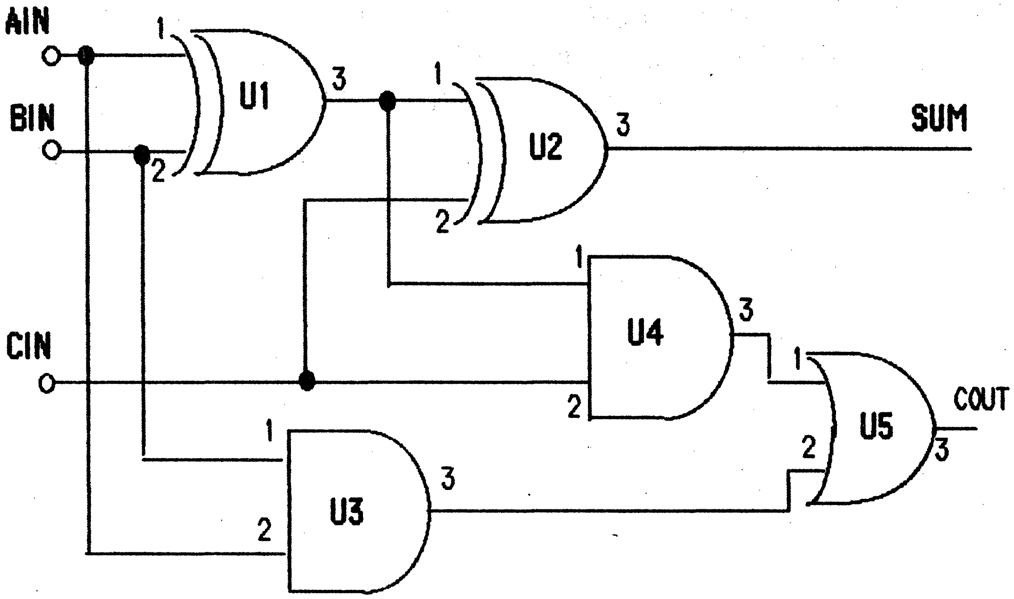
-The time to market for a new product can be shortened by reducing the time spent on developing hardware prototypes as well as making use of simulation data for test generation.

-Higher product quality can be achieved by using simulation to eliminate timing related design errors which often are not detected until the product reachs the field.

How Does A Simulator Work?

There are three major components to a simulation product. They are logic simulation, timing analysis and fault simulation.

Logic Simulation -Logic simulation predicts how a circuit will operate assuming all the devices on the board are good. A set of input values, called stimulus, is specified. The simulator takes that stimulus and moves it through the circuit. The operation of the individual components, or how they propagate their inputs to their outputs, is defined in the models of the components. For each input pattern, the resulting outputs are recorded. Figure 1 shows a simple circuit and the simulation results. The stimulus is applied at AIN, BIN and CIN and the results at the outputs SUM and COUT are recorded.



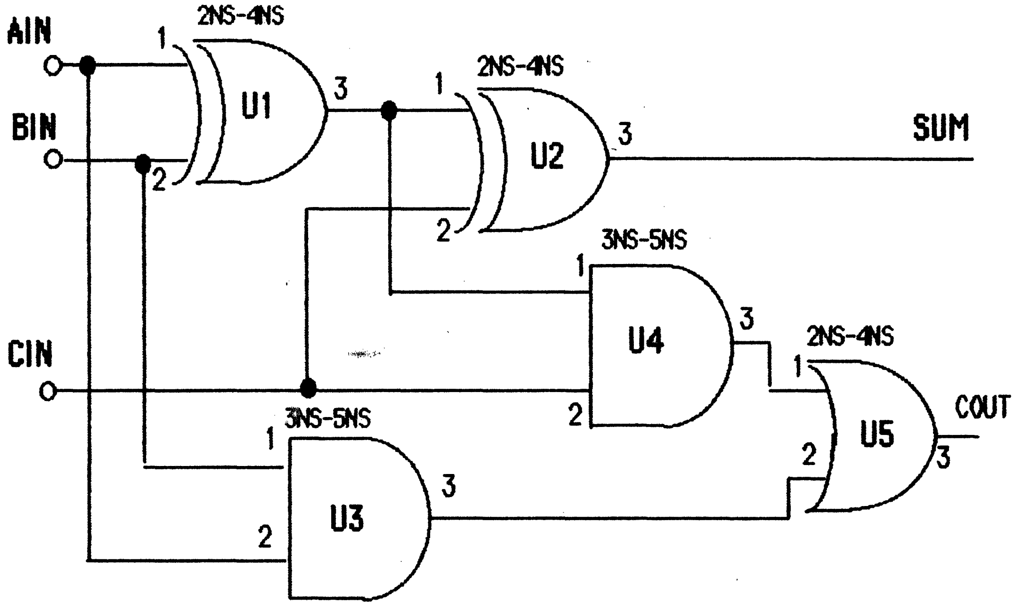
Stimulus

AIN	BIN	CIN	U1.3	U3 .3	U4.3	SUM	COUT
1	0	1	1	0	1	1	1

Figure 1
Logic Simulation

Logic simulation is the first component of simulation the engineer will use as he wants to verify that his design is logically correct before he proceeds any further. A major factor effecting the accuracy of the simulation is the number of logic states used for simulation. A logic state is defined as the combination of the voltage level (1,0,Z,X) on that node and the strength (active, passive, floating) of that level. The more states a simulator can handle, the more accuracy it will have. The trade-off is processing time. More states requires longer simulation time.

Timing Analysis - Timing analysis is used to verify the circuit will perform correctly over the timing range of the components. The operation is the same as for logic simulation, but this time the simulator also keeps track of how long it takes a signal to pass from the input of a device to it's output, called the propagation delay of the device. The circuit can be correct logically, but have timing problems due to the combined effects of the propagation delays of connected devices. The engineer will use timing analysis after he has verified the circuit is logically correct. Two methods are used today. Unit delay timing analysis which gives every device the same timing and MIN/MAX timing which allows every device to have it's own timing. MIN/MAX provides more accurate results. Figure 2 shows the same circuit with the simulation results for MIN/MAX timing analysis. Notice the outputs changes are recorded as happening within a window rather than a fixed value.

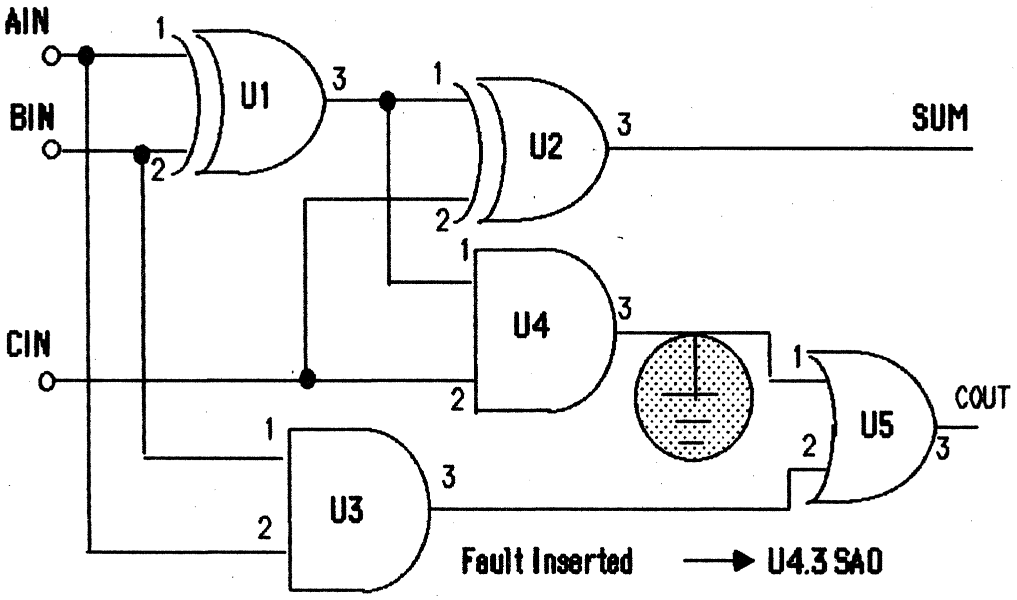


stimulus

AIN	BIN	CIN	U1.3	U3.3	U4.3	SUM	COUT
1	0	1	▨ 2 4	▨ 3 5	▨ 5 9	▨ 4 8	▨ 7 13

Figure 2
Timing Analysis

Fault Simulation - Fault simulation is the process of holding a node in the circuit, called inserting a fault, in a high (1) or low (0) state and running logic simulation again to see what effect that fault has on the outputs of the circuit. If the outputs are different, the fault is considered detected. A node held high is called stuck-at-one (SA1) and a node held low is called stuck-at-zero (SA0). Both of these types of faults are inserted on each node of the circuit. The number of faults detected is an indication of the effectiveness of the stimulus in exercising the circuit. The higher the percentage of detection means a higher percentage of the circuit was exercised. Engineers typically look for 90% to 100% fault detection. Figure 3 shows the same circuit with a fault inserted (U4.3 SA0) and the resulting change in the output states for the same stimulus.



Stimulus

AIN	BIN	CIN	U1.3	U3 .3	U4.3	SUM	COUT
1	0	1	1	0	0	1	0

Figure 3
Fault Simulation

Inputs To the Simulator

There are three essential inputs to the simulator for processing. They are:

Netlist - description of the connections in the circuit and the device types. Manually entered by the engineer or down-loaded from a schematic capture system.

Stimulus - the set of input patterns that will be used to exercise the circuit. Manually generated by the engineer.

Libraries - contain descriptions of the operation (models) of the devices. Libraries are furnished as part of the simulator package. Users will create their own for custom devices. The accuracy of the simulation is very dependent upon accurate models. A more detailed description of models follows.

Models

Models are used to define the operation of an element to the simulator. Most simulators today allow element operation to be defined in a variety of methods, called levels. Listed below are those levels and a brief description:

Level	**Description**
Switch	elements are defined in terms of transistors. Seldom used for board simulations. Only way to accurately simulate MOS IC designs.
Gate	elements are defined as SSI/MSI gates. Examples are AND gates, OR gates, D flip-flop.
Functional	elements are defined as interconnected SSI/MSI gates.
Behavioral	elements are defined in terms of relationship of inputs to outputs. A high level language ("C") is used for the description.
Physical	actual device is used in the simulation. Also called hardware modeling.

The libraries provided with the simulator typically contain switch and gate level models. These are also called primitives. The engineer will use combinations of the gate level primitives to make functional level models of more complex devices. Behavioral models are used for devices who complexity is too high to describe in terms of gates.

Hardware models are used for microprocessors and their support devices. The complexity of these devices makes software models nearly impossible. Hardware modeling allows a device to physically be placed in the circuit for simulation when no software model is available. The device is plugged into a cartridge which the simulator can talk to. When the simulator reaches a point where stimulus has to be propagated through the device, it does so by applying the stimulus and capturing the results in hardware. The simulation can now continue propagating the stimulus with software models for the other devices. Figure 4 shows how the different levels of models apply to a circuit.

Simulators that can work with different levels of models at the same time are called multi-mode simulators. This should not be confused with mixed-mode simulation which means to do analog and digital simulation at the same time.

It should be noted that an engineer using a gate array in a board design will typically have a gate level or functional level model of that device which was developed when the gate array was designed.

Outputs From the Simulator

The results of the good circuit simulation are recorded in a file. Most simulators can also display the signal activity in a waveform display on the CRT. The display looks very much like a logic analyzer display. This is the most useful tool for the engineer to compare the simulation results with what he expected the circuit to do. The simulator does not point out any errors in the circuit. It is up to the engineer to determine, using the waveforms, where the circuit performed differently than he expected. He looks for such things as areas where unknown states resulted on the outputs, bus nodes being driven by more than one device simultaneously (bus contention) or violations of a device's timing requirements.

Fault simulation produces report files which indicate what percentage of the faults are detected and what faults are not detected. If the detected percentage is low (<90%), it indicates the circuit is not being exercised sufficiently. The engineer will usually change his stimulus and repeat the process until he is satisfied with the fault detection. He may also choose to change part of the circuit to make it more testable. Some fault simulators also record the output states for each fault as it is inserted (fault dictionary). This information will be extremely useful in developing a test program for the board.

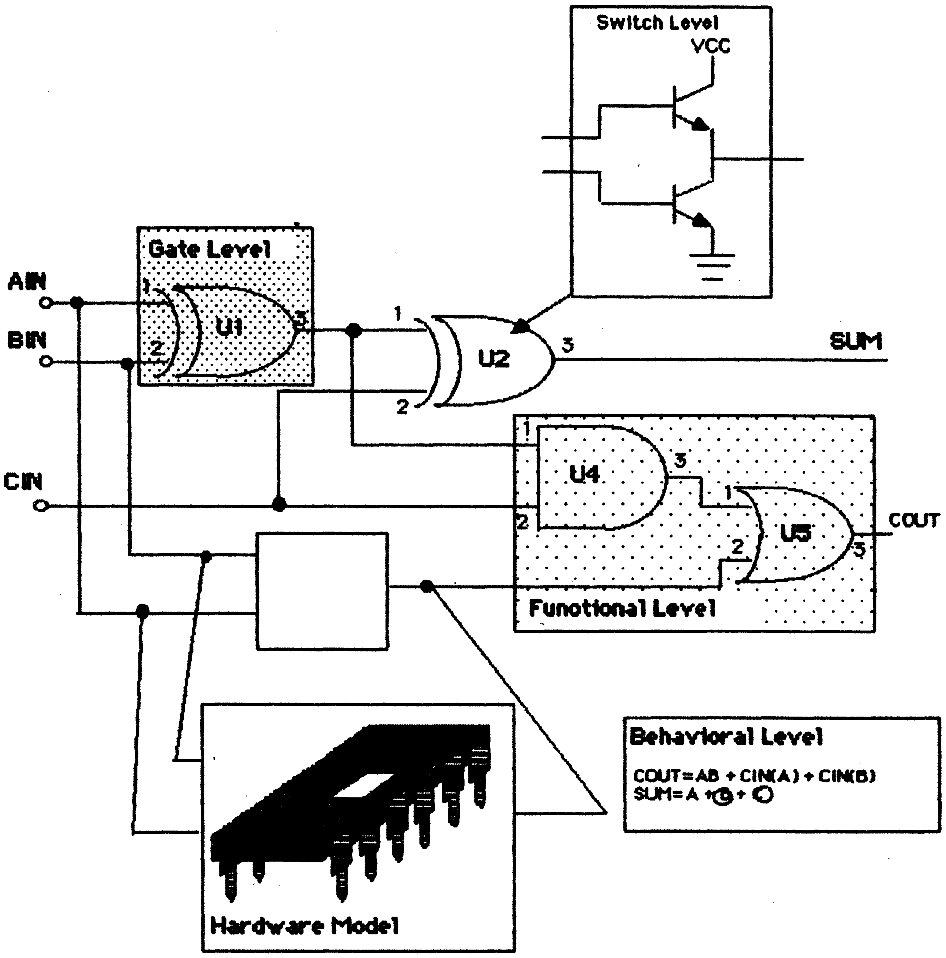


Figure 4
Levels of Models

Simulators On the Market

There are four major simulation products on the market:

Name	Developed By	Typical Use	Available From
CADAT	HHB Systems	Board and IC designs	many OEM's
HILO	GenRad	IC design	many OEM's
LASAR	Teradyne	Board test generation	Valid Logic
TEGAS	Calma (GE)	Military	Calma

What is Telesis Offering for Simulation?

Telesis is now offering the CADAT family of simulation products. CADAT is an integrated logic, timing and fault simulator. The configurations that are available are:

Personal CADAT-CADAT running on the IBM PC/AT. This can be used for board designs with up to approximately 150 IC's. Fault simulation cannot be performed.

CADAT on the 620-For customers with larger boards or the need to do fault simulation but no hardware modeling.

CADAT Hardware Modeler-For customers designing boards containing microprocessors and support devices. The hardware modeler is based on a Sun processor and also runs CADAT. The modeler will handle up to 30 cartridges, allowing 30 different devices to be modeled. The modeler must be configured for the customer's application. The following information must be determined for the configuration:

- which devices will be modeled.
- how many driver/sensor boards are required for those devices.

The hardware modeler can be shared between users. It is not necessary to purchase a modeler for every user.

Summary

Logic Simulation provides the engineer with an analysis tool which can give him a higher level of confidence that his design will function correctly over the entire operating range of the board. Simulation is becoming more prevelent for two reasons. First, as more companies begin using design capture systems, the engineer has a netlist of the design available to him which facilitates the use of the simulator. Secondly, as the operating speeds of boards increase, timing related design errors are a bigger concern. Simulation is the only practical method for the engineer to find these errors before the board is introduced into manfacturing.

Thermal Analysis

Kyle Krauss
Product Marketing
Support Specialist
Telesis, Inc

Thermal Analysis

Introduction

Thermal analysis is becoming increasingly important for many companies as different device technologies and packaging styles are incorporated into new board designs. Many approaches have been taken for thermal analysis ranging from "best guess" to manual calculations to detailed analysis on large mainframe computers. This paper looks at some of the basic components of thermal analysis and what capabilities are offered by the Telesis Thermal Analysis package.

Why Do Thermal Analysis?

Thermal analysis has traditionally been used predominantly be military suppliers to calculate the reliability figures for printed circuit boards. New technologies being used on today's boards will make thermal analysis essential for other manufacturers as well. Those technologies include:

High Speed Devices-Emitter Coupled Logic (ECL) is finding wider spread use on new board designs because of its high operating speed capabilities. ECL devices consume tremendous amounts of power and greatly increase the operating temperatures of ECL based systems.

Surface Mount Devices-Because SMD's must rely on a solder joint at the surface of the board for good electrical contact, they are extremely sensitive to the expansion and contraction of the board do the heat. It is essential that the temperature range of the board in the system is predicted and controlled to avoid stressing the solder joints on the printed circuit board.

High Density Packaging-The use of gate arrays on boards results in much higher circuit density in a single package. A packaging style must be chosen to assure the power generated by the device can be adequately dissipated. By having the thermal information for a board, an engineer can make better decisions concerning the packaging material (plastic vs. ceramic), style of leads (through-hole vs. SMD) and placement of the components.

High Speed Circuits- The operating speeds of new boards are reaching levels where the performance can be effected by the temperature of the board. The engineer will want to know the thermal profile of the board so the most critical timing sections of the board can be located in a cooler section of the board.

Information Required for Thermal Analysis

To perform thermal analysis on a board, information is needed not only about the board and components, but the environment the board will be placed in as well. Specific information includes:

BOARD

-Dimensions of the board (extracted from the drawing database).
-total power dissipation of the board.
-ability of the board to conduct heat (thermal conductivity). This is effected by the material of the board and the number of layers.
-special heat dissipation at the edges of the board (edge conductivity).

COMPONENTS

-dissipation of the component (found in the databook)
-resistivity of the component (found in the databook)
-location on the board (extracted from the drawing database)
-type of component (extracted from the drawing database)
-heatsinks

ENVIRONMENT

-direction of airflow
-velocity of airflow
-board to board spacing
-board orientation
-adjacent board temperature

Temperature Calculations

Finite difference analysis is used to calculate the temperatures on the board. Finite difference analysis is best suited for thermal analysis because it can handle fluid flow problems better than finite element and it requires much less processing time. The calculations are done for all heat transfer mechanisms which are:

Radiation-transfer of heat with no carrying medium (board to board).

Conduction-transfer of heat through physical contact (device to board)

Natural Convection-heat transferred through air (board to air).

Forced Convection-heat transferred through air with the aid of a fan. If forced convection is used, it will be the dominate mode of heat transfer.

Reports and Plots

After the temperature calculations have been run, an isotherm plot can be displayed on the screen with the board and component outlines. The plot shows the temperature characteristics using contours. The number of contours is set by the user. The display allows the user to quickly see what the maximum temperature of the board is and where the hot spots are located. Figure 1 shows an example isotherm display.

A short report can be generated summarizing the thermal data for the board as shown in Figure 2.

A detailed report can be generated with the case temperature and junction temperature of each device sorted by temperatures. The user can easily determine if any component's specifications are being violated. Figure 3 shows an example of the detailed report.

Finally, a sensitive analysis can also be generated which shows what effect a change in a parameter, such as total power dissipation, will have on the overall board temperature.

"What If" Analysis

The real usefulness of a thermal analysis package comes from letting the engineer make changes to the board or environment and quickly see the effects of that change on the board temperatures. If the initial analysis indicated the board was operating too hot, there are several possible methods of lowering the board temperature. The engineer might want to look at what happens if:

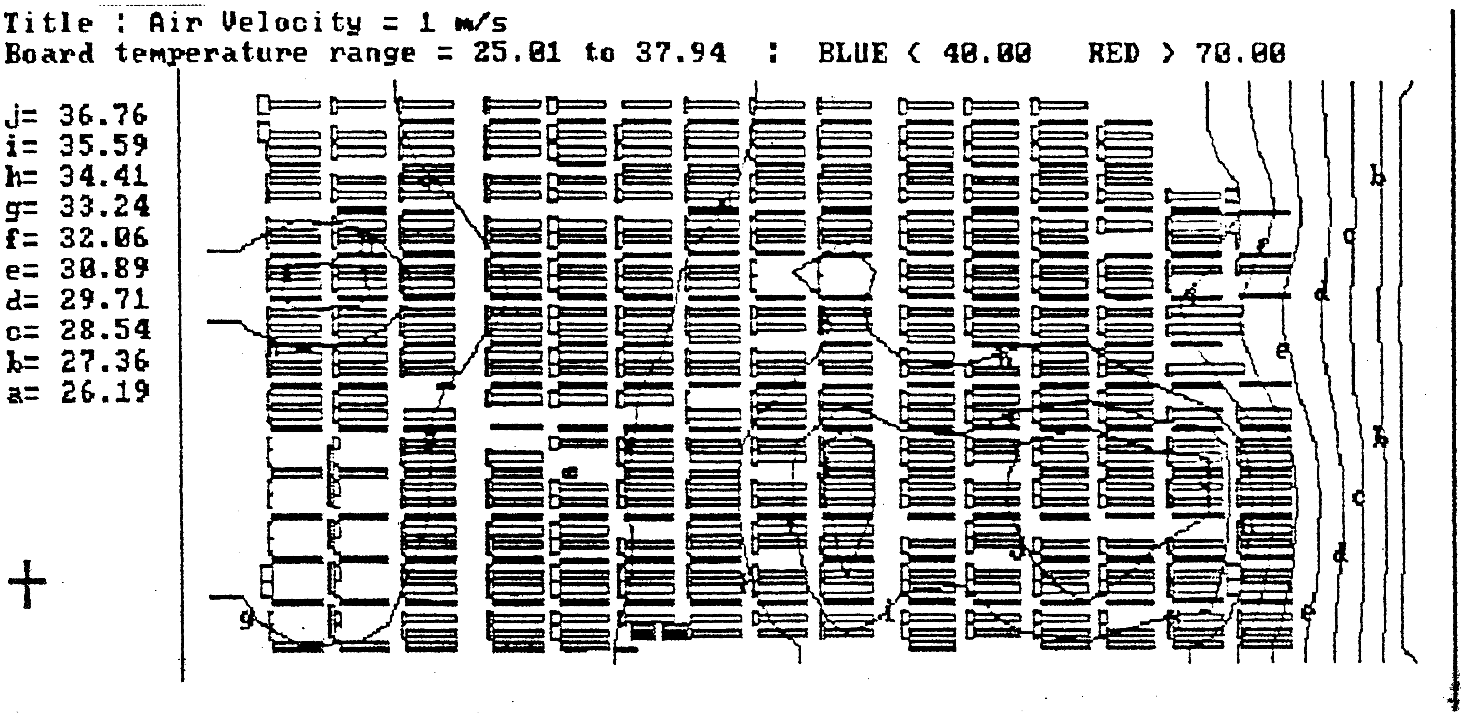


FIGURE 1
ISOTHERM PLOT

```
                     Thermal Design Simulation Report
Air Velocity = 1 m/s



                          SUMMARY OF OUTPUT DATA
                          ======================

          ITEM                                          DATA
          ====                                          ====
Outlet ambient air temperature                          26.5 (C)
Fan/blower induced flow                                 21.5 (CFM)
Maximum PCB temperature                                37.94 (C)
  at X Coordinate                                      36.83 (CM)
  and Y Coordinate                                     29.46 (CM)
Minimum PCB temperature                                25.01 (C)
  at X Coordinate                                      50.80 (CM)
  and Y Coordinate                                     15.24 (CM)
Component with maximum temperature                       16J
  Component case temperature                           77.10 (C)
    at X Coordinate                                     0.00 (CM)
    and Y Coordinate                                    0.00 (CM)
Component with minimum temperature                      RPR9
  Component case temperature                           31.41 (C)
    at X Coordinate                                     0.00 (CM)
    and Y Coordinate                                    0.00 (CM)

                            SUMMARY STATISTICS
                            ==================

          ITEM                                          DATA
          ====                                          ====
Average board temperature                               33.1 (C)
Deviation in board temperature                           2.6 (C)
Total number of components                               993
Average Case temperature                               35.57 (C)
Deviation in Case temperature                           7.34 (C)
Average Junction temperature                           36.31 (C)
Deviation in Junction temperature                      10.90 (C)
```

FIGURE 2
SUMMARY REPORT

Thermal Design Simulation Report

```
Air Velocity = 1 m/s

                              Component List
                        Sorted by Junc Temperature

                                                    Junct.   Case      X       Y
  RefDes            Name                  Power     Temp.    Temp.   Coord.  Coord.
                                          Watts   Degrees-Celsius   Centimeters

  16J          10H016                      0.61     97.8     77.1    26.80   15.43

  17J          10H016                      0.61     97.8     77.1    26.80   14.41

  18J          10H016                      0.61     97.6     76.9    26.80   12.38

  18M          10H016                      0.61     97.6     76.8    35.94   12.38

  19M          10H016                      0.61     97.5     76.8    35.94   11.37

  17M          10H016                      0.61     97.5     76.8    35.94   14.41

  16M          10H016                      0.61     97.5     76.8    35.94   15.43

  19J          10H016                      0.61     97.4     76.7    26.80   11.37

  21M          10H016                      0.61     97.2     76.5    35.94    8.32

  21J          10H016                      0.61     97.1     76.3    26.30    8.32

  16P          10H016                      0.61     97.0     76.3    41.53   12.38

  17P          10H016                      0.61     97.0     76.3    41.53   14.41

  16P          10H016                      0.61     97.0     76.3    41.53   15.43

  19P          10H016                      0.61     97.0     76.2    41.53   11.37

  22M          10H016                      0.61     96.9     76.2    35.94    6.29

  22J          10H016                      0.61     96.8     76.0    26.80    6.29

  21P          10H016                      0.61     96.6     75.9    41.53    8.32

  23M          10H016                      0.61     96.6     75.8    35.94    5.27

  23J          10H016                      0.61     96.5     75.7    26.80    5.27

  22P          10H016                      0.61     96.4     75.6    41.53    6.29

  9A           10H016                      0.61     96.1     75.4     3.68   26.61

  23P          10H016                      0.61     96.0     75.3    41.53    5.27

  9B           10H016                      0.61     96.0     75.2     6.48   26.61

  10A          10H016                      0.61     95.9     75.2     3.68   24.57

  24M          10H016                      0.61     95.9     75.2    35.94    3.24
```

FIGURE 3
DETAILED REPORT

- a fan is added
- the board spacing is increased
- heat sinks are added to the hottest components

In the opposite case, if the initial analysis shows the board operating at a safe temperature, he may want to see what happens if:

- the fan is removed
- boards are moved closer together
- plastic device packages are used in place of ceramic

For the thermal analysis package to be used effectively, it must perform this "what if" analysis with a minimum amount of manual data entry and in a short period of time. Table 1 shows typical changes that are made, what heat transfer modes are effected and how the change is modeled for thermal analysis.

Summary

By answering thermal design questions early in the design cycle, broadly-based trade-offs in packaging, layout and system configuration can be intelligently planned. Telesis Thermal Analyzer has enabled both electronic and mechanical engineers to benefit with significant improvements in overall design parameters and total product reliability.

TABLE 1
TYPICAL ENVIRONMENT CHANGES

FC = FORCED CONVECTION R = RADIATION
COND = CONDUCTION NC = NATURAL CONVECTION

MODELING THERMAL PROBLEM SOLUATIONS

METHOD OF SOLVING THERMAL PROBLEMS	HEAT TRANSFER[1] MODE AFFECTED	HOW IS IT MODELED	WHERE IS IT MODELED
ADD HEAT SINKS	COND/FC	AREA MULTIPLIER	DEVICE
USE THERMAL PASTE	COND	CASE TO BOARD RESISTIVITY	DEVICE
REPLACE BOARD	ALL	DEVICE LOCATIONS	THERMAL EXTRACT
ADD FAN	FC	AIR VELOCITY/LOCATION	ENVIRONMENT
INCREASE FAN SIZE	FC	AIR VELOCITY/LOCATION	ENVIRONMENT
CHANGE FAN LOCATION	FC	AIR VELOCITY/LOCATION	ENVIRONMENT
CHANGE NUMBER OF BOARD LAYERS	COND	BOARD CONDUCTIVITY	BOARD DESCRIPTION
CHANGE BOARD MATERIAL	COND	BOARD CONDUCTIVITY	BOARD DESCRIPTION
USE "COLD WALL" CONSTRUCTION	COND	EDGE CONDUCTIVITY	BOARD DESCRIPTION
CHANGE DEVICE PACKAGING MATERIAL	ALL	JUNCTION TO CASE RESISTIVITIES BOARD TO CASE RESISTIVITIES	DEVICE
CHANGE DEVICE TECHNOLOGY (CMOS VS. TTL)	ALL	POWER DISSIPATION	DEVICE
CHANGE DEVICE PACKAGE TYPE (DIP VS. SMD)	COND/FC	CASE TO BOARD RESISTIVITY AREA	DEVICE
CHANGE BOARD ORIENTATION	FC/RAD	GRAVITY DIRECTION AIR FLOW DIRECTION	ENVIRONMENT
MOVE ADJACENT BOARD	FC/RAD	ADJACENT BOARD SPACING	ENVIRONMENT
CHANGE ADJACENT BOARD TEMPERATURE	FC/RAD	ADJACENT BOARD TEMPERATURE	ENVIRONMENT

GLOSSARY OF

SIMULATION TERMS

GLOSSARY OF SIMULATION TERMS

BACK ANNOTATION

Process of updating schematic to reflect additional information created during the physical layout process including package assignments, device types, locations, and pin numbers.

BEHAVIORAL MODEL

A description of a circuit's outputs purely as a function of that circuit's inputs.

COMMON AMBIGUITY

The skew and signal timing shared by reconverging signals.

EVENTS PER SECOND

A crude measure of simulation speed, involving the number of circuit transitions being evaluated by the simulator in each CPU second. Events per second are dependent on the simulator and CPU.

FAULT SIMULATOR

A simulator for evaluating the results of faulty circuits. Judge is the fault simulator in LASAR.

FLAT DESCRIPTION

Circuit description entirely described at the gate level or below.

FUNCTIONAL MODEL

A description of how data flows through a device in such a way as its outputs can be determined for simulation.

GOOD CIRCUIT SIMULATOR

A simulator for evaluating the expected results of a correctly manufactured circuit. Simul is the good circuit simulator in LASAR.

HARDWARE ACCELERATOR

A technique for accelerating logic simulation by implementing a simulation algorithm in hardware -- can increase throughput at the expense of accuracy.

HAZARD

A condition when a circuit's response cannot be predicted reliably. Hazards preclude repeatable, reliable circuit operation. Timing problems and state conflicts are common causes of circuit hazards.

HIERARCHICAL DESIGN

A design methodology allowing for the description of circuit subsets at varying levels of completeness.

This accommodates TOP DOWN design. Such a methodology allows for interchangeability of precise circuit descriptions with general conceptual descriptions. This design interchangeability is referred to as nesting of references.

LANGUAGE

Syntax for instruction statements allowing the development of user/application program.

LINKS

Communications between data bases of CAD, CAE and ATE.
Example: Back annotation
Post processors
Net list translators

LSRPIPE

A LASAR language for circuit stimuli without tester constraints.

LSRTAP

Standard format for LASAR results for use in other than Teradyne specific applications.

MIXED MODE SIMULATOR

A simulator which can operate on a circuit description composed of any combination of behavioral, functional, or structural circuit elements.

MODCOM

The language in LASAR used for Netlist descriptions and an associated compiler for that language to create a circuit model for simulation.

NETLIST

A compact description of the interconnections between circuit elements.

OPERATING SYSTEM

A dedicated low level software resident on a particular computer which creates and maintains the user environment. Operations include file management, user interface, and the execution of application programs.

PERUSE

A LASAR process for graphically displaying waveforms occurring at any node in a circuit.

PHASER

Pattern language that constrains the programmer to stimulus that the target tester can apply.

PHYSICAL LAYOUT

Translation from schematic to the physical description of interconnects component placement. Required for both boards and devices.

SCHEMATIC CAPTURE

Creation and/or manipulation of a design with a graphics package. This is most important component of Computer Automated Engineering (CAE).

SIGNAL STRENGTHS

A method in simulation for determining the proper results at circuit interconnections, wired nets -- answers the question "Who owns the node?"

SOFTWARE BREADBOARD

LASAR or other simulation of a design in software in lieu of hardware verification.

STATES

A logical condition at circuit nodes, i.e., 0, 1, Z, X, or stored charge.

STRUCTURAL MODEL

A primitive level netlist.

TESTCOM

The language in LASAR used for describing stimulus to the circuit being simulated.

TML

A register transfer language for describing a circuit at the functional level and an associated compiler which creates a structural (gate level) netlist directly from the functional description.

UNIX

An operating system developed and licensed by BELL LABS which is rapidly becoming an industry standard. Most work stations and design automation tools run under this operating system.

VALUES

A list of states, one of which will describe the logical conditions at a particular node for a given copy of the circuit. LASAR operates with greater than 32,000 values.

ECL RULES MANAGEMENT ON THE TELESIS SYSTEM

Buck Titherington

CAD Manager

Scientific Computer Systems

San Diego, California

ABSTRACT

The following paper covers ECL Rules Management as applied to the Telesis system. Some of the topics covered are Net Sequencing and Structure, Impedance Environment, Lumped and Distributed Loads, and Parallelism.

INTRODUCTION

What's the big deal about ECL? It's just another logic family, and you hook 'em all together, right? WRONG!! Well, then, perhaps what you've heard is that it's very difficult to work with, and has all kinds of strange restrictions. THAT'S WRONG TOO!!!

Designing with ECL does require a cooperative effort between the engineer, who is primarily responsible for logic content, and the designer, who is responsible for placement and correct and efficient routing of the design.

In an excellent engineering design, the way the networks are connected can render anything from a very fast product to an inoperative bag of parts. This is wholly dependent on how well a small set of "rules" are observed.

To better understand this, let's look at the things that make ECL fast:

1. highly sensitive to input switching

2. very small voltage swing from Ø to 1

The normal input range to an ECL device ranges from -.8V to -1.8V with a broad linear region in the middle. The linear region is an area where, even though the input has not reached half-rise or half-fail, the circuit is already in the process of switching state. So, for the same reasons that this family is fast, it is also sensitive to bad networks that cause excessive noise.

Please note that twice the word used was "network" and not "net". The word "net", for the purpose of the previous paragraphs, is only implying connectivity, whereas the word "network" implies a well-structured, controlled impedance transmission line. A fully optimized ECL design contains only "NETWORKS".

After making this distinction, in the following, nets will refer to networks. Remember, read "NETS", think "NETWORKS".

Three very important questions should now come to mind -- What makes a good net? How do you control the system to arrive there? What is the cost when you "cheat"?

Though there may be some minor ranking disagreement, here is my list of topics that must be addressed, in order of importance:

1. net sequence and structure
2. impedance environment
3. lumped and distributed loading
4. "wired ors"
5. "Y" nets
6. parallelism

Many of these items can be controlled and/or monitored through use of the ECL TOOLBOX, and some even have multiple solutions.

NET SEQUENCING AND STRUCTURE

The position of a pin within the sequence of a net is determined by the function of the pin in that net. There are three basic functions:

1. Source - senders of signals	(output pins)
2. Destination - listeners	(input pins)

3. Terminator - pull down resistor (terminator pins)

Generally, the correct ECL net sequence is in the order: source(s), destination(s), termination. For the vast majority (9Ø%+) of nets this is all that is required. The exceptional sequences, such as "wired-or" and "Y" nets, will be addressed later.

Now we must consider an odd pin function, the bi-directional. Transceiver pins that have this use are potentially both sources and destinations, depending on the design. Lacking any further information, they have to be placed between the sources (if present) and the destinations to insure that whichever function they are performing at a given moment they are positioned correctly.

And then there is the good old I/O pin. In any other logic family it isn't necessary to pay very much attention to the logic pin. Here, it must have the pin function defined to be correctly sequenced in the net. Think for a moment about an I/O pin that is an output in the local design. Logically, there must be a destination out there somewhere, and because of the "rules" there must be a terminator out there as well. So, for purposes of the local design, it must be treated as a terminator. In the other two cases, input and bi-directional, the pin must be sequenced as a source.

In either case, certain "abnormal" net configurations may be discovered that require reference to company standards or the responsible engineer.

A terminator that is more that 1" in trace from the last load adds one more capacitive load to the net, so as designers, please place terminators liberally over the board (about one pack per IC) so that the interconnect lengths can be optimized. Excess terminators can always be deleted prior to beginning the routing process.

The correct structure for the net is strictly serial, no "T's" or "stubs" of any sort allowed. So, what are "T's" and "stubs"? A "T" is a branch in a trace that occurs between two pins, connecting a third pin. A "stub" is a third trace leaving a pin in the middle of an otherwise sequential net to branch to a third pin. Either of these results in an antenna that allows excess noise injection to the net, and signal reflections within the net; both of these are undesirable. Our design rules specify that stubs of 1" are allowed, but this option is only exercised on bussed backplane signals, where not stubbing them would require the use of two I/O pins.

Incidentally, the use of unstubbed nets is being recommended for higher speed TTL nets as well, for many of the same reasons.

CONTROLLING NET SEQUENCE

We all know designers who have completed ECL designs by using the netlist and schematics, and through painstaking, manual management of the route, have produced good operative designs. These designs were usually small enough to allow this to be considered, and took about four or five times the time of an equivalent sized non-ECL project.

There are a large number of designs that are either too large and/or complex to allow manual completion, and in general the problem can be more uniformly solved through "software"; so let's talk about the Toolbox.

The menu set provided allows for two possibilities for sequencing (scheduling) nets. One is totally internal, and the second allows for dumping data and executing an external sequencing solution.

After the netlist is loaded into the system, and the board is placed, the net sequencing can be controlled completely internally by using the commands **TERMINATOR ASSIGNMENT** , **CREATE SCHEDULE**, and **COMPILE SCHEDULE** and then routed using the resulting schedule. The details for naming conventions and other requirements are included in the **Telesis Operators Manual**.

A second method, that I am more familiar with, is controlled externally using the **CREATE PASS FILES** command. This will create a proposed TTL sequence of connection that may be edited or processed into a valid ECL sequence. The modified sequence may then be reloaded using **LOAD PASS FILES** (detailed pick sequence in TABLE 1). The problem of termination assignment could be addressed internally as in the previous paragraph, or performed externally through several possible forms of function swap.

TABLE 1

----T STATION ROUTER------	-----------CO-ROUTER-------
ROUTER	CO-ROUTER
EDIT-CONNECTIONS	SET UP CO-ROUTER
ECL TOOLBOX	ECL TOOLBOX

EITHER ROUTER

INTERNAL	EXTERNAL
TERMINATOR ASSIGNMENT	CREATE PASS FILES
CREATE SCHEDULE	(assign termination modify externally)
COMPILE SCHEDULE	LOAD PASS FILES
ROUTE W/SCHEDULE	ROUTE BOARD

---- CONTROLLING NET STRUCTURE ----

The net structure for a fully ECL board may be controlled rather simply by setting the "NO T's" switch in the router. For a mixed technology design, a file of net names and/or numbers to be given ECL treatment must be supplied, named **"HIGH-FREQ-NETS"**.

IMPEDANCE ENVIRONMENT

There have been successful wire-wrapped ECL designs. The major problem related to wire-wrap is rooted in the unpredictable impedance of each wire. The impedance is determined by the exact distance from the conductor to the supporting ground or power plane, and the size of the conductor. If you don't know whether the wire is a first or second level wrap, and worse, you can't control it, you've got a problem. Let's avoid "horror stories" and talk about PC.

This task can be accomplished rather easily by:

1. controlling trace width on a per-layer basis

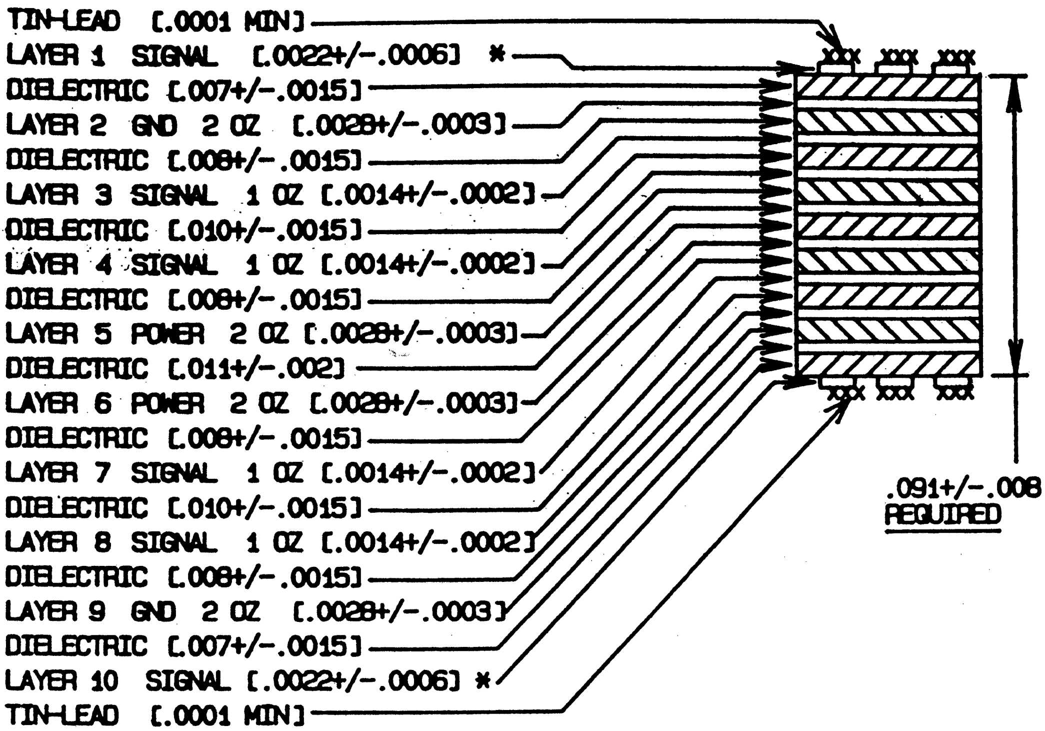
2. controlling the thickness of inter layer media

3. never stacking vertical or horizontal layer pairs together

4. spacing power layers adjacent to all signal layers

5. NEVER breaking up power layers to complete traces

The way we handle the trace width is to put all the trace in at 12 mils, with no other items at that size. Then, instruct the photoplotting house to substitute 1Ø mils on the external layers, and 9 mils on the inner. The rest is covered by sequencing the Gerber files correctly, and by fab notes.



TIN-LEAD [.0001 MIN]
LAYER 1 SIGNAL [.0022+/-.0006] *
DIELECTRIC [.007+/-.0015]
LAYER 2 GND 2 OZ [.0028+/-.0003]
DIELECTRIC [.008+/-.0015]
LAYER 3 SIGNAL 1 OZ [.0014+/-.0002]
DIELECTRIC [.010+/-.0015]
LAYER 4 SIGNAL 1 OZ [.0014+/-.0002]
DIELECTRIC [.008+/-.0015]
LAYER 5 POWER 2 OZ [.0028+/-.0003]
DIELECTRIC [.011+/-.002]
LAYER 6 POWER 2 OZ [.0028+/-.0003]
DIELECTRIC [.008+/-.0015]
LAYER 7 SIGNAL 1 OZ [.0014+/-.0002]
DIELECTRIC [.010+/-.0015]
LAYER 8 SIGNAL 1 OZ [.0014+/-.0002]
DIELECTRIC [.008+/-.0015]
LAYER 9 GND 2 OZ [.0028+/-.0003]
DIELECTRIC [.007+/-.0015]
LAYER 10 SIGNAL [.0022+/-.0006] *
TIN-LEAD [.0001 MIN]

.091+/-.008
REQUIRED

* .5 OZ COPPER + PLATE UP.

LAMINATION STACK-UP

10. THIS NOTE DICTATES THE FINISHED LINE WIDTH. THE FILM PACKAGE MAY OR MAY NOT REFLECT THE FINISH LINE WIDTH. ALL FINISHED SIGNAL LINE WIDTHS WILL BE .010 +/- .002 ON LAYERS 1 AND 10 AND .009 +/- .002 ON LAYERS 3, 4, 7 AND 8. MINIMUM AIR GAP OF .010 REQUIRED BETWEEN ALL TRACES. EXCEPTION TO TRACE AND AIR GAP WILL BE IF TWO TRACES GO BETWEEN .10 CENTER PINS - TRACES WILL "NECK DOWN" AND AIR GAPS ARE REDUCED.

LUMPED AND DISTRIBUTED LOADS

Either "Lumped" or "Distributed" loads are tolerable, but mixing them causes problems.

A distributed load is just what the name implies. The loads are linearly spaced along the length of the net at about three or more inches per segment.

```
3.Ø"     3.5"     4.5"     4.Ø"     4.5"  .5"
S------D------D---------D--------D---------D-T
```

A lumped load is a net with source well spaced from its loads, and whose loads are in relatively close proximity. An example would be a net whose source was ten inches from the first load, and four loads in less than four inches of net.

```
   1Ø.Ø"                 1.Ø" .5"
                           .3"  .1"
S-----------------------D--D-D-D-T
```

Either of these examples are OK. However, if you added another load to the lumped net, an inch away from the source, you would have a mixed consistency. The problem is caused by the lumped area of the net, which changes local impedance, causing signal reflections. In the pure lumped case, no load was back up the net to receive the degraded signal, therefore, no harm was done. In the mixed case, the added load would see the sum of the signal and the reflection, causing distorted results.

```
                         1.Ø" .5"
 1.Ø"     9.Ø"             .3"  .1"
S--D-------------------D--D-D-D-T
```

WIRED OR'S

A "wired-or" is defined as multiple sources in the same net. It derives its name from the fact that one or more sources driven positive will cause the whole net to go positive.

Wired-Or's should never exceed six sources, and the total interconnect length between the sources should be held below three inches. We additionally require that the sources be in the same chip, however, other design rules do not always demand this.

```
      .2"  .4"                  (USE LOADING SPACING
       .1"                       AS SHOWN PREVIOUSLY)
     S-S-S--S--->              <---D---------D-T
```

The length requirement between sources is real, because widely separated sources with multiple sources driven positive will, when several sources drop out, not resync the load fast enough to avoid falling into the linear range.

```
  S-----------S-S-S---->
  1           2 3 4
  +           - + +
```

If S3 and S4 both go negative simultaneously, the down line observed signal may go linear until S1 resyncs.

Use of the wire-or should not be capricious as the propagation delay along a dotted circuit is about double that of a normal net. This is often faster than the trace and a true logic or, but judgement should be exercised.

Y NETS

A "Y" net (sometimes called "wye" net) is a distinct departure from the normal sequence outlined earlier. It is generally used to "de-skew" register control lines by constructing the net with the source(s) in the middle of the net, feeding destinations on both sides, and terminated on both ends. Our design rules require 25-ohm driver(s) as sources in a 5Ø-ohm environment, however, multiple 5Ø-ohm drivers have been used successfully.

```
                    STANDARD "Y"

              T-D--------S-------D-T

       T-D---D----D---D---S---D---D----D--D-T

                    OTHER "Y"

           T-D---------S--S--------D-T
```

PARALLELISM

The problem with parallelism goes back to the "right-hand rule" of magnetic flux, that explains inductance between parallel lines. Because these magnetic fields are additive as more traces run together, if they switch at the same time, any line containing a non-switching signal receives more noise. Eventually, as more lines are bundled, it will be driven to or beyond the switching threshold.

Parallelism is a relatively small problem in a good impedance environment, but a disaster in, for instance, wire-wrap. The distances used for the critical threshold in wire-wrap are typically three parallel wires for three inches. Our PC design rules on a 25 mil grid specify bundles of two lines at 32 inches, and three lines at 16 inches, as a maximum.

CONCLUSION

Now that the router and I have done our thing, how long are the traces, and how many vias are there in each signal; what nets are lumped, etc.? There is a tool provided, called ECL-LOAD-REPORT that reports on: loading, T'ing, and trace length of each segment. Some additional engineering analysis may be required to complete the task, but it does give you a big start on it.

USING AUTOMATIC TERMINATOR ASSIGNMENT WITH SCHEMATIC DRAWINGS

Paul Gillis
Applications Engineer
Telesis Systems Corporation

INTRODUCTION

The intent of this paper is to provide a working knowledge of how to use the terminator assignment feature of the ECL TOOLBOX when a Net-Data-Base has been created using schematic extraction instead of by loading a text netlist. In addition to the standard terminator assignment rules and procedures, the user must contend with some problems that are not present when the Net-Data-Base is created with a text netlist.

The purpose of the automatic terminator program is to make optimum terminator pin selections for user defined nets based on the component placement and availability of terminators on the board. Therefore, the schematic must be created so that when components containing terminator pins are added to the data base, the terminator pins are not assigned to any nets. The schematic must also be modified after extracting the Net-Data-Base so that the connections are appropriate for back annotation and the terminator assignments can be displayed in the schematic drawing.

HOW AUTOMATIC TERMINATOR ASSIGNMENT WORKS

Before explaining in detail the procedures that must be followed, an explanation of how automatic terminator assignment works would provide some insight into how to manage the issues associated with this problem.

The Terminator Assignment program is part of Telesis's ECL TOOLBOX, a group of unique programs that aid in the design of printed circuit boards which contain high frequency circuitry and typically make use of ECL technology. One of the fundamental problems faced by the designer in such a design is to reduce signal reflections that can occur on high frequency signal lines. A resistor connected to the end of a high frequency transmission line can suppress much of this reflection by matching the impedance at the end of the signal line with the impedance at its source.

This resistor, known as a terminator, must be connected to the signal line following some special high frequency design rules. Typically, each pin on a high frequency net can be classified as a source or as a load. The terminator usually connects to the last load on the end of the signal line, which in most cases is the load located farthest from the source. The connection from the terminator to this load should also be very short and direct. This load pin will be established by the program and be referred to as the "most remote load pin". It is this most remote load pin that is used as the search reference when the program searches for a suitable terminator.

To accomplish the goal of finding a terminator for each ECL net, it is neccessary to first place all components on the board, as the program will make its selections based on component placement. The program will evaluate each pin of the net being processed and determine whether the pin acts as a source or a load based on its pin use code. Once the functions of all pins on the net has been determined, the program will then determine which load is the most remote load pin. After the most remote load is identified, the program will search for a terminator using the most remote load as a search reference.

Figure 1 illustrates a most remote load selection for a net containing one source and four load pins. The program would select the pin labeled as the most remote load because the distance from it to the source pin is greater than the distance from any of the other load pins to the source pin.

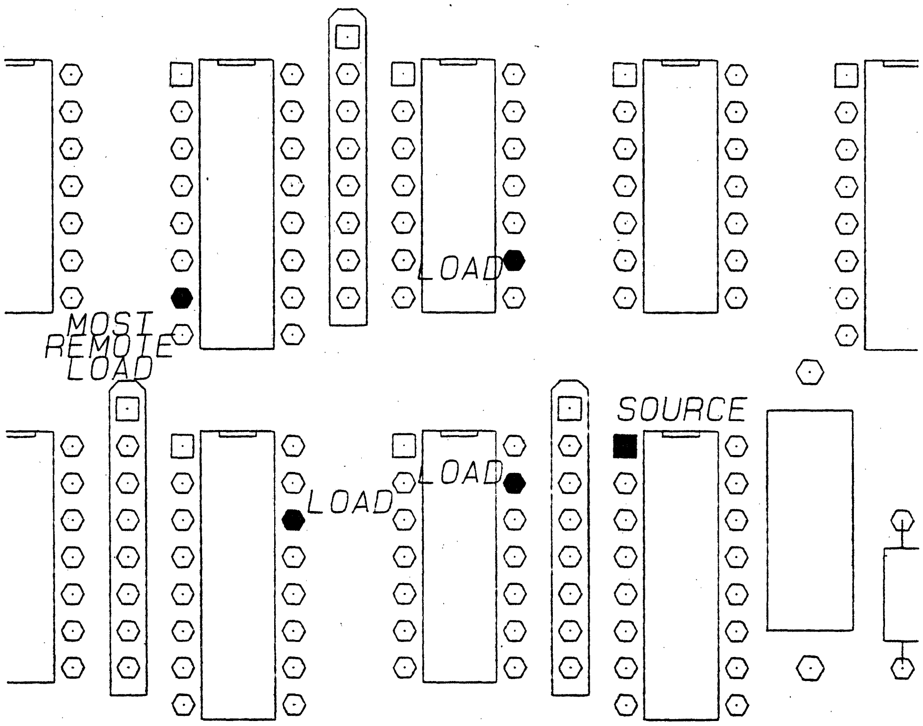


Figure 1

When the program computes the source to load distances, it uses a Manhattan distance rather than a straight line distance. A Manhattan distance is the sum of the X and Y distances between the two pins. The Manhattan distance between the source and load pin in Figure 2 would be 1.3 inches if the X distance was 900 mils and the Y distance was 400 mils. This concept should be kept in mind when placing components strategically to force most remote load and terminator selections from the terminator assignment program.

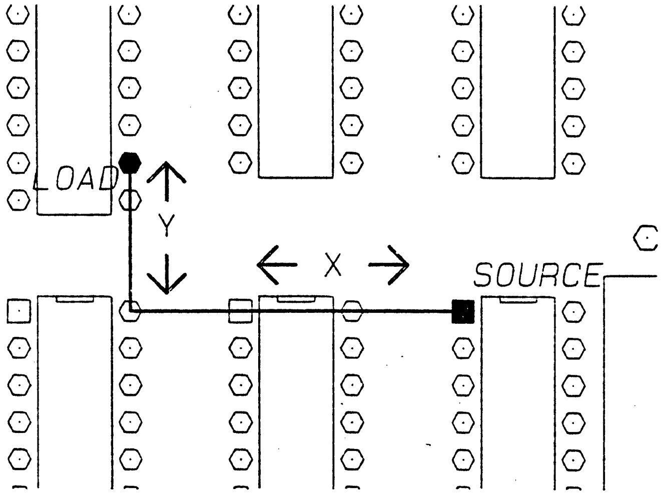


Figure 2

The user also has the option of specifying a preferred maximum distance that terminator should not exceed from the most remote load pin. This is done through a user created text file named "TERMINATOR-CON". For each net listed in the user created HIGH-FREQ-NETS text file, the program will attempt to find an available terminator pin that falls within the maximum terminator distance specified by the user. The program again uses Manhattan distances instead of straight line distances when deciding whether a terminator on the board is within the maximum terminator distance. If none are found within this distance, then a warning message will be placed in the log file named "TERMINATOR-LOG". The program will then continue and select a terminator from among any that are not placed on the board. This allows the user to place the selected terminator on the board so that the maximum terminator distance is not exceeded.

If this attempt at finding a terminator also fails, another attempt will be made in an effort to select a terminator. A search area is computed that is based on the the distribution of terminators on the board. If a terminator is still not found within this search area, then an available terminator will be selected at random from among any placed on the board.

The output from the terminator assignment program is a text file that is formatted such that it is suitable for an incremental load text netlist. This text file is named "TERMINATOR-IN". It contains, for each net, one line that lists the computed most remote load pin, followed by a semicolon, followed by the selected terminator pin. The TERMINATOR-IN file can then be used as input to the load text netlist program to assign the terminator selections to the data base. If the user prefers, the file can first be modified to override program selections. By making the assignments using this indirect method, the user can review the selections made by the terminator assignment program and make changes before the assignments are actually input into the Net-Data-Base.

PREPARATION BEFORE CREATING THE NET-DATA-BASE

Before a Net-Data-Base can be created, it is important that the pinuse codes in the device files or on schematic drawings are correct. Many of the computations made throughout the ECL TOOLBOX programs, including terminator assignment, are based on pin functions which are determined by the pinuse code of a pin. Source pins should be labeled with pinuse codes of "OUT", "OCA", "OCL", "TRI", or "BI". Load pins should be labeled with pinuse codes of "IN", "PWR", or "GND".

Terminator pins are recognized by the system as those pins on components that have a device name whose first ten characters are "TERMINATOR" and that are not common. The pinuse codes of terminator pins should be unspecified, since they do not actually function as a source or load.

SCHEMATIC DRAWING CONSIDERATIONS

When a Net-Data-Base is initially created using a text netlist, components can be created through an entry in the "$PACKAGES" section. Pin to pin connection assignments can made by explicitly entering net information in the "$NETS" section. When creating terminator components, it is desirable to leave terminator pins unassigned so that they will be available to the terminator assignment program for processing. This can be accomplished by simply not adding the terminator pins to any of the nets in the text netlist $NETS section.

When creating components using schematic drawings, the extract netlist program will automatically assign any unannotated pins on symbols that have connections tied to them. It will be necessary then to leave the pin numbers of terminator components unassigned by leaving the terminator pins unconnected, and by leaving reference designators and function designators unannotated. When schematic extraction is run, this will allow components to be created which contain terminator pins but will prevent them from being automatically assigned. The reference designators of terminators should also be left unassigned to allow for back annotation after running terminator assignment.

Later, the terminator pins will have to be connected to the corrosponding high frequency net to represent a net connection, which is necessary to correctly back annotate the schematic. The terminator symbols on the schematic should therefore be placed either close to what the user feels will be the most remote load, they can be grouped together on the schematic, or they can be placed on a seperate schematic sheet. Figure 3 shows terminators placed close to their most remote loads. If the terminator symbols are not added close to the most remote load of the net, additional symbols with the high frequency net names attached will make the task of updating the schematic to prepare for back annotation much easier. This technique is illustrated in Figure 4. Once the above requirements for device files, pinuse codes, and schematic considerations have been met, the extract netlist program can be run on the schematic drawings.

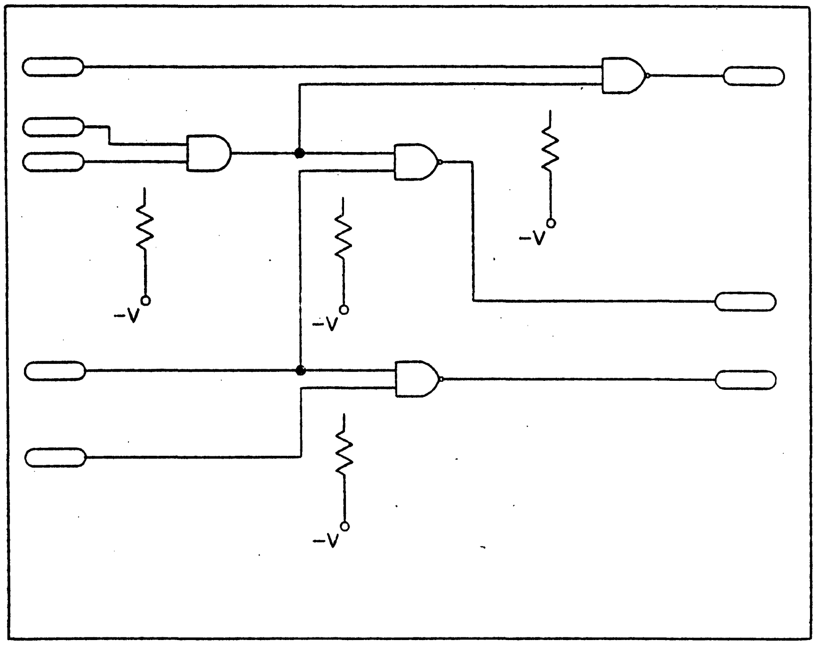


Figure 3

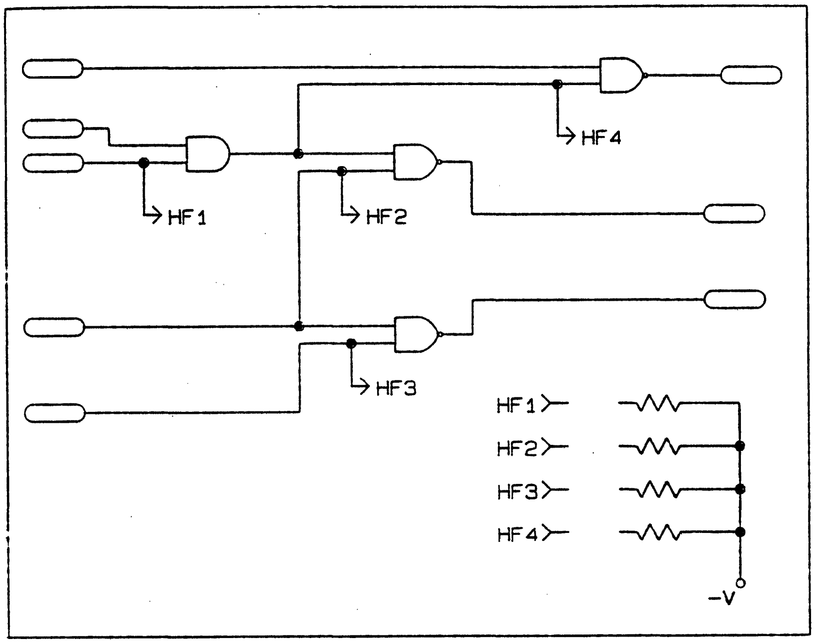


Figure 4

The terminator assignment program is designed to make the most optimum terminator selection for each high frequency net based on the location and availabilty of terminators. In order to perform this task it will be necessary to follow some placement guidelines to insure that optimum terminator selection is obtained.

The typical flow of a high frequency circuit is from the source, or driver, to the loads in "daisy-chain" connections, and ending on a terminator. An example of a net connection exhibiting these characteristics can be seen in figure 5.

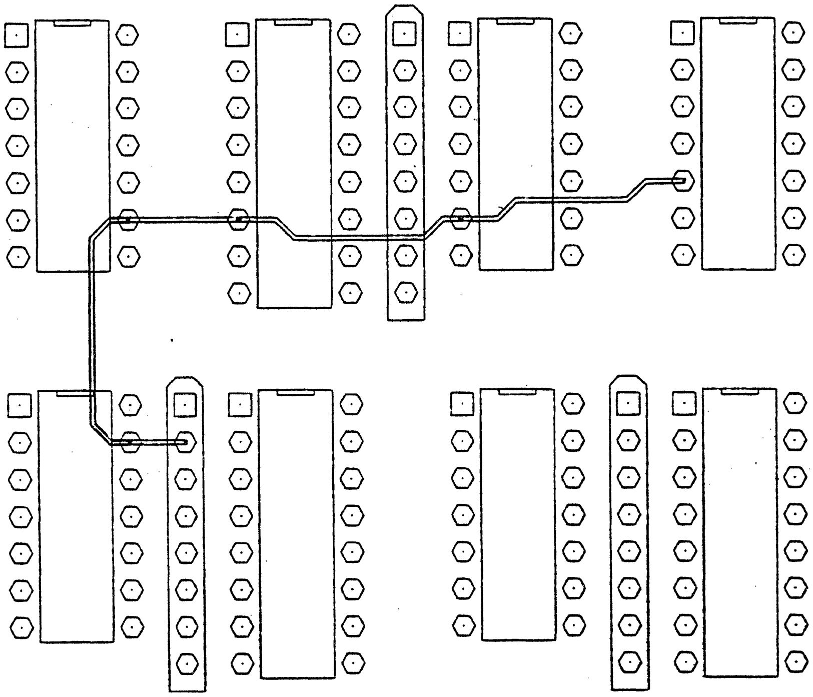


Figure 5

It is generally good design practice to keep in mind the signal flow when placing components on the board. Although the terminator assignment program will always make the most optimum selections, the results may still not be adequate if the program had to work with an extremely poor initial placement. In this case, the user has the option of modifying the placement and running terminator assignment again, or simply changing the most undesirable selections in the TERMINATOR-IN text file.

Another important consideration is to guarantee that there are enough terminators at the locations where they are needed. It is very likely that on a board with a minimum amount of terminators, what may be the best selection for a net being processed would have been a better selection for a subsequent net. If, in figure 6, the highlighted terminator pin was the only one available on that SIP package, it would be selected for whichever of the two nets was processed first. That terminator pin would be better suited for the net with the most remote load that is closest to it, but if the other net was processed first there would be no suitable terminator for either net. If this situation was common throughout a board, it would result in marginal terminator selections for a large amount of ECL nets. To avoid this problem it is suggested that the board be overpopulated with terminators, especially in those places where they may be needed most. Figure 7 shows how the problem in figure 6 could be solved by simply adding an additional SIP package.

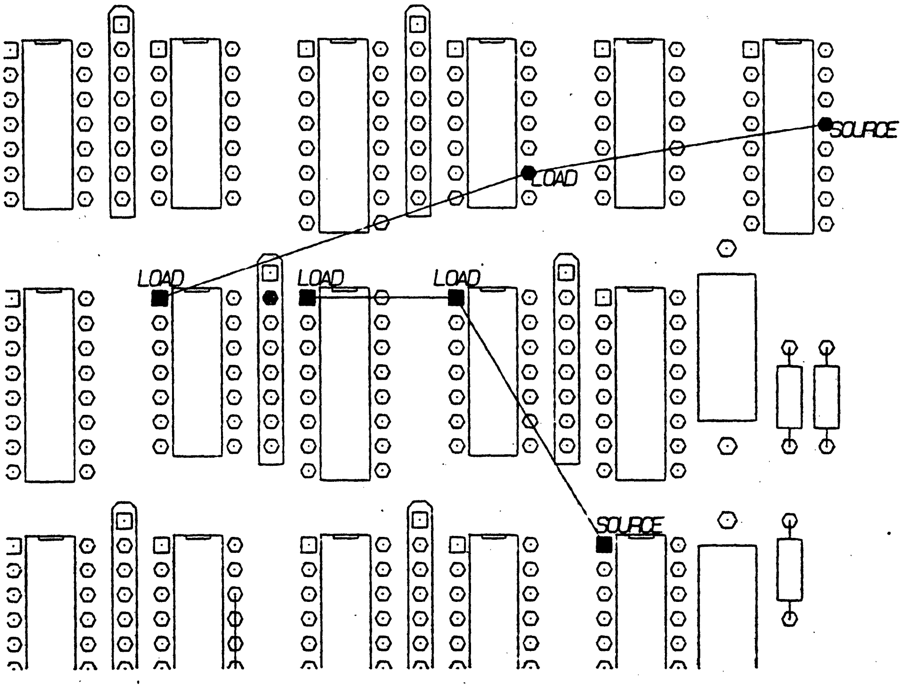


Figure 6

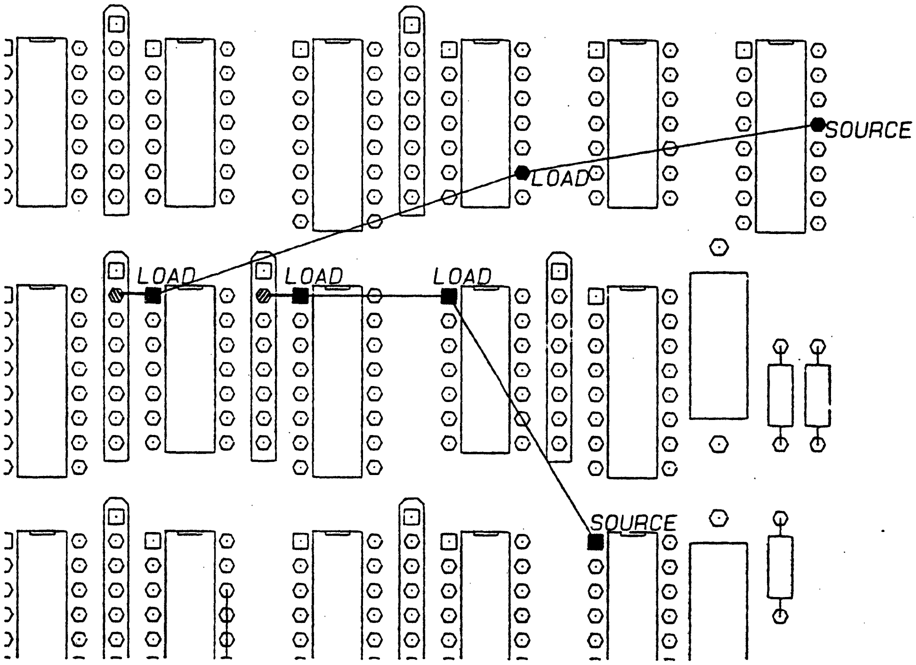


Figure 7

RUNNING TERMINATOR ASSIGNMENT

Once all preparatory work has been done, the terminator assignment program can be run. The requirements are that a board drawing and Net-Data-Base be active, a HIGH-FREQ-NETS text file exist in the current index listing the net names or numbers which should be processed, a TERMINATOR-CON text file exist in the current index if a preferred maximum terminator distance is desired, and a sufficient amount of terminators are available for selection.

The program will begin processing each net listed in the HIGH-FREQ-NETS file. For each net a terminator will be selected, or an error message will be placed in the TERMINATOR-LOG output file. In addition, a warning message will also be placed in the log file for each terminator selection that was made from among any terminators not placed on the board, or for any that were not within the maximum terminator distance but were within the search terminator search area if a maximum distance was specified, or for any that were selected at random because no terminators were within the terminator search area.

As can be seen from these warning messages, some terminator selections may not always be the most desirable. The intent of the program is to make a terminator selection for each net it is asked process. The warning messages in the log file bring to the attention of the user selections that should be reviewed before using the load text netlist program to input selections into the Net-Data-Base. If there are only a few warnings, the TERMINATOR-IN file can be modified to correct only those selections that require it. If there are excessive warning messages, the number of terminators and terminator placement should be examined, and the necessary corrective action taken before running terminator assignment again.

BACK ANNOTATING THE SCHEMATIC

Once terminator assignments are input into the Net-Data-Base using LOAD TEXT NETLIST, the schematic drawing can be back annotated to display the terminator assignments made. Back annotation will update the terminator reference designators as well as the pin numbers.

Before running back annotation, the schematic must be updated to add connections to the terminators. The terminators were deliberately left unconnected before schematic extraction so that terminators would not be automatically assigned, and they can not be back annotated until connection information is added.

It will be a simple task to add the connections to the schematic if the schematic was prepared as recommended. If schematic terminator symbols were placed close to their corresponding most remote load symbol, a connection should be added from the terminator to that load. If the terminator symbols were placed next to symbols that were assigned high frequency net signal names, a connection is added from the terminator to that symbol.

Once this task is completed, back annotation can be run as usual. The terminator symbol reference designators and pin numbers will be updated to reflect the selections made by the terminator assignment program.

SUMMARY

The procedures outlined in this paper will allow schematic drawings to be used effectively when the automatic terminator assignment program will be used to assign terminators. The important steps in the design involve insuring that terminator pins are not connected or annotated before extracting the schematic, running the terminator assignment program and updating the Net-Data-Base, and then adding connection information to the terminator pins so the schematic can be back annotated.

MINISCRIBE CORPORATION
LONGMONT, COLORADO 80501

FLEX-CIRCUIT DESIGN USING PC AND MD TELESIS CAD SYSTEM SOFTWARE

JO ANN VIGIL
FLEXCIRCUIT DESIGNER

FLEX CIRCUIT DESIGN USING THE PCB AND MECHANICAL TELESIS SOFTWARE

1. FLEX-CIRCUIT APPLICATIONS

A FLEX-CIRCUIT IS NO MORE THAN A FLEXIBLE CIRCUIT BOARD. IT IS USED IN MANY COMMON HOUSE-HOLD ITEMS SUCH AS YOUR TELEPHONE, CAMERA, AND IF YOUR HOME HAS A COMPUTER, THE DISK DRIVE INSIDE HAS A FLEX-CIRCUIT.

FLEX-CIRCUITS ARE USED IN TIGHT PACKAGING SITUATIONS WHERE IT MAY BE NECCESSARY TO FOLD, BEND AND WRAP THE CIRCUITRY AROUND OBJECTS.

THE FLEX-CIRCUIT IS MADE OUT OF A PLASTIC MATERIAL, POLYIMIDE, WITH COPPER ADHERED TO ONE SIDE. THE PROCESS OF ETCHING THE IMAGE ONTO THIS MATERIAL IS QUITE SIMILAR TO THAT OF A RIGID PRINTED CIRCUIT BOARD.

A FLEX-CIRCUIT OUTLINE CAN BE ANY SHAPE AND SIZE. THE FINAL PRODUCT IS STAMPED OR PUNCHED OUT USING A DIE RATHER THAN ROUTING AS DONE IN RIGID PRINTED CIRCUIT BOARD MANUFACTURING PROCEDURES.

2. DESIGN CRITERIAS AND VENDOR SPECIFICATIONS

ONCE YOUR FLEX-CIRCUIT OUTLINE HAS BEEN ESTABLISHED YOU CAN BEGIN CALCULATING MINIMUM TRACE WIDTHS AND SPACINGS INTO YOUR USABLE AREA.

MINIMUM TRACE WIDTH=.ØØ6
PREFERRED WIDTH= .ØØ8

MINIMUM TRACE TO TRACE=.006
PREFERRED SPACE= .ØØ8

MINIMUM TRACE TO PAD=.ØØ8
PREFERRED SPACE=.Ø15

MINIMUM PAD OR TRACE TO EDGE=.Ø15
PREFERRED SPACE= .Ø3Ø

THERE ARE THREE THINGS TO KEEP IN MIND WHEN DESIGNING YOUR FLEX-CIRCUIT:

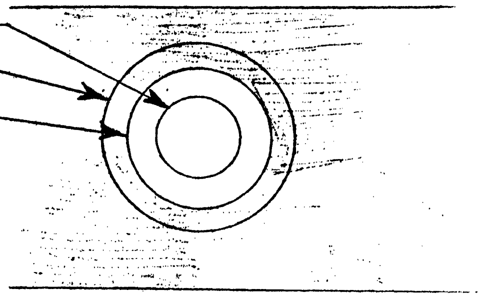
1. DESIGNING OF THE ACTUAL FLEX CIRCUITRY.
2. MAXIMUM AND MINIMUM PAD AND TRACE AREAS.
* 3. COVERLAYER OPENINGS.

* A COVERLAYER ACTS AS A KIND OF SOLDER MASK. A COVERLAYER IS ALSO A PLASTIC MATERIAL, POLYIMIDE.

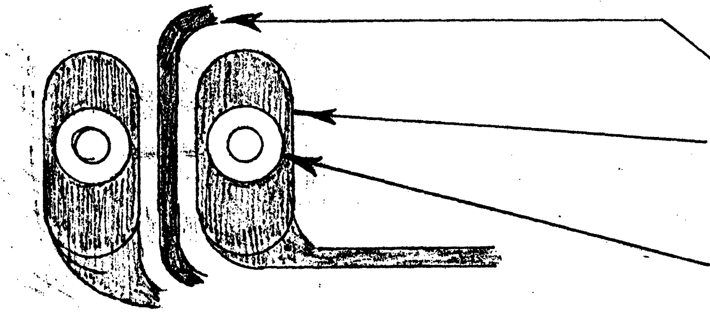
A COVERLAYER OPENING AROUND THE THRU HOLE NEEDS TO BE BIG ENOUGH TO EXPOSE THE HOLE AND ENOUGH PAD TO SOLDER TO, YET NOT BIGGER THAN THE PAD ITSELF, OR ELSE THE PAD CAN LIFT FROM THE POLYIMIDE DURING THE SOLDERING PROCESS.

HOLE=.035 DIA.
PAD=.062 DIA.
COVERLAYER OPENING=.050

THIS IS FOR A STANDARD PAD SIZE, BUT IT IS NOT RECOMMENDED.



PREFERRED PAD DESIGN:



TRACE BETWEEN PADS

MAKE PADS AS LARGE AS POSSIBLE IN USABLE AREA, TO INSURE GOOD PAD CAPTURE.

COVERLAYER OPENING.

3. USING THE MECHANICAL AND PCB PACKAGES TO CREATE A FLEX-CIRCUIT.

A. THE FLEX OUTLINE CAN BE ENTERED INTO EITHER PC OR MD. USE LAYER 33.

B. WHEN MERGING YOUR FLEX-CIRCUIT DRAWING FROM MD TO PC, YOU CAN ONLY MERGE LAYERS 33 AND 34. WHEN MERGING FROM PC TO MD, YOU CAN BRING ALL LAYERS OVER TO MD, BUT YOU MUST THEN CHANGE THE LAYER OF ALL ELEMENTS TO 33 AND/OR 34 IN ORDER TO TAKE THEM BACK TO PC.

C. THE NAME OF THE DRAWING AND THE OVERALL EXTENTS MUST BE THE SAME IN BOTH PC AND MD.

D. A FLEX-CIRCUIT DESIGN DONE USING MD AND PC HAS NO LOGICS. ALL DESIGNING IS GRAPHICAL ONLY. PDRC, NETCOMPARE, AND ANY OF THE CREATE REPORT COMMANDS ARE USELESS.

E. IN MD, BEGIN CREATING YOUR FLEX-CIRCUIT DESIGN BY USING CONSTRUCTION LINES SPACED OUT AND PLACED IN THE AREAS YOU WANT TO HAVE TRACES, PADS, AND COMPONENTS. THEN ADD YOUR LINES AND ERASE THE CONSTRUCTION LINES.

ALL OF THE LINES YOU ARE GOING TO BE PLACING HAVE A WIDTH OF ZERO, SO REMEMBER TO CALCULATE FROM THE CENTER OF YOUR ACTUAL TRACE WIDTH.

F. BUILD ALL OF YOUR COMPONENTS IN MD AND PLACE THEM IN THE AREAS YOU WANT. REMEMBER THAT THE COMPONENTS BUILT ARE FOR PLACEMENT REFERENCE ONLY. WHEN YOU MERGE TO PC, YOU WILL THAN PLACE THE ACTUAL COMPONENT IN THE SAME LOCATION.
REMEMBER TO BUILD YOUR COMPONENTS IN MD ON LAYER 33 OR 34.

G. ONCE ALL OF YOUR LINES AND COMPONENTS HAVE BEEN PLACED YOU CAN

BEGIN ADDING FILLETS AT INTERSECTIONS. THE FILLET COMMAND WILL ROUND CORNERS AND TRIM AT THE SAME TIME.
IF YOU ARE PLEASED WITH THE TOTAL DESIGN YOU ARE NOW READY TO MERGE TO PC.

H. IN PC, OPEN UP A NEW DRAWING WITH THE SAME NAME AND EXTENTS AS YOUR MD DRAWING. PICK THE COMMAND CREATE/MERGE DRAWING. MERGE YOUR MD DRAWING INTO PC AND BEGIN THE FINAL STEPS.

I. USING THE CHANGE COMMAND, CHANGE LAYER AND WIDTH OF YOUR TRACES. PLACE THE ACTUAL PCB COMPONENTS IN THE AREAS YOU HAVE CHOSEN IN MD. DELETE ALL UNNECESSARY LINES.

J. USING THE SHAPES COMMAND, YOU CAN ADD GROUND PLANES AND BUILD YOUR PADS UP WHERE YOU NEED TO.

K. CHECK YOUR DESIGN VERY CAREFULLY, BECAUSE PDRC DOES NOT CHECK LINES OR SHAPES AT THIS TIME.

L. TO CREATE PHOTOPLOT, THE SAME PROCEDURES APPLY:

YOU WILL NEED A PHOTOPLOT-PAR
A PHOTOPLOT-CON
A LAYERSTD
AN APERTURE-TAB
AND -PIN FILES FOR ANY SYMBOLS.

GENERATING ARTWORK

FOR

SPECIAL APPLICATIONS

BY: Penelope Levario
Micro-Rel, Tempe, Arizona

INTRODUCTION

Generating artwork from the Telesis C.A.D. system is part of full utilization of the system. For special applications, such as hybrid technologies, generating an accurate, cost effective, and timely artwork package can be a challenge. However, with a clear understanding of the technology requirements and the system limitations, requirements of any artwork package can be met.

At Micro-Rel we design two types of hybrid technologies each having their own set of artwork requirements.

- Thick Film: These hybrids are produced by screen printing and firing conductive, resistive, and insulation compositions onto a ceramic substrate.

  Artwork Requirement:

  1. an understandable layout
  2. 5:1 and 1:1 reductions of each layer
     - step and repeated images

- Multilayer Co-Fired Ceramic: This process involves screen printing interconnect patterns onto green ceramic tape. The interconnect is made through via's punched in the tape and filled with a metal system. The layers are laminated together and co-fired to form a monolithic network.

Artwork Requirement:

1. an understandable layout
2. 5:1 and 1.18:1 reductions of each layer with tape comparator
   - 1 package with substrate outline
   - 1 package without substrate outline
   - step and repeated images

To meet these artwork requirements, limitations of the system need to be understood.

LIMITATIONS

GRAPHIC

1. During the design process, the requirements for photoplot have to be kept in mind. To achieve accurate photoplots when using add line or add rectangle overlapping is necessary. To meet this requirement the layout is confusing and difficult to check.

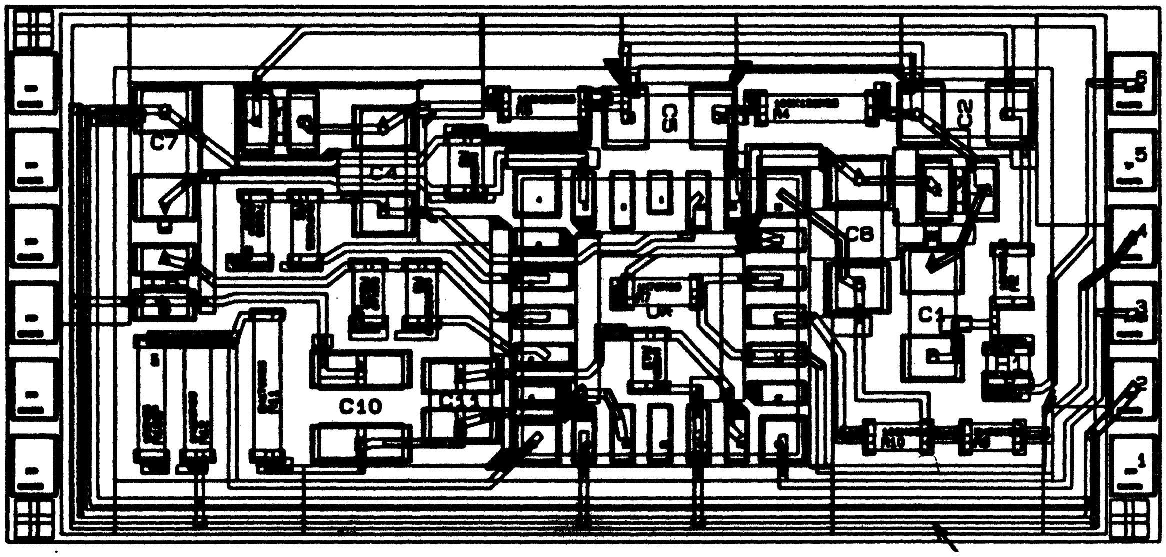
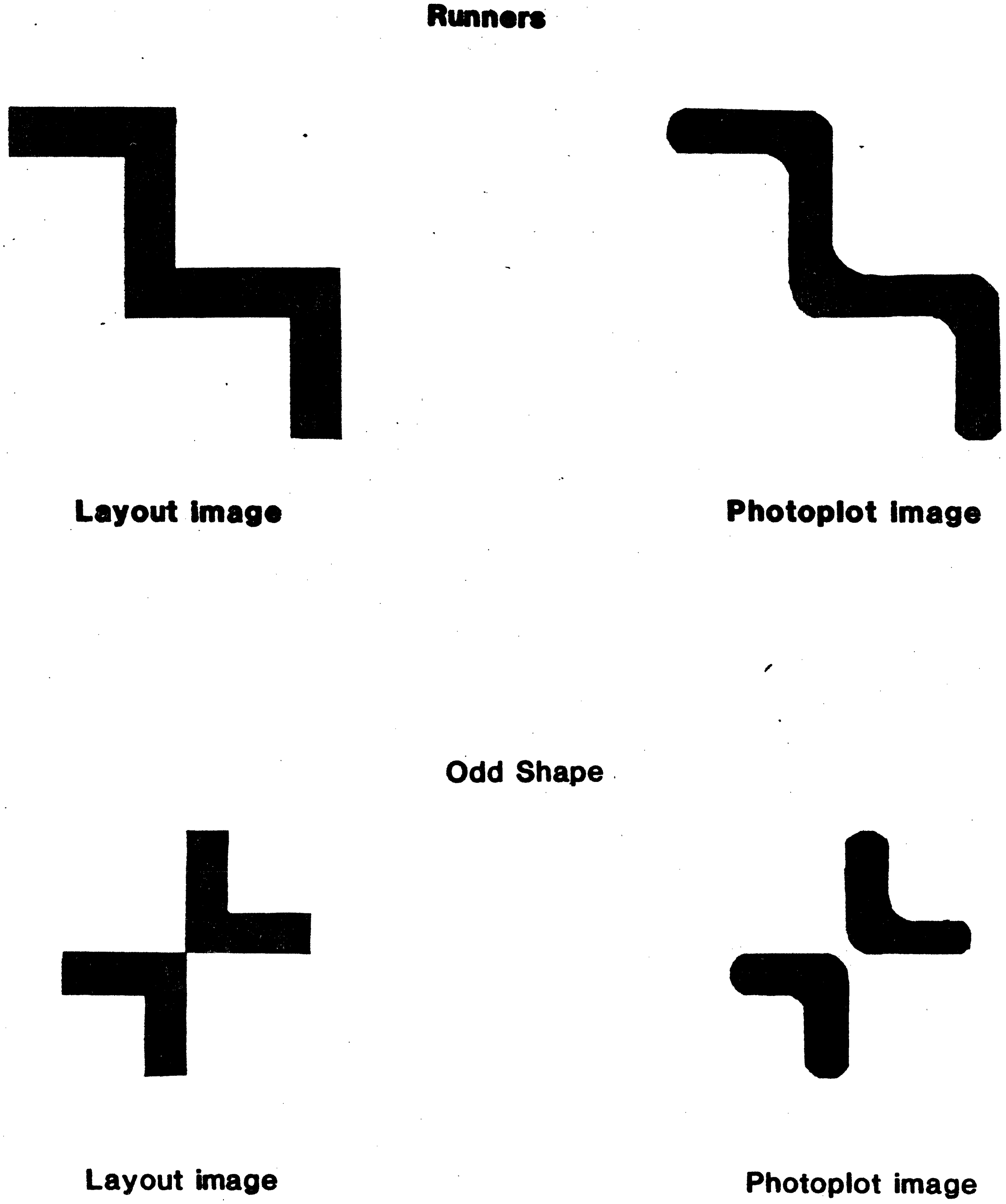


FIGURE 1

2. When using add line for runners and odd shapes the resulting photoplot will produce images with rounded corners and less than sharp definition.



Runners

Layout image

Photoplot image

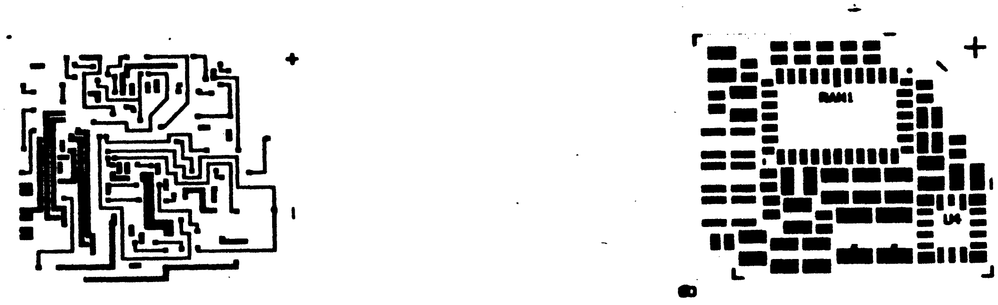
Odd Shape

Layout image

Photoplot image

FIGURE 2

3. The command <u>add rectangle</u> produces images with sharp defined corners, however the photoplot time for a Gerber Plotter is <u>98%</u> more than the <u>add line</u> command.



Layer with ADD LINE

Photoplot time: 1 min.
Cost: $20

Layer with ADD RECTANGLE

Photoplot time: 45 min.
Cost: $80

FIGURE 3

<u>SOFTWARE</u>

1. Co-fired multilayer ceramic technology requires buried via's. The system's via symbol and <u>N.C. Drill</u> cannot be used with a buried via defined symbol.

<u>PHYSICAL</u>

1. The hybrid requirement of a 10:1 system design for accurate artwork, limits stepping capabilities of the system. For multilayer co-fired ceramics step and repeat on the system is impossible.

All these limitations if not resolved can produce less than desirable artwork which is very expensive and time consuming. Therefore, it is important to address each limitation for a solution.

SOLUTION

GRAPHIC LIMITATIONS

1. To produce an understandable layout for checking, large odd shaped areas are defined as they should be for photo plot requirements. However, an outline of the confusing areas is placed on a different layer and used only for check prints.

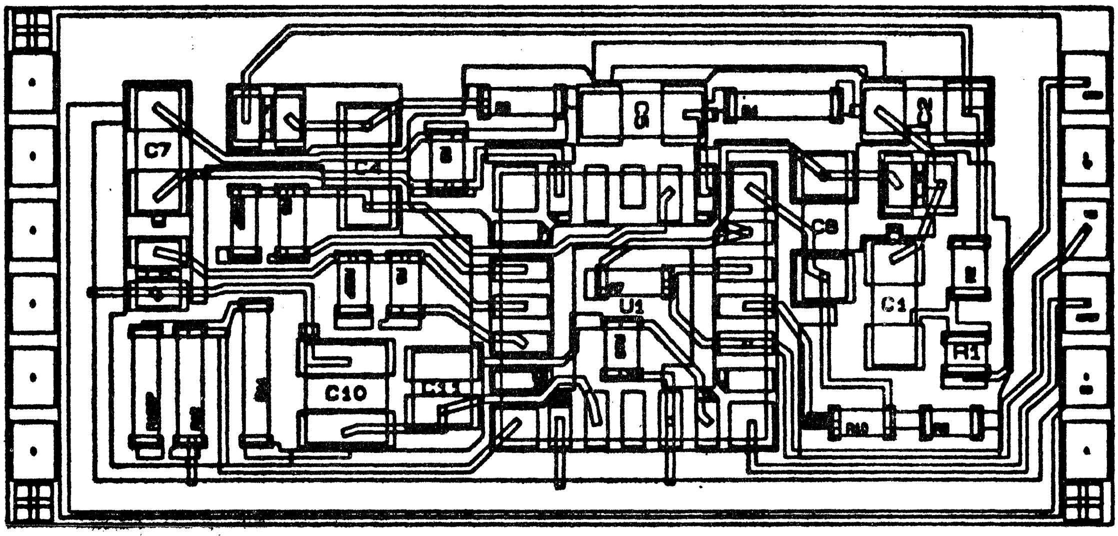


FIGURE 4

2. When using add line for runners and odd shapes, keep in mind that need for overlapping, and be aware of the effects of rounded corners.
3. Use add rectangle only when necessary for images with sharp defined shapes.

SOFTWARE LIMITATIONS

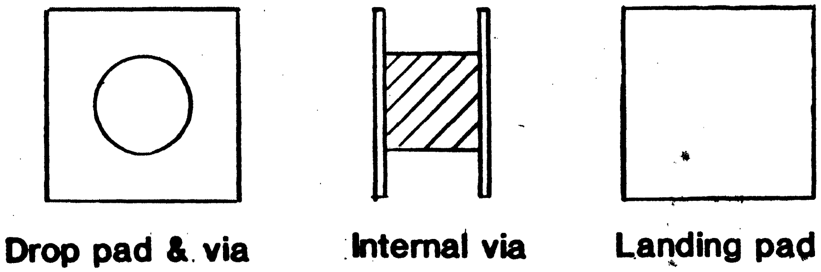
1. To meet the technology requirements for buried vias and via coordinant location, a new via symbol was defined.

   The buried via symbol is made up of:

   1. A drop pad with via

   2. An internal via

   3. A landing pad



Drop pad & via Internal via Landing pad

FIGURE 5

The symbol is constructed for each layer of the hybrid. A drop pad is placed on the layer where it drops from and a landing pad is placed on the layer it drops to. The via hole itself is placed on its own layer which produces a separate photoplot of just the vias. The drop pads and landing pads are photoplotted with the interconnect layers.

The pin file for the via lists the drop pad and landing pad sizes and the via hole is called out as a flash.

Pin file format.

(Pin File For Via Symbol Via-1/2)

(Pin File Name: Via-1/2-Pin)

Pintype

Drill

```
Pad Square=.200 External-Layer-1
Pad Flash=.100 Imbedded-Layer-Via-1/2
Pad Square=.200 Internal-Layer-2
End
```

The layer with the flash can be extracted from the gerber file and the coordinates used to run the via punch equipment for the manufacturing process.

PHYSICAL LIMITATIONS

1. If the physical size of the image and the stepping requirements of the artwork make it impossible to step on the system, the artwork vendor can be a solution.
   - A photoplot can be camera stepped
   - A laser vendor can step the image with his system

As you can see work arounds are not necessarily confined to the system. Solution can also be found with your artwork vendor. For a complex artwork package my solution was a laser photoplot vendor.

The laser plot vendor was able to:

- step and repeat with software which reduced camera stepping charges
- label the stepped images reducing the cycle time of manually labelling each layer
- merge files to produce different artwork packages. For example;
  1. images with and without circuit outlines

2. images with a comparator for manufacturing needs.

▾ reduce cost 50%, due to the lack of design density effecting photoplot time.

Thus, to produce artwork for special applications is not easy, but achievable. The necessary steps are:

1. understand your artwork requirements
2. know the limitations of the system
3. find solutions through system work arounds and the artwork vendor

KIM PATTERSON
NATIONAL SEMICONDUCTOR
(408)721-4120

NATIONAL SEMICONDUCTOR HYBRIDS

AT NATIONAL SEMICONDUCTOR OUR HYBRID SUBSTRATE DESIGN GROUP SUPPORTS ENGINEERING IN THE DESIGN OF THIN FILM CIRCUITS, THICK FILM CIRCUITS AND PRINTED CIRCUIT BOARDS. THIS PAPER WILL ATTEMPT TO DEFINE THE DIFFERENT TECHNOLOGIES USED AND THEN A MORE DETAILED DESCRIPTION OF HOW WE USE THE TELESIS CAD SYSTEM IN THICK FILM HYBRID CIRCUITS.

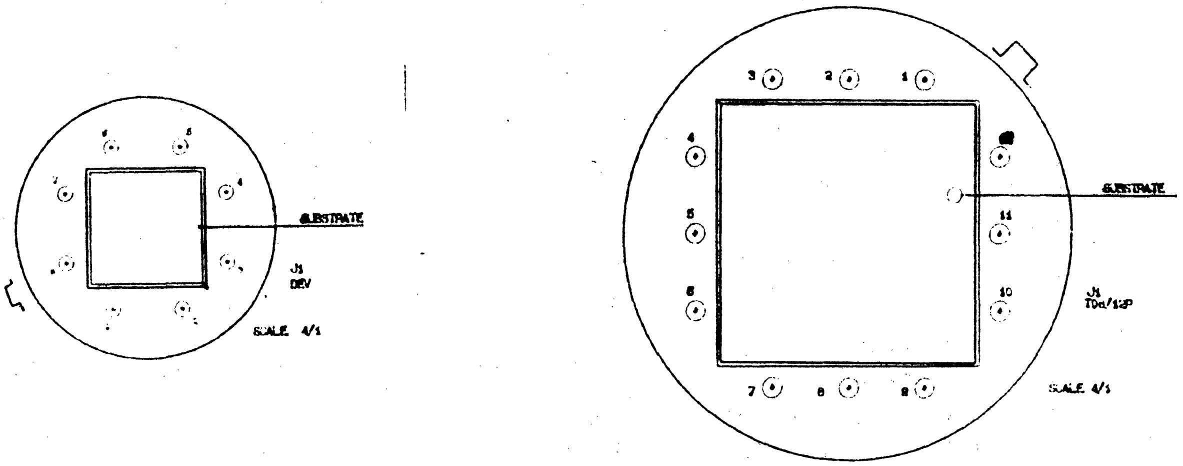
PC BOARDS RANGE IN SIZE AS WELL AS MATERIALS. MATERIALS RANGE FROM THE "NORMAL" FR4 TO PORCELIAN STEEL. TRACE MATERIALS RANGE FROM COPPER TO GOLD AND OUR SIZES RANGE FROM UNDER ONE INCH TO OVER 20 INCHES. THE COMPONENTS USED RANGE FROM SURFACE MOUNT, TO "CHIP AND WIRE" TO THE "NORMAL" COMPONENTS REQUIRING PLATED THROUGH HOLES.

THIN FILM VARIES IN SIZE AND MATERIALS AS WELL AS PACKAGING REQUIREMENTS. THIN FILM SUBSTRATES (OR BOARDS) AT NATIONAL START AS SMALL AS 50 MILS AND GET AS LARGE AS 980 MILS. WE USE VARIOUS MATERIALS INCLUDING GOLD AND ALUMINUM FOR CONDUCTORS, AND ALUMINA AND SILICON FOR SUBSTRATE MATERIAL. THIN FILM MOST ALWAYS USES THE SURFACE MOUNT AND "CHIP AND WIRE" APPROACH.

THICK FILM RANGES IN MATERIALS, SIZE AND PACKAGING REQUIREMENTS. THE SUBSTRATE (OR BOARD) MATERIALS RANGE FROM ALUMINA TO PROCELAIN STEEL AND RANGE IN SIZE FROM 0.150" TO 2.0". THE TRACE OR CONDUCTOR MATERIAL VARY TREMENDOUSLY, FROM GOLD TO SILVER WITH MANY IN BETWEEN. THICK FILM GENERALLY DOES NOT HAVE HOLES AND THUS WE USE THE "CHIP AND WIRE" AS WELL AS THE SURFACE MOUNT APPROACH.

THICK FILM

THERE ARE SEVERAL THINGS IN HYBRIDS THAT DIRECTLY IMPACT THE LAYOUT, THE MOST SERIOUS AND RESTRICTIVE IS THE PACKAGING WHICH DECIDES THE SUBSTRATE SIZE. BELOW IN FIGURE 1 ARE SOME EXAMPLES OF PACKAGES WHICH THE SUBSTRATE INSIDE IS THE ENTIRE LAYOUT AREA.



SINCE THE LAYOUTS ARE DONE ON ONE SIDE OF THE SUBSTRATE, THE SYMBOL GENERATION IS CRITICAL. MOST OF OUR ACTIVE DIE SYMBOLS ARE SET UP WITH THE CONNECT POINTS ON LAYER 1, WITH SOME EXCEPTIONS. TO DETERMINE WHETHER OR NOT THE CONNECT POINTS ARE TO BE ON LAYER 1 OR ANOTHER LAYER, YOU MUST SOMETIMES WORK BACKWARDS. BY THIS I MEAN THAT YOU COMPLETE A LAYOUT USING THE SYMBOLS THAT HAVE CONNECT POINTS ON LAYER 1, (CREATING FLOATING LINES) THEN YOU SEE WHERE YOU WILL NEED TO CHANGE THE LAYERING OF EACH CONNECT POINT. SOMETIMES YOU HAVE ONLY ONE POINT THAT REQUIRES ANOTHER LAYER AND SOMETIMES THERE ARE NUMEROUS POINTS THAT REQUIRE CHANGING. THEN YOU GO TO THE SYMBOL DRAWING AND CHANGE IT PER THE LAYOUT REQUIREMENTS. WHEN THE SYMBOL IS COMPLETE YOU CAN THEN GO TO THE LAYOUT AND REPLACE THE SYMBOL, MOVE THE VERTEXES AND ADD MINOR CONNECTIONS. IT WOULD BE EASIER IF WE COULD USE THE LAYER Ø, HOWEVER THEN IT WOULD SHOW UP ON ALL OF THE LAYERS.

THICK FILM RESISTOR SYMBOLS ARE MOST ALWAYS WITH CONNECT POINTS ON LAYER 1, THIS IS DUE TO THE PRINTING PROBLEMS THAT CAN OCCUR. PRINTING IN "VALLEYS" OF MATERIALS CREATES A UNEVEN THICKNESS AND THUS AN UNRELIABLE RESISTOR. CURRENTLY WE HAVE A RATHER EXTENSIVE LIBRARY INCLUDING RESISTOR SIZES IN 5 MIL INCREMENTS.

LAYOUTS

(INCLUDING PC, THICK FILM AND THIN FILM)

WHEN YOU HAVE PLACED THE COMPONENTS AND THE RATS NEST LINES ARE ON THE SCREEN, SOMETIMES IT IS NEAR IMPOSSIBLE TO DETERMINE IF YOU HAVE THE BEST PLACEMENT. IF WE HAVE CHANGED THE COMPONENTS AROUND AND THINK WE HAVE THE BEST PLACEMENT, WE PRINT OUT THE COMPONENTS WITH THE RATS NEST LINES TO GET A DIFFERENT PERSPECTIVE OF OUR PLACEMENT AT A SCALE OUR DESIGNERS ARE MOST FAMILIAR . THIS HELPS IN TWO WAYS, 1-TO GET A MORE CREATIVE VIEW AND 2- TO SEE IT IN WHAT MOST DESIGNERS ARE ACCUSTOMED TO, A DIFFERENT SCALE AND MEDIUM.

FILM COST REDUCTIONS AT NATIONAL SEMICONDUCTOR

FILM GENERATION CAN BE QUITE COSTLY, AT NATIONAL WE HAVE SEVERAL WAYS TO CREATE FILMS. IF A BOARD OR SUBSTRATE IS UNDER 2 INCHES, AND IS STILL IN THE PROTOTYPE STAGE WE USE RUBYLITH, CUT BY THE HP PEN PLOTTER.THE RUBY SCRIBE THAT WE USE CAN CUT A "D" SIZE RUBY WITH 10 MIL LINES AND SPACES IN APPROXIMATELY 3 MINUTES. THIS SAVES TIME IN TURN AROUND AND FILM COSTS FROM THE PHOTO SERVICE COMPANY AS WELL AS ALLOWING FOR MINOR CHANGES THAT OFTEN OCCUR IN THE SENSITIVE THICK FILM CIRCUITS. WE USE THE PHOTOPLOTTING PROCESS WHEN WE HAVE A BOARD THAT IS OVER 20 INCHES AT A 10/1 SCALE. WE ESTIMATE A COST SAVINGS OF 20% TO 50% IN USING THE RUBYLITH.

SMT CONSIDERATIONS FOR THE TELESIS SYSTEM

Donald DiMatteo

Senior Applications Engineer

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

As the trend in PCB design continues to move toward Surface Mount Technology, it becomes increasingly important for Printed Circuit Designers to have a basic knowledge of the processes and methods used in the manufacture and assembly of SMT boards. Understanding the information required for SMT board construction helps to ensure success and eliminate potential design problems. The following paper provides an overview of three SMT methods that may be used on the Telesis system.

INTRODUCTION

There is no one set of standards or methods that is right for every PC designer using SMT. However, by asking the correct questions, you can easily determine the approach best suited to your situation. Listed below are some of the questions that should be addressed prior to attempting SMT.

- What is the required output from the Telesis system?

- What pieces of artwork are required?

- How will the artwork be plotted?

- Do you have access to a Laser plotter?

- How many different types of packages will be used?

- What type of SMT are you considering?

TYPES OF SMT

There are four types of SMT:

TYPE 1 - SMT ON ONE SIDE ONLY

TYPE 2 - SMT AND THRU-HOLE

TYPE 3 - SMT ON BOTH SIDES AND THRU-HOLE

TYPE 4 - SMT ON BOTH SIDES

If you are designing TYPE 2 or 3 and are using both surface mount and thru-hole, you should be aware that the number of required apertures can easily exceed the maximum (24) that are allowed in the APERTURE-TAB FILE.

WHERE TO BEGIN?

Since the layer standard defines the use of all database layers, it is the obvious place to begin. Once you have determined and specified the database layers, you can begin to build your board symbols. An example of a layer standard follows.

EXAMPLE OF A LAYER STANDARD

```
(FILE NAME = LAYER STANDARD)
DBLAYER 1   COMPONENT-SIDE
DBLAYER 2   SOLDER-SIDE
DBLAYER 3   INTERNAL-SIGNAL
DBLAYER 4   INTERNAL-SIGNAL
DBLAYER 5   IMBEDDED-PLANE GND
DBLAYER 6   IMBEDDED-PLANE +5V
DBLAYER 15  COMPONENT-SOLDER-MASK
DBLAYER 16  SOLDER-SOLDER-MASK
DBLAYER 17  PASTE-MASK
DBLAYER 33  CARD-OUTLINE
DBLAYER 34  PLATING-BAR (BLANKED)
DBLAYER 35  BRD-DIM (BLANKED)
DBLAYER 36  DRAWING-FORMAT
DBLAYER 51  COMP-OUTLINES
DBLAYER 52  COMP-PADS
DBLAYER 53  COMP-PIN-NUM (BLANKED)
DBLAYER 54  REF-DES
DBLAYER 55  DEV-TYPE (BLANKED)
DBLAYER 152 COMP-VALUES (BLANKED)
DBLAYER 89  SILKSCREEN
DBLAYER 9Ø  TOOLING-CORNERS (BLANKED)
DBLAYER 98  NCDRILL-HEADER
DBLAYER 99  NCDRILL (BLANKED)
DBLAYER 1Ø1 RATSNEST
(ROUTING INFORMATION)
DBLAYER 44  VIA INHIBIT
DBLAYER 7Ø  ETCH INHIBIT ALL LAYERS
DBLAYER 71  ETCH INHIBIT LAY 1
DBLAYER 72  ETCH INHIBIT LAY 2
DBLAYER 146 ROUTING AREA
END
```

BUILDING THE BOARD SYMBOLS

There are three main methods of creating SMT pads for board symbols. Any, or all of these may be used on the same design at any time.

METHOD 1 - BUILDING A PAD SYMBOL USING PAD=RECTANGLE

```
SET THE CORRECT LAYER
ADD CONNECT POINT
PAD = .050 WIDE
    = .025 HEIGHT
ENTER THE LOCATION
```

If we use an SO-16, for example, the connect point would be .050 x .025, and the pin file would appear as follows:

SO-16 PINFILE (FILE FOR PACKAGE SYMBOL SO-16)

```
(NAME OF FILE SO-16-PIN)
(TELESIS STANDARD PACKAGE LIBRARY)
PINTYPE  A
PAD RECTANGLE = .050 x .025 COMPONENT-SIDE
PAD RECTANGLE = .050 x .025 PASTE-MASK
PAD RECTANGLE = .060 x .035 COMPONENT-SOLDER-MASK
PIN 1 -16 A
END
```

Note that while three sheets of artwork are generated, only two apertures are required. The same aperture is used for bonding pads as is for the PASTE MASK; a second aperture is used for the SOLDER MASK.

An example of a PLCC44 PIN FILE is shown below:

PLCC44 PIN FILE (6 APERTURES REQUIRED)

```
(FILE FOR PACKAGE SYMBOL PLCC44)
(NAME OF FILE PLCC44-PIN)
(TELESIS STANDARD PACKAGE LIBRARY)
PINTYPE A
PAD RECTANGLE = .Ø25 x .Ø5Ø COMPONENT-SIDE
PAD RECTANGLE = .Ø25 x .Ø5Ø PASTE-MASK
PAD RECTANGLE = .Ø35 x .Ø6Ø COMPONENT-SOLDER-MASK
PINTYPE B
PAD RECTANGLE = .Ø5Ø x .Ø25 COMPONENT-SIDE
PAD RECTANGLE = .Ø5Ø x .Ø25 PASTE-MASK
PAD RECTANGLE = .Ø6Ø x .Ø35 COMPONENT-SOLDER-MASK
PINTYPE C
PAD RECTANGLE = .Ø25 x .Ø75 COMPONENT-SIDE
PAD RECTANGLE = .Ø25 x .Ø75 PASTE-MASK
PAD RECTANGLE = .Ø35 x .Ø85 COMPONENT-SOLDER-MASK
PIN 1 C
PIN 2-6, 18-28, 4Ø-44 A
PIN 7-17, 29-39 B
END
```

METHOD 2 - CREATING A PAD SYMBOL USING THE ADD RECTANGLE COMMAND

SET THE CORRECT LAYER
ADD CONNECT POINT
PAD = POINT OR DIAMOND
PADS SIZE = ENTER
ENTER THE LOCATION

SET LINE WIDTH = Ø
SELECT ADD RECTANGLE
SPECIFY LOCATIONS OF DIAGONAL EXTENTS OF THE RECTANGLE

This method requires no special aperture. A sample pin file follows.

```
(FILE FOR PACKAGE SYMBOL PLCC44)
(NAME OF FILE PLCC44-PIN)
PINTYPE A
END
```

METHOD 3 - CREATING A PAD SYMBOL USING THE ADD LINE COMMAND

```
SET THE CORRECT LAYER
ADD CONNECT POINT
PAD = POINT OR DIAMOND
PAD SIZE = ENTER
LOCATE CONNECT POINT ON SCREEN

SET THE LINE WIDTH
ADD LINE
SELECT TWO LOCATIONS - LINE POINT ENDS ENTER
(NOTE THAT THE LINE EXTENDS HALF THE LINE WIDTH BEYOND THE TWO LOCATIONS
```

This method requires only one aperture. A sample pin file follows:

```
(FILE FOR PACKAGE SYMBOL PLCC44)
(NAME OF FILE PLCC44-PIN)
PINTYPE A
END
```

WHAT ARE THE ADVANTAGES/DISADVANTAGES OF EACH METHOD?

ADVANTAGES OF RECTANGULAR CONNECT POINT (METHOD 1)

1. Fast plotting time - each pad is flashed
2. Easy to create symbols
3. Minimal chance for errors

DISADVANTAGES

1. May require more than 24 apertures
2. May require special (non-standard) apertures
3. If designing Type 3 or Type 4, the mirror symbol command (available in Version 4) will not work.

ADVANTAGES OF CREATING PAD SYMBOL USING ADD RECTANGLE (METHOD 2)

1. Any pad height and width

2. Off-set pads can be created

DISADVANTAGES

1. Increase in plotting time

ADVANTAGES OF CREATING PAD SYMBOL USING ADD LINE (METHOD 3)

1. Fast plotting time
2. Off-set pads can be created
3. Any even width pads (24,26)

DISADVANTAGES

1. Must be careful when creating symbol
2. Round corners on pads (If you are using square apertures, they may be used to create square corners, however, this could result in spacing errors that are not flagged during PDRC).

CONCLUSION

With the release of 3.Ø, the Router and PDRC will recognize the pads, regardless of the method chosen for symbol creation. However, for ARTWORK to be accurately created, the patch titled ARTWORK PRE-RELEASE ØØØ3-P is required.

CARE PACKAGE FOR DESIGNERS ADAPTING TO SURFACE MOUNT TECHNOLOGY

Arn D. Buck

Department Head, PCB Design

Transcom Electronics

Portsmouth, Rhode Island

ABSTRACT

Conscientious thru-hole PCB designers have found that they can design more cost effective products by acquiring a working knowledge of board manufacturing processes. The intent of this paper is to make designers more familiar with SMT assembly and to present guidelines for SMT design.

INTRODUCTION

With the advent of SMT, an indepth understanding of board techniques has become mandatory. To a certain degree, an adept manufacturing engineer could compensate for a poorly designed thru-hole board. However, design deficiencies in an SMT board can easily cause a product to become unproducible.

The following outline shows the basic steps that are required for several types of SMT assemblies. In this outline, references are made to the primary and secondary side of the board. These terms are used because components are no longer restricted to one side of the board. The term "primary side" refers to the side of the board which is now called the component side; the term "secondary" refers to the solder side.

TYPES OF SMT ASSEMBLY

TYPE 1 SURFACE MOUNT ON ONE SIDE ONLY

1. APPLY SOLDER PASTE - PRIMARY SIDE
2. PLACE SMT COMPONENTS - PRIMARY SIDE
3. CURE SOLDER PASTE (PREHEAT)
4. REFLOW SOLDER (VAPOR PHASE, INFRA RED, OR HOT AIR)
5. CLEAN

TYPE 2 SURFACE MOUNT ON ONE SIDE ONLY

1. APPLY SOLDER PASTE
2. PLACE SMT COMPONENTS - PRIMARY SIDE
3. CURE SOLDER PASTE (PREHEAT)
4. REFLOW SOLDER (VAPOR PHASE, INFRA RED, OR HOT AIR)
5. CLEAN
6. INSERT THRU-HOLE COMPONENTS - PRIMARY SIDE
7. WAVE SOLDER
8. CLEAN

TYPE 3 SMT ON BOTH SIDES/THRU-HOLE COMPONENTS ON PRIMARY SIDE

1. APPLY SOLDER PASTE
2. PLACE SMT COMPONENTS - PRIMARY SIDE
3. CURE SOLDER PASTE

4. REFLOW SOLDER
5. CLEAN
6. TURN BOARD OVER
7. APPLY ADHESIVE - SECONDARY SIDE
8. PLACE SMT COMPONENTS - SECONDARY SIDE
9. CURE ADHESIVE
10. TURN BOARD OVER
11. INSERT THRU-HOLE COMPONENTS ON PRIMARY SIDE
12. WAVE SOLDER
13. CLEAN

TYPE 4 SMT COMPONENTS ON BOTH SIDES

1. APPLY SOLDER PASTE - SECONDARY SIDE
2. APPLY ADHESIVE - SECONDARY SIDE
3. PLACE SMT COMPONENTS - SECONDARY SIDE
4. CURE ADHESIVE AND SOLDER PASTE
5. TURN BOARD OVER
6. APPLY SOLDER PASTE - PRIMARY SIDE
7. PLACE SMT COMPONENTS - PRIMARY SIDE
8. CURE SOLDER PASTE
9. REFLOW SOLDER
10. CLEAN

FACTORS THAT AFFECT COMPONENT POSITIONING DURING REFLOW SOLDERING

Solder domes are formed over component lands as the solder paste reaches its melting point during the reflow process. The SMT components will initially float on these domes. If the component lands are properly spaced and sized, the components should remain in position plus or minus a few thousandths of an inch. If these conditions are not met the components may drift, stand on end, pop off the board, or be stressed.

The spacing between lands is critical to successful SMT assembly. Ideally, the lands should be positioned so that the component lands are centered on the apex of the solder dome. If the lands are placed too far apart, each dome will try to draw the component lead into its center. If one side wins this tug-of-war, the component will either tombstone (stand on end), or pop right off of the board. If neither side wins, there is a good possibility that the component will be stressed as it sinks into the solder and the solder hardens. The stress will be directed toward the center of the device's underside. If the stress is great enough, the component will eventually crack. Lands that are spaced too closely will push against each other and cause similar results as those spaced too far apart. In this case, however, the stress on the component would be directed toward the center of the upper side of the device. The package size tolerances should be considered when selecting components to ensure that they are compatible with the lands that are used for other sources.

It is also important to ensure that the mass of solder domes on opposite sides of a device are symmetrical. Components that do not have an even number of leads on opposite sides may have problems unless some countermeasure is taken. The recommended land pattern for an SOT-23 is a good example of such compensation. An SOT-23 is a three-leaded transistor package. The land that is used for the side that has a single lead is approximately 5Ø% larger than the lands for the other two leads. The greater mass of the solder dome on the single lead side offsets the two smaller solder domes on the opposite side and the component will float in without movement.

Vias, ground planes, or heavy traces should not be placed too close to component lands. This situation will attract the apex component land solder dome towards the other solder. The component will have a tendency to follow and be sucked into the larger solder dome. Allowing two components to share a common land will cause the same problem. To prevent this condition, vias should be kept a minimum of Ø.Ø25" away from a land and should connect to it with a trace that is a maximum of Ø.Ø1Ø" wide. If the electrical performance of a Ø.Ø1Ø" trace is not acceptable (i.e. power or ground), multiple Ø.Ø1Ø" traces should be utilized until the performance can be brought up to a satisfactory level. Keep in mind that Ø.ØØ8" line widths and spacings are common in SMT boards to allow traces to run between the lands of active com- ponents.

Another consideration is the width of component lands. Normally, the lands should not be more than Ø.ØØ5" wider than the solderable terminations of a component. Excessive land widths can cause the component to drift off axis and possibly end up on the land of an adjacent component.

If the traces on the board are tin/lead coated, careful thought should be given to their placement under small components. A heavy trace that is "off center" beneath a component can push up on the device while it is floating, forcing it askew. If traces must be run beneath small components (typically passive), they should be kept as narrow as possible and be placed exactly centered between the lands. Using soldermask over bare copper boards is another alternative.

Because of their cylindrical shape, MELF style components are not nearly as stable as counterpart chip styles. Some suppliers of these parts recommend that adhesive be used (even with reflow solder) to keep the MELFs from rolling off their lands or off of the board.

You should now have an understanding of the various factors that can cause component migration during soldering. The next section will take a look at land sizing for both active and passive SMT components.

SMT COMPONENT PACKAGING

The availability of surface mount devices is rapidly catching up with thru-hole components. Passive components are generally available in two basic styles; chip and MELF. The most commonly used chip packages are the 8Ø5, 12Ø6, and 121Ø. The numbers in these package names indicate the body size. For example, an 8Ø5 package description indicates that the device is Ø.Ø8Ø" long and Ø.Ø5Ø" wide and a 12Ø6 package is Ø.12Ø" long and Ø.Ø6Ø" wide.

Active components are available in SOT (small outline transistor), SOIC (small outline integrated circuit), PLCC (plastic leadless chip carrier), LCCC (leadless ceramic chip carrier), and flat pack packages. Within each classification the package size varies according to the number of pins used by the device. The SOT packages may also vary by the power handling capabilities of the device. SOIC packages are available with both "gull wing" leads and "J" leads. "J" leads are usually preferred because of their mechanical strength. Gull wing leads with a pitch of over Ø.Ø4Ø" should be avoided if possible due to their instability.

LAND PATTERNS FOR CHIP COMPONENTS

Although no current industry standard presently exists for chip components, most suppliers do recommend approximately the same land pattern for their 8Ø5 and 12Ø6 packages. These two patterns are shown in figure 1.

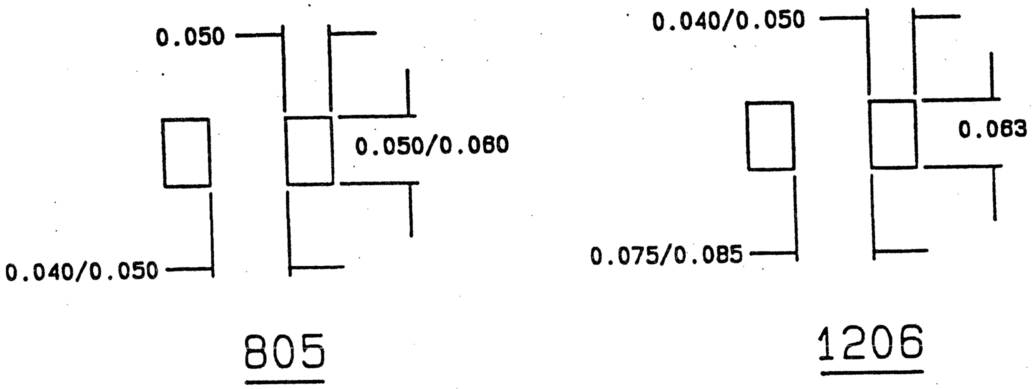
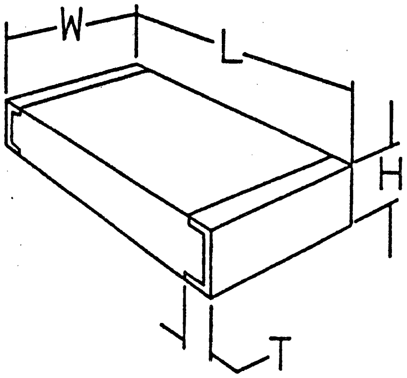


figure 1

The following formula can be used for calculating the land sizes of non-standard chip components.



L = COMPONENT HEIGHT

W = COMPONENT WIDTH

H = COMPONENT HEIGHT

T = SOLDERABLE TERMINATION WIDTH

PAD WIDTH = COMPONENT WIDTH + 0.003"

PAD LENGTH = THE SOLDERABLE TERMINATION WIDTH
+ THE COMPONENT HEIGHT + 0.005"

GAP BETWEEN LANDS = COMPONENT LENGTH - (2X) SOLDERABLE
TERMINATION WIDTH

Until recently, many designers would use different land sizes for wave soldering than for reflow soldering,however, that trend appears to be changing. Many are discovering that the reflow designed lands provide adequate area for wave soldering. This approach also reduces the number of CAD symbols required for SMT design.

LAND PATTERNS FOR SOT DEVICES

The most common SMT transistor package is the SOT-23. The recommended land pattern for this device is shown in figure 2. Note that one land is larger than the other. This is done to enable the device to float evenly during the soldering. Vendor specification sheets should be consulted for other SOT land patterns.

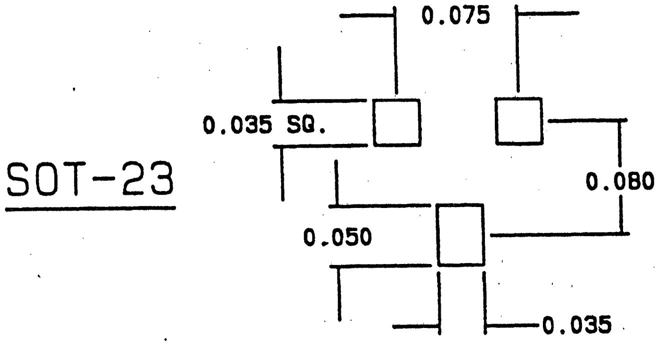
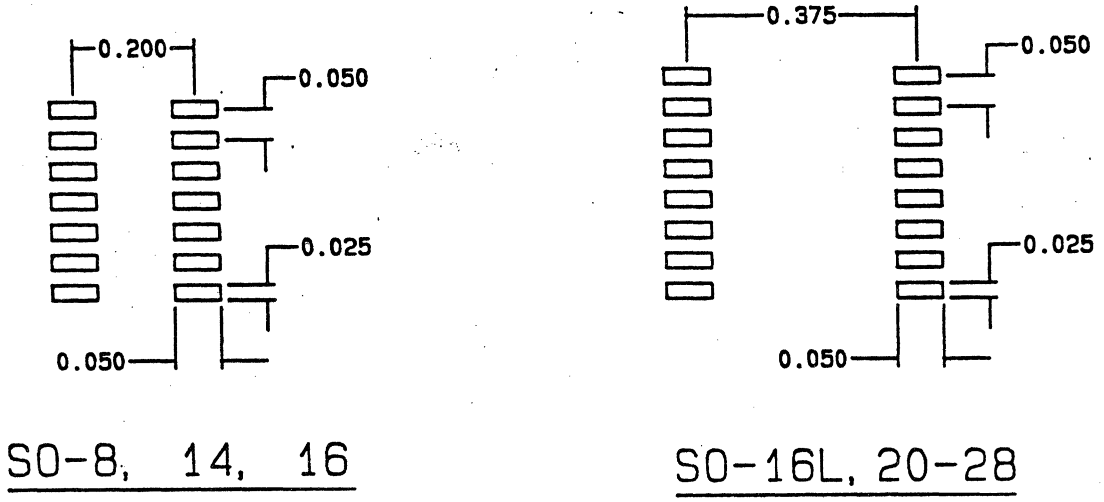


figure 2

LAND PATTERNS FOR SOIC DEVICES

Most SOIC's are available in 8 to 28 pin packages. The pins are oriented in a standard DIP arrangement. The center line spacing between the two rows of 8 to 16 pin packages is 2.ØØ". Some 16 pin SOIC devices are packaged similarly to the 2Ø to 28 pin devices which have a Ø.375" spacing between both rows. These 16 pin devices are referred to as SO-16L's. Figure 3 shows the land patterns for both types of SOIC's.



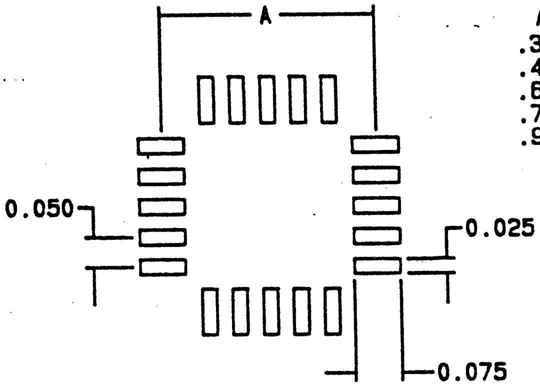
SO-8, 14, 16

SO-16L, 20-28

figure 3

LAND SIZING OF PLCC AND LCCC PACKAGES

SOIC packages with more than 28 leads are rare. Devices that require more leads are usually packaged in PLCC (plastic leadless chip carrier) or LCCC (leadless ceramic chip carrier) configurations. The recommended land pattern for standard PLCC packages is shown in figure 4. The space between rows of pins on LCCC configurations may vary from one supplier to another.



A	I.C. SIZE
.350	PCC-20
.450	PCC-28
.650	PCC-44
.750	PCC-52
.950	PCC-68

PCC PATTERN

figure 4

CONCLUSION

Much of the mystique surrounding SMT is vanishing as companies gain experience with the technology and pass the information on. Those who are just beginning to use SMT should still approach it with caution. SMT products are affected by many factors -- some will only be learned by firsthand experience. One of the best ways of minimizing the potential negative affects of these factors is with good communication between design and manufacturing -- a point that cannot be overstressed.

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

by

Carmen Serna

LARSE

SEPTEMBER 1986

Real Beaudette
Carmen Serna
Larse Corporation

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

1.0 PURPOSES

The purpose of this procedure is to explain how to generate a schematic diagram on the IBM/PC-Telesis Capture System.

The main topics will be Engineering Input, Symbol Creation, Planning and Placement, Processing, Transfer, and Back Annotation.

2.0 ACCURACY OF THE NET-DATA-BASE

It is extremely important to be sure that the information you put into the net data base is accurate before you proceed to take information out of the net data base and use it as the foundation for further work.

Read the sections of your operator manual covering library files and schematics to be sure that you adhere to the stated requirements.

3.0 RELATED DOCUMENTS

A. Telesis Series EDA-1000 Design Capture Systems (Utilities)

B. Telesis EDA-3000 (Creating a New Data Base)

4.0 PROCEDURES

This procedure is comprised of the following sections:

- Engineering input
- Symbols
- Planning and placement
- Annotation of text
- Processing
- Transfer of files
- EDA-300 NET-DATA-BASE
- Back annotation

5.0 ENGINEERING INPUT

A. Engineering Input is submitted in the form of a rough sketch. It is very important to get the latest copy of the engineer'S sketch to keep rework of the schematic to a minimum. This original input should be dated and the date recorded on the project schedule. All future changes and updates should be dated and recorded in order to estimate the hours a job will require and accurately schedule a completion date. (See appendix A.)

B. The engineer's sketch should include the following drawing conventions in order to facilitate the optimum layout of the schematic diagram:

1. Normal signal flow should be from left to right and top to bottom.

2. Indicate preferred pattern of stages and coupling circuits, especially when multiple sheets are necessary.

3. Include a complete material list, identified by the component description and/or company part number. Parts should be selected from the Preferred Parts Data Base Report which is obtainable from the LISTMAT.PAR.

4. New parts not yet assigned a part number or a purchase specification should be reported by the cognizant engineer to the Components Engineer with the following information: manufacturer's part number, project name, and the quantity required.

6.0 SYMBOLS

A. Any part on the PCB which has an electrical connection must also have a symbol on the schematic. Any schematic symbols that do not already exist need to be created. Symbols that do not exist in the SYSTEM.SYM library or those which require revision are created by the following process:

1. A sketch is made of the new or revised symbol, maintaining input to the left and output to the right.

2. Approval is obtained from the CAD System Manager.

3. Symbol is created.

4. New symbol is checked and approved by the CAD System Manager and the cognizant engineer.

5. Revised symbol is checked and approved by the CAD System Manager.

6. Symbols are transferred to the SYSTEM.SYM library by the CAD Lead Operator for immediate and future usage.

B. All symbols must be complete with reference designator, value or device type, pin names or numbers and part number.

1. Reference Designators

Reference designators are text labels of attribute 2, which appear inside a symbol. Reference designators must be 8 characters or less in length. Only the characters "A" through "Z" "0" through "9" "-" "+" "=" "%" "*" "?" and "/" are allowed. (See appendices B, C and D.)

2. Pin Numbers and Pin Names

Pin numbers and pin names are text labels with attributes 1, or 20 through 26, which associate the nodes on a symbol with the pins on a package. Pin numbers must only contain numeric characters that exist in the range 0 to 512. Pin numbers are only required if the operator wishes to assign a logic symbol to a specific slot on a device.

Pin names are similar to pin numbers except they may contain alphanumeric characters (up to 8). The first character of a pin name must be an alpha character (non-numeric), while remaining characters may be numeric. Pin names are required for every logic symbol node and must be the same as specified in the device description file for the physical board component. (See appendices C and D.)

3. Values/Device Type

Component values are labels (VALUE FIELD) with attributes 4, 6, 7 and 55 that are assigned to a symbol. A component label may contain up to 20 characters. Only the characters "A" through "Z" (capitals and lower case) "0" through "9" "+" "." "-" "=" "%" "*" "?" and "/" are allowed. (See appendices B, C and D.)

The value field may have any of the following types of parameters:

NOMINAL-VALUE/SCALE-FACTOR:	attribute 4
COMPONENT TOLERANCE ....:	attribute 6
COMPONENT STRESS ....:	attribute 7
EXTRA IDENTIFIER ....:	attribute 55

ALL ITEMS (LABELS) in the value field must be separated and all items must be located within the symbol's boundaries cell.

Real Beaudette
Carmen Serna
Larse Corporation

4. Part Number

All logic symbols must have a part number. This may be either a part number (a text label with attribute 3), or a user-defined device type. Refer to section 8.0.F. (See appendices B, C and D.)

7.0 PLANNING AND PLACEMENT

Keep in mind that the purpose of a schematic is to show the functions and relations of circuit components through the use of graphic symbols. It will be used for the design and analysis of circuits, instructional purposes, and trouble-shooting. It is important to maintain the original technical significance and to show the wiring of components in a circuit with clarity and simplicity.

Proper selection of schematic symbols serves an additional purpose. The symbols chosen can ultimately expand or restrict software capabilities at the board design level.

STEP 1: PLAN THE SCHEMATIC SHEETS

First plan how many sheets you will create for the schematic. Sketch each sheet to get an idea how you want to arrange your symbols. This will help you to place symbols faster.

Consider the following requirements for sheet one:

A. All notes, tables, and legends.

B. Decoupling Capacitors.

C. Power Supply Circuit.

D. Spares Section.

STEP 2: PLANNING FOR SYMBOL SELECTION

Good planning in symbol selection can have a great impact on the design process. The use of block symbols versus individual gate types determines the potential use of intelligent softwares available with our Telesis EDA-3000. Automatic placement, routing, pin function, and pin swapping are all affected by symbol selection.

Many engineers and draftspersons will prefer block symbols for IC's because they contain all the pin numbers and names within their boundaries and take less space in a schematic diagram. On the negative side these pins cannot be back annotated, therefore, pin swap and function swap options are lost.

Real Beaudette
Carmen Serna
Larse Corporation

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

Fully generic gate type symbols with pin names such as A,B,C, and Y and reference designators such as U* are most adaptable to automatic pin and function swap, two very important options to a designer when a board design is to achieve a high percentile of automatic routing. The importance of proper symbol type selection cannot be over-emphasized.

The goals of achieving high productivity through efficiency, accuracy, and expediency can be realized through the discriminating use of symbols at the schematic stage.

In planning a single sheet or a multiple sheet schematic, the following principles apply:

A. The diagram should be arranged so that it reads functionally from left to right, with the source or input, at the left, and the output at the right.

Multistage diagrams may be laid out in layers, still maintaining the left-to-right and top-to-bottom flow.

B. Lines should be routed as directly as possible with a minimum of direction changes and crossovers.

C. A diagram should be arranged to give prominence to main features.

D. Uniform density of all graphic symbols is desirable; do not crowd in one area while permitting large open areas to exist on a sheet.

E. The final diagram should have a balanced, symmetrical, and pleasing appearance.

STEP 3: PLACEMENT AND INTERCONNECTION

For each sheet of the schematic, place symbols and interconnect them according to your sketch. It is good strategy to place the larger symbols first, then the smaller ones, and leave most of the interconnection until you can see where the symbols will be.

STEP 4: FOLDER PREPARATION

In order to provide a source of materials used in generating the schematic and to make them available to individuals responsible for it's completion, it's checking, and the board design, a folder should be made, containing the following:

A. All engineering input.

B. Purchase specifications.

C. Copies of Device-Files including existing files, newly created files and information on to-be-created files.

D. Data deemed pertinent to the schematic creation.

E. Copies of Project schedules.

8.0 ANNOTATION OF TEXT

STEP 1: SYMBOL ANNOTATION

Reference Designators

Before generating a Telesis netlist from a schematic, the operator must assign reference designators to all symbols. Refer to section 6.B.1. However, the operator may supply "open-series" reference designators ending with an asterisk (*), for example, the reference designator R*. The Telesis Printed Circuit Board Design system (EDA-3000) automatically assigns available reference designators to components when the NET-DATA-BASE is created.

You can annotate the symbols with reference designators and pin numbers at any time during the creation of each schematic sheet. It is probably a good idea to do most of the annotation late in the process, after the symbols and connect lines have been placed, but if you are working with a dense schematic, it could be better to do most of the annotation before starting to add connections so you can tell what area of the schematic you are working with. The symbols you placed in the schematic already have dummy reference designators and pin number labels. Use the UPDATE TEXT command to change the reference designators.

When updating the reference designator, leave a gap of 5 to 10 numerals between sheets for each type of device to ease and simplify the task of future changes and additions. This is not applicable when your company prefers to leave their boards numbered left to right, top to bottom, with schematic annotated accordingly.

Pin Numbers/Pin Names

A. Pin numbers must only contain numeric characters that exist in the range 0 to 512. Pin numbers are only required if the operator wishes to assign a logic symbol to a specific slot on a device. Refer to section 6.B.2.

Pin names are similar to pin numbers except they may contain alphanumeric characters (up to 8). The first character of a pin name must be an alpha character (non-numeric), while remaining characters may be numeric. Pin names are required for every logic symbol node and must be the same as specified in the device description file for the physical board component.

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

With device description files present in the current project or system library on the EDA-300, the system automatically assigns all unassigned logic functions (no pin numbers annotated) to available slots on the device during creation of the net data base file.

B. Common Pins

You can annotate two or more symbols with the same reference designator, and put the same pin number on a connect point in each of them. You normally use this capability of the system to show common pins on each instance of a device such as clear lines and common voltages. For example, a set of resistors in a resistor pack are all tied to a common ground by pin 1.

C. Omitting Pin Number

If you are using device files, you may omit all pin number labels from each logic symbol of the gate type on the schematic. When you use the LOAD TEXT-NETLIST command, the system creates a NET-DATA-BASE using the pin name on the schematic, and assigns physical pin numbers for logical pins from information contained in the device descripton files.

D. Unused/Unconnected Pins

The operator should always use the signal name "N/C" with attribute 5 for unconnected pins. Connecting unused pins to this network suppresses error messages about unused pins when running the Design Check Program. Pins tied to an "N/C" signal name are not generated in the resulting netlist.

Values or Device Types

Component values are labels that are assigned to a symbol. Refer to section 6.B.3. A component label may contain up to 20 characters. Only the characters "A" through "Z" (capitals and lower case) "0" through "9" "+" "." "-" "=" "%" "*" "?" and "/" are allowed.

The value field may have any of the following types of parameters:

NOMINAL-VALUE/SCALE-FACTOR:	attribute 4
COMPONENT TOLERANCE ......:	attribute 6
COMPONENT STRESS ....:	attribute 7
EXTRA IDENTIFIER ....:	attribute 55

All items (LABELS) in the value field must be separated and all items must be located within the symbol's boundaries cell.

SEPTEMBER 1986

Real Beaudette
Carmen Serna
Larse Corporation

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

A. Nominal Value:

This parameter defines the nominal characteristic value of the component(s) in the component field. The format is a dimensionless floating-point number, with a maximum of 5 characters including a possible decimal point, with an optional scale factor. No space is allowed between the number and the scale factor.

B. Scale Factor:

Any of the following scale factor abbreviations can be used immediately after the value.

MEG = 10 EXP 6 (Meg)

K = EXP 3 (Kilo)

M = 10 EXP-3 (Milli)

U = 10 EXP-6 (Micro)

N = 10 EXP-9 (Nano)

P = 10 EXP-12 (Pico)

C. Component Tolerance:

You must enter a tolerance information with an attribute of 6, either in the form n% for symmetrical tolerances or in the form +n%/-n% for non-symmetrical tolerances, where n is a digit from 1-99.

D. Component Stress:

When applicable, you must enter the wattage or working voltages with an attribute of 7.

E. Extra Identifier:

When different components with identical values such as a package (SIP7) of 7 each 10K 2% 1/8W resistors and a package (SIP10) of 10 each 10K 2% 1/8W resistors are used within a schematic set, an extra identifier in the form of 7 each and 10 each must be entered using an attribute of 55.

Part Numbers

All logic symbols must have a part number. This may be either a part number, text label with attribute 3, or a user-defined device type. (See appendices B, C and D.)

SEPTEMBER 1986

Real Beaudette
Carmen Serna
Larse Corporation

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

A device type is a text label with a user-defined attribute between 80 and 99 or 120 and 127. The system prompts for the device type attribute during the TELNET input dialog. Both a part number and a user-defined device type may be present on the same symbol. TELNET will prompt the user for a user-defined device type (attribute). If the user explicitly specifies an attribute, and such an attribute exists on a symbol, then the device type will be the value of that field. Otherwise the device type will be the part number.

Device types must be 18 characters or less in length. Only the characters "A" through "Z" "0" through "9" "-" "+" "=" "&" "*" "?" "." and "/" are allowed.

The TELNET program uses either the part number or the device type as the Telesis device type (to make the association between the schematic symbol and the device to be used on the printed circuit board).

Instead of changing the part number label or adding the device type label to the symbol in the library, the operator can modify the occurrences of the symbol in the schematic.

STEP 2: FUNCTION TYPES

The operator may attach function types to logic symbols on the schematic. Function types are used to differentiate multiple and unlike logic functions on a single device or package. For example, the device 7423 contains both the NOR gate and the expanded NORX gate. By attaching the function type, NOR4, to the schematic symbol representing the 4-input NOR, and NOR4X to the symbol representing the expanded 4-input NOR, the TELNET program will differentiate the logic functions of symbols with the same device type. Function type lables are required for gate and pin assignment of multiple unlike logic functions on a single device when the operator creates the net data base on the Printed Circuit Board Design system.

A function type label is a text label with a user-defined attribute between 80 and 99, or 120 through 127, 80 is standard at Larse. TELNET prompts for the function type attribute during the TELNET input dialog. The function type attribute cannot be the same as the device type attribute (if used). A function type label may contain up to 18 characters or less in length. Only the characters "A" through "Z" "0" through "9" "-" "+" "&" "*" "?" "." and "/" are allowed.

If the operator specifies a function type attribute, and if a symbol has a function type field, the function type will appear in the resulting netlist.

STEP 3: SIGNAL NAMES

Signal names are text labels with attributes 5 and 10 through 19. Signal names must be 18 characters or less in length. Only the characters "A" through "Z" (capitals and lower case) "0" through "9" "." "-" "+" "=" "&" "*" "?" and "/" are allowed.

All signal names must touch the line with which they are associated. All bus lines should be named in the same manner.

When the signal text boundary touches more than one line, system's priority will associate it with the following line:

A. Below the text.

B. To the left of text.

C. Above the text.

D. To the right of text.

STEP 4: UNUSED/UNCONNECTED PINS

The operator should always use the signal name "N/C" with attribute 5 for unconnected pins. Connecting unused pins to this network suppresses error messages about unused pins when running the Design Check Program. Pins tied to an "N/C" signal name are not generated in the resulting netlist.

9.0 PROCESSING

STEP 1: PIN EXTRACTION

Pin extraction is the first of three processes used to generate a Telesis Text Netlist. This netlist will be transferred to the EDA-300 via a RS232 cable and used to generate a Telesis NET-DATA-BASE through the LOAD TEXT NETLIST command. The PINXTR process extracts pin information from your drawing set.

A drawing set means the various drawings (schematic sheets) that make up your schematic set. You can run pin extract on one sheet at a time or have the system run the extraction on all sheets. On a large schematic set, it may be easier to isolate discrepancies if the extraction is run on one sheet at a time. It will also save time when it comes to making changes in the future. In the NETXTR process, these files will be pooled into one file.

When the pin information has been extracted, a text file will be created. If no errors are reported, you are ready to proceed to NETXTR. If errors are found, locate them using a word processor software if available. If not, use the EDLIN.COM then repeat the PINXTR until you obtain a report that is clean and free of errors.

STEP 2: NET EXTRACTION

Net extraction is the second process used to achieve a Telesis netlist. It uses the pin extraction file to create a schematic capture netlist. You will use this netlist to create an EDA-300 netlist.

Using the pinlist file and the drawing, the system creates the NETLIST FILE. This file is in two sections: a symbol section and a signal section.

STEP 3: BILL OF MATERIAL REPORT

Once you have corrected your errors, you will create a List of Materials (BOM). To run the List of Materials (LOM), you need two files. The first is an EDA-1000 netlist, which you created earlier using NETXTR, and the second is the parts data base LISTMAT.PAR.

The parts data base will contain information on the Engineering Preferred Parts to be used on printed circuit boards. When the LOM generator is run, only information about parts used in the schematics being processed will be printed.

The value field of all symbols must have identical text and attributes which match exactly the PART REFERENCE entered in the LISTMAT.PAR data base. You must return to the schematic and fix all symbols text to remedy all errors reported such as NO DATA AVAILABLE.

You will need to rerun the processes discussed PINXTR and NETXTR, then rerun the LOM until all parts are accounted for.

You are now ready to transfer your Bill of Material (BOM) to the Manufacturing Information System, NCA, and to create the TELNET.

STEP 4: TELNET

Once the schematic capture netlist has been created, you must now transform this into the EDA-300 netlist. When the TELNET process is run, an error report will be created. If you find errors in the TELNET, investigate and determine the problem using a word processor or the EDLIN.COM then repeat TELNET until you obtain a report that is clean and free of errors.

SEPTEMBER 1986

Real Beaudette
Carmen Serna
Larse Corporation

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

STEP 5: RENUMBERING

When steps 1, 2 and 3 are completed, renumber the symbols using the upper left to lower right format. The system will accomplish this by the RENUM command. Save your drawing away with the changes made.

You must now go through the three processes again to create a new EDA-300 netlist. This time there should be no errors. The three commands are:

```
PINXTR
NETXTR
TELNET
```

NOTE: It is important that if you renumber your schematic drawings, you rerun the PINXTR, NETXTR, and TELNET processes. The system uses the symbol reference numbers during back annotation to make the appropriate changes to your schematic. If you renumber your schematic without rerunning PINXTR, NETXTR, and TELNET, the symbol reference numbers in the back annotation files will not match the symbol reference numbers of your schematic sheets. This will cause major problems. For example, maybe the symbol reference number 2 was a NOR gate and is now a resistor. In this case, the system will try to update the resistor with the NOR gate information.

STEP 6: DESIGN CHECK

Before the netlist can be transferred to the EDA, there are [three (3) out of five (5)] checking routines that need to be done for all schematics created for our corporation.

1. Check for unconnected pins.

2. Check for nets with only one connection.

5. Generate a sorted list or signal names and cross reference.

If the Design Check Report file shows errors, they must be fixed.

Now rerun PINXTR, NETXTR, TELNET and CHECK. You may have created an error while fixing this, so do not skip the check routine.

STEP 7: ENGINEER REVIEW

After completion of the design check with a no-error report and prior to the transfer of any files to the EDA-300, a copy of the schematic should be given to the responsible engineer for review and approval.

STEP 8: DEVICE-FILES

The Bill of Materials (BOM) should be checked against the Device-Files contained in the current project on the EDA-300. Every part must have a device file. All new and revised files must be checked and approved by the Cad System Manager.

Files may then be copied to another project to be transferred to New-Device-Files at the convenience of the Cad Lead Operator.

Creating and updating the Device-Files is the responsibility of the person who creates or makes changes on the schematic diagram. Updates must occur prior to creating the Net-Data-Base.

10.0 TRANSFER OF FILES

STEP 1: TELNET TRANSFER

Once the schematic has been finished and checked for errors, you are ready to transfer the netlist to the EDA-300. Using the File Transfer capabilities, transfer your TELNET file.

STEP 2: DEVICE-FILES TRANSFER (applicable for MULTI-EDIT SOFTWARE OWNER.)

All Device-Files that you have created using the MULTI-EDIT (ME) Software will be used by the EDA-3000 software when you build your NET-DATA-BASE. They need to be resident on the EDA-300 when the NET-DATA-BASE is built. Using the File Transfer capabilities with the ME batch transfer software, transfer these Device-Files to the EDA-300.

11.0 EDA-300 NET-DATA-BASE

Once the TELNET file and the Device-Files are on the EDA-300 you are ready to start building a project NET-DATA-BASE.

STEP 1: CREATING THE NET-DATA-BASE

Using your netlist and the Device-Files create the NET-DATA-BASE. Once the NET-DATA-BASE has been built, the NET-LOAD-LOG file will appear on the function screen. The first line of the file should indicate that the number of parsing errors was zero. If it did not then you would need to fix the errors reported in the log.

To fix the errors, you may need to return to the schematic on the EDA-1000 or fix a line in a Device-File. Whatever the error is, once fixed you will need to rerun delete the first Telnet in the EDA-300, transfer process, then repeat the LOAD-TEST NETLIST command with the new netlist, Device-Files, etc.

SEPTEMBER 1986

Real Beaudette
Carmen Serna
Larse Corporation

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

Old Net-Data-Bases

When reloading a text netlist, remember to delete all the old net-data-bases in the project or else the OLD ONE will be made twice as big with a more-than-doubling of the time to load it AND the resulting net-data-base will NOT be correct. If the board has been assigned, it should be deassigned at this time.

Special Use of $End in Load-Text-Netlist

A straight-forward way to check that the function specifications in DEVICE FILES on the EDA-3000 match the usage on the EDA-1000 (without doing a full LOAD-TEXT-NETLIST) is to insert a #END statement immediately after the $NETS statement in the Telesis netlist file. Use "@TEXT" on the EDA-3000 after transferring the ".TEL" file (or EDLIN on the EDA-1000 before transferring) to insert a single line containing the keyword "$END" immediately after the "$NETS" keyword, then pick LOAD-TEXT-NETLIST. LOAD-TEXT-NETLIST will then process all the "$FUNCTIONS" section of the netlist, checking that the functions called for are properly defined and will stop before processing any of the nets (thus saving most of the processing time in LOAD-TEXT-NETLIST). After cleaning up any problems with DEVICE FILES, a second run of LOAD-TEXT-NETLIST (with the "$END" removed again) will be necessary to have a full net-data-base created.

STEP 2: CHECKING REPORTS

A. Netlist Report

1. With an error-free NET-LOAD-LOG, you should perform a 100% line-by-line check of this netlist created on the EDA-300 back against your schematic. Every line and signal on the schematic must show up in the NETLIST-REPORT, and vice-versa. This is a key step which must never be skipped for accurate netlists.

2. Check for identical nets on two different net numbers because of spelling difference in net name.

3. Check for net with only one net pin number.

B. Component Report

1. Check all UNUSED pin numbers versus their Device-Files and/or spare gates shown on your schematic.

SEPTEMBER 1986

Real Beaudette
Carmen Serna
Larse Corporation

SCHEMATIC DRAWING CONVENTIONS
EDA-1000 TELESIS, IBM/PC

12.0 BACK ANNOTATION

Back Annotation - What is it?

Back Annotation is utilizing the system to update your schematic drawings to reflect changes made in another portion of the design process. Currently, there are three types of changes that are Back Annotatable.

A. If on your schematic you used generic reference designators, generic pin names, or both, the EDA-300 will automatically assign reference designators and pin numbers during the LOAD TEXT NETLIST process. Once your NET-DATA-BASE is created and correct you will want to return to your schematic drawings and update your symbol's reference designators and pin names to reflect the results of automatic assignment. Back Annotation will accomplish this. (For more information on Automatic Assignment, consult the PCB User Manual Vol 1 NETDB-16).

B. If on the EDA-300 you use the CHANGE-REF-DES command you can use Back Annotation to update this change to your schematic. (For more information on CHANGE-REF-DES, consult the PCB User Manual Vol 1, page PLACE-53-55).

C. If you use the function and pin swapping capabilities available on the EDA-300 Back Annotation can also be used to update your schematic drawings. (For a detailed description of Gate & Pin Swapping, consult your PCB User Manual Vol 1, PLACE-23).

D. When back annotating your schematic, if the following message occurs, do not be alarmed:

   "Warning: The following pins on symbol 10 on sheet 1 \Telesis\____.DWG were never back annotated: 5 6 7 8"

   Be sure to check your back annotation log carefully, to make sure that the warning messages are for pins NOT CONNECTED (N/C).

LARSE

PROJECT SCHEDULE - SCHEMATIC CAPTURE

□ PROTOTYPE
□ PROTOTYPE CHANGES
□ PRODUCTION

PROJECT: ______________
DWG NO.: ______________
BOARD SIZE: ______________
BOARD DENSITY: ______________

ENG: ______________
DFTR: ______________
DATE: ______________
CHARGE NO.: ______________

TASK DESCRIPTION	ORIGINAL INPUT	ADDITIONAL INPUT					HOURS TAKEN														TOTAL HOURS
	DATE / E/TIME	DT/HR	DT/HR	DT/HR	DT/HR	TOTAL	M	T	W	T	F	S	S	M	T	W	T	F	S	S	
PRELIMINARY EVALUATION	/	/	/	/	/																
OUTSTANDING ECO INCORPORATION	/	/	/	/	/																
NEW LOGIC-SYMBOL CREATION	/	/	/	/	/																
SCHEMATIC SET-UP	/	/	/	/	/																
SCHEMATIC ENTRY	/	/	/	/	/																
SCHEMATIC CHECK VERSUS INPUT	/	/	/	/	/																
PIN EXTRACTION	/	/	/	/	/																
NET EXTRACTION	/	/	/	/	/																
LOGIC DESIGN CHECK	/	/	/	/	/																
SCHEMATIC REWORK	/	/	/	/	/																
TELNET TRANSFER TO CAD	/	/	/	/	/																
BOM EXTRACTION	/	/	/	/	/																
BOM TRANSFER TO NCA	/	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
BOM. PRELIMINARY CREATION	/	/	/	/	/																
NEW DEVICE-FILES CREATION	/	/	/	/	/																
NEW DEVICE-FILES TRANSFER	/	/	/	/	/																
LOAD DEVICE-FILES	/	/	/	/	/																
LOAD TEXT-NETLIST	/	/	/	/	/																
REWORK	/	/	/	/	/																
RELOAD TEXT-NETLIST	/	/	/	/	/																
CREATE ALL REPORTS	/	/	/	/	/																
PRINT ALL REPORTS	/	/	/	/	/																
PENPLOT SCHEMATIC, PRELIM	/	/	/	/	/																
CHECK SCHEMATIC VERSUS NETS	/	/	/	/	/																
ESTIMATE PCB DSGN TIME, PRELIM	/	/	/	/	/																
	/	/	/	/	/																
BACK ANNOTATION	/	/	/	/	/																
FINAL REWORK	/	/	/	/	/																
PENPLOT SCHEMATIC, FINAL	/	/	/	/	/																
PROM DOCUMENTATION	/	/	/	/	/																
BOM FORMALIZATION	/	/	/	/	/																
PROJECT BACKUP ON DISKETTES	/	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
PACKAGE RELEASE	/	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

TOTAL ESTIMATED TIME □ TOTAL □ TOTAL SCHEMATIC DESIGN TIME □

APPENDIX "A"

DISCRETES

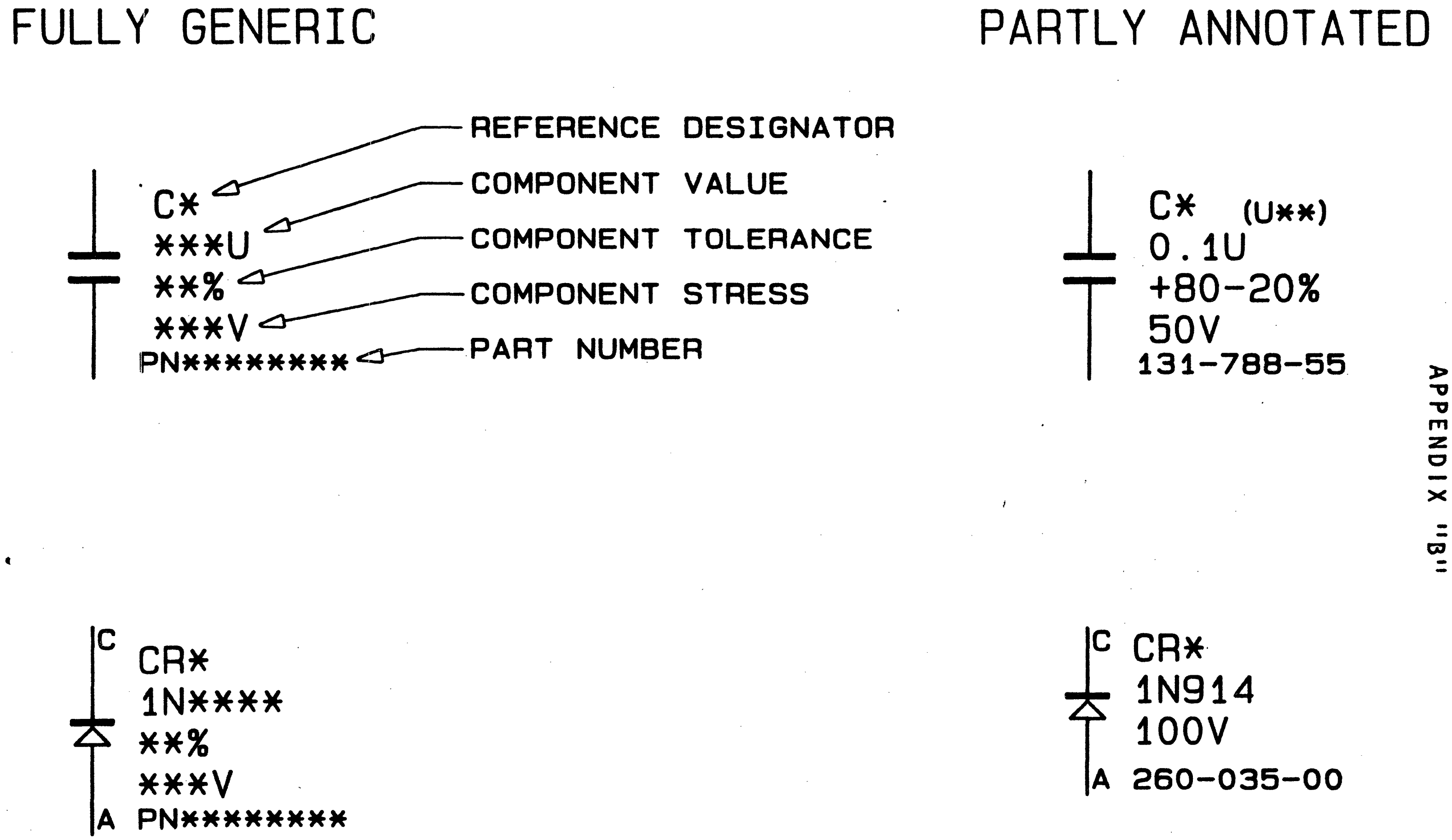
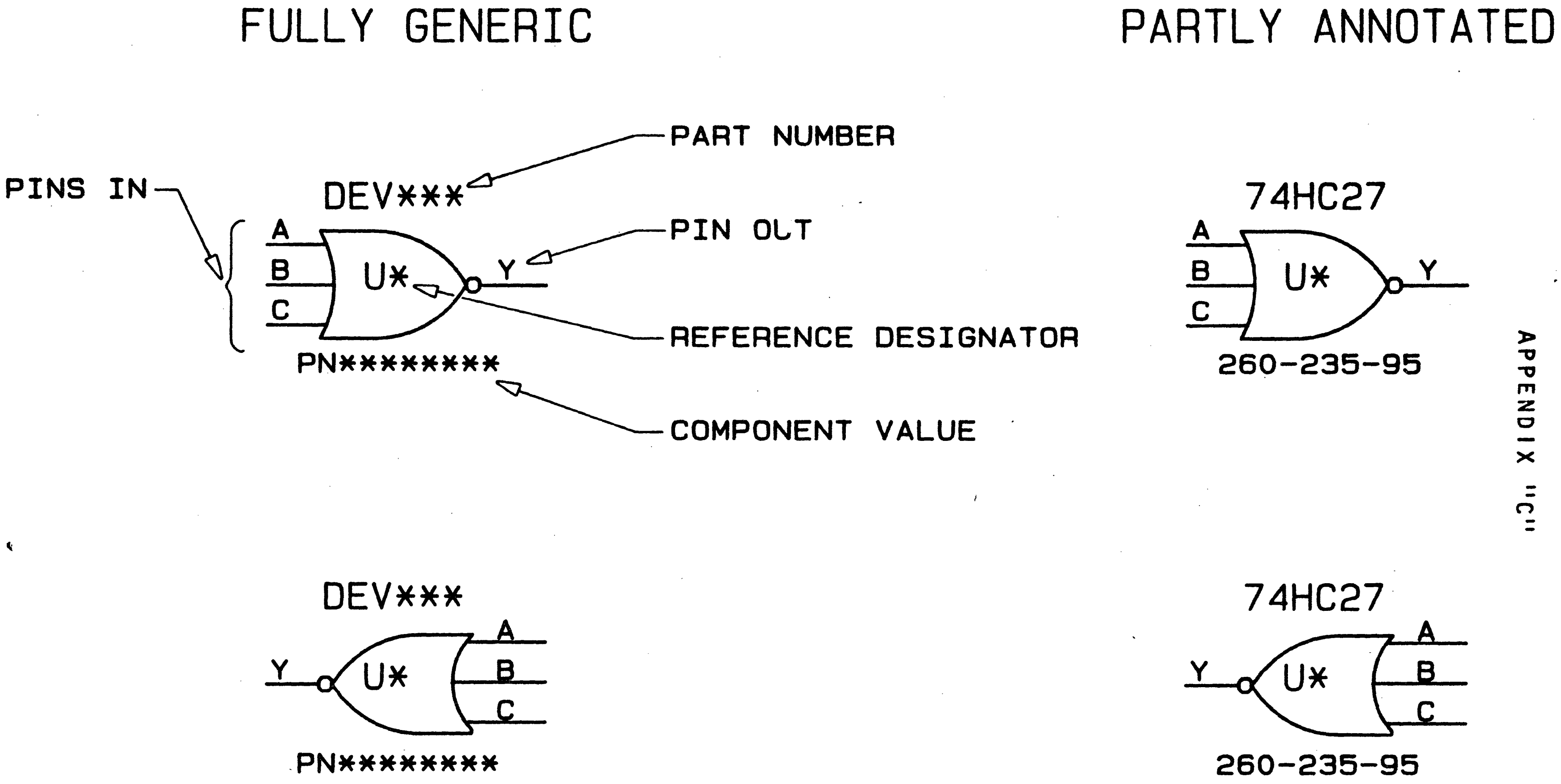


FIGURE 4

IC GATES


FULLY GENERIC

PARTLY ANNOTATED

FIGURE 2

APPENDIX "C"

IC'S

FULLY ANNOTATED EXCEPT FOR REFERENCE DESIGNATOR

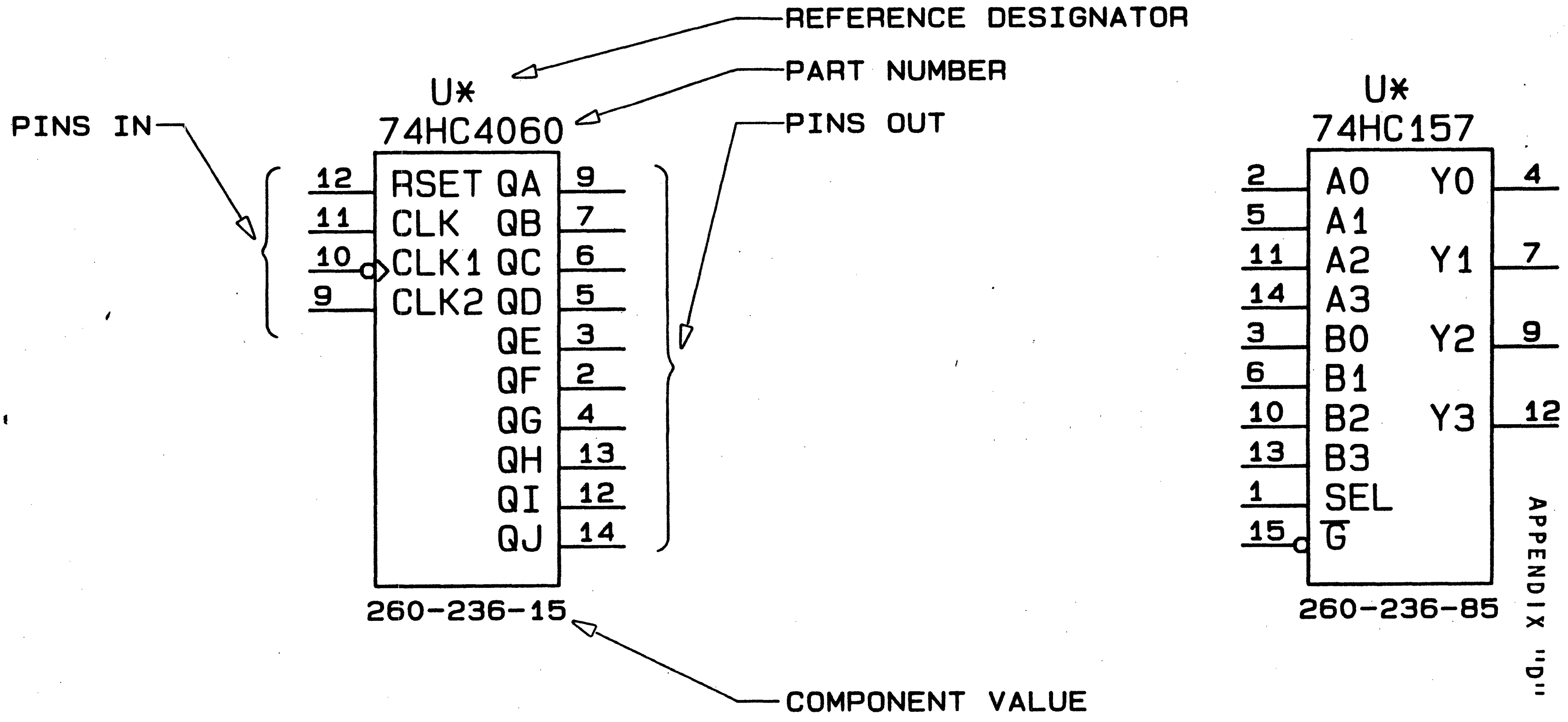
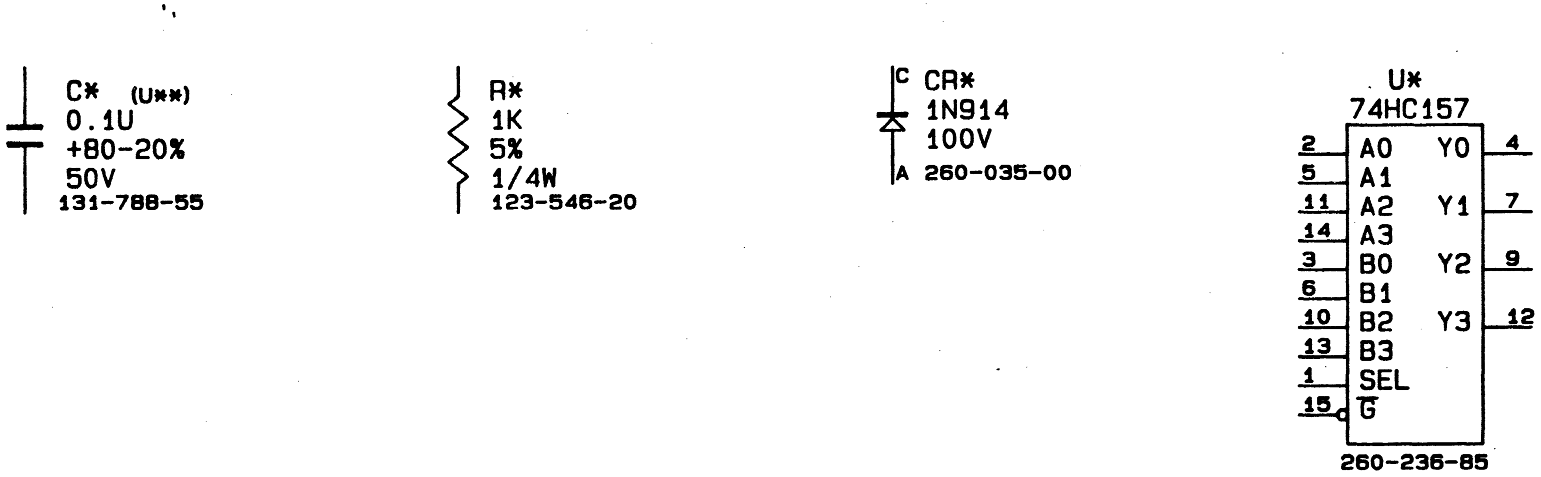
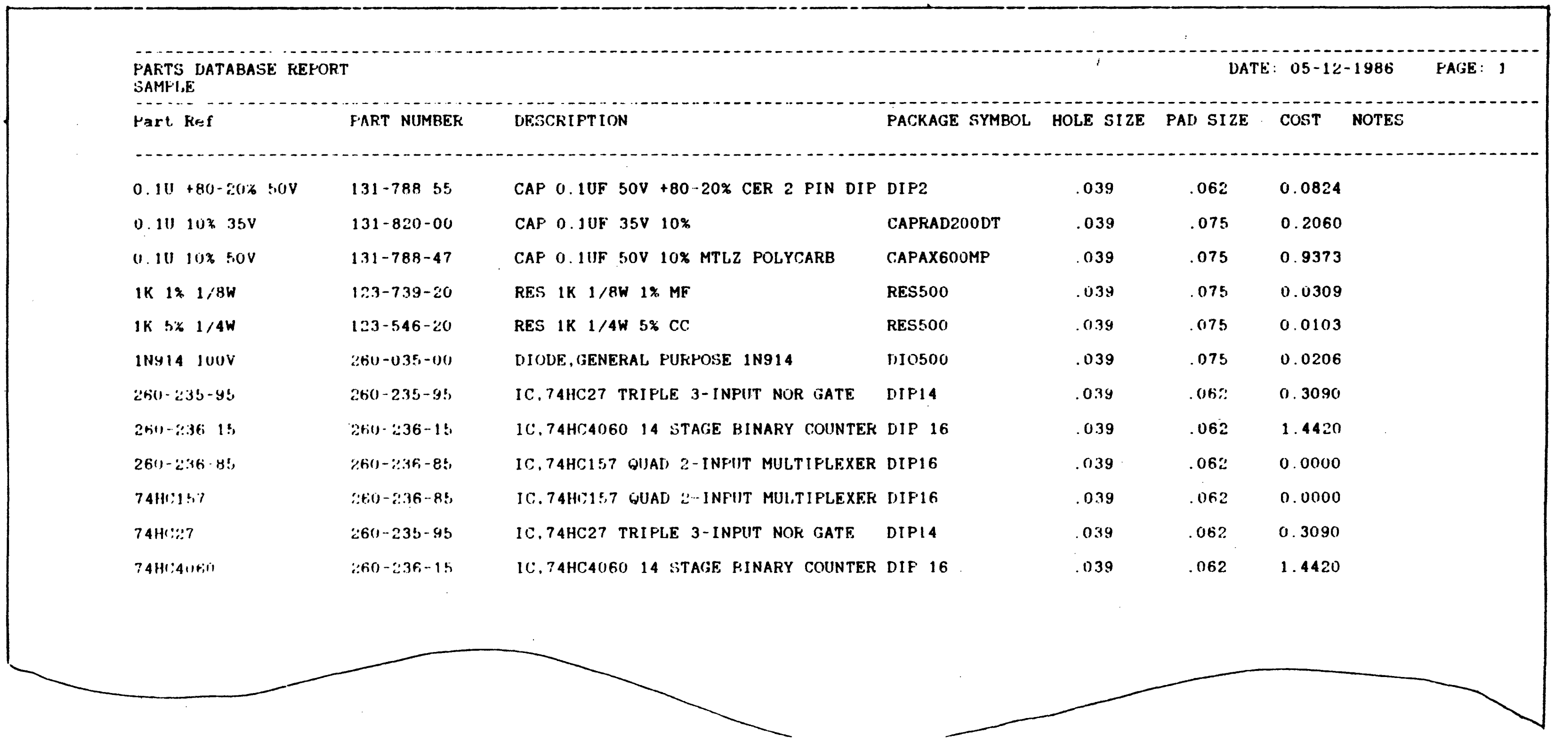


FIGURE 3



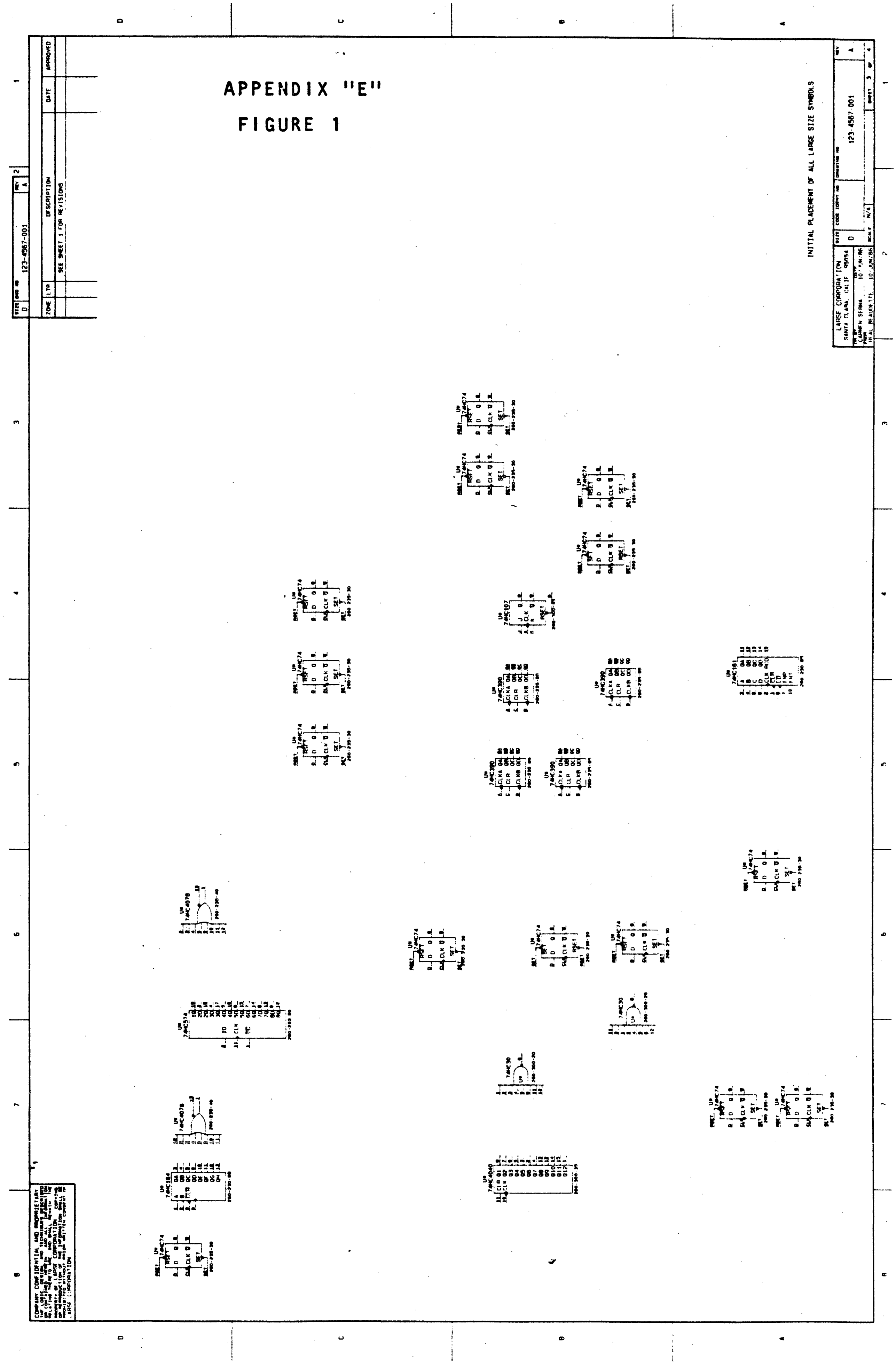


```
PARTS DATABASE REPORT                                                                          DATE: 05-12-1986     PAGE: 1
SAMPLE
Part Ref             PART NUMBER     DESCRIPTION                              PACKAGE SYMBOL  HOLE SIZE  PAD SIZE   COST    NOTES

0.1U +80-20% 50V     131-788 55      CAP 0.1UF 50V +80-20% CER 2 PIN DIP      DIP2              .039       .062     0.0824
0.1U 10% 35V         131-820-00      CAP 0.1UF 35V 10%                        CAPRAD200DT       .039       .075     0.2060
0.1U 10% 50V         131-788-47      CAP 0.1UF 50V 10% MTLZ POLYCARB          CAPAX600MP        .039       .075     0.9373
1K 1% 1/8W           123-739-20      RES 1K 1/8W 1% MF                        RES500            .039       .075     0.0309
1K 5% 1/4W           123-546-20      RES 1K 1/4W 5% CC                        RES500            .039       .075     0.0103
1N914 100V           260-035-00      DIODE,GENERAL PURPOSE 1N914              DIO500            .039       .075     0.0206
260-235-95           260-235-95      IC,74HC27 TRIPLE 3-INPUT NOR GATE        DIP14             .039       .062     0.3090
260-236 15           260-236-15      IC,74HC4060 14 STAGE BINARY COUNTER      DIP 16            .039       .062     1.4420
260-236-85           260-236-85      IC,74HC157 QUAD 2-INPUT MULTIPLEXER      DIP16             .039       .062     0.0000
74HC157              260-236-85      IC,74HC157 QUAD 2-INPUT MULTIPLEXER      DIP16             .039       .062     0.0000
74HC27               260-235-95      IC,74HC27 TRIPLE 3-INPUT NOR GATE        DIP14             .039       .062     0.3090
74HC4060             260-236-15      IC,74HC4060 14 STAGE BINARY COUNTER      DIP 16            .039       .062     1.4420
```

APPENDIX "D"
FIGURE 4

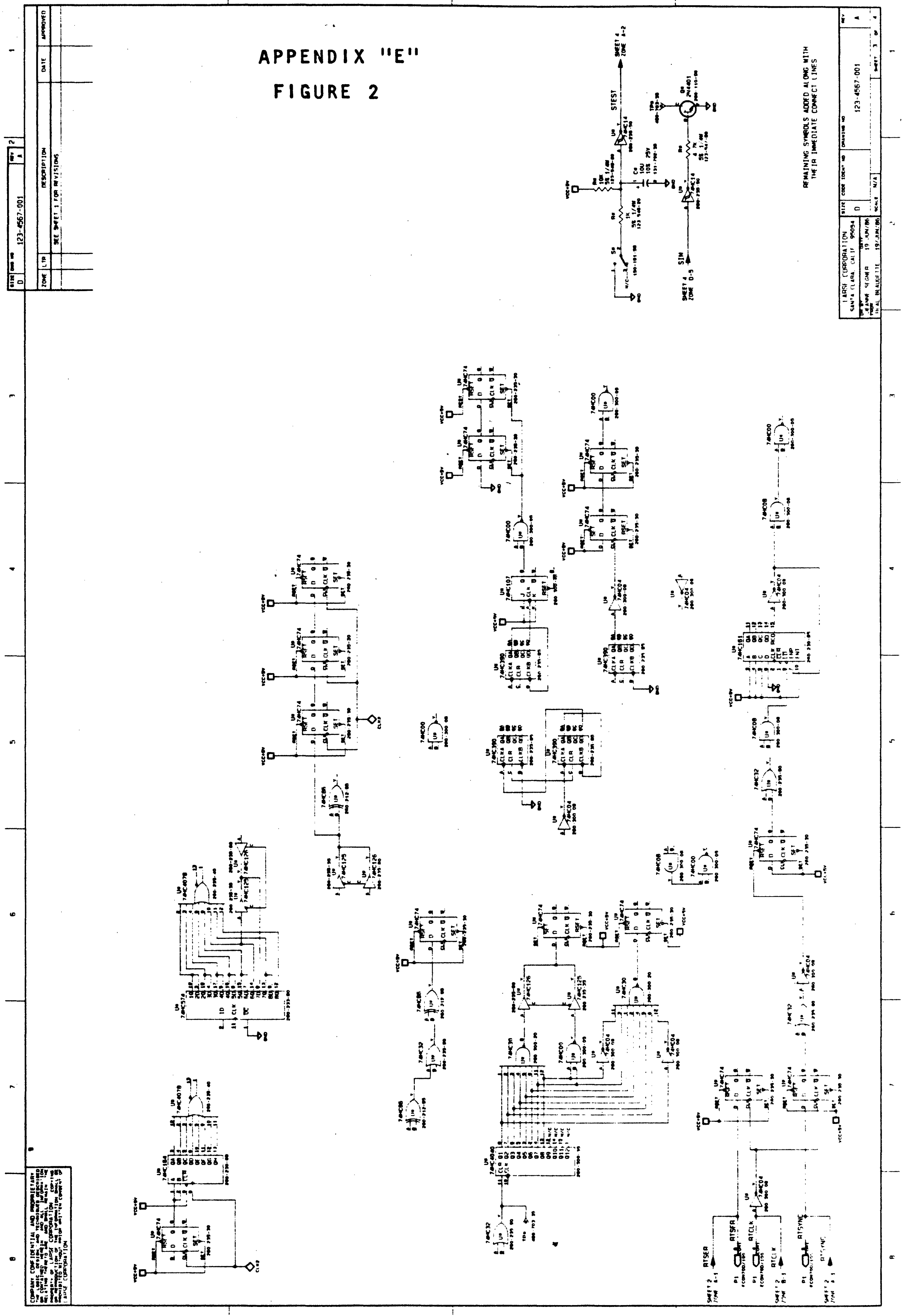
APPENDIX "E"

FIGURE 1



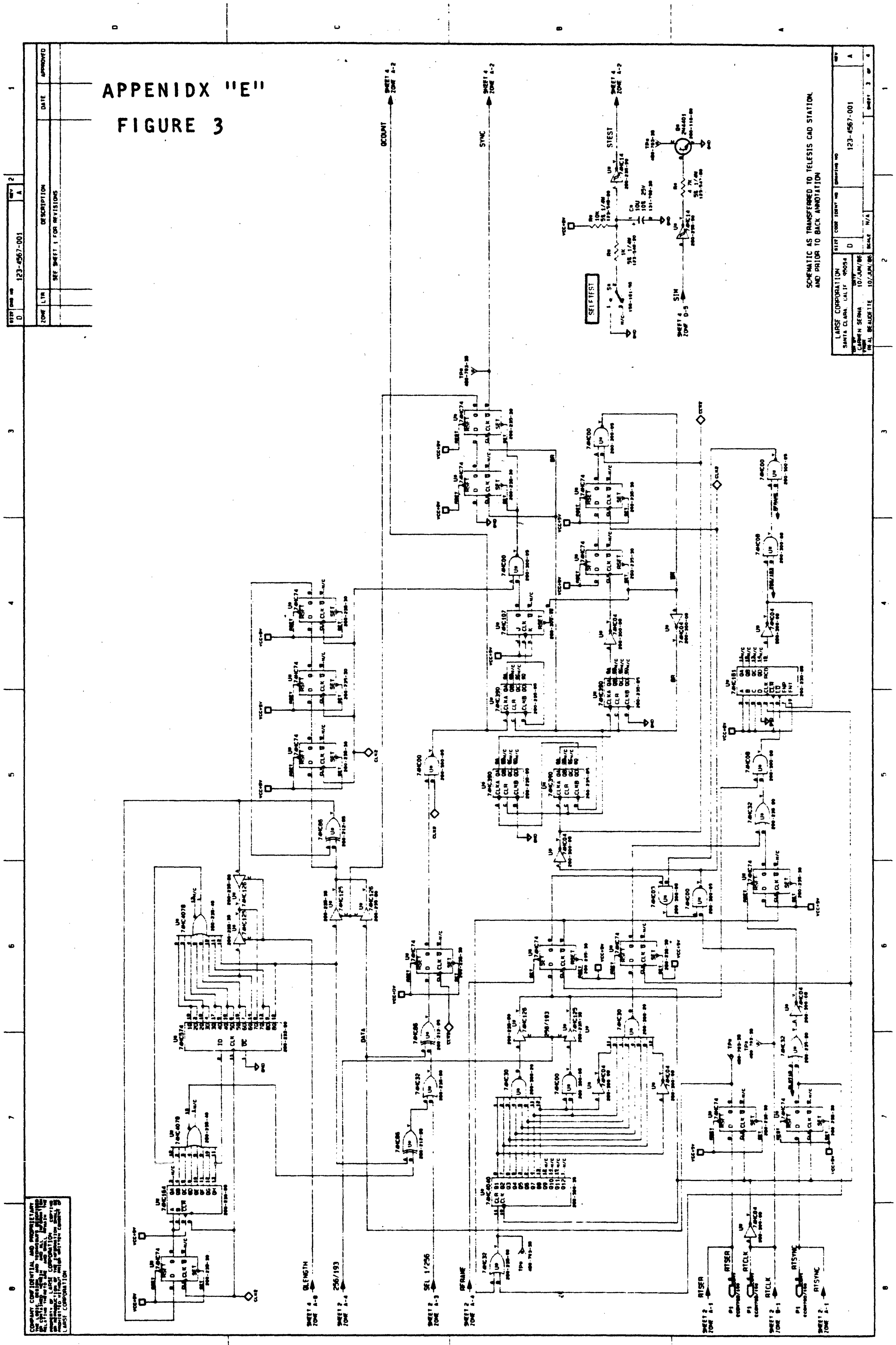
APPENDIX "E"

FIGURE 2



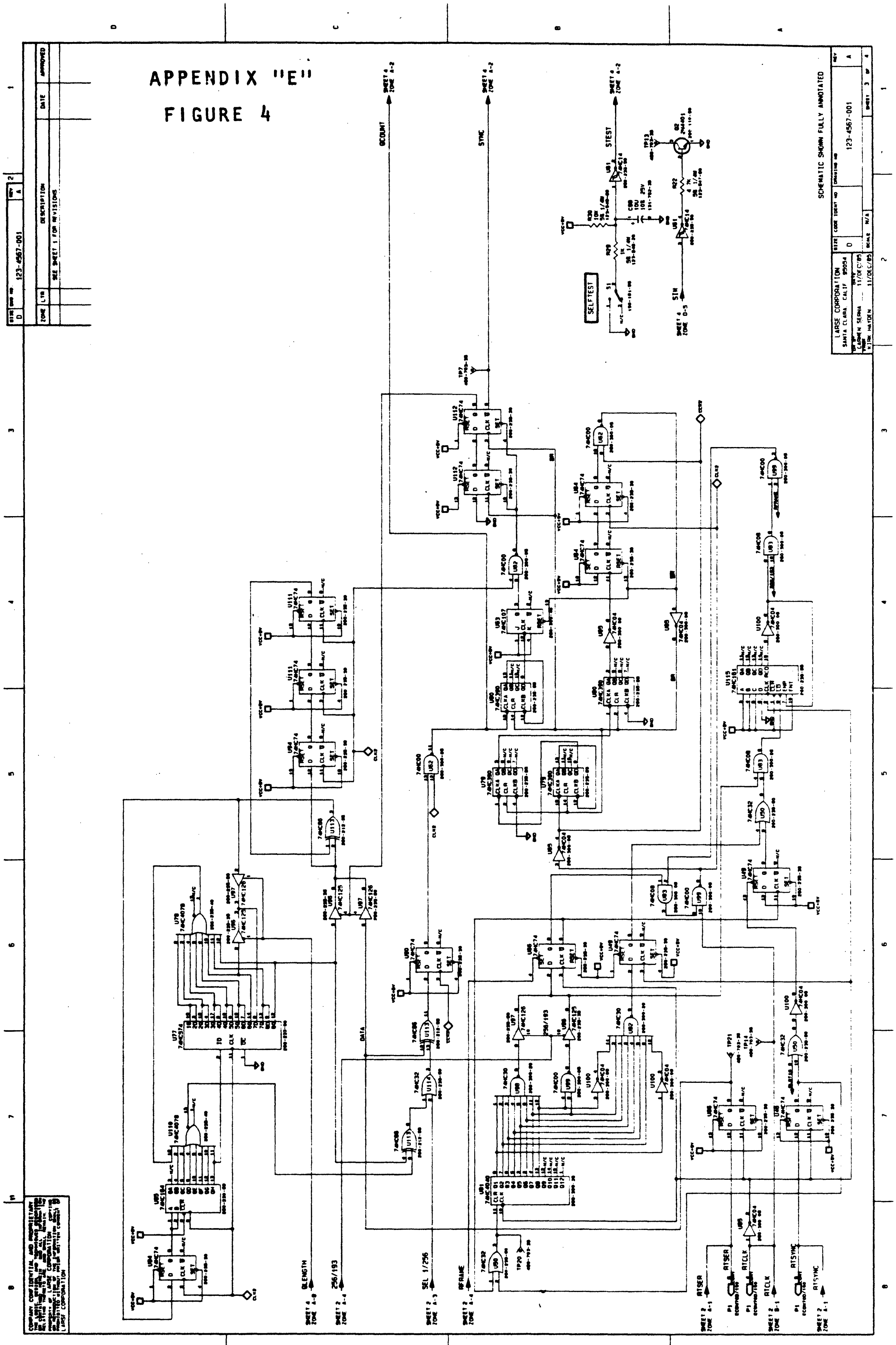
APPENIDX "E"

FIGURE 3



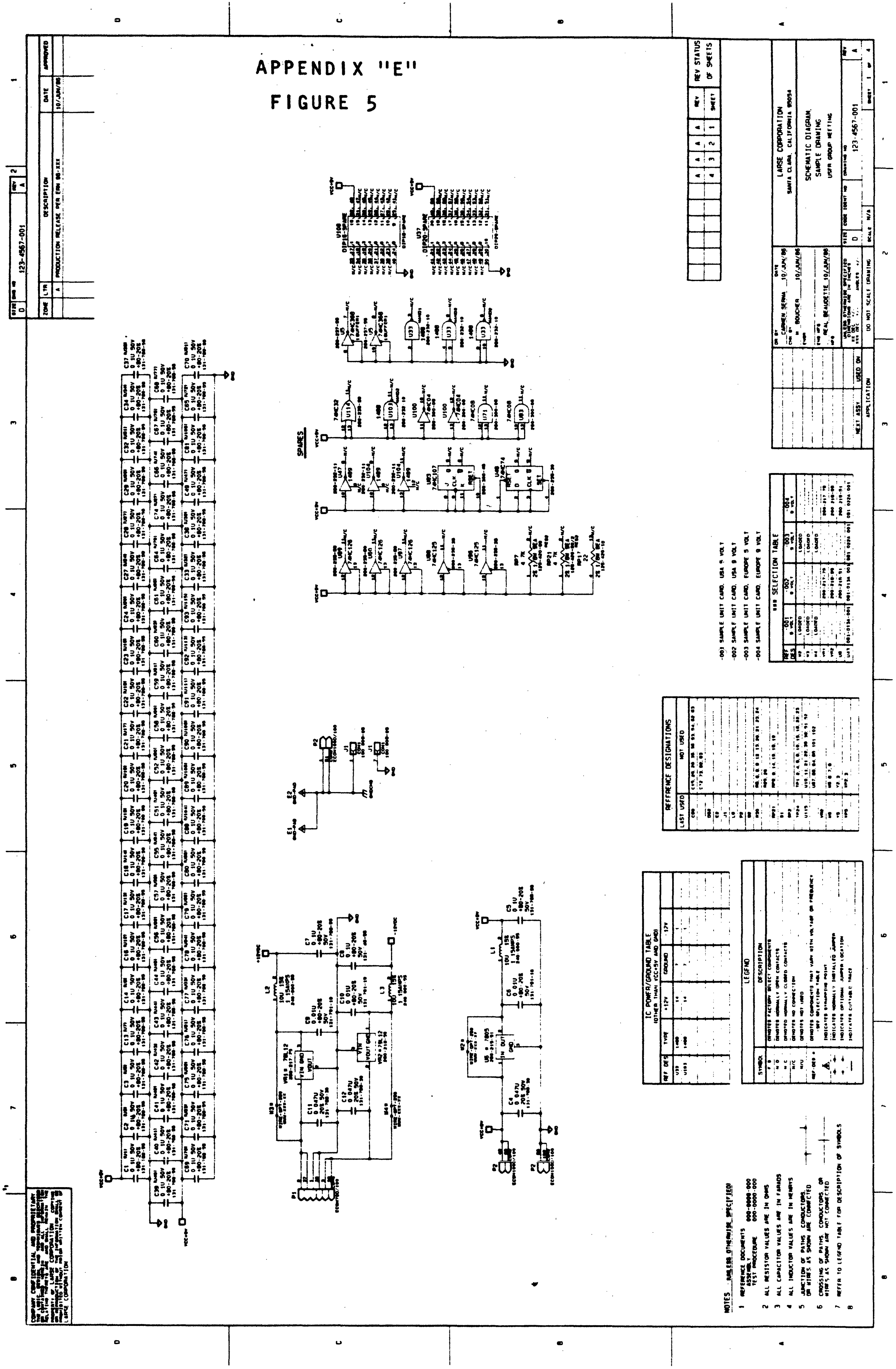
APPENDIX "E"

FIGURE 4



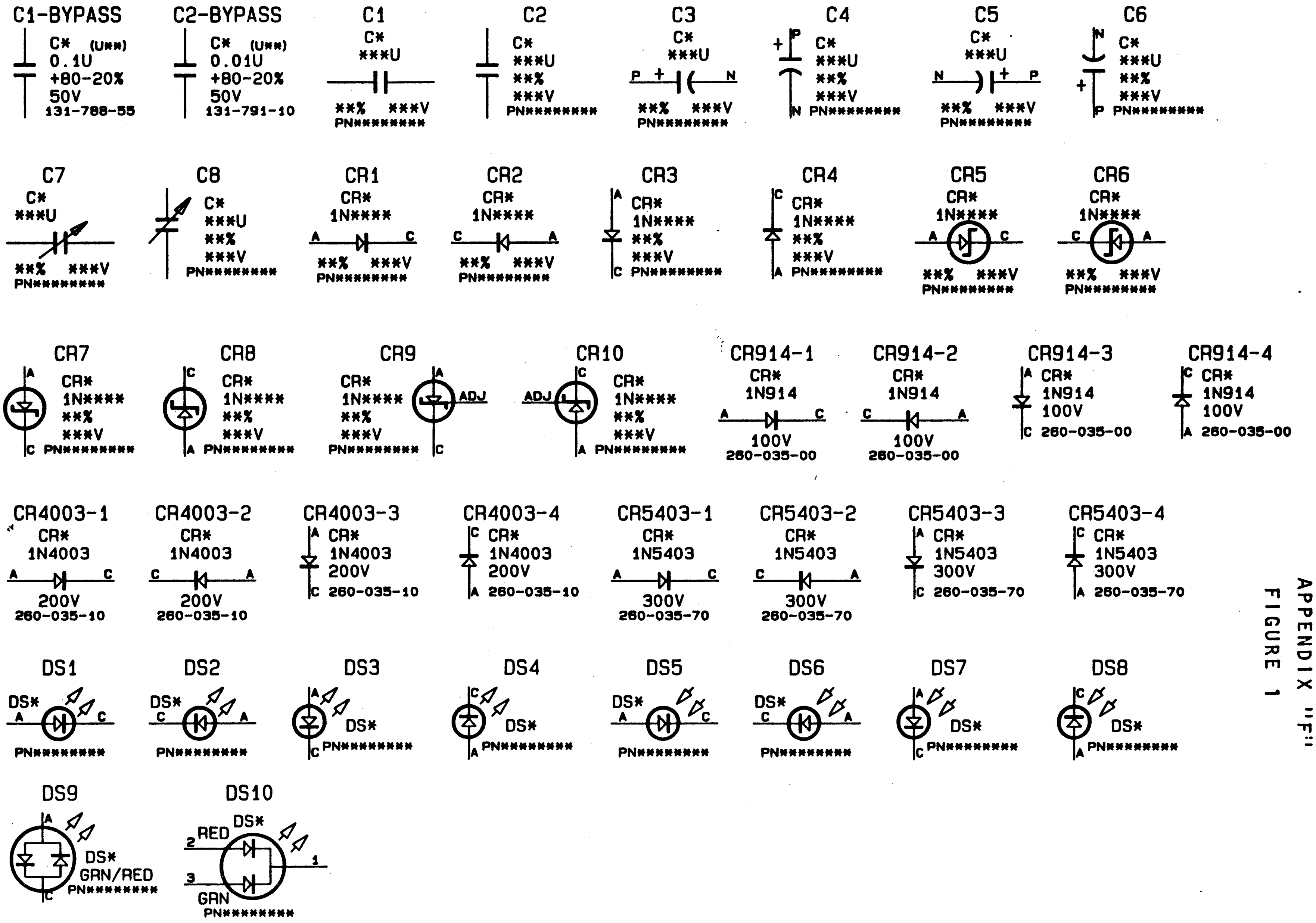
APPENDIX "E"

FIGURE 5

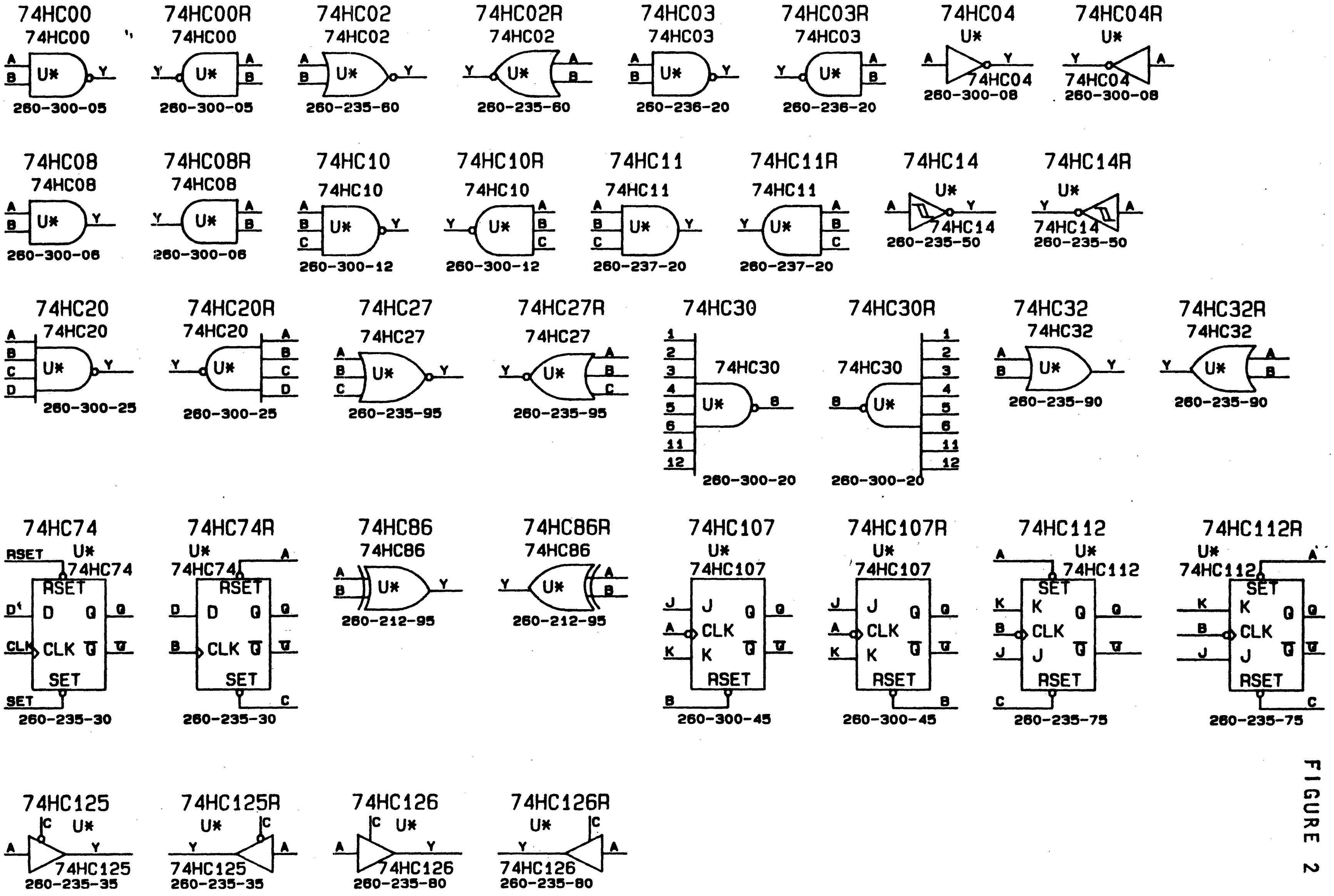


LARSE CORPORATION
SANTA CLARA, CALIFORNIA 95054
SCHEMATIC DIAGRAM,
SAMPLE DRAWING
USER GROUP MEETING
123-4567-001
PRODUCTION RELEASE PER ECN 00-XXX
REV STATUS OF SHEETS
SPARES
REFERENCE DESIGNATIONS
SELECTION TABLE
IC POWER/GROUND TABLE
LEGEND
NOTES UNLESS OTHERWISE SPECIFIED
ALL RESISTOR VALUES ARE IN OHMS
ALL CAPACITOR VALUES ARE IN FARADS
ALL INDUCTOR VALUES ARE IN HENRYS
-001 SAMPLE UNIT CARD, USA 5 VOLT
-002 SAMPLE UNIT CARD, USA 9 VOLT
-003 SAMPLE UNIT CARD, EUROPE 5 VOLT
-004 SAMPLE UNIT CARD, EUROPE 9 VOLT
COMPANY CONFIDENTIAL AND PROPRIETARY
DO NOT SCALE DRAWING

APPENDIX "F"
FIGURE 1

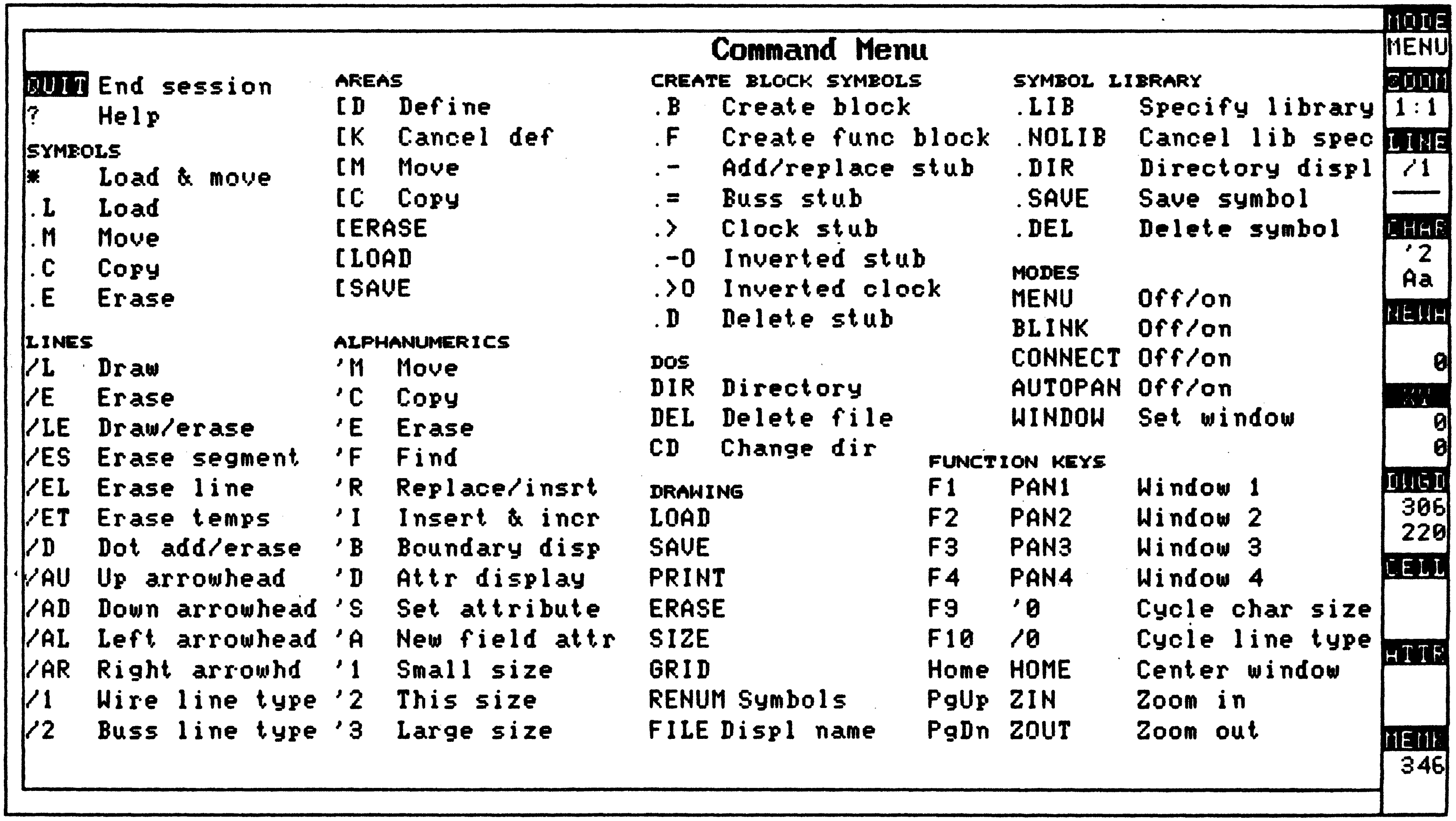


SYSTEM LIBRARY
DIS-GEN2
11-FEB-86



SYSTEM LIBRARY
74HCXX-1
7-FEB-86

APPENDIX "F"
FIGURE 2



Command Menu

```
QUIT End session       AREAS               CREATE BLOCK SYMBOLS         SYMBOL LIBRARY
?    Help              [D   Define         .B   Create block            .LIB     Specify library
SYMBOLS                [K   Cancel def     .F   Create func block       .NOLIB   Cancel lib spec
*    Load & move       [M   Move           .-   Add/replace stub        .DIR     Directory displ
.L   Load              [C   Copy           .=   Buss stub               .SAVE    Save symbol
.M   Move              [ERASE              .>   Clock stub              .DEL     Delete symbol
.C   Copy              [LOAD               .-0  Inverted stub
.E   Erase             [SAVE               .>0  Inverted clock          MODES
                                           .D   Delete stub             MENU     Off/on
LINES                  ALPHANUMERICS                                    BLINK    Off/on
/L   Draw              'M   Move           DOS                          CONNECT  Off/on
/E   Erase             'C   Copy           DIR  Directory               AUTOPAN  Off/on
/LE  Draw/erase        'E   Erase          DEL  Delete file             WINDOW   Set window
/ES  Erase segment     'F   Find           CD   Change dir
/EL  Erase line        'R   Replace/insrt                           FUNCTION KEYS
/ET  Erase temps       'I   Insert & incr  DRAWING                  F1    PAN1     Window 1
/D   Dot add/erase     'B   Boundary disp  LOAD                     F2    PAN2     Window 2
/AU  Up arrowhead      'D   Attr display   SAVE                     F3    PAN3     Window 3
/AD  Down arrowhead    'S   Set attribute  PRINT                    F4    PAN4     Window 4
/AL  Left arrowhead    'A   New field attr ERASE                    F9    '0       Cycle char size
/AR  Right arrowhd     '1   Small size     SIZE                     F10   /0       Cycle line type
/1   Wire line type    '2   This size      GRID                     Home  HOME     Center window
/2   Buss line type    '3   Large size     RENUM Symbols            PgUp  ZIN      Zoom in
                                           FILE Displ name          PgDn  ZOUT     Zoom out
```

```
MODE
MENU
ZOOM
1:1
LINE
/1
CHAR
'2
Aa
NEWH
0
XY
0
0
DWGI
306
220
CELL
ATTR
MEMK
346
```

COMPARING THE EDA 1200 WITH DASH IV

Thomas J. Miller

Manager Applications Integration

Telesis Systems Corporation

Chelmsford, Massachusetts

ABSTRACT

Telesis will soon be releasing a new product based on the FutureNet Dash IV product. The following paper describes the major enhancements made by FutureNet that distinguish Dash IV from the EDA 1200 which was based on Dash III.

INTRODUCTION

Telesis in the past has released two IBM-based schematic entry products that were based on FutureNet products: the EDA 1000 and EDA 1200. We based the EDA 1000 on the Dash II/Strides product and the EDA 1200 on the Dash III/Strides product. Dash III added color using the IBM Enhanced Graphic Display (EGA) system at the expense of a private mouse.

Dash IV closely followed the introduction of Dash III and used the same hardware platform. However, FutureNet made major enhancements in Dash IV that users had been requesting for some time. These enhancements may be categorized into four groups: visual enhancements and operational assists, new operations, drawing conventions, and impacts. The next three sections describe these enhancements in more detail.

VISUAL ENHANCEMENTS AND OPERATIONAL ASSISTS

This section describes those features that make Dash IV users more productive. This does not include new features that increase the total functionality of the system which I discuss in the next section.

1. "Help Menus"

   Extensive help files exist now that offer help on every command. If the user types ? or .HELP without an operand, the system displays the Main Help Menu that can be used to route the user to more specific Help Display Screens. The user may additionally supply the command and be taken directly to a Help Display Screen.

2. "User Profile"

   The user may create a profile file using a DOS text editor that supplies to Dash IV default information on operating states, display states, function keys, drawings, libraries, print options and color assignment. At any point in time while editing, the user may enter the PROFILE command and see a full screen display of the various options.

3. "Libraries"

   Dash IV contains the notion of "reference" and "update" libraries. When a user adds symbols to his/her schematic, Dash IV searches a list of up to ten reference libraries. Symbol editing takes place using the update library. Use the LIB command to specify reference libraries; .LIB must be used for the update library. Please note that the reference libraries must be specified before the update library.

   Dash IV also allows the user to print library directories using the .DIRPR command.

   The new release contains updated symbol libraries as well as one that "meets IEC/ANSII standards". Dash IV supplies an special command to create hat symbols.

4. "Highlighting"

   Look for all the new highlighting that takes place after the cursor stops moving. If it is over text, text will be highlighted on bright background. Its "point of effect" (to be discussed later) is also highlighted.

5. "Tricks"

   I now know some neat editing tricks that may have been there all the time but are now documented in their manual. The first trick involves running a connect line right down the side of the symbol which connects all the pins together. When you move the symbol away from the line, all the pins now have connect lines stretching out to the original line. This is nice for connecting a bunch of pins to the same net.

The second trick is a subtle variation of the first. Butt two symbols together at their connect points. Moving one symbol results in rubberbanding connect lines that join the pin pairs.

As a motivation for the third trick ... how many times have you joined two symbols together with a bunch of parallel connect lines, realized you have to move one, and the connect lines rubberband into beautiful L's, but not where you want them? Supply solder dots at the.speak points and move the symbol. The lines.speak where you want, they L and then the solder dots are removed.

NEW OPERATIONS

1. "Point of Effect"

   Point of effect is one of the most important enhancements that Dash IV has accomplished. It has removed all the old positioning rules associated with alphanumeric strings. The user may now specifically tag graphic elements and then place strings anywhere. Prior to this, users had to be very careful how they positioned signal names on the schematic. Previous versions used internal rules to determine to what connect line a signal name was attached.

2. "New Commands!"

   It is now possible to reflect and rotate symbols and areas (.RE, .R, [RE and [R). Unfortunately, neither maintain connectivity. You can move a vertex (/V), erase entire nets (/EN) and over and underscore strings ('OVER and 'UNDER).

3. "Windowing"

   This has been around, but I don't know how many users are familiar with the operation. A user may declare four "windows" in his/her design. The user supplies a window number and the most magnified view on the screen is turned into a window using the WINDOW command. When the user PANs, he/she selects a window number and the screen is immediately changed to that old window.

4. "Colors"

   The user may select eight colors from a pallette of sixty-four. Colors are associated with "groups" of graphic elements. For instance symbol graphics and connect lines might be in one group; pin numbers and reference designators may be in another. The COLOR command maps colors to groups. The DISPGRP commands map graphic elements to groups.

5. "Field Format Characteristics"

   Users may now right, left and center justify text; set a horizontal or vertical orientation for text; and change the printability of text (it's no longer an attribute).

   There are now seven text fonts.

   Attributes may be changed using the 'CH commands rather than deleting and re-adding.

6. "Bussing"

   Dash IV adds the concept of junctions. In the past, users "T'd" into busses with connect lines to establish connectivity. Now the user may add diagonal or arc junctions into busses to make more pleasing schematics. The /J command performs these operations.

7. "Line Fonts"

   There are now ten line fonts. One and two maintain their old meaning of connect line and bus. Fonts three through ten serve annotation purposes.

8. "Direct Connections"

   In the past, a symbol boundary formed an inviolate border through which nothing could pass. This has now been relaxed. The user may enter a special mode and make a "direct connection" between two pins on a symbol. The connection appears in the netlist. I'm sure this has some use. The /C command enables this mode while the .DCON command toggles the display of direct connections.

9. "Printing"

   There are lots of new print options. The 'PRINT, .PRINT, and PRINTOPT commands have been added.

10. "Layered Alphanumeric Fields"

   Major new functionality will be appearing in the future that will take advantage of this partially implemented feature. Layer fields will replace attributes. They allow the user to add additional information to text that will provide information about electrical properties of components. New post processors will take advantage of layer fields.

   However, be very careful when you use them unless you are writing your own post processors. In fact, FutureNet recommends using the current method of attaching attributes until further notice.

DRAWING CONVENTIONS

There is not much to say here. Some new attributes have been added without a tremendous amount of explanation as to what they are. All attributes above 99 used to be non-printable. This is now accomplished using the 'CH P command mentioned earlier.

IMPACT

As the saying goes, "you don't get nothin' for free". Schematics and symbols are upwards compatable (but not downward) and FutureNet recommends 640 kb instead of the old 256/512.

CONCLUSION

The new release of the Dash Schematic Editor contains many important new features that FutureNet users have been requesting. I believe that the "point of effect" rule, field format characteristics (justification and orientation), and the erase net feature are the most important new enhancements that will increase productivity. In addition, FutureNet plans to move all of their post processors to use "Layered Alphanumeric Fields". This should result in more powerful and flexible post processors.

BASIC STEPS FOR PC DESIGN PROCESS

Debra Carrier
Application Instructor
Telesis Systems Corp
Two Omni Way
Chelmsford, MA 01824

BASIC STEPS FOR PC DESIGN PROCESS

The following is a walkthrough of the basic steps to take to design a printed circuit board using the Telesis system. Following each step is a reference to the Telesis User Manual Volume 1 where further information may be found on each subject.

1. Decide what types of dips, discretes, connectors, and tooling holes you will need for your board. See if any of the symbols already exist in the SYSTEM-LIBRARY.

2. Build schematic symbols for gates, discretes, and connectors. Pick CREATE SYMBOL for each symbol created (do one drawing at a time). Print the -LOG file for each symbol to verify that it was built correctly.

   LIBRARY 2 - 19A.

3. Build a DEVICE file for each type of component used on the board, (7400, 74L508, RC07, etc.). You can get this information from the TTL data book. Use TEXT -> TEXT LEAD THROUGH to create DEVICE files for each component on the board.

   LIBRARY 24 - 27.

4. Open a schematic drawing (ex: NEW DRAWING - SCH-1)

   BASIC 24 - 27.

5. Add symbols to the drawing using:

   ```
        ADD SYMBOL [MENU]
        ADD SYMBOL [NAME]
   or   ADD SYMBOL [USER]
   ```

   BASIC 42 - 44.

6. Update the DEVICE TYPE field (DEV) using TEXT -> UPDATE TEXT.

   BASIC 59 - 61, NETDB - 23.

7. Update the REF DES field and/or the pin numbers (U*, 00 00 00) using the TEXT -> UPDATE TEXT commands. This is an optional step. If you do not update the REF DES and/or pin numbers, the system will automatically annotate those gates and/or components when you pick EXTRACT NETLIST.

   BASIC 59 - 61, NETDB 17 - 22.

8. Now, connect your symbols using ADD CONNECTION. (Set your active layer = to 1.) Make sure that all connect lines added hook up to connect points. Otherwise you will get a "floating end" message.

   BASIC 54 - 56, BASIC 43 - 44.

9. Once you have the drawing completed, pick SAVE DRAWING SAME REV AND CONTINUE. Then pick EXTRACT NETLIST. This will create your "NET-DATA-BASE". When this is done, it will generate an "EXTRACTION-LOG". This report shows any errors. PRINT the EXTRACTION-LOG.
   NETDB 2 - 14.

10. Run CREATE COMPONENT REPORT. This report shows if there are are any unused pins and if there are any errors. PRINT this report and verify all unused pins.

    NETDB 86.

11. If there are any errors, clean them up and re-run EXTRACT NETLIST.

12. If you had let the system annotate your reference designators and pin numbers (see Step 7), then you will have to pick BACK ANNOT CUR. This will update your schematic drawing to match your "NET-DATA-BASE".

    NETDB 80 - 82.

13. When finished, you will save your drawing using SAVE DWG SAME RV or SAVE DWG NEW RV.

    BASIC 68 - 70.

14. If you have additional schematic pages, open a new drawing for each page, i.e., page two would be a new drawing called SCH-2. You would then follow steps 5-13 to complete that page. (You run EXTRACT NETLIST for each schematic page.) For page three of your schematics, you would open a new drawing called SCH-3 and do steps 5-13 again, and so on. Make sure you have a clean extraction-log before proceeding.

15. An alternate method of creating the NET-DATA-BASE is to create a text file that contains all the electrical information about your board. Remember, this is an alternate method. You would not use both schematics and a text netlist to create your NET-DATA-BASE. While the benefit of using a text netlist is that it is much faster than generating schematics, the drawback is that you don´t have formal documentation. To use the text netlist method, the following steps would be taken:

    a. Build any DEVICE files necessary.
    b. Create a text file using this format example:

```
$PACKAGES
(Symbol Name! Device Type; Ref Des  Ref Des)
DIP14! 7410; U1  U[3-5]
RES400! RC07; R1  R4
$FUNCTIONS
(Device Type; Function Designator [# of functions])
74LS00; NAND [1-5]
74L74; FLIP [1-2]
$NETS
(Netname (opt); Ref Des Pin #  Ref Des.Pin #)
+5V; U[3-5].14  U1.[1-3]  R1.1
   ; U1.10  R1.2  NAND2.y
$END
```

NOTES: 1. The lines in the text file contained in parentheses () are comments and are not recognized by the system.

2. The brackets [] designate a range of items. Thus U[3-5] means that there will be a U3 U4 and U5 specified in the $PACKAGES section.

NETDB 36-54

16. Once the netlist is complete, use the command LOAD TEXT NETLIST to create the NET-DATA-BASE. When done, it will generate a NET-LOAD-LOG. This report shows any errors. PRINT the NET-LOAD-LOG.

NETDB 55-61

17. Run CREATE COMPONENT REPORT. This report shows if there are any unused pins and if there are any errors. PRINT this report and verify all unused pins.

18. If there are any errors, fix them and re-run LOAD TEXT NETLIST.

19. Now you will need to build the symbols that are necessary for each package type to be used on your board. (DIP14, RES400, DIP20, CON24 etc.) Pick CREATE SYMBOL for each symbol created (do one drawing at a time). The symbol name must match the package type in the device file. Again, PRINT the log file for each symbol to verify that it was built correctly.

BASIC 42, LIBRARY 9-19A.

20. Build a PIN FILE for each board symbol created (even if it has no pins - the system needs a pin file for each board symbol).

LIBRARY 20 - 37.

21. Open a Board Drawing file (NEW DRAWING - PCB).

BASIC 24 - 27.

22. Place symbols on your drawing using,

manual (ADD SYMBOL -> ASSIGN REF DES), PLACE 52 - 62
interactive placement, PLACE 18 - 25
or automatic placement. PLACE 26 - 50

BASIC 42 - 44.

23. If using manual placement techniques:

A. Pick ADD SYMBOL [MENU] - pick a menu box
ADD SYMBOL [NAME] - type in the name
or ADD SYMBOL [USER] - pick a user menu box

B. ASSIGN REF DES to the reference designators of each symbol (makes the link to the Net-Data-Base - turns U* into U1, U2, etc.).

PLACE 52 - 62.

24. If using interactive placement techniques:

A. Pick PLACE BY REF DES, PLACE BY DEVICE TYPE, PLACE IC´S, PLACE DISCRETES, PLACE IO´S, or PLACE ALL. This will place each component where you specify and "assign" it.

PLACE 18 - 25.

25. If you use the automatic placement commands, there are some prerequisites. These are:

    o All IO´s must be preplaced using manual or interactive techniques.
    o If your board has no IO´s, then at least one component must be preplaced. (It doesn´t matter which one.)
    o A NET-DATA-BASE built with device files.
    o A KEEPIN rectangle placed on Layer 147.
    o KEEPOUT rectangles placed on Layer 40.
    o All component symbols must be in the current project or SYSTEM-LIBRARY.

    PLACE 26 - 50, PLACE - 32, PLACE 3 - 14.

26. You would set your placement parameters, i.e. grid, rotation, etc.

    PLACE 34 - 49.

27. Pick AUTOPLACE BOARD. The system will automatically place your components on the board, and ratsnest to show your connections.

    PLACE 50.

28. Clean up any placements you wish. You can also run interactive or automatic IC, function and/or pin swap at this time to get the best placement you can.

    PLACE 47 - 49.

29. After your board is placed, you can route it. You can either manually route the board using ADD CONNECTION with NETLOCK ON or you can use the automatic router.

    INTERCON 2 - 46.

30. The prerequisites you need for AUTO ROUTE are:

    o Pin Files for every symbol used on the board.
    o A Layerstd file (defines your layers).
    o A PDRC-CON file (Pick INPUT DRC).
    o A NET-DATA-BASE built with device files.
    o A Router Keep-in Rectangle on Layer 146.
    o Router Keep-out Rectangles on Layers 70, 71-84.
    o Via Inhibit Barriers (keepout rectangle) on Layer 44.

    INTERCON 9 - 12.

31. Run the router as often as you like with different parameters to achieve the best completion rate.

    INTERCON 13 - 46.

32. If you want to delete a route - you can pick the command DELETE LAYER. This will prompt you with "which layer". Enter the layer number and the system will delete all the etch on that layer.

33. After you have routed and achieved the best results you can, complete any missing connections by using EDIT CONNECTION within the router pages.

    INTERCON 40 - 42.

34. When done, or at any time during the route clean-up cycle, you can pick VIA ELIMINATION. This will remove any unnecessary vias (feed thrus) from the board.

    INTERCON 39.

35. Run NETCOMPARE to verify that your design is electrically correct. This will check for design drafting errors, extra connections, missing connections, and any unassigned components.

    POST 2 - 7

36. If there are any ECO´s on the board, you would:

    a. "DEASSIGN" affected components on the board.
    b. SAVE DRW SAME REV.
    c. Call up schematic, make necessary changes.
    d. EXTRACT NETLIST.
    e. If EXTRACTION LOG is clean, SAVE DWG SAME REV.
    f. Call up the board dwg, "ASSIGN" affected components, add any new changes.
    g. Change any etches, delete misconnected etch and add connection for new logic.
    h. Run NETCOMPARE.

37. Run DESIGN RULE CHECK (Physical Design Rule Check) to check for any spacing violations.

    POST 8 - 13.

38. When the board is finished and free from errors, you can run the post processes. First, run NC Drill.

    POST 59 - 78.

39. The files needed for NC Drill are:

    o Pin Files for every symbol on the board.
    o NCDRILL-PAR (tells the NC DRILL machines parameters).
    o NCDRILL-FIG (tells which figures you want to represent each drill size).
    o You have to add the drill header section to the board drawing (on Layer 98).

    POST 62 - 69.

40. When all prerequisites are met, pick CREATE NC DRILL. It will create a text file called NCDRILL-TAPE that can be used to punch a paper tape, and will also add drill figures to your drawing which may be pen plotted.

    POST 70 - 71.

41. Next - to create artwork, decide what you want to appear on each sheet of film. For example, on Sheet 1, you might want all components side etch, the board outline, and the drawing format (title block). Create a text file called PHOTOPLOT-CON that specifies which layers appear on each film sheet.

    POST 15 - 58.

42. Prerequisites for artwork generation:

- o Pin Files for every symbol on the board.
- o APERTURE-TAB (tells what size apertures used on Gerber).
- o PHOTOPLOT-CON (tells which DB layers appear on each film sheet).
- o PHOTOPLOT-PAR (tells parameters of Gerber machine).
- o LAYERSTD file.

POST 15 - 43

43. When all complete, pick CREATE PHOTOPLOT. This will create text files for each layer of artwork. The system takes whatever you named each film sheet in the PHOTOPLOT-CON file, and attaches a suffix of -ART for each file´s name. You can copy these files to a mag tape and send this tape to your photoplotting vendor.

POST 44.

44. When the entire board design is done, pick COMPRESS DWG and specify your final drawing. Then copy this drawing file and any other files you want to save (such as device files, pin files, symbols, etc) to mag tape or floppy disk to archive it. You may then delete that project from the system to free up disk space for your next design.

FILE ARCHIVE 25 - 46.

PRINTED CIRCUIT DESIGN

GLOSSARY

Charles F. Powderly, Jr.
Manager, Customer Training
Telesis Systems Corporation

TELESIS

PRINTED CIRCUIT BOARD DESIGN

GLOSSARY

ABSOLUTE COORDINATES - values of X,Y or Z coordinates with respect to the origin of the coordinate system. Contrast with incremental coordinates.

ALGORITHM - a step-by-step procedure for solving a problem.

ALPHANUMERIC - referring to the subset of ASCII characters that includes the 26 alphabetic characters and the 10 numeric characters.

ALPHANUMERIC DISPLAY DEVICE - device capable of displaying a fixed set of letters, digits, and special characters, such ASCII characters in one or more fonts. Allows user to enter commands and receive messages from the system. Device consists of a typewriter style keyboard and a display (CRT) on which the text is viewed.

ANNOTATION - the process of inserting text on a schematic diagram or printed circuit drawing.

APERTURE TABLE - a text file which provides interface information to be used by the photoplotter. The main body of the table is made up of lines that describe the apertures on the wheel.

ARTWORK - silkscreens, solder masks and other films generated on the system and used in the manufacture of printed circuit boards.

ASCII (prononouned "askey") - American Standard Code for Information Interchange, a standard code for expressing the numbers 0-9, letters A-Z (in both upper and lower case), and a variety of common typewriter symbols, such as a period, question mark, and carriage return, in a seven bit format. For example, in ASCII letter a = 110 001, b = 110 0010, A = 100 0001, numeral 5 = 011 1010, $ = 010 0100 and a carriage return = 000 1101. ASCII is used by most computer manufacturers for storing and transmitting alphanumeric data.

AUTOMATIC PLACEMENT - advanced state-of-the-art software to optimally place components on your PC board.

AUTOMATIC ROUTING - another state-of-the-art software program that automatically routes the board per your schematic or netlist.

BACK ANNOTATION - a software program designed to update schematics and supporting documentation on edits or modifications made to a PC board, e.g., pin swaps, reference designator changes, etc.

BAUD RATE - the rate, in bits per second, that data can be transmitted over a serial transmission line (such as a telephone line).

BENCHMARK - when used in computer jargon, it refers to a set of standard designed to verify performance specifications or to compare hardware or software produced by different manufacturers.

BINARY CODE - any system of representing data with zeros and ones only. All data can be expressed in binary code. All computers work only in binary code, although the user never actually sees the code written.

BIT - a single digit, 0 or 1, of a binary code.

BOM (Bill of Material) - this is a parts list for a printed circuit board or equipment drawing from which items such as part type and quantity may be ordered for the manufacturing process.

BUFFER - A storage device used to compensate for a difference in rate of data flow or event timing when transmitting data from one device to another.

BULLET - tie point. A symbol which indicates electrical and logical connection of nets. A bullet does not indicate a physical connection at that point.

BUS - One or more conductors used for transmitting signals, data or power. Often a bus acts as a common connection between several locations.

BYTE - A storage of eight bits, usually treated as a unit. One byte can store one unit of information. Memory capacity of a computer is measured in bytes. Also called a character.

CAD - computer aided design - interactive computer graphic system used in a wide range of design applications.

CAM - computer aided manufacturing - interactive computer graphic system used in conjunction with CAD to produce aids to manufacturing (i.e., NC Drill Tape, automatic component insertions, etc.)

CASSETTE - a standard tape cassette, an inexpensive way of storing programs and data.

CHARACTER - graphic symbol, alphabetic, numeric or special.

CHIP - a single device containing many transistors and other components formed on the surface of a tiny sliver of silicon. Often used synonymously with integrated circuit.

COMMAND DEVICE - provides communication between operator and system. See Alphanumeric Display, LIS.

COMPONENT - a physical part of a circuit, such as a resistor, a capacitor or an IC. The symbol used to represent the physical component is also referred to as the component.

CONNECT POINT - (C-Point) - a location shown graphically as a diamond ( ) used as an attachment point for connect lines. Connect points may have text points associated with them. Connect points are often associated with a symbol.

CPU - central processing unit - the heart of the computer where information is controlled and processed. The portion of the computer which directs the sequence of operations and initiates the proper commands to the computer for execution.

CRT - cathode ray tube - any device which displays data by means of projecting a beam on the inside of an evacuated glass tube.

CURSOR - a graphical indicator displayed on the graphics screen to mark the location of the light pen as determined by the Function Screen.

DATABASE - a collection of information stored and updated in a computer system.

DEFAULT - The system's pre-set values for design parameters (i.e., grid size, active layer, color).

DELETE ELEMENT - to remove from existence.

DIGITIZE - convert from graphical or analog form to representation by discrete values. A drawing must be digitized before a computer can perform operations on it.

DISK/DISC - a flat, circular, magnetic recording device capable of storing large amounts of data. Information can be retrieved from any point on the disk.

DISPLAY SCREEN - a television-like tube used to display text and graphics. While most CAD/CAM systems now use a raster scan, monitors are often referred to as CRT's.

DRILL - a command to tell the system to place a via or feed thru on a board.

DRILL FIGURE _ a figure that indicates a particular size drill to be used for a drill template.

DRILL TAPE - a magnetic or paper tape which contains information to be fed into an N/C drill machine.

EDIT - alter, adapt or refine to bring about conformity to a standard, or to suit a particular purpose.

ELEMENT - the lines, circles, arcs, etc. that make up the schematic and board drawings.

EMULATE - To imitate a computer system by a combination of hardware and software that allows programs written for one computer to run on another.

ENTER - A command used to tell the system to execute a user instruction.

ENTITY - construction curve; primitive part of a drawing. A display element.

ES - electrical schematic

EXTENTS - limits to which somethings extends. The amount of space or surface that something occupies or the distance over which it extends.

FILE - a logical collection of data treated as a unit, which occupies space on a mass-storage device such as a disc or mag tape and has an associated file name (and file type).

FIRMWARE - sets of computer instructions (programs) cast into "Read Only Memory." To change the firmware programming of a computer, the user must replace hardware components.

FONTS (line fonts) - repetitive patterns used to give meaning to a line, such as dashed lines, dotted lines, etc.

FUNCTION KEY - button or switch which enters commands or other keyboard input as opposed to a data item.

FUNCTION SCREEN - a TELESIS exclusive human-engineered interface consistng of a high resolution CRT which automatically displays menus of commands as they are needed.

GRAPHIC TABLET - a surface through which coordinate points or lines can be transmitted to a computer in the manner of writing on paper. A "light pen" or "stylus" is used to "write" on the graphic tablet. The image generally appears on a CRT display.

GRID - network of uniformly spaced points which may appear on an output device; used for locating a position.

HARDCOPY - any information printed on paper, as opposed to apperating on a computer generated display such as CRT.

HARDWARE - all of the various mechanical and electronic components of a computer system, e.g., the electronic chip, printer, monitor, etc.

HIGHLIGHTING - A system feature that "accents" selected graphics to indicate placement or action.

HORIZONTAL LAYER - the layer of the PC board drawing which contains the connections that are predominantly horizontal.

IC - integrated circuit - a tiny complex of electronic components and their connections that is produced on a small slice of material (silicon).

INCREMENTAL COORDINATES - a new set of coordinates measured from a previously derived set of coordinates rather than from the origin (0,0).

INPUT - Used as a verb, a grammatical impossibility accepted placidly by computer people, this is the act of entering information into the computer. As a noun, input refers to data entered into the computer.

INTERCONNECTIONS - anything that connects one item to another. On a PC board, interconnections consist of copper runs connecting pads. In schematic drawings, interconnections consist of lines connecting elements. In electronic drawings, all interconnections are made with connect lines.

INTERFACE - The boundary between two parts of a computer system, often consisting of a piece of electronic circuitry that allows other devices to communicate with each other. Used as a verb, interface means to make one part of a computer system run smoothly with another.

JOYSTICK - A device or lever connected to the computer that moves objects around on a screen. Used with video games. Also referred to as a paddle.

JUSTIFICATION - the exact placement of an item. Text may be left, center or right justified depending upon whether digitized point is to appear on the left, in the center or to the right of the text.

KEEP-OUT - a keep-out defines the boundaries of a PC board and areas inside of the board that are not to be used for routing and placing. Keep-outs are rectangles and are used by Autoplace and Autoroute.

KEYBOARD - A menu made up of alphanumeric characters similar to the keys of a typewriter that is displayed on the Function Screen.

KEYPAD - a menu of numbers used to give explicit coordinates to the system for placement of graphics that is displayed on the Function Screen.

KILO (K) - a prefix meaning thousand. Used before the word byte to denote memory capacity.

LABELS - used to associate various characteristics and values with a symbol or part.

LAYER - logical concept used to discriminate (separate) group(s) of data within a given drawing. Layering enables the operator to specify derived display elements to be visible. May be thought of a series of transparencies arranged in any order yet having no depth. Layering allows a more efficient use of the database.

LIBRARY - a collection of PARTS or other files.

LIGHT PEN - see graphic tablet.

LINE - graphic representation connecting two or more points.

LINE FONT - a special pattern used to represent a line.

LIS - (Large Interactive Surface) - an automated drafting table used to plot and/or digitize drawings.

MAIN FRAME - an overused, ill-defined term which refers to the central processing unit(s) of a large data processing system.

MEMORY - a device or series of devices capable of storing information into the computer temporarily or permanently in the form of patterns of the binary numbers "0" and "1." In many personal computers, memory can be expanded by adding hardware.

MENU - a selection of command choices displayed on the Function Screen and chosen with the light pen.

MICROPROCESSOR - a central processing unit (CPU) containing a single chip.

MINI-COMPUTER - like main frame, this is another bad term. A mini is smaller than a main frame, and generally employs a central processor unit capable of handling 16 bit (2 byte) words rather than the 32-bit work typical of main frame computers.

MODEM - derived from the words modulate-demodulate. A device attached to a computer to convert the computer's digital signals into signals for transmission to other computers over telephone lines.

MONITOR - a television receiver or cathode ray tube (CRT) used to display computer output.

NC DRILL - numerically controlled drilling of all component and feed through holes on a printed circuit board.

NC INSERTION - numerically controlled insertion - automatic control of machines which place physical components on a PC board or which direct the placement.

NECK - a command that reduces the width of an interconnect to fit between components.

NET - all the electrical connections of a particular signal.

NET COMPARE - a feature of the TELESIS system designed to check that the logical schematic and board connections match.

NET-DATA-BASE - a collection of logical connection, component device descriptions, and other "intelligent" informatin that describes your circuit.

NET LIST - a complete listing of all electrical connections.

NETLOAD - an alternative method of generating your netlist information. Allows your to forego drawing schematics on the system.

NET LOCK - a feature of the TELESIS system that ensures that a connection cannot be made to the wrong net.

OFF-LINE - equipment or devices in a data processing system which are not under the direct and immediate control of the central processing unit.

OFFSET ORIGIN - the distance from the origin of a part to the point referenced by the NC Drill and the photoplotter.

ON-LINE - equipment or devices in a data processing system which are under the direct and immediate control of the central processing unit.

OOPS - a command that tells the system to ignore the previous input.

OPERATING SYSTEM - the primary control program of a data processing system. Written in assembler language, the operating system is unique to each central processor unit design. The program controls the execution of programs within the CPU and controls the flow of data to and from the memory, and to and from all peripheral devices in the data processing system.

ORIGIN - the XO, YO point from which all coordinates are measured.

OUTPUT - information or data transferred from the internal memory to the computer to some external device such as the screen or printer.

PAD - an area of plated copper on a PC board to which leads of components are soldered.

PC - (printed circuit) - a circuit for electronic apparatus containing conductive material forming paths from terminal to terminal on an insulating surface.

PERIPHERALS - the various pieces of a computer system that can be hooked up in different ways to the central processing unit and memory which form the system's input and output devices, such as printers, disk drives, etc.

PERSONAL COMPUTER - a small computer based on a microprocessor. Not all microcomputers can be dedicated to single tasks as diverse as controlling a machine tool or a video game.

PHOTOPLOT - a photograph of printed circuit board graphics used for a particular process. For example, separate photoplots are used for etching each side of a circuit board.

PHOTOPLOT WHEEL - a wheel in a photoplotter containing apertures of various sizes and shapes through which light must pass to reach the film.

PHYSICAL DESIGN RULE CHECK - A program developed to check for design violations on PC boards (i.e., line-to-line spacing, line width and line-to-pad spacing.)

PICK - the combined use of the light pen and Function Screen to place graphics or select a menu box.

PIN - the lead or connection point for electronic components.

PIN FILE - a text file that specifies shape, size, layer, and drill information or component pads.

PIXEL - derived from picture to element. The smallest point on a CRT device that can be controlled independently by the computer. The more pixels per square inch, the higher the resolution or clarity of the CRT screen image.

PLOTTER - any device that produces hardcopy or graphical data. There are two main types, vector plotters, which draw with ordinary ballpoint pens and ink and electrostatic plotters, which product output that resembles Xerox copies.

POINT - a narrowly localized place having a precisely indicated position.

PRE-PLACED LINES - runs or lines that are placed on the PC board before using the Autorouter. Lines may be inserted using the ADD CONNECTION command.

PRINTER - a device for producing paper ("hard") copies of data output by a computer.

PROGRAM - a series of instructions carried out by the computer in sequence. The program must be written in a language the computer speaks.

PROM (Programmable Read Only Memory) - Non-volatile memory chip that allows a program to reside permanently in a piece of hardware.

PROMPT - any message or symbol from the computer system informing the user of possible actions or options.

PROTOCOL - a set of rules governing the exchange of data between devices in a data processing system.

QUEUE - a waiting line or area.

RAM - (Random Access Memory) - an array of semiconductor devices for temporary storage of data during the computation process. Individual bytes (or words) of data may be stored in or retrieved from any point in RAM.

RASTER - a CRT display that resembles a television set image. The image is defined by a beam that sweeps the CRT at a constant rate along horizontal lines. Raster displays may be changed and updated very rapidly.

RATSNEST - a program on the TELESIS system that displays the shortest connect paths.

REFERENCE DESIGNATOR - text used to uniquely identify a component and to associate the component with a particular schematic element.

REPAINT - redraw a display image on the output device.

ROM - (Read Only Memory) - similar to RAM, except that the computer user may retrieve data only. The information is inserted in ROM by the manufacturer in accordance with his own or the purchaser's specifications.

ROUTING - to place interconnections on a printed circuit board.

SCALE - to change the magnitude of a variable(s) in a uniform way; the ratio of a real thing's magnitude to the magnitude of the model; analog of the real thing.

SEGMENT - a portion of a line or element.

SIGNAL HIGHLIGHTING - an operation that visually identifies the connection points of a net in a PC board. The command required is HIGHLIGHT NET.

SIGNAL NET - see Net List.

SILKSCREEN DIAGRAM - as used here, the silk screen diagram shows the outline of the PC component. It may be used to obtain a pattern on the PC board by using the silk screen process.

SIP (Single-In-Line Package) - an assembly consisting of two or more SMC's mounted on a common substrate. The SMC's are interconnected to each other and to pins at the edge of the substrate for through-hole mounting on a PWB.

SMC (Surface-Mounted Component) - a component designed to be mounted and soldered to pads on the surface of a PWB rather than inserted into through-holes in a PWB.

SNAP OFF DISTANCE - the distance between the surface the PWB and the screen when they are mounted in a screen printer. After the squeegee deflects the screen to the PWB and passes over, the screen must "snap-off" to the original position.

SOFTWARE - programs, procedures, and associated documentation for data processing systems. In general, the software associated with a computer system costs more to develop than the hardware. Of course, reproducing software is very inexpensive.

SOIC (Small Outline Integrated Circuit) - a package in which an integrated circuit chip can be mounted to form a surface-mounted component. It is made of a plastic material which can withstand high temperatures and has leads formed in a gull-wing shape along its two longer sides for connection to a PWB footprint.

SOLDER BALLS - Very small balls of solder that separate from the main body of solder which forms the joint. Primarily caused by excessive oxides in the past which inhibit solder coalescence during reflow.

SOLDERABILITY - a qualitative measure of the ability of PWB pads of or of a component's leads or pads to be completely wetted by molten solder.

SOLDERING, REFLOW - a process in which solder paste is deposited upon pre-tinned pads on a PWB and the component's pre-tinned leads are placed upon the paste. When the assembly is heated to the proper temperature, the solder paste melts and the solder on the leads and pads reflows to form a solder fillet.

SOLDERING, VAPOR PHASE REFLOW - a type of reflow soldering in which the PWB assembly is passed through a vaporized inert fluorcarbon. The latent heat given up when the fluorocarbon condenses causes the solder to reflow.

SOLDERING, WAVE - a method of soldering complete assemblies where the PWB with components mounted is passed through one or more waves of molten solder which

is continuously moving to maintain fresh solder in contact with the PWB.

SQUEEGEE - A rubber blade used in screen printing to wipe across the screen to force the solder paste through the screen mesh onto the footprint.

STYLUS - see graphics tablet.

SUBSTRATE - The base material which forms the support structure of an IC or PWB.

SURFACE MOUNTING - See SMC.

SYMBOLS - a representation of something by reason of relationship, association or convention. An arbitrary or conventional sign used in writing or printing relating to a particular field to represent operations, elements, relations or quantities.

TCE (Temperature Coefficient of Expansion) - the rate of expansion of a material, measured in ppm/°C, when the material's temperature is increased. The TCE of a substrate must closely match the TCE of a solidly mounted SMC to prevent mechanical stress on the solder joints.

TERMINAL - a device for providing input to and output from a computer. Usually consisting of a keyboard and screen in the same box.

TEXT FILE - a file containing text. A text file may be created by the user

with the CREATE/EDIT TEXT command or it may be created automatically by a system command such as EXTRACT NETLIST.

TEXT POINT - (T-Point) - a location shown graphically as a triangle ( ) which contains default parameters for subsequent placement of text. A text point may be associated with a connect point, a nodal subfigure or may stand alone.

THERMAL MANAGEMENT - consideration and use of the variousmethods for dissipating head produced by operating electronic equipment so that the operating temperature specification of the components is not exceeded.

THERMAL RESISTANCE - Indicates the rate of change of junction temperature with power dissipation. Measured in °C/W.

THICKNESS - as applied to text, thickness is the line width and is controlled by the aperture on the photoplotter wheel, or the pen on the penplotter.

THROUGH-HOLE BOARD - A PWB which has holes through it so that component leads can be inserted through the board and soldered on the other side.

TRAP - an automatic transfer of control to a known location. To catch or take in as if in a trap. An imaginary square about a digitized location.

TRAP SIZE - one half the size of the imaginary square forming the trap.

USER-FRIENDLY - What all computer "illiterates" hopes for: a computer system that is easy and non-threatening to use and understand.

VERTICAL LAYER - the layer of the PC board drawing which contains the connections that are predominantly vertical.

VIA - a feedthrough or plated through hole on a printed circuit board.

VIEWS - provide a selected view of a part.

VOIDS - Cavities inside the solder joint formed by gases which are released during reflow or by flux residues which fail to escape from the solder before it hardens.

WIDTH - a measurement taken at right angles to the length and depth. For text, width is the size of the character measured horizontally.

ZOOM - to proportionately enlarge or decrease the size of the display entities by rescaling.

TELESIS AUTOPLACEMENT

Garrett T. P. Cavanaugh

Product Manager

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

The following document covers the theory behind Autoplacement including the processing program, pin-pair distance formula, and how the placement algorithm works. It also provides a description of parameters not generally found in the user documentation, and methods for increasing Autoplacement effectiveness.

INTRODUCTION

This paper is intended for use by customers who would like a more indepth understanding of how to utilize the Telesis Autoplacement package.

PROCESSING ORDER

The Telesis Autoplacement system operates in two stages; set-up and execution. During set-up, autoplacement performs a number of functions necessary for its execution. These functions build scratch files for the program to use. These run in two separate programs; CREGEO (CREate GEOmetry) and CREPIN (CREate PINlist).

The CREGEO program pays particular attention to the geometry of the board; the location and identity of any pre-placed components; the size and number of any placement keep-out rectangles.

The CREPIN program creates a list of components connected to other components; the packages required to place the board and whether they exist in the project or system library; and the package sizes and shapes. This list can be extremely large and hence very time-consuming to construct. For example, an 18 component board would produce the following matrix:

	UØ1	UØ2	UØ3	UØ4	UØ5	UØ6	UØ7	UØ8	UØ9	U1Ø	U11	U12	U13	U14	U15	R1	R2	*P1
*P1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	2	4	Ø	Ø	Ø	Ø	Ø	6	Ø	Ø	N
R2	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	N	
R1	Ø	Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	N		
U15	Ø	Ø	1	1	1	1	1	1	1	1	1	1	Ø	Ø	N			
U14	1	5	1	1	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	N				
U13	Ø	Ø	4	Ø	Ø	Ø	4	Ø	Ø	3	Ø	Ø	N					
U12	Ø	Ø	Ø	5	5	Ø	Ø	Ø	Ø	Ø	Ø	N						
U11	Ø	Ø	Ø	Ø	3	Ø	3	Ø	Ø	Ø	N							
U1Ø	Ø	Ø	Ø	Ø	Ø	7	1	Ø	Ø	N								
UØ9	1Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	N									
UØ8	Ø	11	Ø	Ø	Ø	Ø	Ø	N										
UØ7	Ø	Ø	Ø	Ø	Ø	Ø	N											
UØ6	Ø	Ø	12	Ø	Ø	N												
UØ5	8	8	8	8	N													
UØ4	4	4	4	N														
UØ3	2	2	N															
UØ2	1	N																
UØ1	N																	

This matrix is used to determine the component that should be placed next. The numbers in the matrix reflect the number of interconnections existing between any two components. For example, there are 7 connections between U1Ø and UØ6. Connections to a class I/O component are multiplied by the I/O weight parameter.

The following items should be noted about the matrix:

- It is inconsequential whether UØ2 is connected to itself.

- The number of entries in the matrix is equal to: n/2*(n-1), where "n" is the number of components (placed and unplaced) on the board.

- All of the components -- IC's, Discretes, and I/O's -- are shown in the matrix.

- The number of connections between UØ6 and U1Ø is equal to the number of connections between U1Ø and UØ6, resulting in a triangular matrix.

- An '*' denotes a preplaced component.

PIN-PAIR LENGTH FORMULA

The formula used in the placement of any component on the system is referred to as the Pin-Pair Distance formula. It is calculated as follows:

```
 -----------------------------------P2  (x2,y2)
 |                                  |
 |                                  |
dY                                  |
 |                                  |
 |                                  |
 ------P1                           |
       |                            |
       |<----------dX-------------->|
```

Given two pins with coordinates (x1,y1) (x2,y2). There are two numbers:

dX = Ah*Rh*ABS(x1-x2)

dY = Av*Rv*ABS(y1-y2)

Where: Rh = horizontal position weight

Rv = vertical position weight

Ah = horizontal Aspect ratio

Av = vertical Aspect ratio

If delta-grid-x> delta-grid-y then:

Ah = 1.0

Av = delta-grid-x/delta-grid-y

ELSE

Ah = delta-grid-y/delta-grid-x

Av = 1.0

The aforementioned variables are used when calculating the pin-pair distance:

M = dX+dY-Ø.5*(ABS(dX-dY))

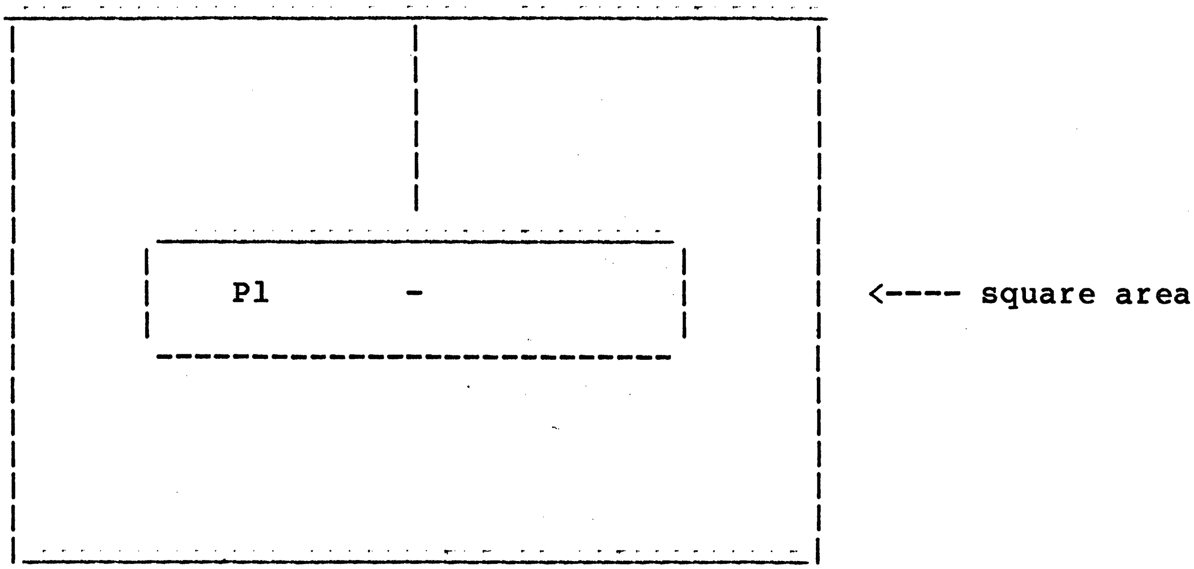
The formula calculation considers all components about to be placed. "M" is the distance between each pin on the component to be placed and the closest placed pin in the same net. During all calcualtions, an attempt is made to minimize "M" at every possible location for any given component.

PLACEMENT ALGORITHM

During Autoplacement, the component most heavily connected to an already placed component is selected as the next location for placement. This is determined from the connectivity matrix and the components already in place on the board. For example, in the matrix shown earlier, U15 is the component to be placed next, not UØ6 or UØ3.

The component is temporarily placed at the first available grid location and its pin-pair distances calculated. (Available is defined as a grid point existing in such a manner that when a component is placed on the grid point, its footprint will not be violated by other components).

The component is then moved to the next available location and its pin-pair distances recalculated. This process continues until all available grid locations have been tested. The location with the lowest calculated "M" is the one chosen for component placement. The most heavily connected component is U15, making it the first component to be placed.



The first iteration examines all the grid points in this area. If a fit is found, then the "best" fit is chosen.

If no fit is found, the area is increased by one grid in all directions and a suitable location is again looked for.

The next most heavily connected component is chosen and the process is repeated. For example, the next components to be placed would be UØ9 and then UØ1.

PARAMETERS

This section covers the requirements needed to run Autoplacement, the contents of the PLACE-CON, and a discussion of some of the parameters.

REQUIREMENTS

There are a number of questions that need to be answered prior to entering the Autoplacement menu pages. Specifically, these questions are:

1. Has the NET-DATA-BASE been created with device files that contain the PACKAGE line?

2. Are the board symbols in the SYSTEM-LIBRARY or PROJECT?

3. Is there a single COMPONENT KEEP-IN rectangle defined?

4. Are there optional footprints defined, and if so, what layer are they on?

5. Are there optional component keep-out rectangles defined, and if so, what layer are they on?

6. Have the I/O components been placed?

7. Is the component grid an even multiple of the router grid?

8. Has the trap size been set? (This is not required for placement).

9. Have the colors been assigned? (This is not required for placement).

PLACE-CON

The following is a list of the parameters that may be set during Autoplacement. Please refer to the Telesis User Manual for a description of what they are, and how they can be set.

```
RUN_IC_PLACEMENT                                NO
RUN_FUNCTION_IMPROVEMENT                       YES
RUN_IC_IMPROVEMENT                             YES
RUN_DEVICE_PLACEMENT                            NO
RUN_DEVICE_PLACEMENT                            NO
RUN_PIN_IMPROVEMENT                            YES
RUN_DISCRETE_PLACEMENT                          NO
IC_GRID_ORIGIN                          7700  6400
IC_GRID_SIZE                             600  1400
DEVICE_GRID_ORIGIN                      7700  6600
DEVICE_GRID SIZE                         600  1400
DISCRETE_GRID_ORIGIN                    7500  6400
DISCRETE_GRID_SIZE                       400   400
ALLOW_IC_ROTATION_0_DEG                        YES
ALLOW_IC_ROTATION_90_DEG                        NO
ALLOW_IC_ROTATION_180_DEG                       NO
ALLOW_IC_ROATATION_270_DEG                      NO
ALLOW_DEVICE_ROTATION_0_DEG                     NO
ALLOW_DEVICE_ROTATION_90_DEG                    NO
ALLOW_DEVICE_ROTATION_270_DEG                  YES
ALLOW_DISCRETE_ROTATION_0_DEG                  YES
ALLOW_DISCRETE_ROTATION_90_DEG                 YES
ALLOW_DISCRETE_ROTATION_180_DEG                YES
ALLOW_DISCRETE_ROTATION_270_DEG                YES
DEVICE_NAMES                                  C0C9
MAX_IC_IMPROVEMENT_TIME                         60
MAX_FUNCTION_IMPROVEMENT_TIME                   60
POSITION_WEIGHTS                        1.00  3.00
I/O_WEIGHT                                    3.00
KEEPOUT_LAYERS                                  40
```

DISCUSSION

There are five sets of parameters that deserve a closer analysis:

POSITION WEIGHTS 1.00 1.00 - As discussed earlier, these two parameters are used in the pin-pair distance formula to skew the placing of components in a specific direction. It is advantageous to set them equal to the aspect ratio of the PCB, unless there is just reason to do otherwise.

I/O_WEIGHT 3.00 - This parameter is used when the connectivity matrix is built. The specified number is a multiplier to the number of connections going to any class I/O component. For example, if there are 4 connections from edge connector J3 to IC U72, then an I/O weight of 3.ØØ would yield a value of 12 in the matrix. This would guarantee that components talking directly to the edge connector would be placed first.

IC_GRID_SIZE 6ØØ 14ØØ - These numbers are used to calculate the Aspect ratios in the pin-pair formula. They correct the skew to the placement formula by the body size of an IC.

IC_GRID_ORIGING/SIZE and DEVICE_GRID_ORIGIN/SIZE If there are decoupling capacitors used in the circuit that will be placed directly over the IC's, then:

1. DEVICE_GRID_SIZE = IC_GRID_SIZE and

2. IC_GRID_ORIGIN X Y = DEVICE_GRID_ORIGIN X (Y+offset)

This offset is usually 1ØØ mils.

ALLOW IC ROTATION - These parameters are not mutually exclusive and can be set correspondingly while still achieving the desired results.

METHODS OF USE

There are six recommended methods for using Autoplacement, all which have been tested and used effectively. (It is not recommended that you pick PLACE BOARD and expect the board to be placed correctly). The methods are:

A. **PREPLACE CRITICAL COMPONENTS** - As designers and engineers, you know where certain components should be placed. Component placement is critical to the success of Autoplacement and its ability to place the board acceptably. The recommended components for preplacement are I/O components, components attached to busses, memory arrays, and components that you know the placement of such as clock circuits, and voltage regulator circuits.

B. **THE LARGE WINDOW METHOD** - Another way to use Autoplacement is with an artifically large component keep-in rectangle. This allows the computer to place the components in a given location and also allows for board overlap. The components are placed so that the relationship between all components is easily visible. Minimal time is required to move the components back onto the board.

C. **THE SMALL GRID METHOD** - If there are edge connectors on both the top and bottom of the PCB, then an artificially small placement grid can be used. For example, an IC_GRID_SIZE of 1ØØ, 1ØØ could be used. This will increase the processing time slightly but will still produce the same results as the Large Window Method.

D. **LARGE WINDOW/SMALL GRID METHOD** - Combining the above methods will produce remarkable results. For example, Telesis had a potential customer who produced the netlist and the placement of a particular board (we weren't shown the board). Use of the Large Window/Small Grid method resulted in the Autoplacement version being almost identical to the manually placed board.

E. **ITERATE** - The Iterate method requires Autoplacement to run "N" number of times with interaction between each run. To use this method, first run Autoplacement on the entire board. Next, examine the Ratsnest lines. Delete any that do not "look right" using the DELETE COMPONENT command and then repeat the process. The results are generally acceptable once the process has been repeated three times.

F. **INCREMENTAL NET-DATA-BASE BUILD** - Another method of running Autoplacement involves using an incremental build of the NET-DATA-BASE with a corresponding placement of the board. Engineers usually design circuits in blocks, while Designers tend to place these blocks in logical groups. Or, the engineer designs a schematic sheet while the designer places it. To incrementally build and place the NET-DATA-BASE and BOARD, first LOAD TEXT NETLIST with $ADD on one schematic sheet. Then, Autoplace just this portion of the board, ensuring that you place the component keep-in accordingly. These steps should be repeated for the entire circuit.

This method decreases the size of the connectivity matrix and so reduces the set-up processing time.

GENERAL AUTOPLACEMENT HINTS

Following are some useful hints for increasing the effectiveness of the Autoplace program:

A. **PUT UP AN IC GRID** - This is an especially helpful guide to Interactive Placement and its relationship to Autoplacement. First ADD LINE, and then STEP & REPEAT in two directions showing where the IC grid should go. This can be done on DBLAYER 147 and will help determine whether there are adequate grid locations for all IC's on the board.

B. **DECREASE DENSITY IN MIDDLE** - Adding thin keep-out rectangles towards the middle of the board wil artificially create the space for routing channels.

C. **DISCRETE PLACEMENT WITH ONLY ONE ROTATION** - To gain time during Autoplacement, run discrete placement with one only one rotation allowed. 18Ø degree rotations can then be performed by running PIN SWAP.

D. **MULTIPLE AUTO-SWAP PASSES** - It is better to run a number of swapping executions than to run one long execution. For instance, if you have a night run devoted to running the auto-swap algorithms, divide the night into three equal lengths of time, and run Autoplacement three times using an EXECUTE MENU. Between each Autoplace run you should save a version of the drawing, rename PLACE-LOG to P-L, change the revision of P-L to agree with the revison of the board that has been saved, and change any necessary placement parameters.

   The next morning, call up each of the boards, a revision at a time. The TOTAL VIRTUAL WIRE LENGTH will decrease with each subsequent run of Autoplace. After three swapping executes, you will reach an acceptable improvement.

CONCLUSION

With a little thought and preplanning, you will find the Telesis Autoplacement package an effective tool.

TELESIS PLACEMENT IMPROVEMENT USING

FUNCTION SWAP/PIN SWAP

Debra Carrier
Application Instructor
Telesis Systems Corporation
2 Omni Way
Chelmsford, MA 01824

TELESIS PLACEMENT IMPROVEMENT USING FUNCTION SWAP/PIN SWAP

INTRODUCTION

The Telesis placement improvement capability, known as function swap/pin swap, allows users to selectively improve the placement of components, functions within components, and pins within functions (or gates). This may be done either automatically using commands within the automatic placement menu pages, or interactively using commands within the interactive placement pages.The placement improvement programs enable the user to have the most efficient placement of circuit logic prior to routing the board. This increases router efficiency and decreases post route clean up time by reducing the number of missing connections.

In order to utilize the improvement capabilities the system needs information provided by a NET-DATA-BASE that was created using DEVICE FILES that contain a PIN SWAP line. Once complete the NET-DATA-BASE must be in the current design project.

The NET-DATA-BASE contains all the information about nets, pins, reference designators and device types to be used in the board design. The DEVICE FILES provide information relative to the logical subdivision of a device (i.e. the gates or functions) including which pins within the gate (if any) may be swapped.

The NET-DATA-BASE containing the DEVICE FILES may be created using 1 of 2 methods. It can come from a schematic drawing using the EXTRACT NET LIST command or it can be created from a text file input using the LOAD TXT NETLIST command.

The following is an overview of the steps necessary to utilize placement improvement commands. A more detailed description follows.

I. Update DEVICE FILES with PINSWAP line if necessary.

II. Create a NET-DATA-BASE using EXTRACT NETLIST or LOAD TXT NETLIST commands.

III. Place components on board using manual, interactive or automatic techniques.

IV. Fix or free desired components, functions and/or pins.

V. Run function swap, IC swap and/or pin swap. Re-run for desired results.

VI. Back annotate all.

I. If the DEVICE FILES were created prior to EDA 3000 rev 1 software release, then an additional line containing the pinswap information must be added to permit pin swap. This may be done by editing the device file using TEXT LEADTHRU. The PIN SWAP line appears within the PINORDER section. It must appear before the FUNCTION lines. See Figure 1. Information about which pins are swappable may be obtained from the same place that you got the rest of the pinorder information - your TTL DATA BOOK FOR DESIGN ENGINEERS.

NOTE 1: The PINSWAP line is not necessary to run IC swap or function swap.

NOTE 2: Please reference the Telesis User Manual LIBRARY-43 through 62 for a more detailed description of DEVICE FILES.

Device File 7400 DEVICE 7400

```
PACKAGE       DIP14

CLASS         IC

PINCOUNT      14

PINORDER      7400 A B Y

PINUSE        7400 IN IN OUT

PINSWAP       7400 A B

FUNCTION G1   7400 1 2 3

FUNCTION G2   7400 4 5 6

FUNCTION G3   7400 9 10 8

FUNCTION G4   7400 12 13 11

POWER         +5V ; 14

GROUND        GND ;  7

END
```

Figure 1

II. If schematic drawings have been used to create the NET-DATA-BASE, when you command EXTRACT NETLIST, the system automatically assigns all unannotated functions on the schematic to required components in the NET-DATA-BASE. The reference designators and pin numbers are then updated by the BACK ANNOTATE CURRENT command. An unannotated function is one which has the DEVICE TYPE updated while the REFERENCE DESIGNATOR and pin numbers are left in their generic state of U*, 00, etc. An annotated function is one which has the DEVICE TYPE, REFERENCE DESIGNATOR and the pin numbers updated to reflect specific package slots. The functions that are annotated by the user before EXTRACT NETLIST are considered "fixed" by the system. (i.e. the system will not automatically swap these functions unless the user "frees" them prior to running automatic function swap). Any function that was unannotated by the user (that is "free" to be assigned by the system) will be free to be function swapped. This affects functions only; IC swap and pin swap are considered free for swapping whether the function was annotated or unannotated.

If the NET-DATA-BASE was created using LOAD TEXT NETLIST, the same thing is true. Any function that is pre-assigned by the user within the $PACKAGE section, will be considered "fixed" by the system, while any unassigned function, (those specified within the $FUNCTION section), will be "free". Again, these may be "freed" or "fixed" by the user prior to running automatic function swap.

<u>NOTE 1:</u> Spare function slots in components of pre-assigned functions are not considered open for unassigned functions.

<u>NOTE 2:</u> Please reference the Telesis User Manual NETDB-11 through 14 and Place-34 through 36 for more information.

III. Now place the board using manual, interactive and/or automatic commands. You <u>may</u> run automatic swapping commands in conjunction with automatic placement commands, however, if you do, do not use the FIX ALL command on the AUTOMATIC PLACEMENT menu prior to executing AUTO-PLACE BOARD. Fixing components, functions and pins with swapping specified will cause the AUTO-PLACE BOARD execution to stop, with incomplete board placement.

<u>NOTE 1:</u> Please see the Telesis User Manual Place-1 through 67 for further instructions in using these commands.

IV. To run swapping automatically once the board is placed, go into the automatic placement menu pages and "FIX" or "FREE" desired components, functions and/or pins. Use the LIST or HIGHLIGHT commands to verify what is FIXED or FREE.

NOTE 1: To run the FIX/FREE and AUTOSWAP commands, an automatic placement keep-in rectangle must be placed on layer 147 even if the components were placed manually or interactivity.

NOTE 2: Please see the Telesis User Manual Place-35 and 36 for more detail on "FIX" and "FREE".

V. After you have set the correct items "FIXED" and "FREE" you can control what type of improvement occurs by setting function swap, IC swap and/or pin swap within the placement parameters to YES, and running AUTOPLACE BOARD. The system will then improve the placement so that the shortest vertical and horizontal ratsnest lines occur. You may re-run any of the swapping programs as often as desired until the optimum results are achieved. The PLACE-LOG lists all swapping that was made.

NOTE 1: If the design was ratsnested prior to running automatic swap commands the program moves attached ratsnest lines as it swaps components, functions and pins. When all commands specified under AUTO-PLACE BOARD are finished running, the system re-ratsnests the design.

NOTE 2: Please reference the Telesis Users Manual Place-47 through 49 for further detail.

VI. After running FUNCTION SWAP or PIN SWAP programs, the current drawing of the designed board no longer matches the schematic drawings. When you are finished running the swapping programs, save the drawing, and then run the BACK ANNOTATE ALL command. This will automatically update the schematic drawings to match the gate and pin information in the NET-DATA-BASE.

NOTE 1: Please reference the Telesis Users Manual NETDB-79 through 82 for more information on back annotation.

VII. In addition to automatic swapping routines, you may also swap components, functions and pins interactively. Within the INTERACTIVE PLACEMENT MENU pages are commands which allow you to highlight swappable functions and swappable pins. Gates may be swapped by "picking" two functions in similar device types or you may specify SWAP BY REFDES.PIN#. You may swap pins by using the SWAP PIN command. If you use any interactive swapping commands, you will still need to run BACK ANNOTATE ALL to make the schematic drawings consistant with the board drawing.

NOTE 1: You may use interactive swap commands before or after running automatic swap to achieve desired results. For example: interactively swap critical components, functions and/or pins, then FIX those items in automatic placement menu pages before running automatic swap on the rest of the board design.

NOTE 2: The fact that a function was annotated (fixed) or unannotated (free) during NET-DATA-BASE creation has no bearing on the interactive function swap commands. All functions, components and pins are considered free for interactive swap.

CHECK SHEET FOR AUTOMATIC SWAPPING

PREREQUISITES:

- o NET-DATA-BASE created with DEVICE FILES that contain a pinswap line.
- o an active, placed PCB drawing with the NET-DATA-BASE present.
- o a component placement keep-in rectangle on layer 147.

OPERATION SEQUENCE: (Once prerequisites are met)

- o AUTOMATIC PLACEMENT
- o set desired components, functions and/or pins FIXED or FREE
- o set PLACEMENT PARAMETERS for RUN FUNCTION SWAP, RUN IC SWAP and RUN PINSWAP to Y to run or N to not run
- o AUTO-PLACE BOARD
- o PRINT PLACE-LOG
- o DONE
- o SAVE DRAWING
- o DONE
- o BACK ANNOTATE ALL

CHECK SHEET FOR INTERACTIVE SWAPPING

PREREQUISITES:

o NET-DATA-BASE created with DEVICE FILES

o an active placed PCB drawing with the NET-DATA-BASE present

OPERATION SEQUENCE:

o INTERACTIVE PLACEMENT

o COMP/FUNC/PIN SWAP

o execute swapping commands desired

o DONE

o SAVE DRAWING

o DONE

o BACK ANNOTATE ALL

NEW ROUTER FEATURES IN RELEASE 3.Ø

Roger Cleghorn

Telesis Systems

Chelmsford, Massachusetts

INTRODUCTION

Included in release 3.Ø are many new features in the router. Each of these will be described in this paper.

1. NEW PARAMETERS

In release 3.Ø, there were a number of new routing parameters added and a few of the existing ones were changed. The SET PARMS menu was expanded from one page to two. These two pages are shown in figure 1.

The remainder of this section will describe each of the new or modified parameters.

1.1 GRID SIZE

This parameter has been enhanced to allow a different grid to be specified in the x and y directions. The menu box now prompts for an x grid size and then a y grid size. If no response is entered for the y grid size, it is assumed that the y grid is to be the same as the x grid.

This parameter is stored in the ROUTER-CON and COROUTER-CON files as:

```
X_GRID_SIZE n
Y_GRID_SIZE n
```

The old syntax of:

```
GRID_SIZE n
```

is still recognized and indicates a grid size of "n" in both the x and y directions.

1.2 CHANNEL PREF

This new parameter allows the user to indicate preferred routing channels. The user can define a pattern of channel preferences that will be repeated across the board. A separate preference pattern is allowed for the x direction and the y direction.

There is a separate prompt for each direction. The user can respond to each prompt with a preference pattern, where low preference is indicated by the letter L and high preference is specified by an H. For example, the response

L L H H L

indicates a pattern of 5 preferences that is to be repeated across the board. Assuming that the routing origin is at (0,0) and that the grid size is 20 mils, the sequence of preferred and non-preferred channels would be as follows:

channel location	preferred
0	NO
20	NO
40	YES
60	YES
80	NO
100	NO
120	NO
140	YES
160	YES
180	NO
200	NO
220	NO
240	YES
260	YES
280	NO
300	NO

The most common use for this parameter is to give preference to the routing channels that run between the component pins. For example, if the component pins are located at a multiple of 100 mils (e.g., 0, 100, 200, 300), the above preference pattern causes the routed lines to fall on the two channels that run between the pins (e.g., 40, 60, 140, 160, etc.). This is illustrated in figure 2.

If the user does not respond to either of the x or y channel preference prompts, it is assumed that there is no channel preference in that direction. This is how a user turns off channel preference once it has been set.

This parameter is saved in the ROUTER-CON and COROUTER-CON files as follows:

```
X_CHANNEL_PREFERENCE H/L ...
Y_CHANNEL_PREFERENCE H/L ...
```

1.3 DEF LINE WIDTH

This parameter defines the default routing line width; that is the width that is to be used for any net that does not have an explicit line width defined. It is the same as the old ROUTER LINE WIDTH menu box. The name has been changed to match up with the new MAX LINE WIDTH box.

This parameter is stored in the ROUTER-CON and COROUTER-CON files as:

```
DEFAULT_LINE_WIDTH n
```

1.4 MAX LINE WIDTH

Normally the largest line width that may be routed is equal to the difference between the grid size (or the smaller of the x and y grid sizes if they are different) and the line-to-line spacing. Any net with a line width larger than this value is skipped by the router. For example, on a 25-mil grid with a 13-mil line-to-line spacing, no net with a line width larger than 12 mils is attempted.

This new parameter allows the user to set a maximum line width to be routed that is different from the standard default. The user is prompted for a single integer line width. This value can be either larger or smaller than the default maximum line width.

This parameter is saved in the ROUTER-CON and COROUTER-CON files as:

```
MAX_LINE_WIDTH n
```

1.5 TS ALLOWED

This new parameter allows the user to specify whether or not the router should be prohibited from forming Ts. If Ts are not allowed, a routed connection will only start and end at pins. This parameter can be used in conjunction with the ECL Toolbox to achieve daisy-chain routing.

This menu box will ask if Ts are allowed. The legal responses are Y (for yes) and N (for no).

This parameter is stored in the ROUTER-CON and COROUTER-CON files as:

```
TS_ALLOWED YES/NO
```

1.6 P1 PIN KEEP AWAY, P2 PIN KEEP AWAY and P5 PIN KEEP AWAY

The pin keep away menu boxes will now prompt for two different pin keep ways. The first prompt:

```
pin keep away for pins in the net
```

requests a value that is used for all the pins in the routing window that are a part of the net being routed. The second prompt:

```
pin keep away for pins not in the net
```

allows the user to optionally specify a different value for all the other pins in the window. Figure 3 illustrates the meaning of these two different pin keep away values.

If the user does not respond to the second prompt, it will default to the same value as specified for the first prompt. This makes the pin keep aways function just as they do today.

These parameters are saved in the ROUTER-CON and COROUTER-CON files as:

```
              P1_PIN_KEEP_AWAY value1[:value2]
              P2_PIN_KEEP_AWAY value1[:value2]
P5_PIN_KEEP_AWAYS value11[:value12] value21[:value22] ...
```

1.7 P1 TOLERANCE and P2 TOLERANCE

These two new parameters allow the user to specify the delta x or delta y tolerance that is used to select connections for pass 1 or pass 2. The meaning of these two tolerances is illustrated in figure 4.

Each of these menu boxes prompts the user for a tolerance. The response must be an integer value in board drawing units (i.e., English or metric). If either of these parameters is not specified, a default value of 100 mils (or 254 metric units) is assumed.

These parameters are saved in the ROUTER-CON and COROUTER-CON files as:

```
P1_TOLERANCE n
P2_TOLERANCE n
```

1.8 P1 JOG LIMIT, P2 JOG LIMIT and P5 JOG LIMITS

These new parameters allow the user to specify a limit to the number of jogs that are created in a connection. Any routed connection with more than the specified number of jogs is failed. A value of -1 for any of the limits indicates that any number of jogs are allowed.

These parameters are saved in the ROUTER-CON and COROUTER-CON files as:

```
P1_JOG_LIMIT n
P2_JOG_LIMIT n
P5_JOG_LIMITS n1 n2 ...
```

1.9 P5 VIA LIMITS

With this new parameter the user may limit the number of vias that are allowed in a routed connection for each pass 5 execution. A value of zero totally disallows vias and a value of -1 indicates that there is no limit and, therefore, any number of vias are allowed in a single connection.

This parameter is only meaningful if the pass 5 vias allowed parameter is set to YES. If vias are not allowed in a pass 5 execution, the via limit parameter is ignored.

This parameter is saved in the ROUTER-CON and COROUTER-CON files as:

```
P5_VIA_LIMITS n1 n2 ...
```

1.10 P5 OFFSET VIAS

This new parameter specifies whether or not all the vias created by an execution of pass 5 should be shifted off grid. If requested, they will be shifted down and to the left, if possible. The size of the shift is a function of via size, grid size, minimum line to pad spacing, and the MAX LINE WIDTH parameter. The actual shift is given by the following formula:

```
SHIFT_SIZE = ((MAX_LINE_WIDTH + (2 * LINE_TO_PAD_SPACE) +
            VIA_DIAMETER + 1) / 2) MOD GRID_SIZE
IF (SHIFT_SIZE > (GRID_SIZE / 2)) THEN
        SHIFT_SIZE = GRID_SIZE - SHIFT_SIZE
```

The reason for shifting vias is that it will sometimes cause the via to block fewer routing channels. Two cases in which this is true are shown in figures 5A and 5B.

The default for any pass 5 execution is to not shift the vias.

This parameter is saved in the ROUTER-CON and COROUTER-CON files as:

```
P5_OFFSET_VIAS YES/NO ...
```

1.11 P5 VIA GRIDS

With this parameter the user can specify a special grid for vias in both the x and y directions. This restricts vias from being placed anywhere except where the via grid and the routing grid line up. For example, a via grid of 1ØØ mils on a routing grid of 25 mils means that vias are only placed on every fourth routing grid.

The user is prompted for a list of via grids in the x direction and then for a list of y via grids. Each response should consist of a list of grid sizes, one for each pass 5 execution. A value of zero for any execution is a special case that indicates no via grid, i.e., that a via can be placed on any routing grid. If no response is specified for either prompt, it is assumed that there is no via grid restriction in that direction.

The most common use for via grid is to force vias to line up with the component pins in the pass 2 direction of the board during the early pass 5 executions (e.g., line up horizontally with the component pins of vertically placed components). For multi-signal layer boards, this parameter is frequently used in conjunction with channel preference to force vias to fall on the same grid as the component pins and then give preference to routing lines in the channels between the pins and vias.

This parameter is saved in the ROUTER-CON and COROUTER-CON files as:

```
P5_VIA_GRID_SIZES xgrid1 ygrid1 xgrid2 ygrid2 ...
```

1.12 RE-RATSNEST

This new parameter allows the user to control whether or not the board should be re-ratsnested after the router has completed or the corouter has updated the drawing. An answer of Y indicates that re-ratsnesting should be executed; an N specifies that there is to be no ratsnesting.

If the user is planning to run the router several times, it is more efficient to only re-ratsnest the drawing after the last route.

This parameter is saved in the ROUTER-CON and COROUTER-CON files as:

RATSNEST YES/NO

1.13 FAILURE LIST

This new parameter allows the user to control whether or not the failure summary report is added to the ROUTER-LOG file. A response of Y indicates that the report is to be written; an N response eliminates the report.

If a user is planning to run the router with several sets of parameters, it is generally more convenient to turn off the lengthy failure report until the last run.

This parameter is saved in the ROUTER-CON file as:

ROUTER_SUMMARY YES/NO

No failure report is generated for the corouter.

2. IMPROVED PIN ESCAPE TECHNIQUE

The method that has been used until release 3.Ø in the router for directing etch as it exits from a pin has been somewhat limited. This feature is controlled by the user setable parameter known as PIN KEEP AWAY (now known as net pin keep away). Previously, if the user specified a pin keep away of zero or greater, the etch was forced to escape in the opposite direction of the user defined component orientation. For example, if the user specified that the components were vertical, all routed etch would exit or enter a pin horizontally. If the user specified a pin keep away of -1, the etch was allowed to escape in all four directions.

This method worked fairly well when all the component pin rows on the board run in the same direction. Unfortunately, this almost never happens. Even in the best case where all the components are DIPs oriented in the same direction, the connector pins are frequently running in rows opposite to the DIP pins. The situation of course gets much worse when the board contains components like chip carriers or pin grid arrays that contain rows of pins running both horizontally and vertically. Whenever there are many pin rows running in the "wrong" direction, the user is forced to turn off the directional pin escape feature (i.e., set pin keep away to -1) which allows etch to enter or exit all pins in any direction. This usually results in many of the pins being routed incorrectly.

For net pin keep away values that are greater than zero, the router will no longer assume that the preferred escape direction is opposite to the component orientation. Instead, it locates the pins that are adjacent to the pin being routed and then forces the etch to escape in the direction opposite to the direction of the adjacent pins. For example, if the adjacent pins are to the left and right, the allowed escape directions will be up and down. This means that the escape direction is no longer dependent upon the user specified component orientation and instead is based on the direction of the pin rows.

Another improvement in the way the router handles pin keep aways is that it will never let a pin keep away completely block the routing space between two adjacent pin rows. For example, if the user has specified a pin keep away of 2 but the number of available routing grids in a given direction is only 2, the pin keep away will be automatically lowered to 1 in that direction.

3. SWITCHING PREFERRED ROUTING DIRECTION

Single-sided connector pins create a unique routing problem. This is especially true for those pins that are on the side of the board that has a routing direction that is not going away from the pins. For example, if the connector is at the bottom of the board, the connector pins on the horizontal layer are often not optimally routed. After the router exits from the pin, it tends to either immediately place a via which blocks the pin on the other layer or it turns and goes horizontally which blocks many of the adjacent pins. This is illustrated in figure 6A. It would be preferable for the router to continue away from the conector for some distance before turning or placing a via.

A new feature is available in release 3.0 to help eliminate this problem. This feature consists of defining a rectangular area on a layer in which the router will reverse the normal routing direction. To solve the connector problem, this area is defined immediately above the connector pins that do not escape in the correct direction. This means that the router will favor moving away from the connector until it leaves the defined rectangular area, which is the desired effect. The results of using this switch rectangle are shown in figure 6B.

This feature is available in both the router and corouter. It is invoked by adding rectangles to the drawing that define the area(s) in which the router should reverse its preferred routing direction. These rectangles must be on drawing layers 221-234, where layer 221 corresponds to routing layer 1, 222 corresponds to layer 2, etc.

<table>
<tr><td>DONE</td><td></td><td></td><td></td><td>MENU-&gt;</td><td>LIST</td><td></td><td>CANCEL</td></tr>
<tr><td colspan="2">ROUTER GRID SIZE</td><td colspan="2">CHANNEL PREF</td><td colspan="2">SKIP PASS 1</td><td colspan="2">SKIP PASS 2</td></tr>
<tr><td colspan="2">ROUTING LAYERS</td><td colspan="2">DEF LINE WIDTH</td><td colspan="2">P1 WINDOW EXPAN</td><td colspan="2">P2 WINDOW EXPAN</td></tr>
<tr><td colspan="2">COMP ORIENTATION</td><td colspan="2">MAX LINE WIDTH</td><td colspan="2">P1 PIN KEEP AWAY</td><td colspan="2">P2 PIN KEEP AWAY</td></tr>
<tr><td colspan="2">DIAGONAL ALLOWED</td><td colspan="2">TS ALLOWED</td><td colspan="2">P1 TOLERANCE</td><td colspan="2">P2 TOLERANCE</td></tr>
<tr><td colspan="2">STATUS MSG FREQ</td><td colspan="2"></td><td colspan="2">P1 JOG LIMIT</td><td colspan="2">P2 JOG LIMIT</td></tr>
</table>

<table>
<tr><td>DONE</td><td></td><td></td><td>&lt;-MENU</td><td></td><td>LIST</td><td></td><td>CANCEL</td></tr>
<tr><td colspan="2">NUM P5 EXECUTES</td><td colspan="2">P5 WINDOW EXPAN</td><td colspan="2">P5 ROUTER TYPES</td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2">P5 LAYER PAIRS</td><td colspan="2">P5 JOG LIMITS</td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2">P5 PIN KEEP AWAY</td><td colspan="2">P5 VIA LIMITS</td><td colspan="2"></td></tr>
<tr><td colspan="2"></td><td colspan="2">P5 JOG SIZES</td><td colspan="2">P5 OFFSET VIAS</td><td colspan="2">RE-RATSNEST</td></tr>
<tr><td colspan="2"></td><td colspan="2">P5 VIA ALLOWED</td><td colspan="2">P5 VIA GRIDS</td><td colspan="2">FAILURE LIST</td></tr>
</table>

Figure 1 - SET PARMS menu pages for Router/Corouter in release 3.Ø

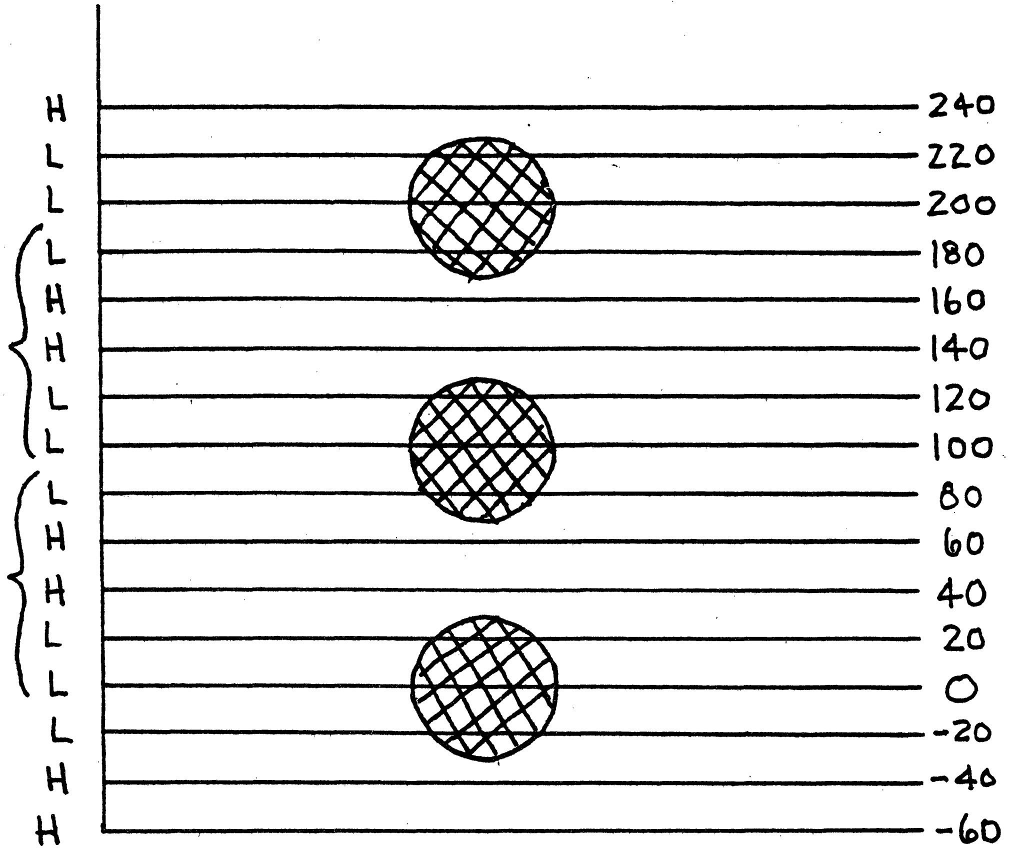


Figure 2 - Channel preference pattern of L L H H L on a 2Ø mil grid. Note that the preferred channels are those going between the pins.

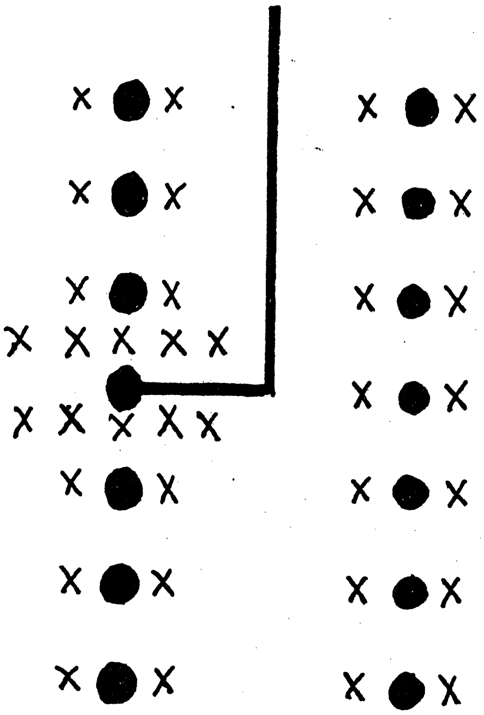


Figure 3 - Effect of the following pin keep away values:
net pin keep away = 2
other pin keep away = 1

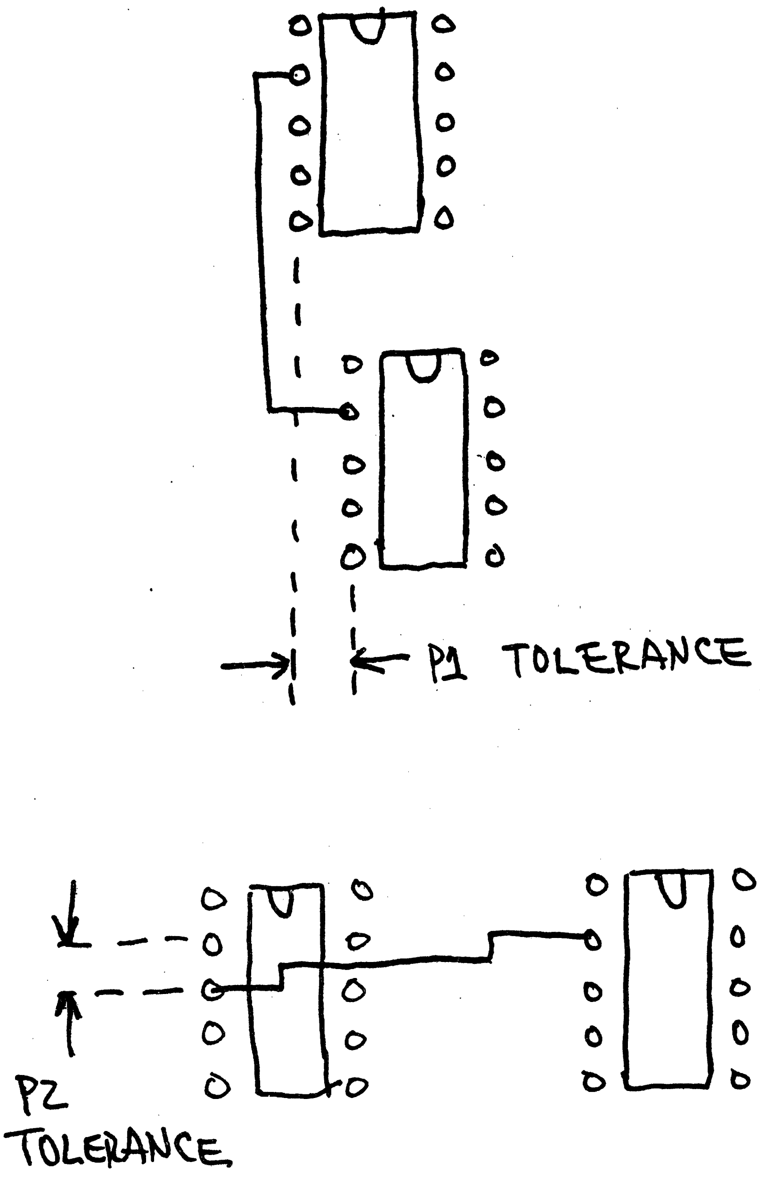


Figure 4 - Definition of P1 and P2 tolerances.

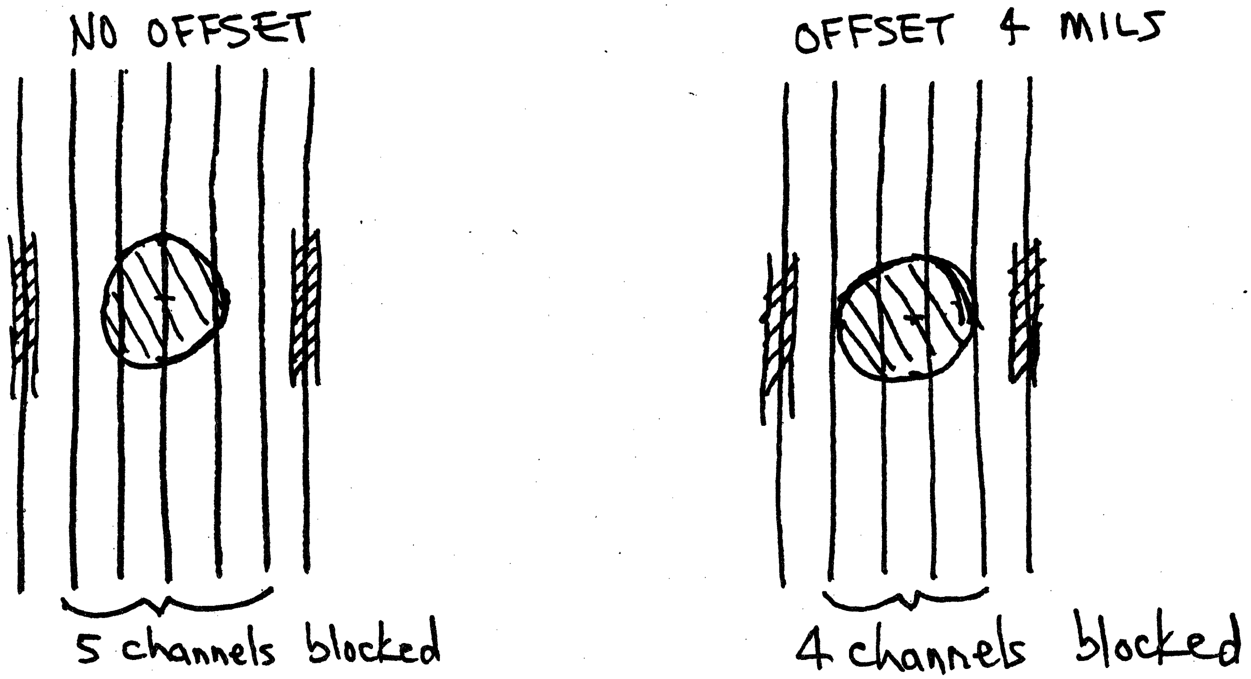


Figure 5A - Offset vias with the following design rules:
62 mil via diameter
2Ø mil grid
1Ø mil line width
1Ø mil line-to-pad spacing

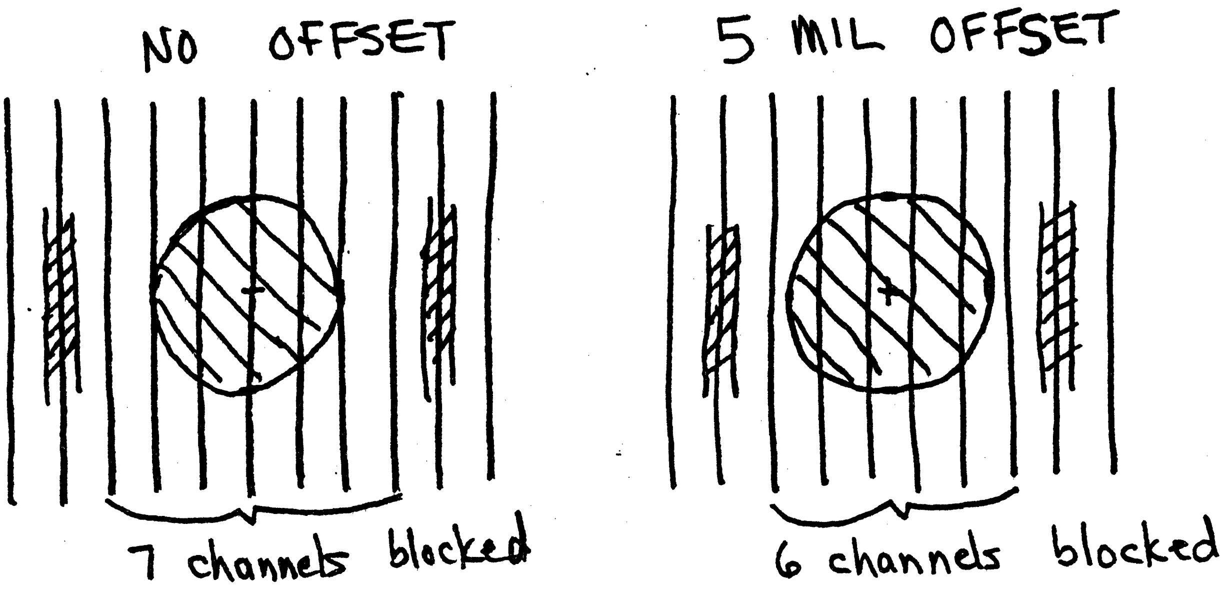


Figure 5B - Offset vias with the following design rules:
40 mil via diameter
10 mil grid
10 mil line width
10 mil line-to-pad spacing

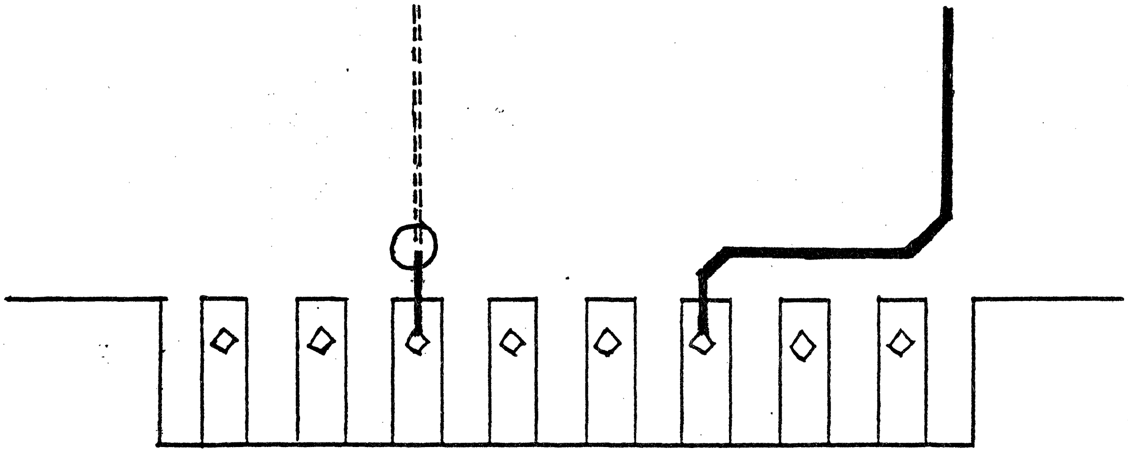


Figure 6A - Routing to connector fingers on a horizontal layer with no direction switch rectangle.

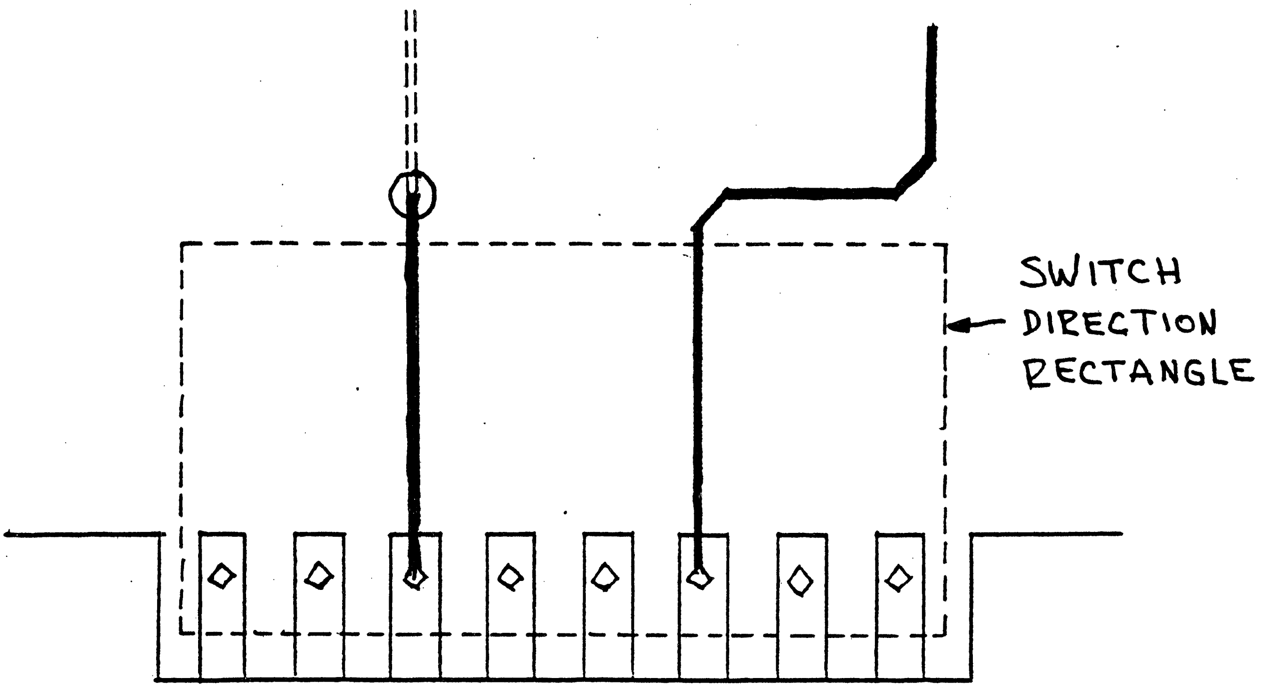


Figure 6B - Routing to connector fingers on a horizontal layer with a direction switch rectangle above the fingers.

INSIGHT -- AN EXPERT SYSTEM'S APPROACH TO ROUTING

Robert Fisher

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

Insight is a form of the Telesis router that contains all of the existing capabilities of the standard Telesis router -- capabilities which have enabled our benchmark people and many of our customers to obtain excellent results on a large number of boards. Insight allows a user to achieved the desired results without knowledge of how the router works. It accomplishes this by utilizing an exper system's approach by incorporating a large database of rules governed by the characteristics of different board types. In addition, it incorporates Release 3.Ø routing parameters into the router control parameters. The following paper outlines the Insight router in detail.

"INTRODUCTION" When we became aware that many of our customers were not using the router, we were surprised. We had seen so many boards that had obtained excellent results. We found that one reason the router wasn't being used was that some customers specialized in designs that don't lend themselves to automatic routing (such as hybrids). However, the majority of customers bypassing the router did so because they either found the parameters too difficult to set, or because they were unhappy with the results -- this was generally due to the fact that the optimum parameters were not selected.

Since the parameters selected by our in-house `experts' are based on the characteristics of the board, we decided to develop an expert system approach to determining parameter settings and controlling the router.

HOW IS INSIGHT USED?

A certain amount of basic information is required to accurately determine the control parameters. This is obtained by asking the user a series of questions, the answers to which the user must already know in order to lay out the board. Some of the questions pertain to information already existing within the Telesis system. However, we elected to make the Insight query session almost independent of the rest of the Telesis system. This allows the future option of running it on a separate editing station, such as the IBM-PC.

The output of the interactive session with Insight is a ROUTER-CON, or CO-ROUTER-CON file. This file replaces any previously generated file of the same name and may then be utilized by the router.

To illustrate its use, it is helpful to examine the set of Insight questions, listed in FIGURE 1. To understand the reason(s) for asking the questions, it is beneficial to refer to the sample ROUTER-CON form shown in FIGURE 2. If each of the routing parameters appearing in the control file is considered separately, the following discussion applies.

FIGURE 1 - INSIGHT SYSTEM PROMPTS

1. The system may ask a series of questions. Each question may be answered by typing QUIT to exit, or RESTART to answer the questions all over again.

2. What are the board dimensions? Please enter board width (in INCHES); Please enter board height (in INCHES).

3. Are there finger connectors on the edge of the board? (Y or N).

4. What are your routing layers? (Use 1-16 only). Please use a space to separate each layer number; Routing layers:

5. Which layer is the component side?

6. Which direction is the etch going on the component side? (H or V or N).

7. Which direction is the etch going on the solder side? (H or V or N).

8. What is the default line width? (in MILS).

9. What is the primary component orientation? (H or V or N).

1Ø. Are the components mostly IC types, or discretes? (I or D).

11. Do any of the components have pins on 4 sides? (Y or N).

12. Do most components have pins on 4 sides? (Y or N).

13. What is the typical IC pad height and width? Please enter the pad height (in MILS): Please enter the pad width (in MILS):

14. What is the distance between PIN 1 and PIN 2 (in MILS) of a typical IC? If there are no ICs, enter 1ØØ.

15. What is the line-to-line spacing requirement? (in MILS).

16. Which router will you be using? (RSX or COROUTER).

17. Do you wish to perform "SAVE_DWG_NEW_REV_AUTO" between multiple executes of PASS 5? (Y or N).

FIGURE 2 - FORM OF ROUTER-CON AND COROUTER-CON

```
GRID_SIZE
HORIZONTAL_LAYERS
VERTICAL_LAYERS
COMPONENT_ROTATION
DIAGONALS_ALLOWED
SKIP_PASS_1
P1_WINDOW_EXPANSION
P1_PIN_KEEP_AWAY
SKIP_PASS_2
P2_WINDOW_EXPANSION
P2_PIN_KEEP_AWAY
NUMBER_P5_EXECUTIONS
P5_PIN_KEEP_AWAYS
P5_WINDOW_EXPANSIONS
P5_LAYER_PAIRS
P5_VIAS_ALLOWED
P5_JOG_SIZES
P5_ROUTE_TYPE
P5_VIA_LIMITS
P5_VIA_GRID_SIZES
X_CHANNEL_PREFERENCE
Y_CHANNEL_PREFERENCE
DEFAULT_LINE_WIDTH
```

GRID -- Insight automatically determines the optimum grid to use based on the spacing between adjacent pins, linewidth, line-pad spacing, line-line spacing, "typical" pin size, and via size. If a non-uniform grid is advisable to allow multiple traces between pins, Insight recommends it. Since some people don't permit such a grid, Insight asks whether a non-uniform grid is acceptable. If the user responds in the negative, the "best" uniform grid for that board is substituted.

An example of a non-uniform grid follows:

Component Pin Size = 60 mils

Line-Pad Spacing = 8 mils

Line-Line Spacing = 8 mils

Linewidth = 8 mils

These design rules allow two traces between pins, however, no uniform grid accommodates this possibility. Since the two traces must be entered at 42 and 58 mils to fit between pins spaced 100 mils apart, a grid such as

21 21 16 21 21

repeated every 100 mils is necessary. Insight automatically generates such a grid for the user's approval.

Channel Preference -- One of the desirable parameters for many multi- layer boards is channel preference. Insight first determines whether it is advisable for the given board. This determination is based on the number of routing layers, grid size, component orientation, predominant component type (DIP, 4-sided, discrete), and component spacing. If channel preference is to be used, its values are based on the grid size and via size.

Routing Layers (H/V) -- For a board with an even number of routing layers, Insight alternates the settable layers between horizontal and vertical. However, some users have fixed requirements for the routing directions of the Component and/or Solder sides of the board. If two layers have been user-set to the same direction, Insight assigns the other direction to the first two unassigned layers. It then alternates the remaining layers.

If the board contains an odd number of routing layers, Insight first determines the direction of the odd layer.

It then utilizes the same procedure as for an even number of layers, adding the odd layer at the end. The direction of the odd layer is parallel to the component orientation. If the orientation is not specified, the aspect ratio of the board is utilized (based on the board dimensions indicated by the user).

Window expansion -- The PASS 1 and 2 window expansions are based on the grid size. For PASS 5, this parameter is determined by examining the grid size and the number of routing layers.

Pin Keepaway -- The pin keepaway parameters are determined from the same data as the window expansion.

Jog Size -- Jog size is also based on the same factors used for window expansion.

Via Grid Size -- With via grid size, there are two decisions to be made; whether or not to use it at all for the board, and if so, the values that should be set. Both of these decisions are based on grid size, number of routing layers, component orientation, types of components (DIPs, SMDs, discretes), and spacing between adjacent component pins.

Routing layers -- In addition to the obvious decisions (using only the specified layers, and forming horizontal/vertical pairs), Insight uses component/solder side definitions and the knowledge as to existing connector fingers, to assist in the determination of layer pairing and the order in which the router uses these pairs.

Router Type -- From the large number of boards routed by Telesis personnel, it has become evident that the "line" routing option often has an adverse affect on the results. As a result, Insight always specifies "maze" only.

WHAT ABOUT USERS OBTAINING GOOD RESULTS WITHOUT USING INSIGHT?

Several users have been obtaining good results all along. They know what the optimum parameter values are for their boards. Of course, they may continue to set the router control parameters themselves. It is recommended, however, that Insight be tested for comparison. If superior results are achieved without Insight, please forward the data to Telesis so that we may utilize it. We will be looking to continually improve the rules in the Insight database.

It is also possible to change the values of any parameters that Insight generates. Again, it is recommended that you try the parameters as they are first. If you elect to modify the Insight parameters, be aware that there are interdependencies among some of the parameters. For example, if the grid size is changed, then window expansion, jog size, and channel preference may be affected.

NEW FEATURES AVAILABLE THROUGH INSIGHT

One unique feature of Insight exists for the ROUTER-CON case (as opposed to COROUTER-CON). The router's execution time is a related to the size of the routing bit map. If the bit map is too large to fit entirely in the memory of the EDA-3ØØ or EDA-7ØØ, there is a potential slowdown in the router's operation. This decrease in efficiency may be reduced, or in many cases, eliminated entirely for multi-signal layer boards, by making each execution of PASS 5 a separate computer run. For example, if the router is set up to run one PASS 5 execution and then stopped and restarted for each successive pass, this will require a separate router set up for each execution, but will result in a much smaller bit map. This is true because the size of the bit map increases linearly with the number of routing layers -- a four layer board has a bit map that is twice the size of a bit map for the same board routed over two layers only. The end result is a much faster overall routing time.

Insight recognized the amount of memory available on the system it resides on. It also has the data necessary to calculate the size of the router bit map required for the board, if all layers are specified at once. If it determines that the bit map will be too large to fit completely in memory, it will generate a new form for the ROUTER-CON file. The router recognizes this form and will caused a separate router run to be performed for each PASS 5 execution of the multi-signal layer board.

POTENTIAL PITFALLS

Care should be exercised in generating the Router Keepin rectangle whenever Channel Preference is invoked. This is true whether parameters are manually generated, or done through Insight. If the lower left coordinate of the keepin rectangle is not a multiple of 1ØØ, the Channel Preference values must be readjusted. For Insight, the assumption is made that the keepin satisfies this requirement. Examples of acceptable coordinates are:

100   -2200   13300   0

and undesirable values include:

10   335   -70

DOES INSIGHT WORK FOR ALL BOARDS?

There are some sets of design rules for which Insight has not yet been optimized. If you find that you have a board that this is true for, please contact Telesis so that we may obtain the data and augment the Insight database with the necessary information. For example, Insight does not currently generate parameters of a 14 signal layer board. It will inform the user of this if an attempt is made to specify 14 layers. Insight does not handle 14 layer boards because we have no 14 layer board available as a test case. If we receive the appropriate data, we can experiment with it to determine the best way to enhance Insight.

CONCLUSION

What does all of this mean to the Telesis User Community? In short, Insight asks the user a set of simple questions. It then combines the answers to those questions with the knowledge of Telesis routing experts, resulting in routing parameters that are optimized for the user's board. The results of using Insight on some boards may be seen in the examples that follow:

Eq. ICs	Connections	Customer fails	Insight fails
798	4861	160	3
774	5543	227	3
359	2224	86	0
111	538	33	16
186	1116	22	6
1381	8774	168	42

Insight is a dynamic router, i.e., it will be continually enhanced. Some of the items currently being considered are the handling of more situations (such as 1Ø or more signal layers), the incorporation of some ECL controls, and the generation of metric parameters.

MULTIPLE ROUTER EXECUTIONS IN ROUTER-CON

Roger Cleghorn
Telesis Systems Corporation
September 1986

1. INTRODUCTION

The ROUTER-CON file is a text file that is used to save the current router parameter values. In release 3.0 the format of this file has been expanded so that a sequence of router runs can be defined in a single ROUTER-CON file. This feature was added to be used by the Insight Router, but can also be invoked by a user if they edit the file correctly.

In this paper the format for defining multiple runs of the router in a ROUTER-CON file will be described.

2. DEFINING PARAMETERS IN THE ROUTER-CON FILE

In release 3.0, the ROUTER menu box flips to a page that contains the commands AUTO ROUTER and INSIGHT ROUTER. The AUTO ROUTER command functions exactly as the ROUTER command did previously. When selected, it first reads the ROUTER-CON file (if one exists) to determine the most recent routing parameter values and then flips to the menu with all the routing commands (i.e., ROUTE BOARD, ROUTE WINDOW, etc.).

The most common way to change parameter values is through the SET PARMS menu page. Another method for doing this is to edit the ROUTER-CON file with the text editor before picking the AUTO ROUTER menu box.

In the ROUTER-CON file each parameter is defined on a separate line. The format of each of these lines is a keyword, starting in column one, followed by the parameter value(s). For parameters that have a large number of values (such as P5 VIA GRID SIZES), the list of values can become quite long and therefore the line of the file will become long. The only limit on the length of the line is introduced by the KEY EDIT TEXT program, which limits lines to 255 characters.

The list of all the legal ROUTER-CON line formats is shown in the table below. The first word of each line in the table is the keyword. The text of each keyword has been designed to make it obvious which parameter it is defining. The lower case words following the keyword must be replaced by the desired parameter value. When a value is shown as two uppercase letters separated by a slash (e.g., Y/N), this indicates that one of the two letters is to be selected.

```
X_GRID_SIZE size
Y_GRID_SIZE size
HORIZONTAL_LAYERS layer1 layer2 ...
VERTICAL_LAYERS layer1 layer2 ...
X_CHANNEL_PREFERENCE L/H ...
Y_CHANNEL_PREFERENCE  L/H ...
DEFAULT_LINE_WIDTH width
MAX_LINE_WIDTH width
COMPONENT_ROTATION H/V
DIAGONAL_ALLOWED Y/N
TS_ALLOWED Y/N
ROUTER_SUMMARY Y/N
RATSNEST Y/N
STATUS_FREQUENCY number
SKIP_PASS_1 Y/N
P1_WINDOW_EXPANSION size
P1_PIN_KEEP_AWAY npka[:opka]
P1_TOLERANCE size
P1_JOG_LIMIT number
SKIP_PASS_2 Y/N
P2_WINDOW_EXPANSION size
P2_PIN_KEEP_AWAY npka[:opka]
P2_TOLERANCE size
P2_JOG_LIMIT number
NUMBER_P5_EXECUTIONS number
P5_WINDOW_EXPANSIONS size1 size2 ...
P5_LAYER_PAIRS layer11 layer12 layer21 layer22 ...
P5_JOG_SIZES size1 size2 ...
P5_PIN_KEEP_AWAYS npka1:opka1 ...
P5_VIAS_ALLOWED Y/N ...
P5_JOG_LIMITS number1 number2 ...
P5_OFFSET_VIAS Y/N ...
P5_VIA_LIMITS number1 number2 ...
P5_VIA_GRID_SIZES xsize1 ysize1 xsize2 ysize2 ...
```

Any line of the file that does not begin with one of the legal keywords will be ignored. This means that comments can be added as long as they can not be interpreted as a key word. To avoid adding commands that might conflict with future new key words, it is suggested that all comment lines begin with a semicolon.

3. INVOKING ACTIONS IN THE ROUTER-CON

If the user selects the INSIGHT ROUTER command instead of the AUTO ROUTER command, the system expects that all routing actions to be performed are defined in the ROUTER-CON file. Because of this, there is no page flip after the INSIGHT ROUTER command. It simply executes all the routing actions defined in the ROUTER-CON file. The routing summary information for the entire execution will be output to a single ROUTER-LOG file.

The general format of the ROUTER-CON file required by the INSIGHT ROUTER command is as follows:

```
{commands defining parameters for first router action}
{command defining first router action}
{commands defining parameter changes for second router action}
{command defining second router action}
{commands defining parameter changes for third router action}
{command defining third router action}
                  .
                  .
                  .
```

This sequence of defining parameters followed by a router action can be as long as necessary; there is no practical limit.

The commands that define router actions are similar to those described above for specifying parameters, i.e., a keyword followed by values. A list of the legal commands is shown in the table below:

```
ROUTE_BOARD
ROUTE_WINDOW xleft ybottom xright ytop
ROUTE_NET net1 net2 net3 ...
VIA_ELIMINATE
SAVE_DRAWING_OLD_REV
SAVE_DRAWING_NEW_REV rev
```

As illustrated above, the specified routing actions are executed in the order in which they appear in the ROUTER-CON file. Also, those commands that require router parameters must be preceded by the all the commands that will define the necessary parameters values. Once a parameter has been specified, it remains unchanged until it is redefined by a subsequent command in the file. Therefore, after all the parameters have been specified for the first routing action, only those that are changing must be defined for subsequent routing actions.

The commands SAVE_DRAWING_OLD_REV and SAVE_DRAWING_NEW_REV have been included to allow the user to save the state of the drawing between runs of the router. These commands function identically to the SAVE & CONTINUE menu commands. It is highly recommended that these commands be included in any routing sequence defined in a ROUTER-CON. They will minimize the amount of routing that will be lost if the system should crash during the execution of an INSIGHT ROUTER command.

The ROUTER-CON file shown on the next page illustrates the use of all the action commands. When the user picks the INSIGHT ROUTER box, this file will cause the following sequence of events to occur.

[1] Route several critical nets with a route by net.

[2] Use a route by window to route the straight connections in a selected area of the board.

[3] Save the drawing as a new rev named TEMP.

[4] Route the remainder of the board with a route board command. After this route is completed, a summary of the router failures will be listed in the ROUTER-LOG file and the board will be re-ratsnested.

[5] Save the drawing again as rev TEMP. This will replace the previous rev TEMP.

[6] Perform via elimination.

EXAMPLE ROUTER-CON FILE WITH MULTIPLE ROUTER RUNS

```
;        set parameters for route by net
;
X_GRID_SIZE 25
Y_GRID_SIZE 25
HORIZONTAL_LAYERS 1
VERTICAL_LAYERS 2
COMPONENT_ROTATION V
DIAGONAL_ALLOWED Y
DEFAULT_LINE_WIDTH 12
SKIP_PASS_1 N
P1_WINDOW_EXPANSION 6
P1_PIN_KEEP_AWAY 0
SKIP_PASS_2 N
P2_WINDOW_EXPANSION 6
P2_PIN_KEEP_AWAY 2
NUMBER_P5_EXECUTIONS 2
P5_WINDOW_EXPANSIONS 10 20
P5_LAYER_PAIRS  1 2 1 2
P5_JOG_SIZES 8 20
P5_PIN_KEEP_AWAYS 0 -1
P5_VIA_LIMITS 4 -1
ROUTER_SUMMARY N
RATSNEST N
;
;        execute route by net
;
ROUTE_NET CLOCK1 CLOCK2
;
;        change the parameters for routing an area of board
;
NUMBER_P5_EXECUTIONS 0
;
;        execute route by window
;
ROUTE_WINDOW 3000 2500 600 4500
;
;        save the drawing as rev temp
;
SAVE_DRAWING_NEW_REV TEMP
```

```
;
;        change the parameters for routing remainder of board
;
NUMBER_P5_EXECUTIONS 3
P5_WINDOW_EXPANSIONS 10 20 40
P5_LAYER_PAIRS  1 2 1 2 1 2
P5_JOG_SIZES 8 20 40
P5_PIN_KEEP_AWAYS 0 -1 -1
P5_VIA_LIMITS 4 -1 -1
ROUTER_SUMMARY Y
RATSNEST Y
;
;        execute route board
;
ROUTE_BOARD
;
;        save drawing as rev temp
;
SAVE_DRAWING_OLD_REV
;
;        perform via elimination
;
VIA_ELIMINATE
```

CO-ROUTER TESTING RESULTS

Bobbi Smith

CAD Supervisor

Gould/SEL, Plantation, FL

Edited by, Jackie Benincasa

Applications Engineer

Telesis Systems, Chelmsford, MA

ABSTRACT

A full understanding of the speed and improved completion rates that are possible when using the Telesis Co-Router is best attained through experimentation on a sample board that is processed repetitively. The following paper outlines the techniques used by Gould/SEL Division. It is intended primarily for users who are unfamiliar with the Telesis Co-Router.

INTRODUCTION

For experimental purposes on the Co-Router, we used a 2Ø chip board and co-routed repeatedly over the course of a week. One parameter was changed for each co-route and the drawing was always updated and penplotted so that we could clearly observe and analyze the results. Such testing enabled us to understand the co-router, the new parameters, such as Channel Preference and Via Grid, and how these parameters affect the routing process. Thus, when we began designing the first production board, we were able to choose the appropriate parameters.

BACKGROUND INFORMATION

Gould's printed circuit boards tend to be 9Ø% digital, utilizing 4-6 routing layers and 2 imbedded planes. We sometimes use gate arrays and always use electromechanical parts such as LEDs and switches as well as connectors.

In the past, routing with the RSX router did not often yield 1ØØ% completions. Using the Co-Router, and with a better understanding of the parameters, we now expect to achieve 100% completion on our boards. Since delivery of the EDA-620, we have routed six boards; five of these were routed to 100% completion. For example:

```
---------------------------------------------------------------
BOARD 1  IC Count = 41; PCB Size = 3.4" x 15"; Density = 1.18
         Signal Layers = 2; Number of Connections = 212

RSX ROUTER  200 Completes; 12 Fails; 8 Hours to Route
620 ROUTER 212 Completes;  0 Fails; 1.5 Hours to Route
---------------------------------------------------------------
BOARD 2  IC Count = 298; PCB Size = 8.5" x 15"; Density = .44
         Signal Layers = 4; Number of Connections = 1541

RSX ROUTER  1505 Completes; 36 Fails; 12 Hours to Route
620 ROUTER 1541 Completes;  0 Fails; 2.5 Hours to Route
---------------------------------------------------------------
BOARD 3  IC Count = 781; PCB Size = 15" x 18"; Density = .34
         Signal Layers = 8; Number of Connections = 5543

RSX ROUTER  5293 Completes; 250 Fails; 120 Hours to Route
620 ROUTER 5543 Completes;   0 Fails;  10 Hours to Route
---------------------------------------------------------------
```

BEGINNING THE ROUTING PROCESS

The most critical step to a successful route begins with placement. The following is a brief synopsis of our placement procedure.

All I/O connectors, and all mechanically fixed parts, such as switches, are placed interactively. Using the schematic, the ICs directly related to those connectors are placed on the board. All remaining parts are then placed outside the board outline and ratsnested. The dynamic ratsnest is used to interactively guide the correct placement of ICs on the board. The major objective of this process is to achieve a ratsnest that is as completely horizontal and vertical as possible.

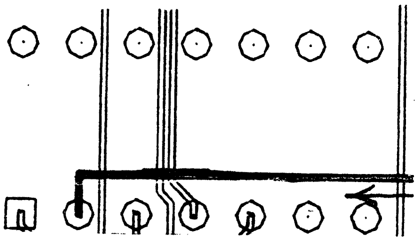
Next, we penplot the ratsnest. Congested areas and long diagonal lines are studied for possible placement improvement, using the dynamic ratsnest as a guide to a more preferable location. At this stage, we hand stitch all high-speed (non-ECL) and clock lines for fixed length and location. Any remaining diagonal ratsnest that we feel would fail to autoroute, is hand stitched at this time.

Experience has proven that the more horizontal and vertical the ratsnest appears, the more successful the router completion will be. Additionally, the number of vias added will also be lower. Another placement consideration is to adhere to a 100 mil grid during the placement process, being careful to insure that the router keep-in rectangle origin is also on 100 mil grid.

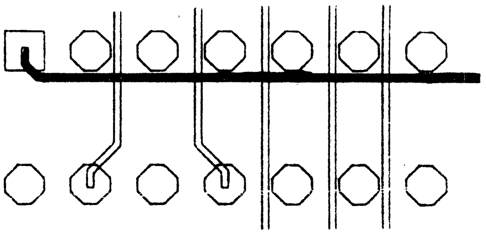
CO-ROUTING

The first phase in the routing process involves passes 1 and 2 (horizontal and vertical connections made without vias). The primary goal in this phase is NOT to block off any possible future channels. Pin keep-away and P1 and P2 tolerance, as well as channel preference assisted in this goal:

PREFERRED
(BOTTOM ROW LEAVES PADS
FREE FOR FUTURE ROUTES)



NON-PREFERRED
(TOP ROW OF PADS ARE
BLOCKED ON ONE SIDE)



In general, P1 and P2 should complete 50% or better, and repetition of these passes with increased tolerances has not been productive on Gould designs. Therefore, one iteration each of P1 and P2 prepares our boards for P5. Finally, any I/O connections that are not completed during P1 and P2 are interactively completed prior to running P5. The following is a typical set of co-router parameters used at Gould for a dense, eight signal layer board:

```
GRID SIZE: X = 20, Y = 20
CHANNEL PREFERENCE: X =(L L H H L), Y = (L L H H L)
ROUTING LAYERS: HOR = (10,8,6,2), VER = (9,7,3,1)
LINE WIDTHS: DEF = 10, MAX = 10
DIAGONALS: YES, TS: YES, COMP ROT: HORIZONTAL, RE-RATSNEST: YES
*******PASS 1*****************         *********PASS 2*****************
WINDOW EXPANSION: 5                    WINDOW EXPANSION: 5
PIN KEEP-AWAY: NET = 0,OTHER = 0      PIN KEEP-AWAY: NET = 2,OTHER = 2
TOLERANCE: 100                         TOLERANCE: 100
JOG LIMIT: -1                          JOG LIMIT: -1

***************************PASS 5 EXECUTIONS*****************************
WINDOW EXPN: 10    10    10    10    20    20    40    40    80    100
LAYER PAIRS: 3:6   7:8   9:10  1:2   1:10  2:9   3:8   6:7   1:8   1:2
NET P K A  :  0     0     0     0    -1    -1    -1    -1    -1    -1
OTHER P K A:  0     0     0     0    -1    -1    -1    -1    -1    -1
JOG SIZES  :  8     8     8     8    16    16    20    20    40    40
JOG LIMITS : 10    10    10    10    -1    -1    -1    -1    -1    -1
VIAS ALLOW : YES   YES   YES   YES   YES   YES   YES   YES   YES   YES
VIA LIMITS :  2     2     4     4     6     6    -1    -1    -1    -1
OFFSET VIAS: NO    NO    NO    NO    NO    NO    NO    NO    NO    NO
X VIA GRID : 100   100   100   100   100   100   100   100    0     0
Y VIA GRID : 100   100   100   100    0     0     0     0     0     0
ROUTER TYPE:  M     M     M     M     M     M     M     M     M     M
```

The last phase in the routing process is Pass 5. We have used from five to fifteen executes of Pass 5, depending on the density and the difficulty of the route. When P1 and P2 were routed, layer pairs were processed hierarchically; 1&2, 3&4, 5&6. On Pass 5, we chose the opposite, using 5&6 etc., as the least amount of etch exists on these last layers. Layer pairs are not mixed unless the route is in the final stage, and still not at 100% completion. This has happened only once since Gould began using the co-router, and we usually achieve 100% completion without crossing the original layer pairs.

Our basic technique has been to execute the layer pairs with Channel Preference, Via Grid, and small parameters, and then route without the Via Grid, but with more open parameters for window, jog, etc. This technique generally yields 100% completion. If it had not, the next step would be to resubmit the least used layers in crossed pairs for the final attempts. While this action was necessary on the RSX Router, it hasn't been necessary using the co-router.

CONCLUSION

We have been delighted at Gould with the routing results obtained using the 620 Co-Router. The increased speed has enabled us to better understand the software, and so our use of the Co-Router has been more productive. The new parameters appear to have been directly responsible for the completions we have achieved -- these completions were what we had anticipated when deciding to purchase the co-router.

ADVANCED TEXT EDITING

Garrett T. P. Cavanaugh

Product Manager

Donald DiMatteo

Applications Engineer

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

One of the most powerful, yet frequently underutilized tools on the Telesis workstation is the Text Editor. The following paper covers advanced text editing features currently available to the Telesis User.

INTRODUCTION

Prior to implementing the techniques outlined in the paper, the reader should be able to:

- Log the text editing terminal in on the EDA-300/700
- Open and close a text file using the text editor
- Understand the commands used in the User Documentation
- Feel confident about current knowledge level of the Text Editor

In addition, the reader should become familiar with the following conventions for this paper:

- When a character is followed by a "**kp**" it should be typed in on the keypad instead of the keyboard.
- Any **standalone** character should be typed in on the keyboard instead of the keypad.
- A "**cr**" indicates a carriage return.

- A **CTRL-[x]** indicates that you should type the control key and a letter simultaneously.

- An **ENT** refers to the enter key located on the lower right of the keypad.

- A **DEL** refers to the delete key located on the upper right of the keyboard.

- An **LF** describes the linefeed key on the right of the keyboard.

- An **sp** refers to the space bar at the bottom of the keyboard.

- **Square brackets** denote something in a command line that must be defined, for example, a buffer name, key name or file name.

HELPFUL COMMANDS FOR THE TEXT EDITOR

HELP - To access **HELP** for the Text Editor, type **PF2**, or **PF1 7kp HELP ENT.**

SEARCH - To search for a word, type **PF1 PF3 [word] PF3.** To locate additional occurences of the same word, type **PF3** again.

DELETION - Characters can be deleted in two ways. Typing **,kp** will delete the character underneath the cursor; typing **DEL** (the key in the upper right of the keyboard) will delete the character to the left of the cursor.

When deleting entire words, typing **-kp** will delete the word the cursor is on; typing **LF** will delete the word to the left of the cursor. If you wish to delete an entire line of text, typing **PF4** will remove the line to the right of the cursor, including the carriage return and linefeed. To delete an entire section, type **.kp** and move the cursor through the section that you want deleted, and then type **6kp.**

MOVING THE CURSOR - The following commands allow the user to move the cursor:

- The **four** arrow keys

- **1kp** - moves the cursor to the next word

- **2kp** - moves the cursor to the next end-of-line

- **PF1 4kp** - moves the cursor to the bottom of the text file

- **PF1 5kp** - moves the cursor to the top of the text file

- **7kp** - moves the cursor to the next section

- **8kp** - moves the cursor to the next page

ADVANCING/BACKING UP - To **advance** the cursor in the default direction, type **4kp**; to **reverse** the cursor default direction, type **5kp**. The following commands for advancing and reversing words, lines, sections and pages assume that if you are advancing you have already set the cursor in the default direction using 4kp. If you are reversing, the commands assume that you have already set the cursor in the default direction using 5kp.

⌽ To **advance or reverse the cursor one word at a time**, type 1kp. (Remember, if you are advancing, you should already have typed **4kp**, if you are reversing, you should already have typed **5kp**).

⌽ To advance or back up **one line at a time**, type **2kp.**

⌽ To advance or back up an **entire section**, type **8kp.**

⌽ To advance or back up an **entire page**, type **7kp.**

⌽ To **move to the top** of the text file type **PF1 5kp.**

⌽ To **move to the bottom** of the text file type **PF1 4kp.**

⌽ To **delete an entire section**, move the cursor through the section that you want to delete and type **6kp.**

JUGGLING TEXT - Substituting a word (or phrase) can be accomplished two ways.

- The first way is to type the following:.kp **new_word 6kp PF1 PF3 old_word PF3 PF1 ENT.**

- The second way is to type **PF1 7kp S/old word/new word/W ENT.** Note: this will replace the word or phrase for the entire file.

To **cut and paste** an entire section, move the cursor to the beginning of the section you want to cut, and type **PF1 .kp.** Finally, move the cursor to the end of the section you want to cut (it will highlight) paste and type **PF1 6kp.**

To **find the RSX name of a text file** first get into the project and pick **PROJECT INDEX **** ENT** on the function screen. The RSX names will be listed next to the Telesis names.

To **merge two text files** first obtain the RSX name of the text file you want to merge. Edit the text file you will be merging into and then type **PF1 7kp INCL [rsx_name].TXT ENT** .

To **open up a new text buffer,** type **PF1 7kp =[BUF]. ENT** . ([BUF] is the name of the buffer). Please note that a text buffer is an alternative "scratch pad" for performing text editing. It can be opened while inside the editor, enabling you to edit two or more files simultaneously. The contents of the text buffers will be forgotten and the buffers automatically cleared once you exit/quit the editor.

To **determine what text buffers you have open** type **PF1 7kp SHO BUF ENT.** To edit the buffer, type **PF1 7kp =[buffer]. ENT** . To **file the buffers** on the disk type **PF1 7kp WRITE [rsx_name].TXT ENT.** Finally, to return to the original text file, type **PF1 7kp=MAIN.ENT.**

MISCELLANEOUS COMMANDS

- To **repeat a command [x] times** type **PF1** [a keyboard number specifying the number of times you wish to repeat the command] and then the command that you want to repeat.

- To **change the case of a letter** type **PF1 1kp.**

- There are two methods for **defining a special key;** you can type **CTRL-K** gold [the letter you wish to redefine] as [the keypad commands you want to substitute for the key]. NOTE: the commands must end in a period. The second method of defining a special key is to type **PF1 7kp def key gold** [the letter you want to define] as [the line editing commands you want to súbstitute for that key]. The line editing commands must be bounded by single quote marks and end in a period.

LINE MODE - Line mode is another method of editing a text file that does not require the keypad. To access line mode, type **CTRL-Z.** The computer will respond with an * in the lower left hand corner. To return to keypad mode, type a C **cr.**

Once you are in the line mode, **HELP** can be accessed by typing **HELP cr.** This will list the available commands in the line mode. To get HELP for a **specific** command, type **HELP command.** For example: *HELP SUBSTITUTE - you need not type the entire command, just enough of it to be unique. The line edit mode automatically numbers each line. To **determine what line number you are on,** type ***T.** The following is an example of the SUBSTITUTE command: *SUB/OLD/NEW 1 THRU 1Ø - this will replace the old text with new text on lines 1-1Ø.

TEXT EDITOR APPLICATIONS

TERMINATOR-IN CONVERSION - To convert a **TERMINATOR-IN** file to a **PASS5** file, first locate all instances of $ by typing **PF1 PF3 $.** Next, type the following items four times: **PF3 PF4** (this will delete $DELETE, $ADD, $PINS, and $END). Then move to the top of the file using **PF1 5kp.** Once you are there, type **.kp sp 6kp PF1 PF3; PF3** (you are replacing

a semicolon with a space) PF1 ENT. Finally, move to the top of the file again using **PF1 5kp** and then type **with a carriage return, semicolon, space).**

REMOVING VALUE FIELDS FROM THE BOM - To remove a value field from a BOM report, you will first need to define a key that will delete the 34-54 character locations. This is accomplished by going to the end of the current line, advancing to the beginning of the next line, advancing 34 characters and deleting the next 20 characters. For example, type **CTRL-K PF1 D 2kp rt-arrow 34 (right arrow) 20 (,kp). ENT .** This will result in the following message across the bottom of your screen: **EL+C34(+C)20(D+C).** Each time that PF1 D is typed, the 34 through 54 characters will be deleted. This PF1 D may also be preceded by PF1 999 to repeat the command 999 times.

```
   project ECL-FLASH            rev  8/86            18-SEP-86   16:13:23
                                        BOM REPORT                page      1
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     package    |      device       |       component      | comp    |ref-des |total
      name      |       type        |         value        | class   |        |
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  DIP16          10102                                      IC        U23
                                                                      U22          2
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  DIP16          10105                                      IC        U9
                                                                      U19
                                                                      U18
                                                                      U17
                                                                      U16
                                                                      U15
                                                                      U14
                                                                      U13
                                                                      U12
                                                                      U11
                                                                      U10         11
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  DIP16          10106                                      IC        U7
                                                                      U6           2
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  DIP16          10107                                      IC        U3
                                                                      U2
                                                                      U1           3
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  DIP16          10109                                      IC        U8
                                                                      U5
                                                                      U4           3
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  DIP16          10111                                      IC        U21
                                                                      U20          2
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  ECON24/156     CON                                        IO        J1           1
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The results are:

```
  project ECL-FLASH            rev  8/86              18-SEP-86    16:13:23
                                    BOM REPORT                     page      1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     package     |        device         | comp     |ref-des  |total
      name       |         type          | class    |         |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 DIP16            10102                   IC         U23
                                                     U22             2
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 DIP16            10105                   IC         U9
                                                     U19
                                                     U18
                                                     U17
                                                     U16
                                                     U15
                                                     U14
                                                     U13
                                                     U12
                                                     U11
                                                     U10            11
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 DIP16            10106                   IC         U7
                                                     U6              2
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 DIP16            10107                   IC         U3
                                                     U2
                                                     U1              3
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 DIP16            10109                   IC         U8
                                                     U5
                                                     U4              3
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 DIP16            10111                   IC         U21
                                                     U20             2
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 ECON24/156       CON                     IO         J1              1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

TYPING NETLISTS - **To simplify the typing of a netlist,** begin by typing one of the sample lines of the file. For instance, **DATA-BUSØØ; U[1ØØ-131].1** . Then place this line (including the carriage return) in the paste buffer. Next paste this line x amount of times, and manually correct the addresses and pin numbers **(i.e. DATA-BUSØ5; U[1ØØ-131].6** . You will find that correcting those two numbers will be faster than typing the entire line.

TYPING PIN USE LINES - To **easily type in a device file** that has, for example, 16 inputs and 8 outputs, begin by typing the beginning of the pinuse line: **PINUSE 74LSXXX.** Next, type **.kp INsp 6kp PF1 16 PF1 6kp** (this will repeat the paste 16 times). Then, type **.kp OUTsp 6kp PF1 8 PF1 6kp** (this repeats the paste command 8 times).

SEMI-AUTOMATIC FORM FEED - To **utilize the form feed in a semi-automatic manner,** type **PF1 66 down-arrow, advance the arrow up to a convenient form feed location and type a CTRL-L.**

DEVICE FILE CONVERSION - If for example, you have a 74LSØØ device file and need a 74HCTØ$, an easy method for obtaining it would be to do a **COPY TEXT FILE** from 74LSØØ to 74HCTØ$ on the EDA-3ØØ/7ØØ. Next, edit the text file now named 74HTCØØ and then type **PF1 7kp S/74LS/74HCT/W ENT.** The conversion of an entire set of 74LSxxx device files to 54ALSxxx device files requires a seven step process.

1. Perform a **COPY TEXT FILE** from 74LSxxx to 54ALSxxx on the EDA-3ØØ/7ØØ of all the device files. This will be the most time consuming portion of the process. It is advisable to do this in a separate PROJECT so that you may save it off as your 54ALSxxx device file library. This may also be performed by first copying the entire project and then performing **CHANGE TEXT FILE NAME** on all of the device files.

2. Obtain the **RSX** names of all the device files you have renamed.

3. Create an RSX text filed name **MACRO.TMP** . This is accomplished by typing **EDT MACRO.TMP cr** . This will not show up in your current index. You will be able to use the same commands as before. Type the following into the file: **S/74LS/54ALS/W EXIT.** Character strings in this search and replace command can be substituted. **File** this two-line text file by typing **PF1 7kp EXIT ENT** .

4. Create a second RSX text file named **DEVICE.TMP** by typing **EDT DEVICE.TMP cr.** Type the following into the editor:.kp **EDT ØØØØØØxxx.TXT,MACRO.TMP cr 6kp PF1 [n] PF1 6kp.** Where [n] is the number of device files you wish to convert. This command will generate an [n] line file, by creating a cut and paste section and then repeating it [n] times. Next, manually change all of the **"ØØØØØØxxx"** text strings to their Telesis counterparts and file it away by typing **PF1 7kp EXIT ENT .**

5. Type the following on your terminal: **@MACRO.TMP**

6. The entire process only takes a few minutes but you **shouldn't attempt to do anything else on the system during this time.**

7. Finally, ask your boss for a raise -- you've just increased your productivity 1Ø zillion to 1.

DELETING COMMENTED LINES - To **delete commented lines from a pass file**, first define a key that finds all occurences of a "cr(" . Then advance to the "(" and delete that line. For example, type **CTRL-K PF1 R PF3 lft-arrow PF4 . ENT** (this defines the key GOLD R). You should see the following at the bottom of the screen: **""+CD+NL.** Finally, type **PF1 PF3 cr(PF1 999 PF1 R** -- this sets the search buffer and then repeats the key 999 times.

CONCLUSION

The Telesis Text Editor is a powerful tool -- one that with a little perseverance will help you to increase your productivity.

CREATING THE NET-DATA-BASE

Don Deily

Senior Member, Technical Staff

Frank Boyle

Product Marketing Specialist

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

The following document describes the Telesis NET-DATA-BASE, how to create it when using various types of schematic/logic input, and how to get back annotation data for the Telesis EDA-1ØØØ Design Capture System or non-Telesis CAE systems.

INTRODUCTION

The reader should be familiar with the Telesis EDA-3ØØØ PCB Design system, including text netlist grammar, but desire a deeper understanding of the NET-DATA-BASE file, and how to utilize it more productively. The following is a dictionary of definitions used within the paper.

DEFINITIONS

COMPONENT: an instance of a particular device, with a reference designator (refdes). For example, component with refdes U6, device type 74LSØØ, contained in package DIP14; component with refdes R5, device type RESISTOR, contained in package RES-4ØØ.

DEVICE: the electronic functionality contained in a physical package in a PC board. For example, a 74LSØØ is a device -- one that is contained in a DIP14 package. Devices always contain at least one function, and may contain any number of functions. The 74LSØØ contains four NAND2 functions; an axial resistor contains one function. You tell the EDA-3ØØØ system which device is in a particular package by attaching a DEVICE TYPE label to it. DEVICE FILES describes devices (see next definition).

DEVICE FILE: a text file you create that describes a device to the EDA-3000 system. It tells how many pins the device has, what package it uses, and how its function pins are arranged. Device files are described in another session.

FUNCTION: the logical unit of electrical function in the NET-DATA-BASE. In the EDA-3000 software, a schematic symbol ALWAYS represents only one function. One example of a function is a NAND2 gate. Every component contains one or more functions. For instance, a 74LS00 contains four NAND2 functions. A function always contains one or more pins.

FUNCTION DESIGNATOR: the unique name that each function instance must have. They appear in the $FUNCTIONS section of the text netlists. They are used to name unassigned functions and functions that will be back annotated, since they must have names that are recognized by both the system and the user.

NET: in the NET-DATA-BASE a net has a signal name and points to the list of pins it contains.

NET-DATA-BASE: See the next section of this document.

PACKAGE: the physical package you put on the PC board. For example, a DIP14 is a package; a RES-400 (axial resistor with 400 mil separated leads) is also a package. In an EDA-3000 PC board drawing packages are represented with symbols. Each physical package contains an electrical component such as a 74SL00 or 74244. Each component contains at least one function. The 74SL00 for example, contains four NAND2 functions. Each NAND2 function contains three pins.

PIN: in the NET-DATA-BASE, a pin is associated with both a logical function and a component, so it has a pin name and a pin number. A pin may also be a member of a net.

PIN FILE: a text file created by the user (one for each package symbol name) that tells what pad should be flashed on each board layer for every pin in the package.

REFERENCE DESIGNATOR: the unique name that each component instance must have. Each reference designator (refdes) in the input netlist will cause the creation of a component instance in the NET-DATA-BASE with that refdes. Each package on the PCB drawing has a text label of type refdes which takes on the value of the corresponding component refdes in the NET-DATA-BASE.

VALUE: a text string that specifies an electrical quantity of a component. For example, a resistor may have the value "5K 5%". How a value is assigned to a component depends on how the NET-DATA-BASE is created. In a schematic, the ATTACH VALUE label command should be used on the schematic symbol When using the text netlist, the value should be placed as a third field in $PACKAGES, or in the second field in the $FUNCTIONS section. (See the text netlist format section of the EDA-3000 User Manual).

WHAT IS THE NET-DATA-BASE?

The NET-DATA-BASE is the file that contains the logical and connectivity information about the printed circuit board designed using Telesis EDA-3000 software. When commands such as RATSNEST, ROUTE BOARD, COMPARE NET, or CREATE PHOTOPLOT need to know the logic of the design, they look in the NET-DATA-BASE.

The NET-DATA-BASE file consists of four sub-files (you can't actually see them as they are contained in the file management entry NET-DATA-BASE). These subfiles are:

COMPONENTS SUBFILE: contains all of the components of your board design.

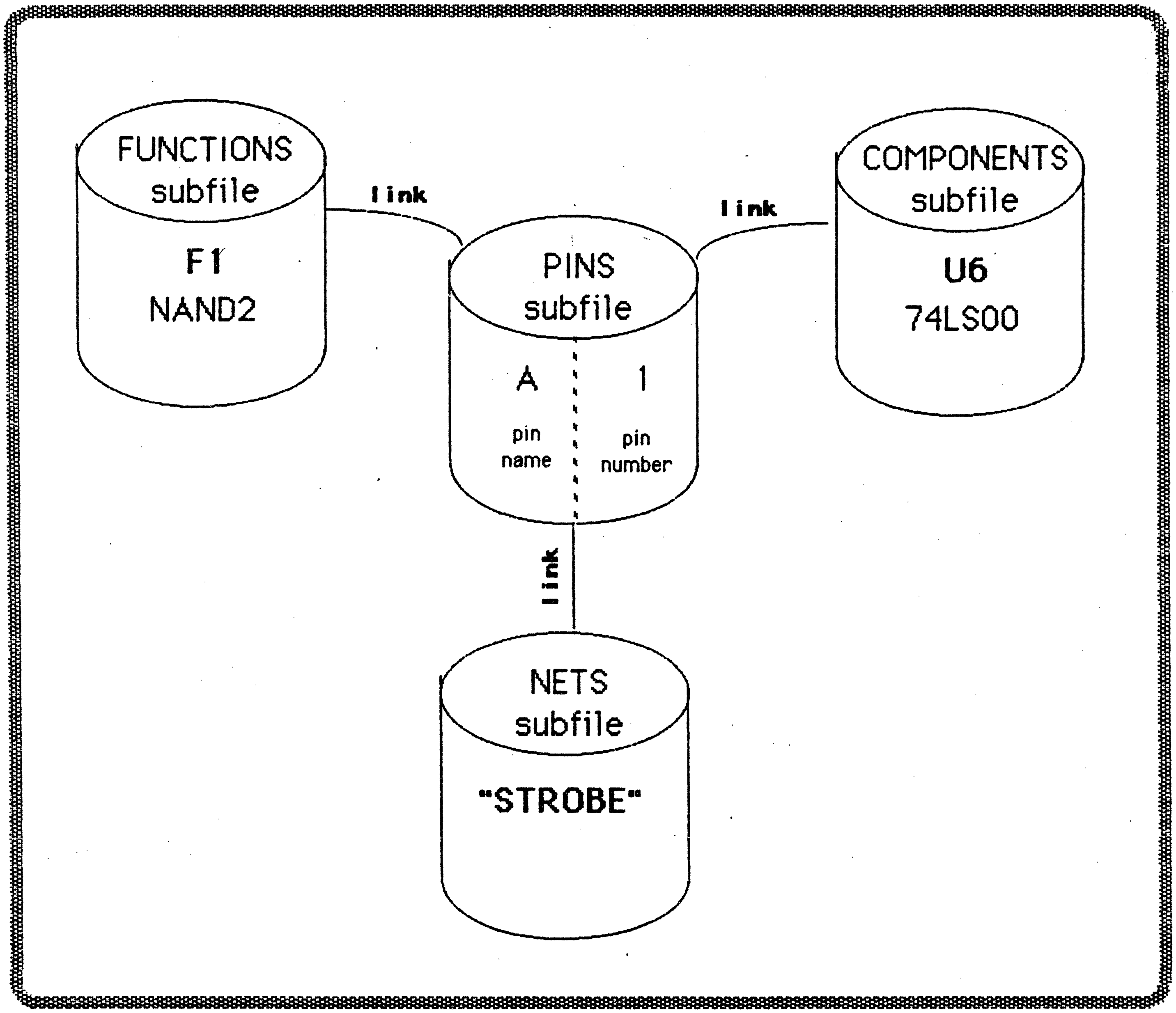
PINS SUBFILE: contains all of the pins in your board design. Each pin belongs to a function, a package, and a net. (This will be explained later).

NETS SUBFILE: contains all nets of the circuit. Each net has a net name and points to all of the pins in the net.

FUNCTIONS SUBFILE: contains all of the functions contained in all of the components of the circuit. For example, all of the NAND2 gates of all SN7400's.

All of the different revisions of the NET-DATA-BASE in one project are considered to represent the SAME circuit. (Possibly different ECO's of the circuit). Each time the NET-DATA-BASE is updated, all subsequent commands that use its circuit data (such as RATSNEST, ROUTE BOARD, COMPARE NET) will use the newest version in the project index.

NET-DATA-BASE subfiles



NET-DATA-BASE	
1986	Telesis

HOW DO I CREATE THE NET-DATA-BASE?

The NET-DATA-BASE is created using one of the following commands:

- EXTRACT NETLIST - from an active schematic drawing
- LOAD TEXT NETLIST - reading a text netlist file you have prepared.

Normally, you wouldn't use both of these commands in the same project. The appropriate use of these commands has been described below.

EXTRACT NETLIST FROM EDA-3ØØØ SCHEMATICS

This is the most straightforward way to create a NET-DATA-BASE. You begin by following the menu path for SCHEMATIC DRAWING, and create as many different schematic drawings as needed. As you complete each drawing, pick the EXTRACT NETLIST command on the menu while the drawing is active, and the system will automatically insert the new schematic data into the most recent version of the NET-DATA-BASE file. EXTRACT NETLIST will automatically assign any unassigned functions in the schematic and allow function and pin swapping if the DEVICE FILES have been set up correctly. (DEVICE FILES will be covered in another section).

The Telesis EDA-3ØØØ User's Manual, Operator's Manual pages NETDB-18 through NETDB-82 gives the details of the text netlist grammar. This document assumes that you are familiar with that material.

Before you create your text netlist input, there are three decisions you must make:

1. to back annotate or not to back annotate
2. to use device files or pin files
3. to assign functions manually or automatically

We will look at each of these decisions separately:

TO BACK ANNOTATE OR NOT TO BACK ANNOTATE

In general, text netlists come from two sources:

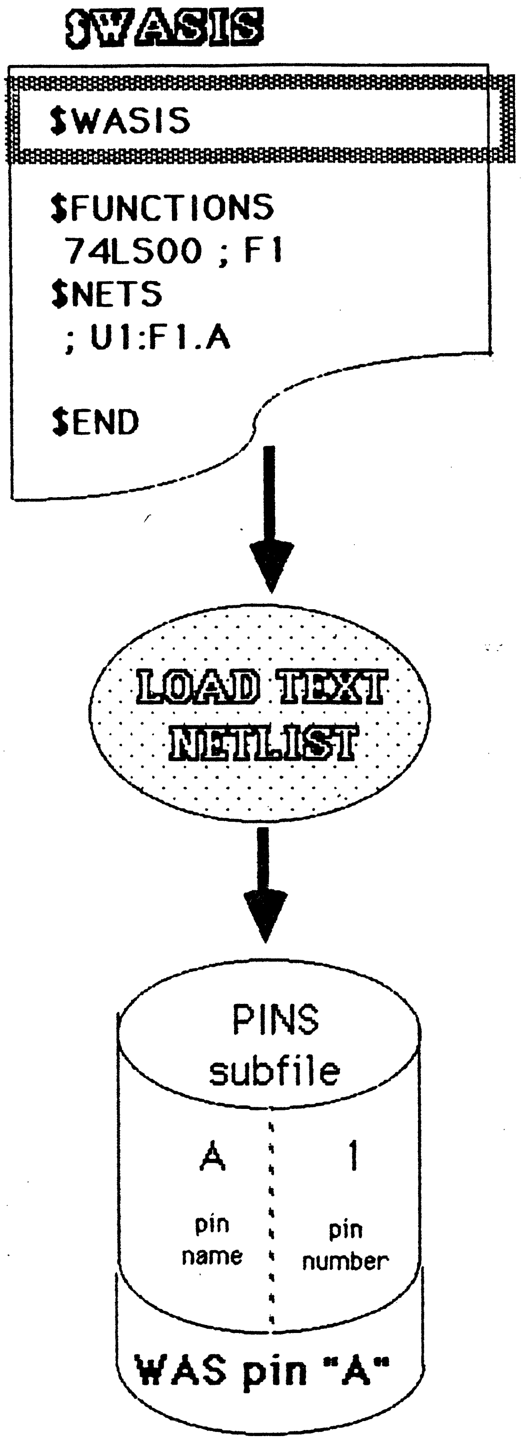
- NON-BACK ANNOTATEABLE - such as a netlist derived from a hand drawn set of schematics, or some automated method that simply supplied the netlist without any expectation of back annotation
- BACK ANNOTATEABLE - such as the netlist from a Telesis EDA-1ØØØ or other (Daisy, Mentor, or Valid) CAE systems where back annotation is possible and required.

The EDA-3ØØØ can supply a back annotation file to the EDA-1ØØØ or other CAE systems, but only if the netlist has been formatted correctly as described below. The difference is whether or not the keyword $WASIS is there or not.

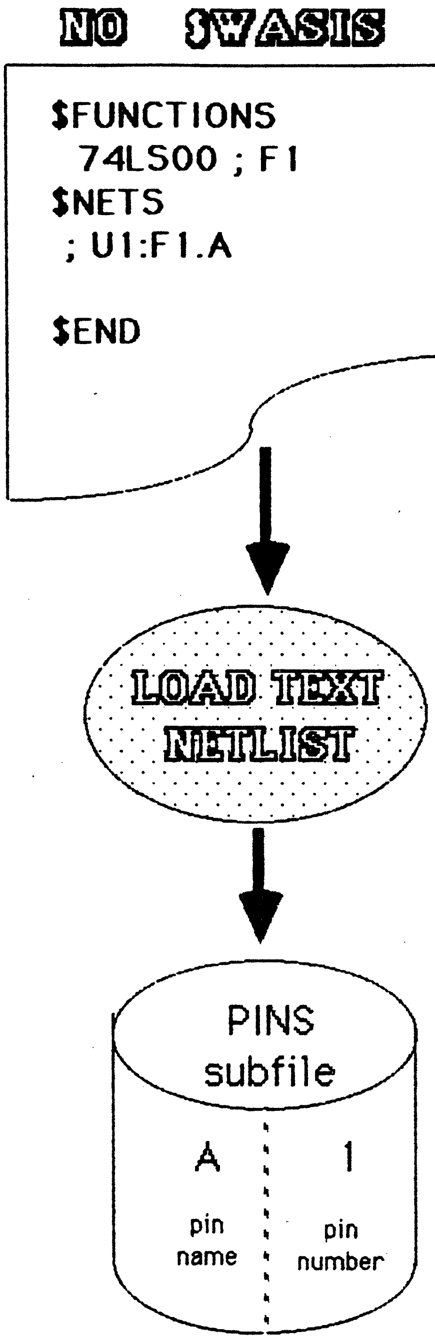
THE $WASIS KEYWORD

The $WASIS keyword at the start of the text netlist signals the LOAD TEXT NETLIST command to remember the original pin number for every pin in the NET-DATA-BASE. (The original pin identifier is stored in a special field associated with the pin). Then, regardless of what function and pin swapping you do during the PCB layout, the NET-DATA-BASE always keeps the original (WAS) pin number. The command CREATE BACK ANNOTATION FILE will then have the data it needs to create a correct file. If the text netlist doesn't have the $WASIS keyword, LOAD TEXT NETLIST won't store the original pin number in the special field, and CREATE BACK ANNOTATION FILE won't have the necessary data.

The following diagram shows the difference between the two methods:



```
$WASIS

$FUNCTIONS
 74LS00 ; F1
$NETS
 ; U1:F1.A

$END
```



```
$FUNCTIONS
 74LS00 ; F1
$NETS
; U1:F1.A

$END
```

$WASIS/no $WASIS

1986 Telesis

TO USE DEVICE FILES OR PIN FILES

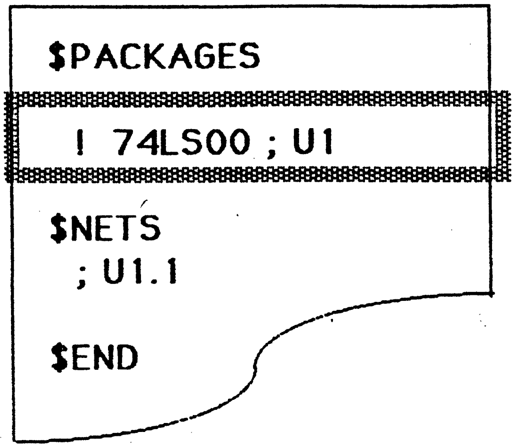
Whether you use DEVICE FILES or PIN FILES to describe your components in the netlist affects the ability to perform function and pin swapping in your PCB drawing.

DEVICE FILES: If you use device files to describe your components, then you have to put additional data into the system, but, you get greater intelligence as a result. You will be able to command interactive and automatic function and pin assignement and swapping. The details of how to use device files will be covered in another session.

PIN FILES: If you use the pin files to describe your components (which generally requires less work at set-up time), the system won't know anything about the internal structure of your components so it can only place and route them. It won't perform any function or pin swapping.

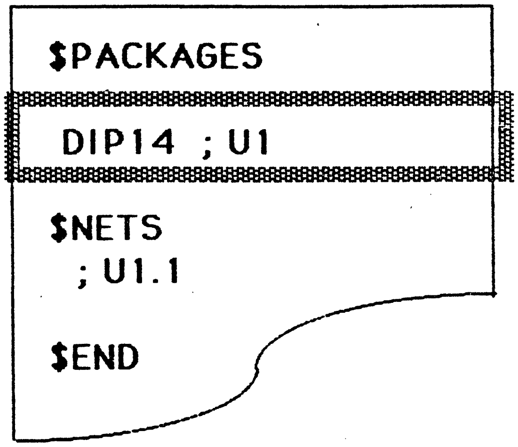
The following diagram shows the differences in format between a netlist with device files and one with PIN files.

NETLIST with DEVICE FILES



```
$PACKAGES
 ! 74LS00 ; U1
$NETS
 ; U1.1
$END
```

NETLIST with PIN FILES



```
$PACKAGES
DIP14 ; U1
$NETS
 ; U1.1
$END
```

DEVICE vs PIN FILES

1986 Telesis

TO ASSIGN FUNCTIONS MANUALLY OR AUTOMATICALLY

All components listed in the $PACKAGES section of the netlist have reference designators given by the netlist, and are therefore preassigned. All functions listed in the $FUNCTIONS section are not preassigned which makes them candidates for automatic assignment. However, if the LOAD TEXT NETLIST process finds a "reference designator: function designator" combination in the $NETS section for a function, then it considers it to be preassigned. Similarly, if the process finds a "pin name:pin number" combination for a pin in the $NETS section, it is considered preassigned. Therefore, it is possible that the entire circuit be fully assigned although a $PACKAGES section is not present.

The following table provides the possible combinations and their effects:

LOAD NETLIST:

ASSIGNED / UNASSIGNED BACK ANNOTATION

REF DES		FUNCT DES		PIN			ASSIGNED		BACK ANNOTATABLE
				#		NAME	FUNCTION	PIN	
U1			.	1			yes	yes	no
U1	:	F1	.	1	:	A	yes	yes	yes
U1	:	F1	.	1			yes	yes	yes
U1	:	F1	.			A	yes	no	yes
		F1	.	1	:	A	yes	yes	yes

ASSGN/ANNO TABLE

1986

Telesis

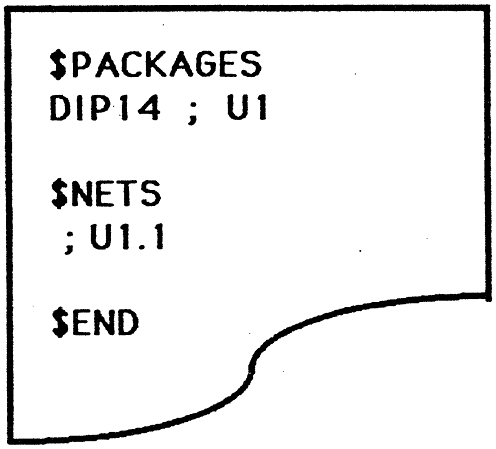
SUMMARY OF NETLIST INPUT COMBINATIONS

The following are examples of all the allowed combinations of netlist input data:

- Assigned, No Swapping, No Back Annotation
- Assigned, Swapping, No Back Annotation
- Assigned, Swapping, Back Annotation
- Unassigned, Swapping, Back Annotation
- Unassigned, Swapping, No Back Annotation

LOAD NETLIST

ASSIGNED
NO SWAPPING
NO BACK ANNOTATION



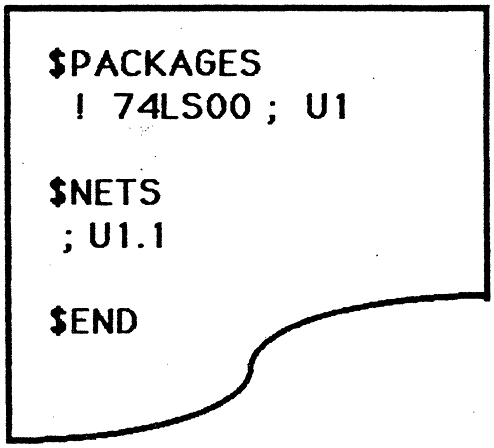
```
$PACKAGES
DIP14 ; U1

$NETS
 ; U1.1

$END
```

This example uses PIN files for component data, so swapping data is not loaded. Since $FUNCTIONS section is not present, function designators are not loaded, and so no Back Annotation data can be generated.

NETLIST INPUT FMT	
1986	Telesis

LOAD NETLIST

ASSIGNED
SWAPPING
NO BACK ANNOTATION



```
$PACKAGES
 ! 74LS00 ;  U1

$NETS
 ; U1.1

$END
```

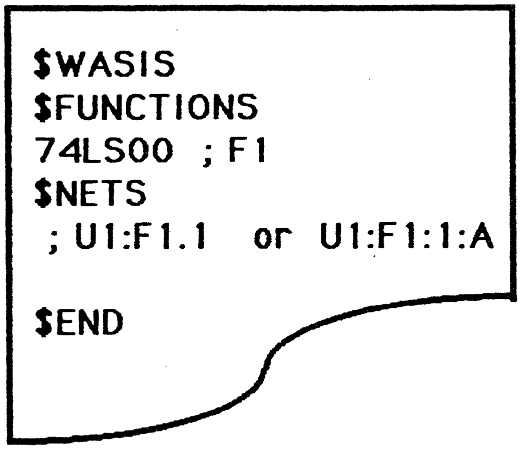
This example uses DEVICE FILES for component data, so the swapping data is loaded. Since $FUNCTIONS section is not present, function designators are not loaded, and so no Back Annotation data can be generated.

NETLIST INPUT FMT

1986 Telesis

LOAD NETLIST

ASSIGNED
SWAPPING
BACK ANNOTATION



```
$WASIS
$FUNCTIONS
74LS00 ; F1
$NETS
 ; U1:F1.1  or U1:F1:1:A

$END
```

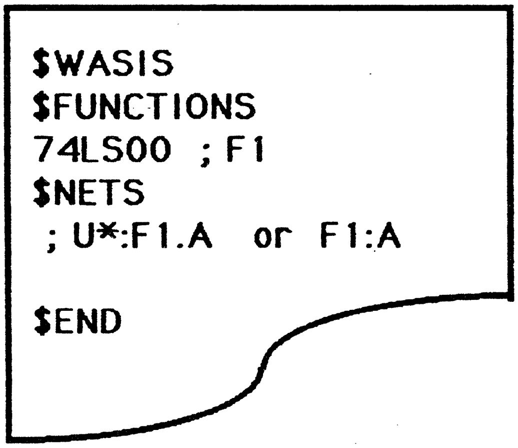
This example uses DEVICE files for component data, so swapping data is loaded. Since $FUNCTIONS section is present, function designators are loaded, and so Back Annotation data can be generated. The $WASIS keyword causes the original pin number to be stored in a special field in the NET-DATA-BASE for the Back Annotation output.

NETLIST INPUT FMT

1986 Telesis

LOAD NETLIST

UNASSIGNED
SWAPPING
BACK ANNOTATION



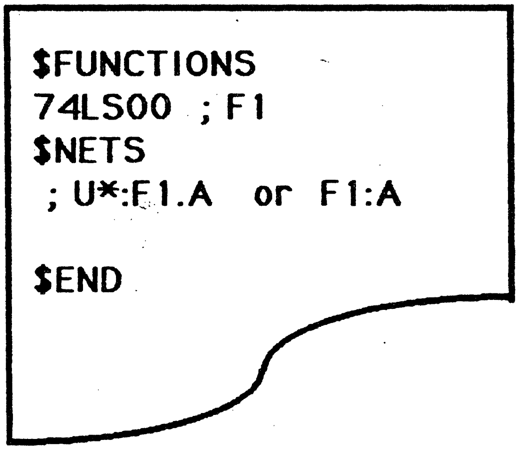
```
$WASIS
$FUNCTIONS
74LS00  ; F1
$NETS
 ; U*:F1.A   or  F1:A

$END
```

This example uses DEVICE files for component data, so swapping data is loaded. Since $FUNCTIONS section is present, function designators are loaded, and so Back Annotation data can be generated. The $WASIS keyword causes the original pin number to be stored in a special field in the NET-DATA-BASE for the Back Annotation output. Since no explicit reference designators are present, LOAD NETLIST assigns the functions, either using the series "U*" or the current default.

NETLIST INPUT FMT	
1986	Telesis

LOAD NETLIST

UNASSIGNED
SWAPPING
NO BACK ANNOTATION



```
$FUNCTIONS
74LS00 ; F1
$NETS
 ; U*:F1.A  or  F1:A

$END
```

This example uses DEVICE files for component data, so swapping data is loaded. Since $FUNCTIONS section is present, function designators are loaded, and so Back Annotation data can be generated. Since no $WASIS keyword is present, the special "was" pin field is not loaded into the NET-DATA-BASE, and so Back Annotation output cannot be created. Since no explicit reference designators are present, LOAD NETLIST assigns the functions, either using the series "U*" or the current default.

NETLIST INPUT FMT	
1986	Telesis

CONCLUSION

The Telesis NET-DATA-BASE allows you to input your circuit data in a variety of ways, enabling you to select data sources and the desired amount of automation.

ACKNOWLEDGMENTS

Frank Boyle, Charlie Powderly, Garrett Cavanaugh, and Henk Versteeg all contributed to this document.

REFERENCES

1. Telesis User's Manual, Vol. 1: Operator's Manual, "Introduction", Creating a NET-DATA-BASE.

2. Telesis User Group Meeting: "DEVICE FILES" (Friday, October 17, 1986)

DEVICE FILES

Don Deily

Senior Member, Technical Staff

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

The following paper describes how to use device files on the Telesis EDA-3000 Printed Circuit Design system with:

- EDA-3000 schematics
- manually entered netlists
- EDA-1000 netlists

INTRODUCTION

Prior to implementing the techniques defined in the paper, you should be familiar with the EDA-3000 and know how to use EXTRACT NETLIST, AUTOMATIC PLACEMENT, and ROUTE BOARD. If you are not familiar with the EDA-1000, then you can skip the sections pertaining to it.

The topics covered in the paper are:

- logic symbols
- device files
- logic symbols in relation to device files
- back annotation
- devices that are tricky such as the 244 series, connectors, and resistor packs.

DEFINITIONS

COMPONENT: an instance of a particular device with a reference designator. For example, component with reference designator U6, device type 74LS00, contained in package DIP14; component with reference designator R5, device type RESISTOR, contained in package RES-400.

DEVICE: the electronic functionality contained in a physical package in a PC board. For example, a 74LS00 is a device that is contained in a DIP14 package. Devices always contain at least one function, and may contain any number of functions. The 74LS00 contains four NAND2 functions. An axial resistor contains one function. In schematics, you specify the device type with a DEVICE TYPE label (by menu on the EDA-3000, by attribute 3 on the EDA-1000); in netlist input with the DEVICE TYPE field in the $PACKAGES or $FUNCTIONS section. You specify which device is in a particular package by attaching a DEVICE TYPE label to it. DEVICE FILES describes devices (see next definition).

DEVICE FILE: a user-created text file that describes a device to the EDA-3000 system. It defines how many pins the device has, what package it uses, and how its function pins are arranged. A device file name is always the same as the device it describes. For instance, device 74LS00 gets its data from the text file named 74LS00.

FUNCTION: the logical unit of electrical function in the NET-DATA-BASE. On the EDA-3000 and the EDA-1000, a schematic symbol **ALWAYS** represents only one function. One example of a function would be a NAND2 gate. Every device contains one or more functions. For example, a 74LS00 contains four NAND2 functions. A function always contains one or more pins.

FUNCTION DESIGNATOR: the unique name that each function instance must have if there is a $FUNCTIONS section in the text netlist. Function designators name unassigned functions, and functions that will be back annotated, since they must have names that are recognized by both the system and user. Function designators are not used in EDA-3000 schematics, because the function identifiers are internal binary codes.

NET: in the NET-DATA-BASE, a net may have a none-blank name called a signal name, and it points to the list of pins that make up the net.

NET-DATA-BASE: the file that contains all logical and connectivity information about a printed circuit board drawing. When commands such as RATSNEST, ROUTE BOARD, COMPARE NET, or CREATE PHOTOPLOT need to know the design logic or how it is connected, they look in the NET-DATA-BASE.

PACKAGE: the physical package placed on the PC board. For example, a DIP14 is a package; a RES-400 (axial resistor with 400 mil separated leads) is also a package. Packages are represented with symbols on an EDA-3000 PC board drawing. Each physical package contains an electrical component, such as a 74LS00 or 74244. Each component contains at least one function. The 74LS00, for example, contains four NAND2 functions. Each NAND2 function contains three pins.

PIN: in the NET-DATA-BASE, a pin is associated with both a logical function and a component, so it has both a pin name, and a pin number. A pin may also be a member of a net.

PIN FILE: a user-created text file that specifies what pad should be flashed on every board layer for each pin in the package. One text file should be created for each package symbol name.

REFERENCE DESIGNATOR: the unique name that each component instance must have. Each reference designator (refdes) in the input netlist causes the creation of a component instance in the NET-DATA-BASE with that refdes. Each package on the PCB drawing has a text label of type refdes, that takes on the value of the corresponding component refdes in the NET-DATA-BASE.

VALUE: a text string that specifies an electrical quantity of a component. For example, a resistor may have the value "5K 5%". How you define a component value depends on how the NET-DATA-BASE is created: In a schematic, use the ATTACH VALUE label command on the schematic symbol; in a text netlist, put the value as a third field in the $PACKAGES or as a second field in the $FUNCTIONS section. (See the text netlist format section of the EDA-3000 User's Manual).

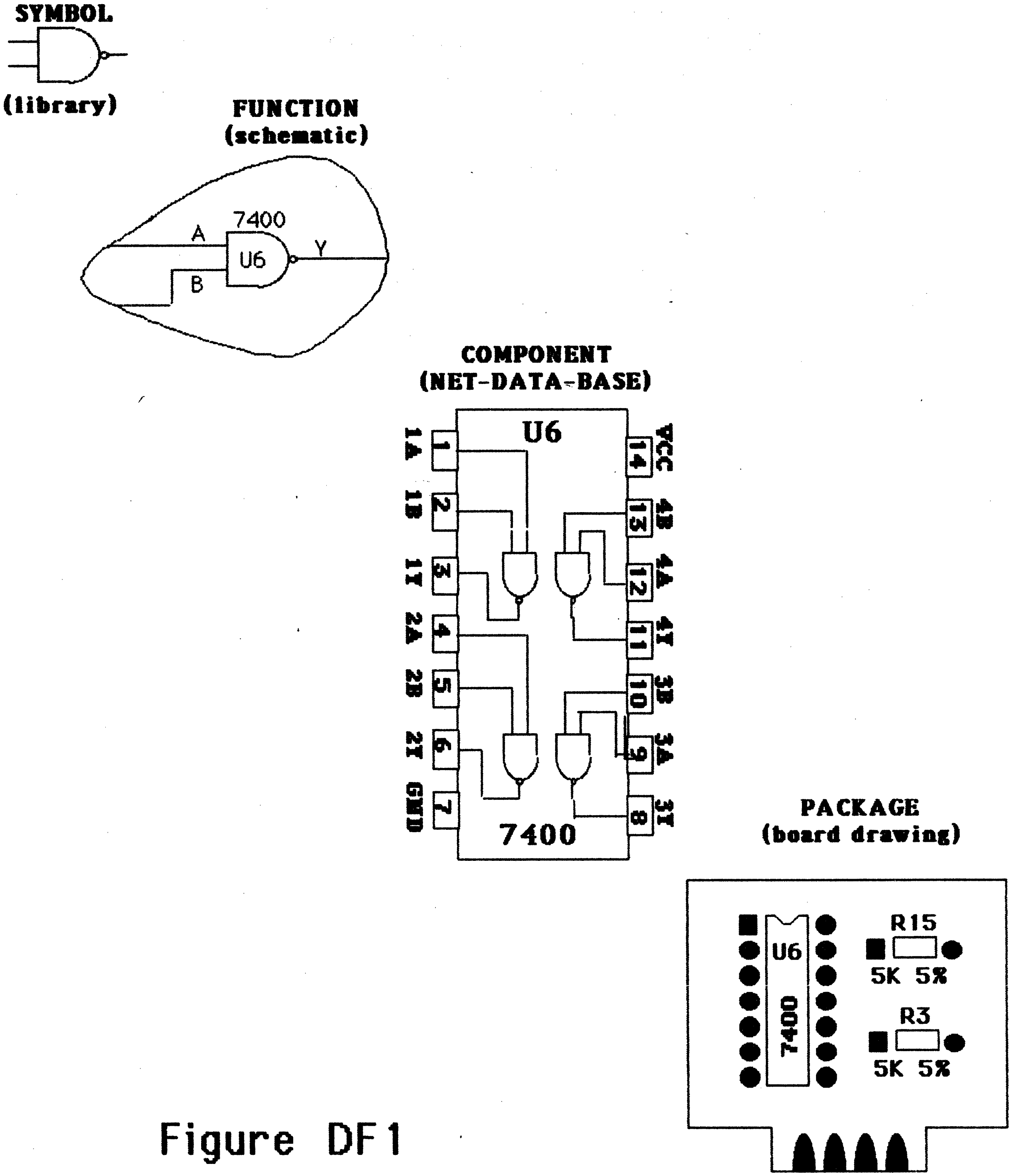

SYMBOL
(library)
FUNCTION
(schematic)
7400
A
B
U6
Y
COMPONENT
(NET-DATA-BASE)
U6
1A
1B
1Y
2A
2B
2Y
GND
VCC
4B
4A
4Y
3B
3A
3Y
1
2
3
4
5
6
7
14
13
12
11
10
9
8
7400
PACKAGE
(board drawing)
U6
7400
R15
5K 5%
R3
5K 5%

Figure DF1

WHAT ARE DEVICE FILES?

The system needs to know what logic functions exist in each device it contains, and what pins are swappable within each function. DEVICE FILES are your way of providing this information to the system. DEVICE FILES are text files that are supplied from the Telesis library, or are created using information about the device being described.

The information required to create a device file is:

- Physical PACKAGE (14 pin DIP, 4ØØ mil resistor, etc.)
- CLASS (IC, IO, or DISCRETE device)
- How many pins there are (PINCOUNT)
- What types of logic functions exist in the device and how their logical pins correspond to the device pin numbers (PINORDER)
- What pins are swappable (PINSWAP)
- How the pins are used (PINUSE IN, OUT, etc.)
- How to connect any POWER and GROUND pins

The system uses the information supplied by the device file for:

- automatic function and pin assignment
- logical design rules check (LDRC)
- power and ground pins
- component packaging
- pin and function swapping

A sample device file would be:

```
PACKAGE DIP14
CLASS IC
PINCOUNT 14
PINORDER 7400  A  B  Y
PINUSE   7400 IN IN OUT
PINSWAP  7400 A B
FUNCTION G1 7400 1 2 3
FUNCTION G2 7400 4 5 6
FUNCTION G3 7400 9 10 8
FUNCTION G4 7400 12 13 11
GROUND       GND;    7
POWER        +5V;   14
END
```

WHERE DO I GET DEVICE FILE INFORMATION?

The EDA-3000 system provides a start-up library of device files that compliments the EDA-3000 and EDA-1000 symbol libraries. (See Telesis User Manual, Volume 2, TELESIS LIBRARY, for the symbols and device files that are supplied).

You can find the information for many device files in a book that is probably already on your bookshelf, **The TTL Data Book for Design Engineers.** For example, look up the section on "SSI GATES... LOGIC & PIN ASSIGNMENTS (TOP VIEWS)" for the 7400 series. What you will see appears in FIGURE DF2.

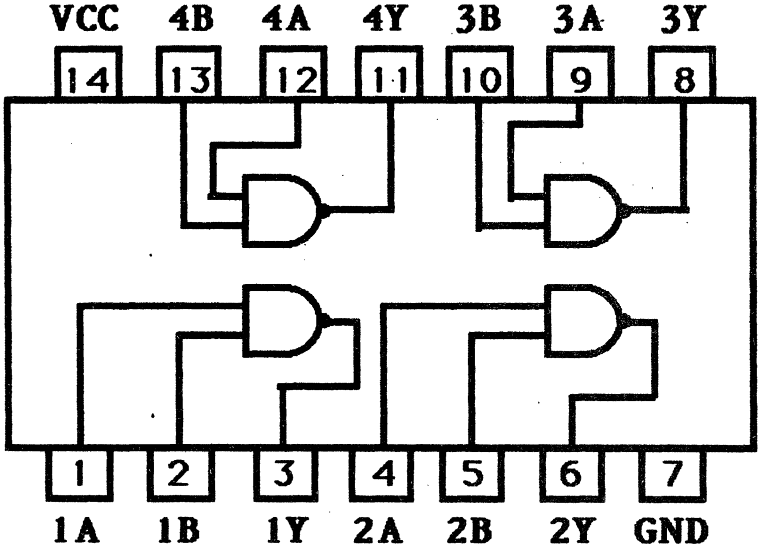


Figure DF2

This shows designers how the four logical NAND2 gates are arranged in a 14-pin Dual In-Line Package. The four gates in this diagram are examples of FUNCTIONS. Each gate can be represented on a schematic diagram by ONE SYMBOL (no more -- no less!). The 74ØØ DIP has places for four NAND2 functions; the places are called FUNCTION SLOTS.

Creating or editing device files can be done in one of three ways:

1. The Telesis Keyboard Text Editor. Since device files are text files, the text editor lets you key in new device files and edit existing ones.

2. The TEXT LEADTHRU command. This command contains an EDIT DEVICE FILE selection that has a set of menu picks that let you add/delete/modify sections of a device file and check the file for correct format when storing it.

3. You can also use the text editor on the EDA-1ØØØ and send the text device files to the EDA-3ØØØ.

WHERE DO I STORE MY DEVICE FILES?

The two commands that read device files, EXTRACT NETLIST and LOAD TEXT NETLIST, first look for them in the current project, and then in the SYSTEM-LIBRARY. Both commands use the device files to help create or update the NET-DATA-BASE in the current project. After they are done, the device files can be archived onto floppy or tape and then deleted to allow for more disk space.

SCHEMATIC LOGIC SYMBOLS

Let's review the information required by a logic symbol to correctly represent, for example, the NAND2 gate of a 74ØØ:

- three connect points; one for each pin of the gate

- each pin (connect point) must have one pin number attached

- each pin may have one pin name attached

- one reference designator

- one device type

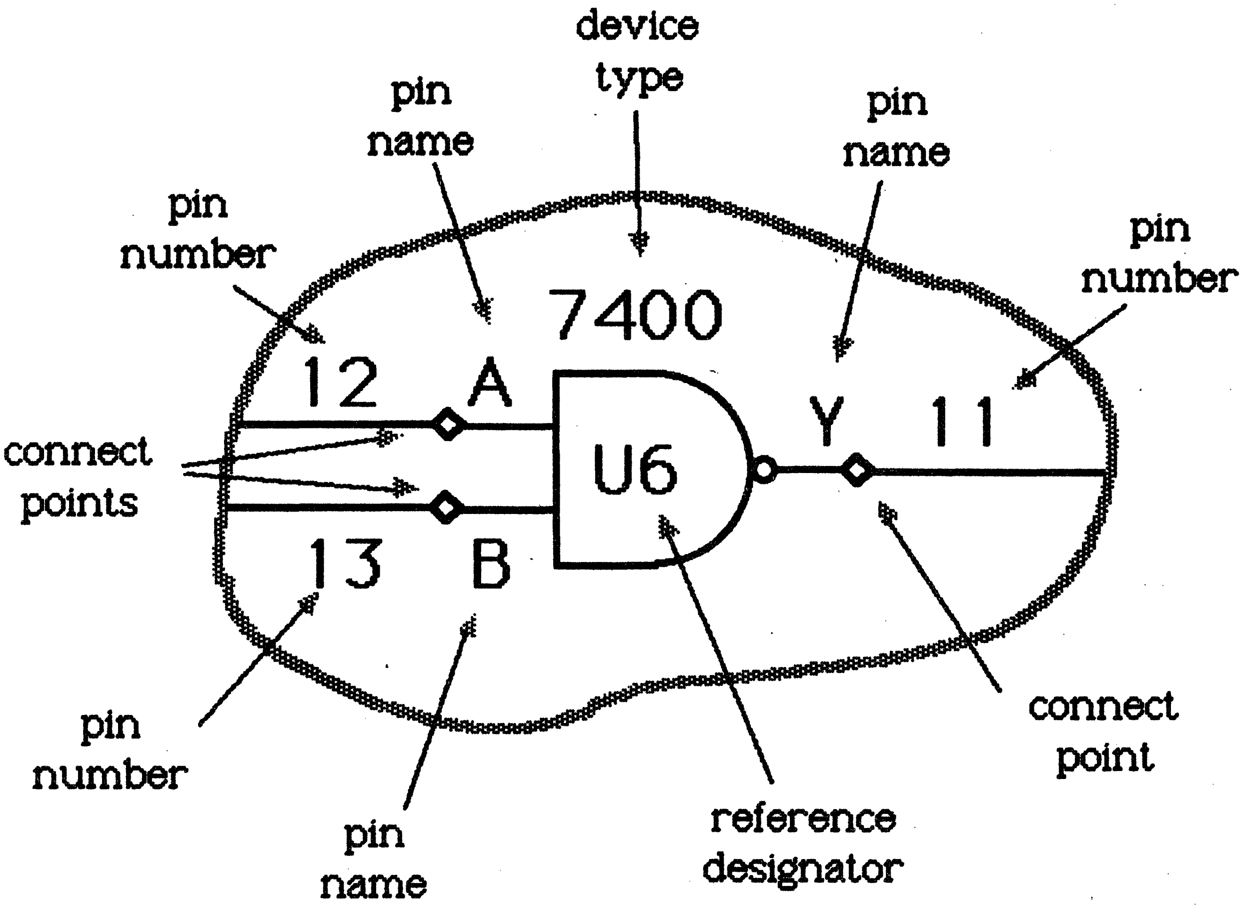


Figure DF3

SYMBOL ANATOMY	
1986	Telesis

other logic functions in the drawing, and then add the required interconnections among them. When you execute EXTRACT NETLIST, the command automatically assigns all the unassigned functions on your schematics to components it has added to the NET-DATA-BASE. If you are creating the NET-DATA-BASE from a netlist using LOAD TEXT NETLIST, it performs exactly the same assignment process.

Let's see how the device files "maps out" the relation between the schematic functions and the component:

```
PACKAGE        DIP14
CLASS          IC
PINCOUNT       14
PINORDER       7400  A  B  Y
FUNCTION G1    7400  1  2  3
FUNCTION G2    7400  4  5  6
FUNCTION G3    7400  9 10  8
FUNCTION G4    7400 12 13 11
POWER  +5V;    14
GROUND GND;     7
END
```

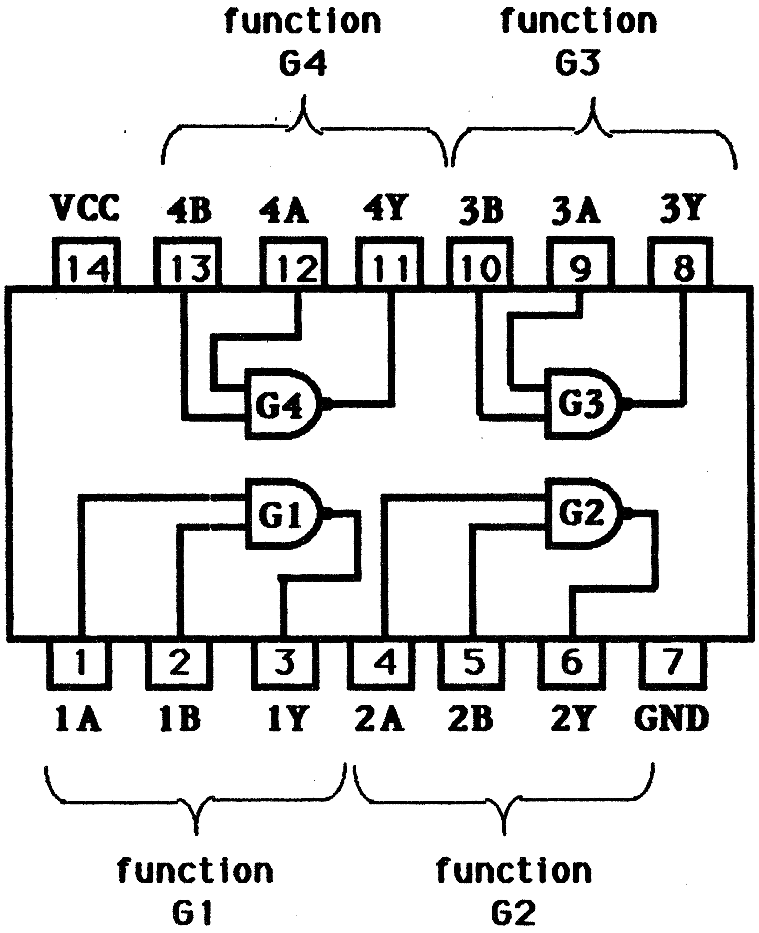


Figure DF4

DEVICE FILE MAP

1986

Telesis

Notice that the PINORDER line and the FUNCTION lines use the function type "74ØØ", as does the device type. This is the most convenient method to use when all the functions in a device are the same type. Additional information is required on the logic symbol (or, in the netlist), if there are two or more different types of functions on the device. This is shown in the first illustration in the examples section that follows.

HOW DO I CREATE A DEVICE FILE?

Now, let's build a device file using the 741Ø. Let's assume that you have no idea what a 741Ø is and that you plan to use EDA-3ØØØ features that require all the device file data. Locate the 741Ø device in your **TTL DATA BOOK FOR DESIGN ENGINEERS**. You should see what appears in FIGURE DF5.

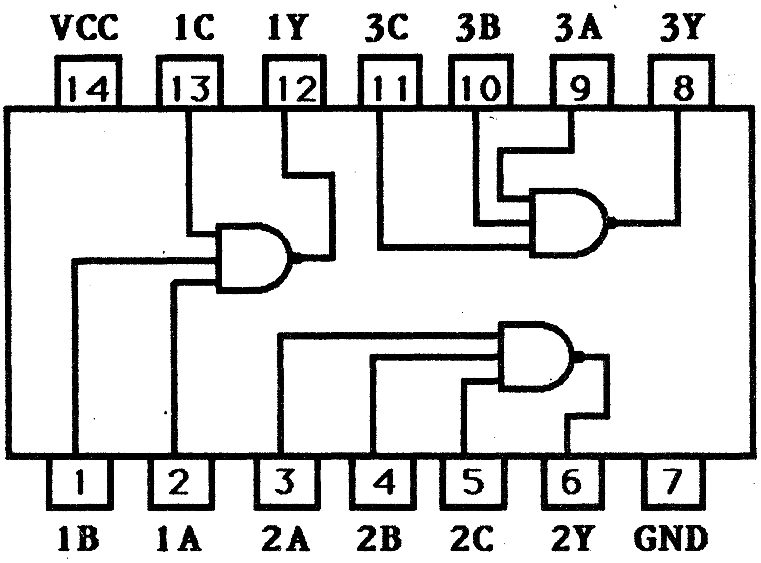


Figure DF5

7410	
1986	Telesis

You can see that the schematic symbol for a 7410 is a 3-input NAND gate, and that it uses a DIP14 package symbol in the board drawing. You can now build the device file.

NAMING A DEVICE FILE

The rules are:

- It is a normal EDA-3000 file name (18 characters maximum, 4 characters maximum revision)
- The name must **MATCH EXACTLY THE DEVICE TYPE LABEL ON THE SCHEMATIC SYMBOL OR NETLIST DEVICE TYPE FIELD/fR. We've named this device file "7410" and added the keywords as shown below:**

```
============================================================
|   PACKAGE DIP14                                          |
============================================================
```

The PACKAGE keyword gives the name of the package symbol that will be used in the board drawing. A symbol name can be 1-14 characters long, using allowed Telesis file manager name characters.

```
=============================================================
|   CLASS IC                                                |
=============================================================
```

The CLASS keyword tells the placement class of the device. It can be IC, IO, or DISCRETE:
IC for integrated circuit element (usually a DIP, SIP, or SMD die).

- **IO for input/output connectors (edge connectors or plugs).**
- **DISCRETE for all other elements (resistors, capacitors, transistors, diodes, etc.)**

```
|   PINCOUNT    14                                                |
```

PINCOUNT tells the number of pins on the physical package. For the DIP14, it is 14. Devices like op-amps, or clock chips, for example, that are in DIP packages but only have 4 pins, numbered 1,7,8, and 14, would have a pincount of 4, not 14.

```
|   PINORDER  741Ø   A  B  C  Y                                   |
```

The PINORDER keyword line opens a PINORDER section of a device file, and may contain keyword lines such as PINUSE and PINSWAP. It always contains at least one FUNCTION line. One question that must be answered when creating the PINORDER section is: "Are all the functions in this device of the same type?" Many devices, such as the 741Ø, have all the same type of functions -- in this case, NAND3's . Therefore, you will only need one PINORDER section, with three FUNCTION lines inside it. Some devices have multiple function types. For example, a 7423 has an NOR4 and an NOR4X. In that case, two PINORDER sections are required, one each for the NOR4 and the NOR4X.

The PINORDER line lists the function type of the function being defined and its pin names. All other lines in the PINORDER section list facts about the functions of this type in the component, with their pin numbers listed in the same order as the PINORDER line. Remember, the pin names are labels that you attach to the pins in schematic symbols.

```
|   PINUSE   741Ø   IN   IN   IN   OUT                                |
```

PINUSE codes are used by Logical Design Rules Check and some special commands. Pinuse codes are placed in the same order as the pin names in the PINORDER line. The pinuse codes you can use are:

- IN (input, receiving, or load)

- OUT (output, sending, or source)

- BI (bidirectional -- source or load)

- TRI (tristate -- high, low, or off)

- OCA (output collector AND -- several outputs can be tied together and the net acts as an AND gate (a dot-or gate).

- OCL (output collector LOW -- several outputs can be tied together and the net will act as an OR gate (a dot-or gate)).

- POWER - on a power pin. May connect to other POWER pins on the same net and loads (IN).

- GROUND - ground pin. May connect to other GROUND pins on the same net and loads (IN).

- NC - no connect. This pin has no electrical function.

```
=============================================================
|     PINSWAP    741Ø A  B  C                               |
=============================================================
```

Lists the names of pins that are swappable with each other. You may add any number of PINSWAP lines, each listing some of the functions' pins. Obviously, you can't have the same pin name in two different PINSWAP lines. Automatic and interactive pinswap use the pinswap information.

```
==============================================================
|    FUNCTION G1  741Ø  1  2  13  12                         |
==============================================================
```

Defines one function slot of the function type. That is, lists the pin numbers of the one instance of the function given in the PINORDER line. There are as many FUNCTION lines in one PINORDER section as there are function slots of that function type. For example, there are three FUNCTION lines of the 741Ø. The format of the FUNCTION line is:

```
===============================================================
| FUNCTION slot-name function-type pin-# pin-# pin-# ......   |
===============================================================
```

The slot-name is one to four characters long, with alphanumeric characters (Ø to 9, A to Z) being used only. The slot-name is used by

placement and other programs to communicate with you about the device, so it must be unique. No other slot-name in this device file can have the same name.

The pin numbers listed must be in the same order as their corresponding pin names in the PINORDER line. There must also be the same number of pin numbers as pin names, as the system matches the names with the numbers when creating each function instance in the NET-DATA-BASE.

```
=================================================================
| POWER     +5V  ;  14                                          |
=================================================================
```

lists pins that are not associated with any function in the component but must be tied to a power net. The format is:

```
-------------------------------------------------------
POWER  signal-name   ; pin-#  pin-#   pin-# ....
-------------------------------------------------------
```

The signal-name may contain blanks. There must be at least one pin-number and there may be any number of pin-numbers in the list. They MAY NOT be pin-numbers that occur in any FUNCTION line in the device files. There may be any number of POWER lines.

```
=================================================================
| GROUND  GND ;  7                                              |
=================================================================
```

lists pins that are not associated with any function in the component, but must be tied to a ground net. The format is:

```
-----------------------------------------------------------------
GROUND signal-name ;  pin-#  pin-#   pin-# ....
-----------------------------------------------------------------
```

The signal-name may contain blanks. There must be at least one pin-numbe and there may be any number of pin-numbers on the list. There MAY NOT be pin-numbers that occur in any FUNCTION line in the device file. There may, however, be any number of GROUND lines.

```
=================================================================
| NC  3  11                                                     |
=================================================================
```

lists component pin-numbers not to be connected to any net. The format is:

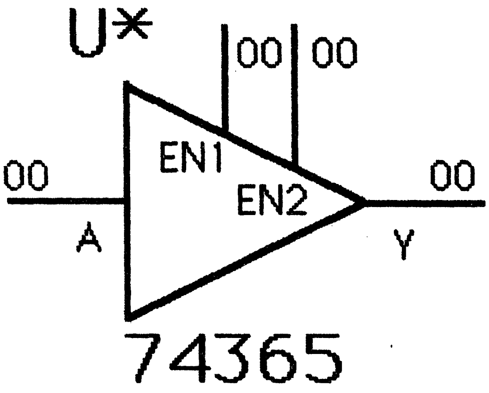
```
----------------------------------------------------------------
NC  pin-number  pin-number   pin-number .....
----------------------------------------------------------------
```

If there is an NC section, it must contain at least one pin number, and may contain any number.

The complete 741Ø device file looks like this:

```
PACKAGE      DIP14
CLASS        IC
PINCOUNT     14
PINORDER     741Ø     A     B    C   Y
PINUSE       741Ø    IN    IN   IN  OUT
PINSWAP      741Ø     A     B    C
FUNCTION G1  741Ø     1     2   13  12
FUNCTION G2  741Ø     3     4    5   6
FUNCTION G3  741Ø    11    1Ø    9   8
POWER         +5V ;  14
GROUND       GND  ;   7
END
```

SCHEMATIC



PACKAGE

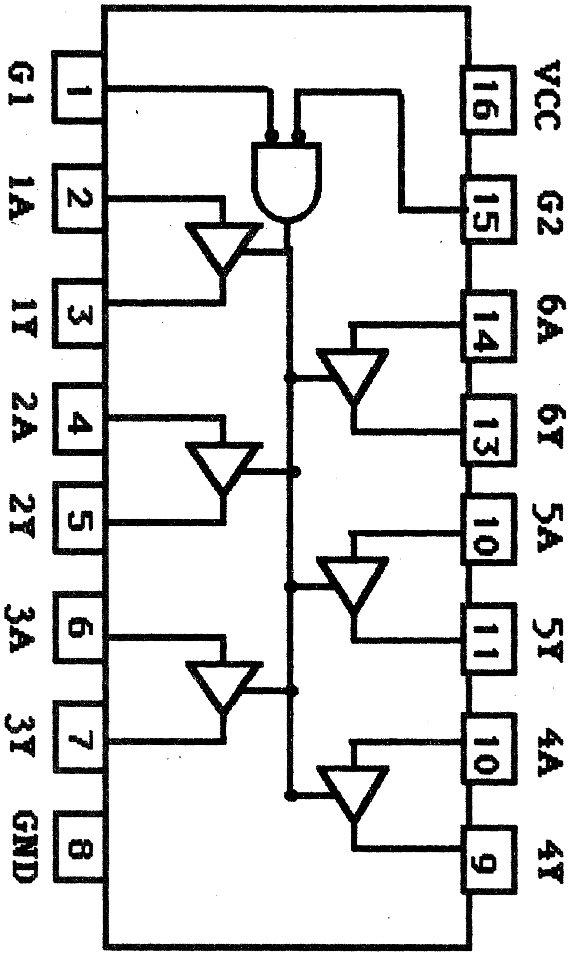
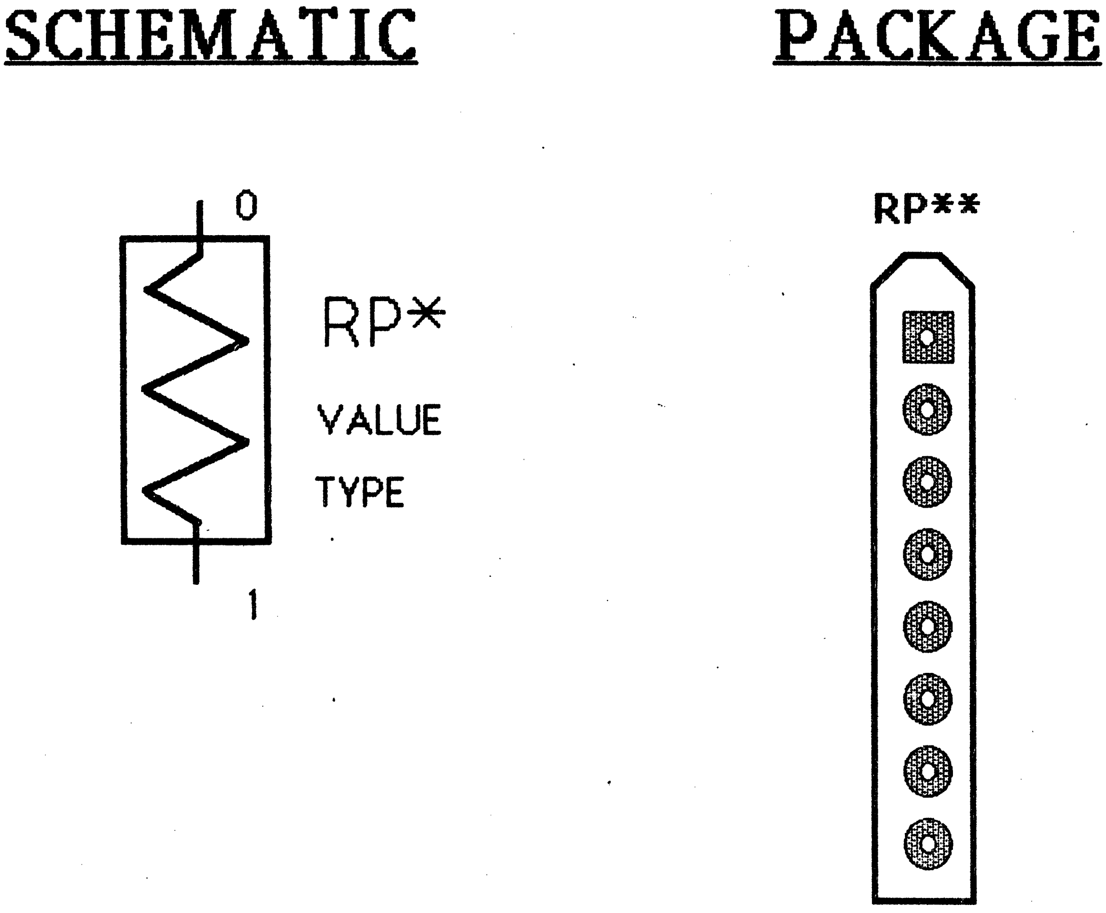


Figure DF6

74365	
1986	Telesis

Notice that the schematic symbol has four pins. EN1 and EN2 will be common to all the drivers in the component. Therefore, simply list the pins in each function in the same manner:

```
PACKAGE     DIP16
CLASS       IC
PINCOUNT    16
PINORDER    74365   A   EN1  EN2  Y
PINUSE      74365  IN   IN   IN   OUT
PINSWAP     74365       EN1  EN2
FUNCTION G1 74365   2     1   15  3
FUNCTION G2 74365   4     1   15  5
FUNCTION G3 74365   6     1   15  7
FUNCTION G4 74365  14     1   15  13
FUNCTION G5 74365  12     1   15  11
FUNCTION G6 74365  1Ø     1   15  9
POWER +5V; 16
GROUND ;    8
END
```



SCHEMATIC

PACKAGE

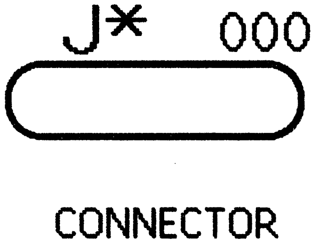
Figure DF7

RESISTOR PACK	
1986	Telesis

RESISTOR PACK - FIGURE DF7

```
PACKAGE     SIP8
CLASS       DISCRETE
PINCOUNT    8
PINORDER    RP8   A   B
PINSWAP     RP8   A   B
FUNCTION R1 RP8   1   2
FUNCTION R2 RP8   1   3
FUNCTION R3 RP8   1   4
FUNCTION R4 RP8   1   5
FUNCTION R5 RP8   1   6
FUNCTION R6 RP8   1   7
FUNCTION R7 RP8   1   8
END
```

SCHEMATIC



PACKAGE

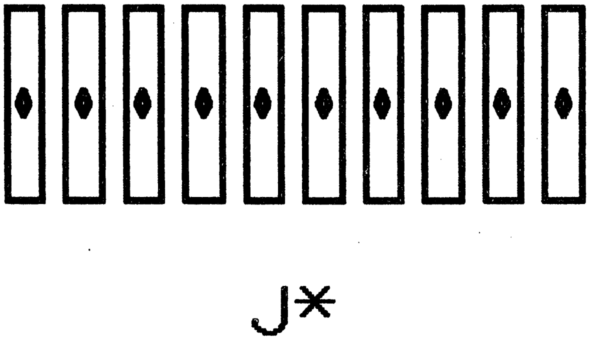


Figure DF8

EDGE CONNECTOR	
1986	Telesis

CONNECTOR - FIGURE DF8

THE EDA-3000 applies a special rule to CLASS IO devices: if there is no PINORDER section, then EACH PIN IS STORED AS A SEPARATE FUNCTION. Since there are typically single IO pins represented by symbols scattered around the schematics, this allows you to do so without having to create a large device file with many FUNCTION lines in it, one for each pin.

```
PACKAGE  ECON/24
CLASS    IO
PINCOUNT 24
END
```

EDA-1ØØØ SYMBOLS AND DEVICE FILES

Garrett P. Cavanaugh

Product Manger

Don Deily

Senior Member, Technical Staff

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

The following paper deals with EDA-1ØØØ schematic symbols, their corresponding device files, and their respective benefits and limitations.

INTRODUCTION

This paper examines four different classes of schematic symbols; the nand gate, 74244, resistors, and connectors. Examples of how to use the standard EDA-1ØØØ Library, showing the corresponding device files will be provided for each class. A discussion of the benefits and limitations for each method is included as well.

There are certain questions that should be answered prior to implementing the standards suggested in the paper. Specifically, these questions are:

- Do you require ANSI standard symbols?
- Do you use Automatic gate swapping?
- Do you use Automatic pin swapping?
- Do you use Interactive gate swapping?
- Do you use Interactive pin swapping?
- Do you use Back Annotation?
- Do you design your own schematics, or obtain them from an outside source?

- Is loading an area (vs. a symbol) an issue at your company?

- Do you use Logical Design Rule Checking?

Once these questions have been addressed, certain solutions will become more favorable to you and your company than other, equally valid, yet undersirable alternatives.

GIVENS AND ULTIMATE TRUTHS

There are certain facts that every designer must know and adhere to when using the EDA-1ØØØ package:

1. Every schematic symbol on the IBM must have a corresponding FUNCTION LINE in a device file.

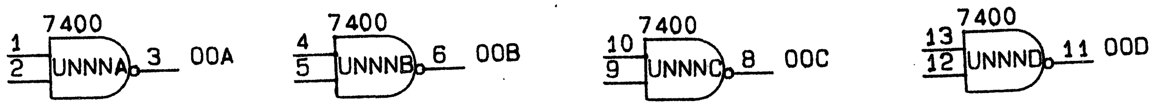
2. Every EDA-1ØØØ schematic symbol must have a device name (attribute=3) a reference designator (attribute=2, it doesn't need to be assigned), and pin names/numbers (attributes=1,2Ø,21,22, or 23 -- they must be unique names inside the symbol).

3. It's better to use generic pin names, for example, A,B,Y, rather than pin numbers, as pin numbers will prevent the pinswapping of netlists created on the EDA-1ØØØ.

4. Every EDA-1ØØØ may optionally have a function type (attribute=user-defined).

5. EDA-1ØØØ passes only one piece of information to the EDA-3ØØ/7ØØ -- either the pin name, or the pin number. This information is used for swapping and back annotation.

NAND GATES (74xx00)



```
(7400)
(THIS DEVICE AGREES WITH THE EDA-1000
 SYMBOLS OOA - OOD)
PACKAGE DIP14
PINCOUNT 14
CLASS IC
PINORDER 7400 A B Y
PINUSE 7400 IN IN OCA
PINSWAP 7400 A B
FUNCTION S1 7400  1 2 3
FUNCTION S2 7400  4 5 6
FUNCTION S3 7400 10 9 8
FUNCTION S4 7400 13 12 11
POWER +5V; 14
GROUND GND; 7
END
```

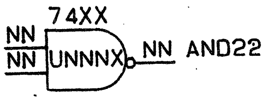
PROS/PLUSES

- callable from EDA-1000 standard library
- pins swappable
- only reference designators are backannotateable (if left unassigned)

CONS/MINUSES

- gates are not swappable because of preassigned pin numbers
- reference designators must be changed (globally or locally)
- not an ANSI standard symbol

AND 22



```
(7400)
(THIS DEVICE AGREES WITH THE EDA-1000
 SYMBOLS AND22)
PACKAGE DIP14
PINCOUNT 14
CLASS IC
PINORDER 7400 A B Y
PINUSE   7400 IN IN OCA
PINSWAP  7400 A B
FUNCTION S1 7400 1 2 3
FUNCTION S2 7400 4 5 6
FUNCTION S3 7400 10 9 8
FUNCTION S4 7400 13 12 11
POWER +5V/ 14
GROUND GND; 7
END
```

PROS/PLUSES

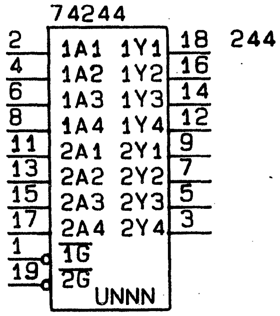
- callable from EDA-1000 standard library
- gate swappable
- pins swapable
- reference designators and pins are backannotateable

CONS/MINUSES

- reference designators must be changed (globally or locally)
- device name must be updated
- pin names must be changed (globally or locally)
- not an ANSI standard symbol

74LS244

[METHOD 1]



```
(74244)
(THIS DEVICE FILE AGREES WITH
THE EDA-1ØØØ SYMBOLS 244)
PACKAGE DIP2Ø
PINCOUNT 2Ø
CLASS IC
PINORDER 74244 2 4 6 8 11 13 15 17 18 16 14 12 9 7 5 3 1 19
PINUSE 74244 IN IN IN IN IN IN IN IN OUT OUT OUT OUT OUT OUT OUT OUT IN
FUNCTION S1 74244 2 4 6 8 11 13 15 17 18 16 14 12 9 7 5 3 1 19
POWER +5V; 2Ø
GROUND GND; 1Ø
END
```

PROS/PLUSES

- callable from the EDA-1ØØØ standard library

- only the reference designators are back annotateable

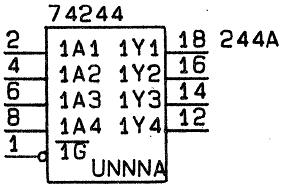
CONS/MINUSES

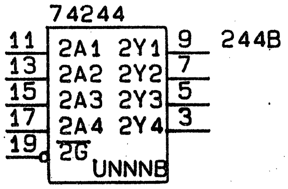
- gates and buffers are not swappable

- not an ANSI standard symbol

74LS244

[METHOD 2]





```
(74244)
(THIS DEVICE AGREES WITH THE
 EDA-1ØØØ SYMBOLS 244A - 244B)
PACKAGE DIP2Ø
PINCOUNT 2Ø
CLASS IC
PINORDER A 2 4 6 8 18 16 14 12 1
PINUSE A IN IN IN IN OUT OUT OUT OUT IN
FUNCTION S1 A 2 4 6 8 18 16 14 12 1
PINORDER B 11 13 15 17 9 7 5 3 19
PINUSE B IN IN IN IN OUT OUT OUT OUT IN
FUNCTION S2 B 11 13 15 17 9 7 5 3 19
POWER +5V; 2Ø
GROUND GND; 1Ø
END
```

PROS/PLUSES

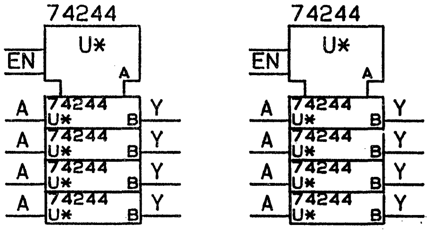
- callable from the EDA-1ØØØ library as a symbol
- only the reference designators are backannotateable
- only a block of four buffers is swappable

CONS/MINUSES

- individual gates and buffers are not swappable
- not an ANSI standard symbol
- function designator must be added to symbol

74LS244

[METHOD 3 -- ALTERNATE A. (TWO DISSIMILAR FUNCTIONS)]



```
(74244)
(THIS DEVICE FILE AGREES WITH
 SPECIAL TELESIS SYMBOLS)
PACKAGE DIP20
PINCOUNT 20
CLASS IC
PINORDER A EN
PINUSE A IN
FUNCTION S1 A 1
FUNCTION S2 A 19
PINORDER B A Y
PINUSE B IN OUT
FUNCTION G1 B 2 18
FUNCTION G2 B 4 16
FUNCTION G3 B 6 14
FUNCTION G4 B 8 12
FUNCTION G5 B 11 9
FUNCTION G6 B 13 7
FUNCTION G7 B 15 5
FUNCTION G8 B 17 3
POWER +5V; 20
GROUND GND; 10
END
```

PROS/PLUSES

- callable from the library as an area
- reference designators and pin names are backannotateable
- individual gates and buffers are manually swappable
- may be built to resemble ANSI standard symbol

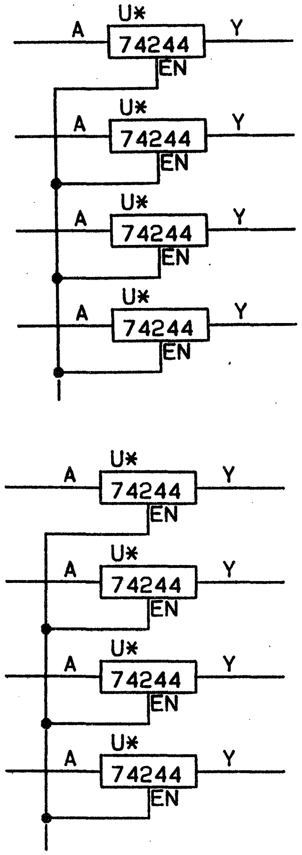
CONS/MINUSES

- autoswap will not swap these properly (no common enables)
- function descriptor must be added to all symbols

74LS244

[METHOD 3 -- ALTERNATE B. (EIGHT 3-PIN FUNCTIONS)]

```
(74244)
(THIS DEVICE FILE AGREES WITH
 SPECIAL TELESIS SYMBOLS)
PACKAGE DIP 2Ø
PINCOUNT 2Ø
CLASS IC
PINORDER 74244 A Y EN
PINUSE 74244 IN OUT IN
FUNCTION G1 74244 2 18 1
FUNCTION G2 74244 4 16 1
FUNCTION G3 74244 6 14 1
FUNCTION G4 74244 8 12 1
FUNCITON G5 74244 11 9 19
FUNCTION G6 74244 13 7 19
FUNCTION G7 74244 15 5 19
FUNCTION G8 74244 17 3 19
POWER +5V; 2Ø
GROUND GND; 1Ø
END
```



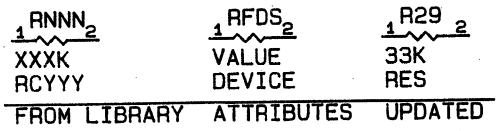
PROS/PLUSES

- callable from the library as an area
- reference designators and pin names are backannotate-able
- individual gates and buffers are manually swappable
- autoswap will swap these properly

CONS/MINUSES

- must remember to add the connections to the symbols prior to filing
- must have 8 reference designators and devices shown
- not an ANSI standard symbol

RESISTORS



```
(RES)
(THIS DEVICE FILE AGREES WITH THE
PACKAGE RES400
PINCOUNT 2
CLASS DISCRETE
PINORDER RES A B
PINUSE RES BI BI
PINSWAP RES A B
FUNCTION S1 RES 1 2
END
```

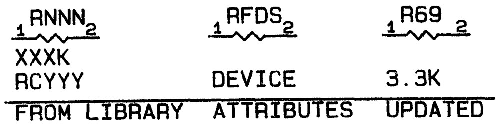
PROS/PLUSES

- callable from the library as a symbol
- pins are swappable
- reference designators and pins are backannotateable

CONS/MINUSES

- do not use automatic gate swapping
- library figure must be changed to properly use pin swapping

R1-R2 (where the value is used as the device field)



```
 (some given value)
 (THIS DEVICE AGREES WITH
  THE EDA-1ØØØ SYMBOLS R1 - R2)
PACKAGE RES 4ØØ
PINCOUNT 2
CLASS DISCRETE
PINORDER some_given_value A B
PINUSE some_given_value BI BI
PINSWAP some_given_value A B
FUNCTION S1 some_given_value 1 2
END
```

PROS/PLUSES

- callable from the library as a symbol

- functions and pins may be swapped manually or automatically

- reference designators and pins are back annotateable

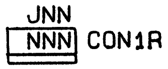
CONS/MINUSES

- library figure must be changed to properly use pin swapping (pin names changed to A & B)

- library must be changed to change value attribute to device attribute (value field changed to device field)

- device file library grows exponentially

EDGE CONNECTORS



```
(CONN)
(THIS DEVICE FILE AGREES WITH THE
 EDA-1ØØØ SYMBOLS CON1L or CON1R)
PACKAGE CONN-whatever
PINCOUNT you_supply
CLASS IO
END
```

PROS/PLUSES

- callable from the library as a symbol

- functions and pins may be swapped manually, automatically, or with floating connectors

- pins are backannotateable

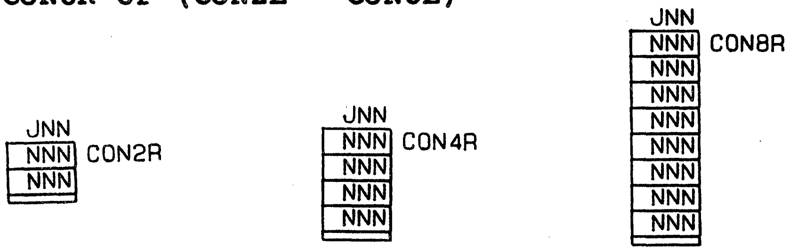
CONS/MINUSES

- library figure must be changed to add device field

- creates one function per pin in the device file

- only these symbols may be used in schematics

CON2R - CON8R or (CON2L - CON8L)



```
(CONN)
(THIS DEVICE FILE AGREES WITH THE
 EDA-100) SYMBOLS CON1L or CON1R)
PACKAGE CONN-whatever
PINCOUNT you_supply
CLASS IO
PINORDER A 1 2 3 4 5 6 7 8
PINSWAP A 1 2 3 4 5 6 7 8
FUNCTION S1 A 1 2 3 4 5 6 7 8
PINORDER A1 1 2
PINSWAP A1 1 2
FUNCTION S1 A1 9 10
PINORDER B 1 2 3 4 5 6 7 8
PINSWAP B 1 2 3 4 5 6 7 8
FUNCTION S1 B 11 12 13 14 15 16 17 18
PINORDER C 1 2 3 4
PINSWAP C 1 2 3 4
FUNCTION S1 C 19 20 21 22
END
```

PROS/PLUSES

- callable from the library as a symbol
- functions and pins are swappable manually or automatically
- pins are backannotateable

CONS/MINUSES

- library figure must be changed to add device field
- library figure must be changed to add function designator field
- you have to build one device file per connector and it will generally only be used once.
- each PINORDER section must agree EXACTLY with the schematic symbol

CONCLUSION

Once you have determined a certain set of standards, the EDA-1ØØØ Schematic Capture package linked to the EDA-3ØØ/7ØØ provides an effective design.

A HISTORY OF SPLIT PLANES ON THE TELESIS SYSTEM

Jay Kenney

Telesis Systems Corporation

1. INTRODUCTION

During the past few years, several methods have surfaced to produce split planes on the Telesis workstation. Each one had its own advantages and disadvantages. This paper describes the different solutions to split-planes as they evolved on the Telesis system. But first, a little background information on imbedded and split planes.

Board designers often utilize imbedded planes in an effort to gain more routing "real-estate" and to provide shielding between signal layers. By taking some of the board's larger nets (measured in number of nodes) and connecting them to an internal plane, the user eliminates the need for a large number of routed connections on his signal layer. Typically, power and ground nets are the ones imbedded onto an internal plane because they are typically the largest nets on a board.

In addition to "freeing-up" routing real-estate, imbedded planes act as a shield between two signal layers. Because of the large mass of copper on the internal plane, etch on the signal layer above is protected from from noise and interference from the signal layer below and vice-versa.

In spite of their benefits, imbedded planes do add one more physical layer to a board design, raising the board's manufacturing cost. For boards with multiple power and ground signals, the addition of an imbedded plane for each signal is very costly. To avoid this, designers have been splitting these planes so that one plane is shared by multiple signals. The plane is still a mass of copper but now the mass of copper is separated into sections, one for each net that the user wishes to imbed.

On the Telesis system, an imbedded plane is declared in the LAYERSTD file with an entry such as:

```
DBLAYER 5   IMBEDDED-PLANE   +5V
```

With this mechanism, a user can specify one and only one net name for an imbedded plane.

When CREATE PHOTOPLOT processes pins defined on dblayer 5 (from the above example), it checks the net of that pin. If the pin's net matches the layer's net (from LAYERSTD), a "thermal-relief" is flashed and a connection is made to the surrounding copper on that layer. If the pin's net does not match the layer's net, an "anti-pad" is flashed and the pin is isolated from the surrounding copper. Thus, any pad that is on the "+5V" net will be flashed as a thermal relief while all other pads will be flashed as anti-pads.

So we see that the CREATE PHOTOPLOT program handles imbedded planes naturally, but needed some user intervention in order to be able to include multiple nets on one plane.

2. EDITING -ART FILES

2.1 How it worked ...

The first (and simplest) method of doing split planes was to edit the gerber (-ART) files on the Telesis system using the text editor. The basic idea was to create one gerber file for each net on the split plane and merge them by using the "cut and paste" feature of the text editor.

For each net on the split-plane one Gerber file was produced which treated that net as if it were the only net on that plane. Each gerber file would have a thermal-relief section for its own net nodes and antipads for all other nets' nodes.

By using advanced text editing techniques, the user would edit one of the gerber files, calling it the master, and would cut and paste the thermal-relief aperture sections from the other gerber files into the master. For each thermal-relief section added, the user would have to then delete all the anti-pads in the master that corresponded to the newly added thermal-reliefs. Remember, the master originally contained antipads for all those thermal-relief sections being added.

2.2 It had its good points ...

This method was a pretty straightforward way of getting split-planes. It was easy to understand, and reliable. Besides it was the first way people could think of to solve this problem, and it did solve the problem. However ...

2.3 It had its disadvantages ...

Several factors made this method of doing split-planes obsolete pretty quickly. The first drawback was its reliance on fairly advanced text editing skills. As powerful a skill as it may be, not too many people feel comfortable juggling several gerber files between buffers and cutting and pasting between files.

Secondly, this method was time consuming. Searching for and deleting the corresponding antipads was very tedious and took too long, in spite of the fact that the searching and key-defining features of the text editor could be used to help automate the process.

This method also required the user to manually separate the nets on his split-plane by placing his components so that certain nets fell in certain areas of the board. Lines had to be added that would be photoplotted and would separate the nets when the film was photographically reversed.

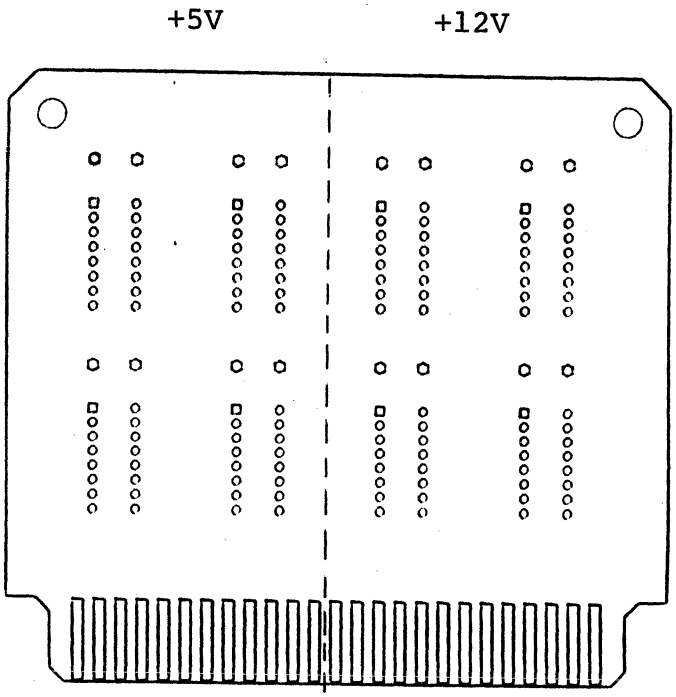
There had to be a better way ...

3. PHOTOGRAPHIC MERGE OF FILMS

3.1 How it worked ...

This method of creating split-planes was based on the Telesis Application note. It involves using ADD RECTANGLE to split the nets on the plane, creating multiple films and photographically reversing and merging all the films.

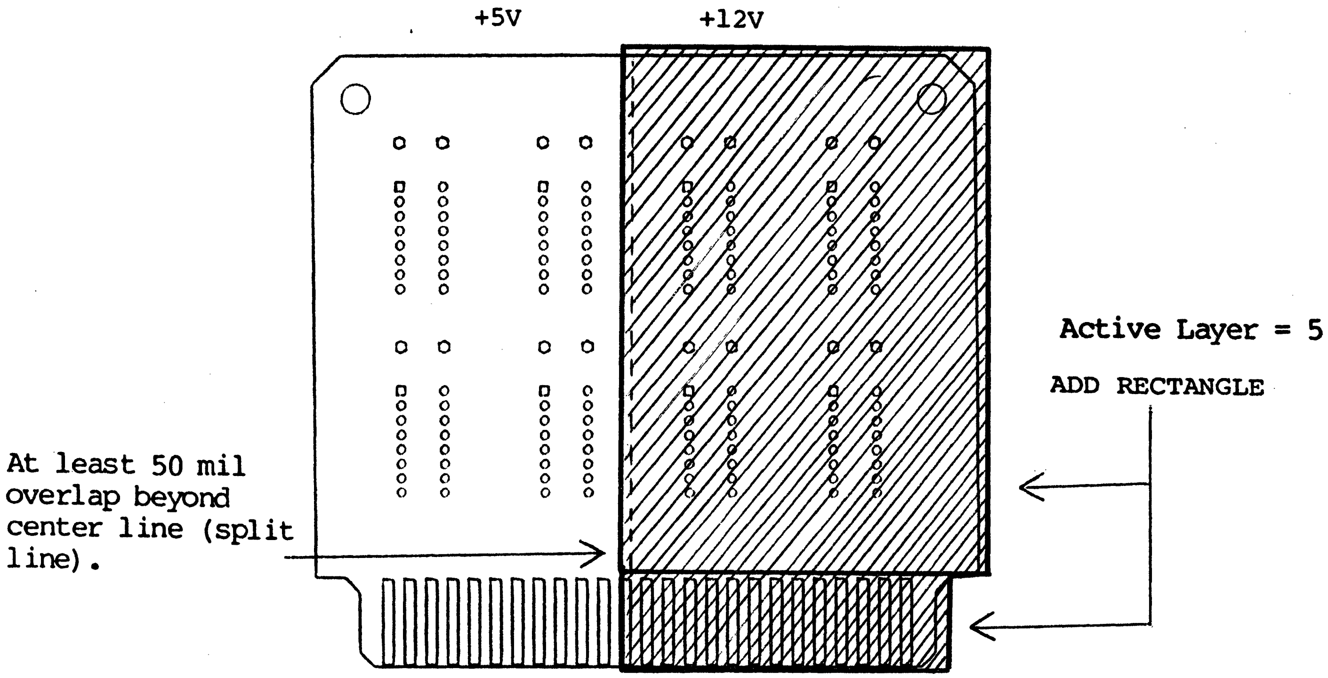
This method starts with the user placing his components so that the pins of each voltage net (or ground net) fall into different regions of the board and are therfore easily separable. (See example below)



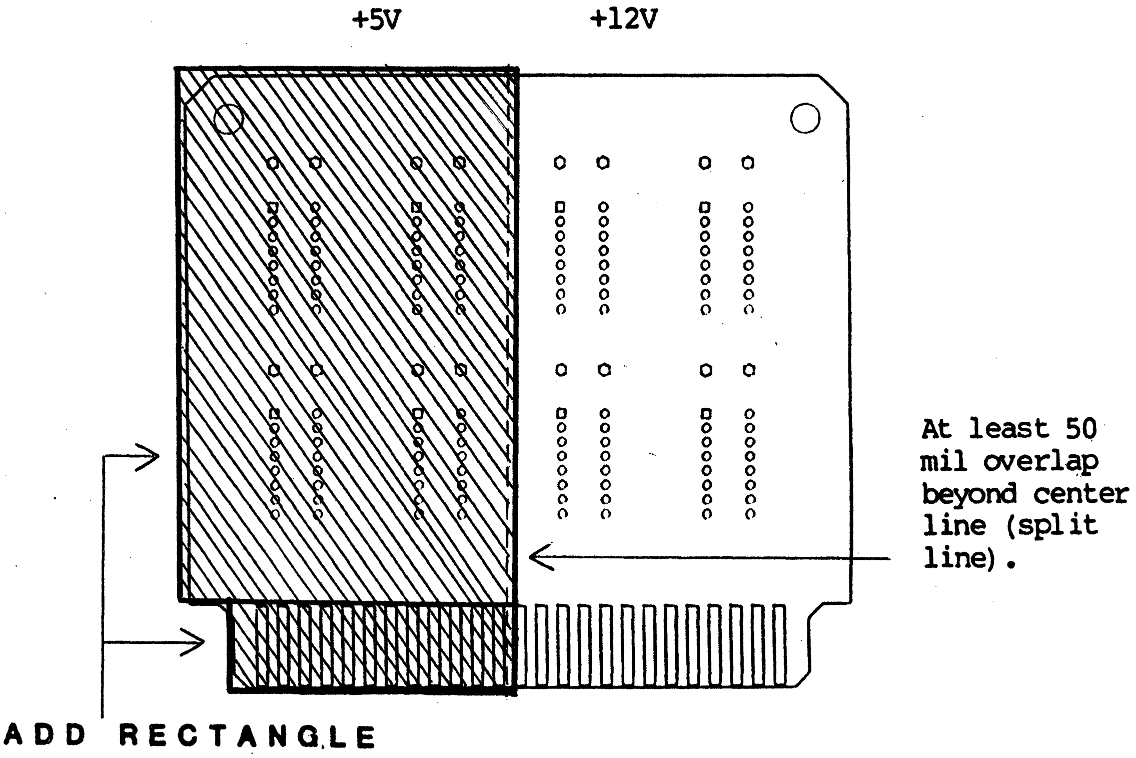
The LAYERSTD was set up with the split-plane nets declared on different layers, as follows:

```
DBLAYER 5   IMBEDDED-PLANE   +5V
DBLAYER 6   IMBEDDED-PLANE   +12V
```

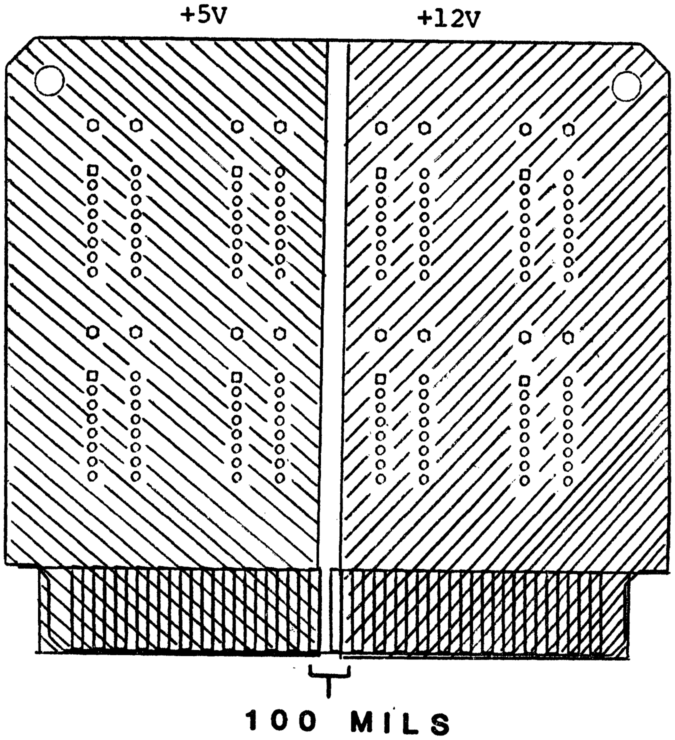
Then, for each net on the split-plane, the user set his active layer to that net's corresponding layer in the LAYERSTD file. Let's use dblayer 5 and net +5V for our example. A rectangle (or several rectangles) was added to cover the entire region of the board, **EXCEPT** the area occupied by the current net (+5V). In this case, a rectangle had to cover the entire area of the +12V side to the board edge. Furthermore, the rectangles for each net had to overlap the center line (between nets) by approximately 50 mils to insure separation of the planes. (see example below)



This was repeated for the +12V net, covering all areas of the board with rectangles on layer 6 except the +12V area. (see example below)



At CREATE PHOTOPLOT time, separate films were created for each of the split-plane layers. The photoplot vendor was instructed to phototgraphically reverse all films and combine them to produce the desired split-plane film. (see example below)



3.2 **Advantages ...**

This method improved upon the previous one because it did not require any text editing at all. The CREATE PHOTOPLOT program and the vendor did most of the work. All the user had to do was add rectangles and run the program. There was less room for error than there was by manually editing and changing gerber files.

3.3 **There were still some drawbacks ...**

While this method was a step in the right direction, there were still a few problems. This was a rather difficult process to visualize. (The user had to cover the entire board with rectangles EXCEPT for the area of the board that the net occupied?)

Then there was the cost. Plotting all those films and all those rectangles could be expensive. Reversing the films and merging them also added to the cost.

Besides that, some people were not satisfied with the quality of photographically merged films.

4. MERGING NETS BEFORE CREATE PHOTOPLOT

4.1 How it worked ...

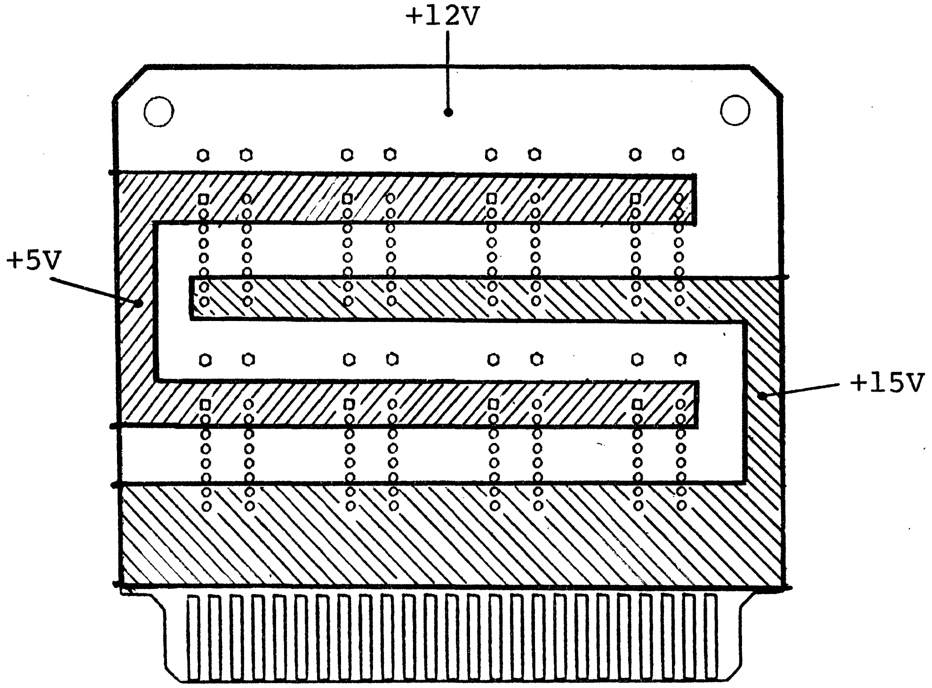
This method was presented at the 1985 Telesis User's Group Meeting. Basically, it involves changing all the nets on the split-plane to one net name. That net name would be the name specified in the LAYERSTD as the imbedded plane. All the pads on all those split-plane nets would then be flashed as thermal-reliefs since they are all on one net now, and that net name matches the name in the LAYERSTD.

Let's walk through an example to illustrate this method. Suppose we wanted to incorporate +5V, +12V and +15V on one imbedded plane.

The net-data-base has been created as usual, either from text netlist or from a schematic. It reflects all three voltages (+5V, +12V and +15V). All of the usual Telesis programs (DRC, netcompare, etc) have been run and its time to run CREATE PHOTOPLOT. So far, nothing unusual.

The next step is to isolate the voltage nets from one another. This is done by adding a 50 mil (the width varies with the design) around each net, isolating it from the other voltages. Highlighting the nets one at a time is helpful in figuring where the lines should go.

These lines should be extended beyond the board's edges to insure that the nets were truly isolated. In this example, these isolation lines divide the plane into three regions and each region contains all the pins from one voltage and no pins from the other voltages. (see example below)



The board drawing should now be saved. At this point, an INCREMENTAL NETLIST file is created to delete two of the three nets and add them back in as the third net. In our example, let's delete +5V and +12V and add them back in as +15V. The INCREMENTAL NETLIST could look like this:

```
( INCREMENTAL NETLIST FILE )
()
( Delete nets +5V and +12V )
$DELETE
$NETS
  +5V
  +12V
$PINS
()
( Add the +5V and +12V pins to the +15V net )
$ADD
$NETS
$PINS
  (+15V) P1.2; P1.1 J[1-4].1                 ( This is +5V ) ,
               P1.[4-6] J[1-4].5 J[1-4].6   ( This is +12V )
$END
```

Now its time to run CREATE PHOTOPLOT. We set up the PHOTOPLOT-CON file so that the -ART file will contain the necessary layers. The layer names are reflected in the LAYERSTD file:

```
( LAYERSTD )
DBLAYER 1   COMPONENT-SIDE
DBLAYER 2   SOLDER-SIDE
.
.
DBLAYER 4   IMBEDDED-PLANE +5V
DBLAYER 5   IMBEDDED-PLANE +12V
DBLAYER 6   IMBEDDED-PLANE +15V
.
.
DBLAYER 33  BOARD-OUTLINE
```

The PHOTOPLOT-CON file might look like this:

```
( PHOTOPLOT-CON )
ARTWORK SPLIT-PLANE
DBLAYER 5
BOARD-OUTLINE
END
```

When CREATE PHOTOPLOT is run, all of the pads on +5V and +12V, which are now on +15V, get flashed as thermal-reliefs. The gerber file called SPLIT-PLANE-ART (in this case) is created and should be penplotted. The resulting penplot should be checked to be sure that the separation lines and flashes appear in the proper regions of the board.

4.2 **Advantages ...**

This method of producing split-plane artwork is superior to the previous methods in several ways. First of all, the Create Photoplot program need only be run once so the user saves some time.

Secondly, no tricky photographic tricks are necessary, because all the proper thermal-reliefs are flashed in one film.

Lastly, any number of nets could be put on one imbedded plane easily as long as the nets could be easily separated by lines.

4.3 **Disadvantages ...**

On the other hand, this method requires that the net-data-base be changed either by INCREMENTAL NETLOAD or by reloading a brand new text netlist (with all voltage nets renamed to one net).

Not only is changing the net-data-base a time consuming program, but many people felt uneasy about deviating from the standard design process.

5. THE SPLIT-PLANE PARAMETER

5.1 How it works ...

Very recently, Telesis enhanced its CREATE PHOTOPLOT program to handle multiple nets on one imbedded plane. The user still needs to isolate his split-plane nets from one another with the use of separation lines, but it is no longer necessary to modify his net-data-base in any way to obtain the correct results from CREATE PHOTOPLOT.

This is how it works. The net-data-base is created as usual (via text netlist or schematic). Then the board is designed as usual up to the point where the user is ready to obtain artwork files.

At this point, the user must divide his imbedded plane into one region for every split-plane net, using the same technique outlined in the previous method (HIGHLIGHT NET, ADD LINE, etc).

Now it is time for the user to indicate to the CREATE PHOTOPLOT program that he wishes:

1. to create a split plane
2. which layer is will reside on
3. which nets will be included in that split plane.

All this information is passed to the CREATE PHOTOPLOT program by means of a new parameter in the PHOTOPLOT-PAR:

```
SPLIT-PLANE
```

The format of this new parameter is as follows:

```
SPLIT-PLANE <layer> <1st net # ... 10th net #>
```

After the keyword "SPLIT-PLANE", the user indicates the layer that the split-plane will be based upon. After the layer number, he may indicate up to ten nets to be included on that split-plane. These nets are indicated by NUMBER (not net name). For example:

```
( PHOTOPLOT-PAR )
COORDINATES ABSOLUTE
FORMAT 5.3
MACHINE-OFFSET 0,0
SPLIT-PLANE  5  38  39  40  41  42  43  44  45  46  47
END
```

This example shows how the user can inform the CREATE PHOTOPLOT program that layer 5 is to be used as a split-plane. Now, if the user includes "DBLAYER 5" in his PHOTOPLOT-CON file, any pins on nets 38, 39 .. 47 will be flashed as a thermal-relief instead of antipad.

Notes on this enhancement:

- "SPLIT-PLANE" is an optional entry in the PHOTOPLOT-PAR file. If the user does not add it, CREATE PHOTOPLOT will behave as usual, allowing only one net per layer (as specified in LAYERSTD).

- The layer numbers (first entry in the SPLIT-PLANE line) are assumed by the program to be valid layer numbers. The program only checks to be sure that the layer entry is an integer.

- Nets can be referenced by number only. Net names cannot be used. The numbers are available from the NETLIST-REPORT. Furthermore, there is no check made on the validity of the net number. The user must be sure that the net number exists. If it does not exist, no error will be reported and no action will be taken.

- A maximum of 10 net numbers can be specified on a "SPLIT-PLANE" entry. Any more than 10 will result in a warning in the PHOTOPLOT-LOG and only the first 10 will be read.

- The user can specify 10 split-plane nets with the SPLIT-PLANE entry and one more in the LAYERSTD bringing the total number of nets allowed on a plane to eleven.

- The order in which the nets are specified is unimportant.

- The "SPLIT-PLANE" entry itself can occur a maximum of 5 times in a -PAR file, specifying a maximum of 5 different layers that can have split-planes on them.

- The order in which these SPLIT-PLANE entries occur in the -PAR file is unimportant. However, if there are more than five SPLIT-PLANE entries in the -PAR file, only the first 5 will be recognized.

- This enhancement does not perform any geographical checking to make sure that all pins of one net are in the same copper area. Furthermore, our PDRC program cannot help detect errors on imbedded planes. It is up to the user to be sure that he doesn't have a +5V pin (for example) in his +12V section of his split-plane.

5.2 **Advantages ...**

This method improves upon all previous methods. The user is no longer required to do any time-consuming modifications to his net-data-base to get all of his split-plane nets merged into one net.

No tricky, costly photographic merges are required.

5.3 **Disadvantages ...**

As I see it, there is one disadvantage inherent in this method. The net must be specified by number instead of by name. This requires the user to have a NETLIST-REPORT in front of him to determine the net number for a net of a given name. This also requires him to change this parameter after certain ECOs, if those ECOs would cause insertion or deletion of nets, thereby changing net numbers. If this was done, the user would need to generate a new NETLIST-REPORT.

Also, the user might expect this type of parameter to be found in the PHOTOPLOT-CON file where layer information is stored instead of the PHOTOPLOT-PAR.

6. FUTURE DIRECTIONS

What kind of improvements could be made in the future?

Well, someday we may be able to provide tools to generate the "separation lines" between the regions of the split-plane that contain different nets.

Someday, we may also be able to develop tools that will flag DRC errors caused by pins located in the wrong region on the split-plane.

A BEGINNER'S GUIDE TO NETWORKING

Christina Runtagh

Product Manager, Systems

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

The term "networking" means different things to different people. The following document will provide a general definition of a network, what its purpose is, and how it can be used in a Telesis environment. It will also cover various networking features, some of the problems that may arise, the networking products currently available from Telesis, and how they can be adapted to your needs.

INTRODUCTION

What is Networking? Networking came into existence to provide a more efficient means of sharing expensive computer resources, common data files, and distributed applications. Engineering, manufacturing, finance and other company departments needed to easily share various pieces of information. This need posed the unique problem of crossing barriers between different computer types, different operating systems, and different file structures. The solution had to overcome time, and physical distance constraints.

To meet these needs, communications networks were devised to provide users with easy access to shared data, while still allowing for file security and information integrity. Briefly, the basic purpose of a network is to provide users on a variety of systems with mutual access to a wide range of devices and to share data among them.

TYPES OF NETWORKING

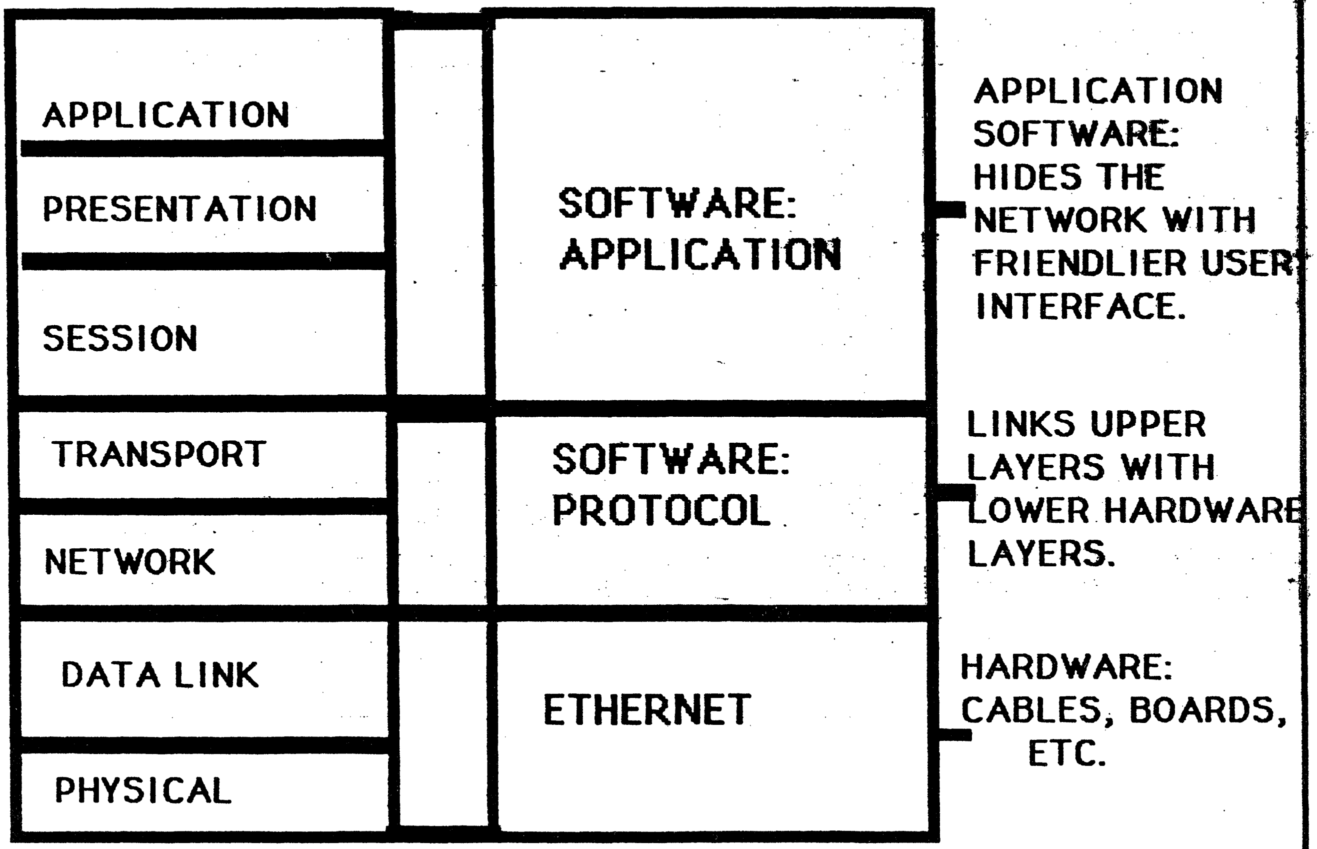
There are several networking types that are becoming industry standards: Ethernet, token-passing ring, and token-passing bus.

The token-bus technique is used mainly to communicate with automated equipment in factory automation applications. Token-ring is best known by its non-announcement by IBM last year. Its characteristics and marketplace are presently unclear. Ethernet is by far the most widely recognized and the one chosen by Telesis. Ethernet allows networking connections to the operating systems of the most popular minicomputers. It is optimized by high speed data exchange within a moderately sized geographical area (< 2.8 kilometers).

HOW DOES NETWORKING OPERATE?

Ethernet represents the physical links between systems; the board cables, transceivers, etc. A network also requires various levels of software interfaces. The networking model is comprised of 7 levels (see Figure 1). While it is not necessary to understand how these levels operate, or interact, they are helpful in understanding the terms frequently used in networking discussions. For example, the protocol, or convention governing the message interchange is an important part of the networking structure (Protocol is levels 3 and 4 of our model).

The protocol used by Telesis products is TCP/IP. It is the latest standard by the Department of Defense. It is important to know the protocol used on a given network as networks using different protocols may be incompatible. For example, if a Telesis network using TCP/IP is connected to an existing network using DEC-NET, the Telesis stations can only communicate with the system that is connected to both stations. (See Figure 2).



APPLICATION SOFTWARE: HIDES THE NETWORK WITH FRIENDLIER USER INTERFACE.

LINKS UPPER LAYERS WITH LOWER HARDWARE LAYERS.

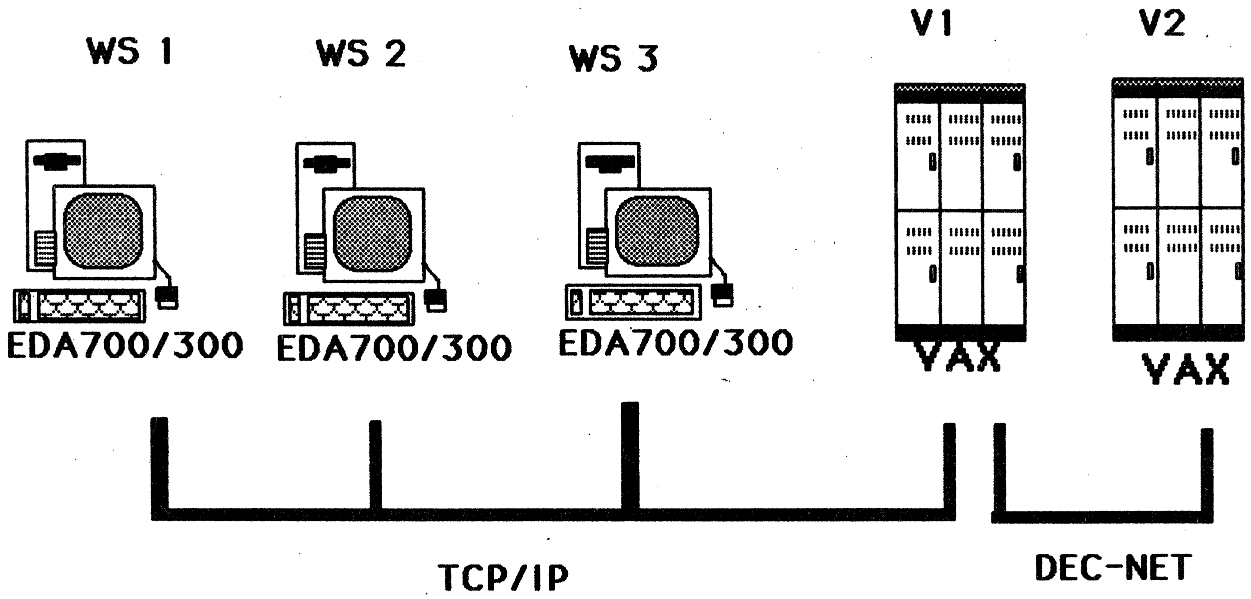
HARDWARE: CABLES, BOARDS, ETC.

FIG 1.

ISO NETWORKING REFERENCE MODEL

NETWORKING

1986 Telesis



WS1-WS3 CAN SHARE DATA WITH V1 AND EACH OTHER.
V1 CAN SHARE DATA WITH WS1-WS3 AND V2.
V2 CAN ONLY SHARE DATA WITH V1.

FIG. 2

NETWORK COMPATIBILITY EXAMPLE

NETWORKING

1986 Telesis

NETWORKING FEATURES

The following network features are provided by some, or all of the Telesis networking products:

1. SHARING PERIPHERALS - One of the major features of a network is that it allows users shared access to expensive peripherals such as printers, plotters, and mag tape drives, that are seldom in constant use on any one engineering workstation. It is far more economical to tie the workstations together via a network, than to purchase additional peripherals for each station.

   Sharing peripherals between "like nodes" or stations that have similar hardware and operating systems presents one set of networking problems. Peripherals in this environment can generally be used without user intervention. The peripherals are transparently accessed over the network, so users need not be concerned with instructing the system to plot on node A, or print on node C.

   Sharing peripherals between "unlike nodes" is a two or three step process. Unlike nodes are represented by an EDA-7ØØ and a VAX-78Ø. Peripherals on the VAX can be indirectly accessed by the user on an EDA-7ØØ by first transferring the file(s) that are to be printed, copied to tape, etc., on to the VAX, and then using standard VAX commands to perform the desired actions.

2. TRANSFERRING DATA - The transfer of data from one station to another is called file transfer. This feature is offered by the Telesis Basic Networking option, and VAX file transfer option. It allows the transfer of all files in any projects, drawings, execute files, etc., from one node on the net to any other node on the net.

3. LIBRARY CONSISTENCY - By selecting one disc to serve as the library storage node, all users can be certain of always having the latest library symbols providing the following conventions are set up:

   - the selected node is known to all

   - library updates are made regularly to the library storage node, and update reports are distributed to all concerned

- prior to working on a job, each library part needed is transferred from the library storage node (with updates made as necessary)

Since libraries tend to be extremely large, using this approach has an added benefit -- not all systems are burdened with the need for the extra disc space consumed by the library. Only those library parts required for the current job need reside on the disc.

The library storage node can be a normal workstation. However, because of the amount of disc space required, plus the amount of transfers that may take place, it is advisable to use a VAX as the library storage node if one is accessible.

4. REMOTE LOG-IN - Remote Log-in is another feature offered when networking to "unlike nodes". This feature allows a user on one system to log in and use the "unlike" system. For example, with remote log-in, a user can utilize the editing terminal on the Telesis workstation as if it were a terminal tied into the VAX. A user can log in, using any allowable VAX account passwords, and execute any legal VAX commands. This feature is particularly helpful when combined with file transfer to/from the VAX. The user can transfer a text file, log in to the VAX, execute the PRINT command on the VAX, delete files, clean up directories, log out again, and be back on the Telesis workstation -- without moving anything but his fingers.

POTENTIAL NETWORKING PROBLEMS

While networking has opened the door to data sharing between various computers and operating systems, it has also created a new set of potential problems:

Networking remote log-in features require that users be knowledgeable in the language of the computer being addressed (in this case, VMS). The user must also be proficient in operating all the programs chosen for use on the other system (i.e. PRINT, ARCHIVE, etc.). Thus, competence in a wide range of operating systems may become a necessity.

All operating systems have their peculiarities. One of the most obvious of these is file management. Creating directories, changing directories, listing, and deleting files are all basics that should be learned. Unfortunately, they generally differ from system to system.

Naming conventions must also be taken into consideration. The Telesis workstation allows 18 character names, with 4 character revisions. The allowable character set has an abundance of special characters. Directories are used to separate file categories. VMS has a 32 character limit, and a very restricted character set. MS/DOS (the operating system of IBM/PCs) has an 8 character file name limit. Both VMS and MS/DOS have a numeric revision field that is not user-controllable. Because of the wide diversity of the naming conventions, it's best to be aware of them prior to naming a file that may eventually be transferred to a different system. This will eliminate frustration later, and the time consuming process of renaming the files.

DIFFERENT OPERATING SYSTEMS

FILE PROTECTION - Providing users with access to any file on a network, and the ability to transfer those files has generated a variety of file protection problems. One of the most common problems is work loss resulting from two people accessing and modifying the same file. For example, if user A and user B are working on Drawing X simultaneously, and both transfer it to the library storage node, the user completing the file transfer process first will lose his work. The file will be overwritten when the second file is transferred back. To alleviate this problem, file access should be restricted to one person only at any given time. A simple, manual check-in, check-out procedure can be easily implemented. If you are using a VAX for your storage node, the account/password protection package can also help to prevent this problem.

NETWORK COMPATIBILITY - As outlined earlier, not all networks are created equal. Simply having an existing network in house, does not guarantee that any other network can be included, or that any networking software package will work. Network features and compatibility depend largely on the hardware, protocol used, and the type of application software involved. For example, you cannot add a DEC-NET package to an existing TCP/IP network.

The concept of a gateway was formed to provide access to other networks. Gateways can be thought of as a check point between two otherwise incompatible networks. A gateway requires that both networking protocols and the appropriate hardware exist on the same node. The protocol must be intelligent enough to identify only those information packets that pertain to it. Additionally, it must not discard those packets that are not in its format, or worse yet, crash the system.

CURRENT TELESIS NETWORKING PRODUCTS

Telesis currently offers networking products that allow peripheral sharing. Project, drawing, and file transfer to Telesis workstations as well as the VAX is also allowed (See FIGURE 3).

REMOTE PERIPHERAL OPERATION (RPO)

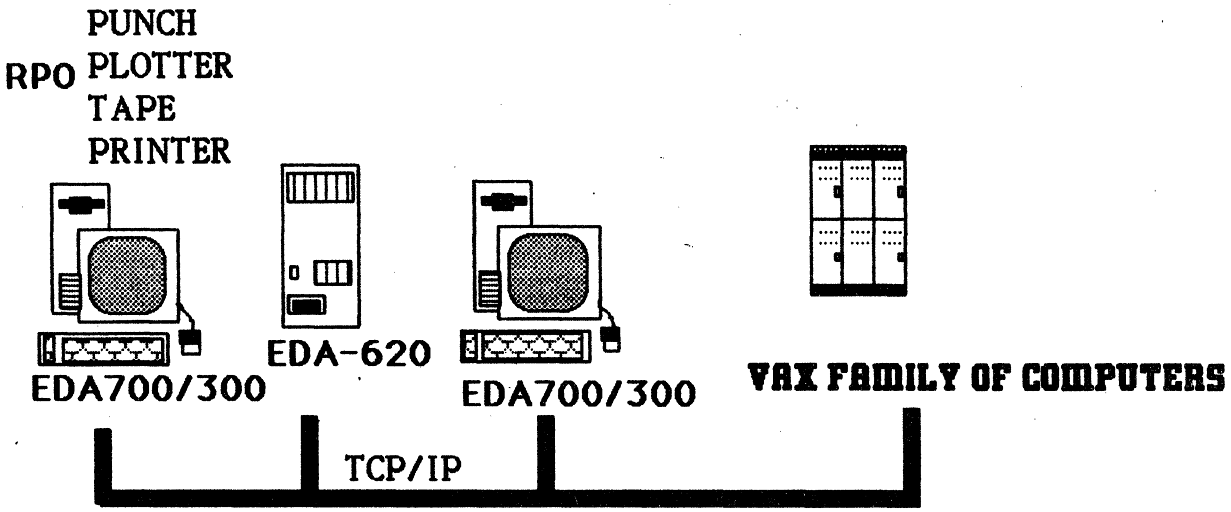
This option provides an economical method of sharing a common set of peripherals among all Telesis EDA-3ØØ and EDA-7ØØ workstations on a network. The peripherals that are to be shared across the network must be physically connected to one workstation (known as the HOST system). The other systems, called REMOTE systems, use the Host system's peripherals transparently over the network. Remote systems may have their own local peripherals, however, local peripherals can only be accessed by the station that they are physically connected to. The actual throughput when using RPO is identical to running on a local peripheral. That is, a plot that takes three minutes to run on a local plotter, will also take three minutes over the network.

The following peripherals can be shared over the network:

- magnetic tape unit
- pen plotter
- NC punch
- matrix printer

Some peripherals cannot be shared over the network and therefore are connected as local peripherals:

- Text Editing Terminal
- Digitizer
- Floppy Disc
- Hard Disc
- Modem



RPO PUNCH
PLOTTER
TAPE
PRINTER

EDA700/300

EDA-620

EDA700/300

VAX FAMILY OF COMPUTERS

TCP/IP

T-T FILE TRANSFER:
PROJECTS
DRAWINGS
TEXT-FILES

VAX FILE TRANSFER

FIG. 3

TELESIS COMMUNICATIONS PRODUCTS

NETWORKING PRODUCTS	
1986	Telesis

RPO does not allow the direct sharing of peripherals on the VAX. Some VAX peripherals can be shared indirectly, by first using the VAX file transfer option to transfer the desired files and then directly accessing them from the VAX.

T --> T FILE TRANSFER

This option allows the transfer of projects, netlists, schematics, PCB, and all other Telesis files from one Telesis workstation to another. By picking the appropriate menu boxes, the user creates a list of the files to be transferred between any two systems on the network. Other menu boxes are used to perform the actual transfer.

T ---> EDA-620

The use of the Co-Router requires that information concerning the board to be routed be passed to the EDA-620. The resulting routing information must also be passed back from the EDA-620 to the Telesis system so that the drawing can be updated to reflect its routed state. Both of these networking operations are handled transparently when the "CO-ROUTE BOARD" and "UPDATE DRAWING" menu boxes are picked.

Status information about the various co-routes that may be running on an EDA-620, is also retrieved via the network.

VAX FILE TRANSFER

The latest Telesis networking option provides the capability to transfer Telesis projects, schematics, netlists, PCB drawings, etc., between a Telesis EDA-300 or EDA-700 workstation and a VAX. This allows users to take advantage of the many features offered by the VAX. Projects, or specific files may be controlled by the VAX password protection/access package. The large disc capacity of the VAX, plus any existing VAX printers and mag tapes may be utilized, as well as the use of existing file management and archiving procedures that are already set up for the VAX environment. And, as mentioned earlier, library consistency and gateways to other networks are also possible.

COMMON USER PROBLEMS

The following outlines some of the most common problems or misunderstandings currently encountered with Telesis networking products.

STATIC - Static can present a major problem for networking environments, as well as for all workstations involved. This is especially true in areas where carpeting is used. Static usually manifests itself as an audible spark that shoots from the operator to the workstation on a network. The static charge generally produces a "DEVICE ERROR" message on the workstation that is performing some network function.

One way to alleviate the static problem is by installing grounding straps to the workstation chassis. Operators should be advised to touch the strap prior to beginning work on the station.

Another alternative is to use non-static carpeting wherever necessary. The least expensive, but still effective solution is to use one of many anti-static sprays available on the market.

MULTIPLE PERIPHERALS - At some sites, a need exists for more than one printer or plotter to service the number of workstations on the network. There are two ways to solve this problem: Since each workstation can have a peripheral attached directly to it, additional printers, plotters, etc., can be added as local peripherals to a node on the network. This approach still allows file transfer among all the nodes on the network, and peripheral sharing of all peripherals attached to the HOST by nodes not having local peripherals. The limitation is that only the workstation that has the local peripherals can access those peripherals.

Another alternative is to break the network down into several separate networks, each with their own HOST with their own peripherals attached. The advantage to this approach is that many stations now have access to the peripherals. However, file transfer can only be accomplished between workstations existing on the same net. File transfer between non-networked systems must be performed by archiving to tape or floppy.

SHARABLE PERIPHERALS - A common question asked in the Telesis Customer Support Center is, "How do I get my VT1ØØ on the net?". The response is that you cannot. As described earlier in the RPO feature section, only the following peripherals can be shared over the network: magnetic tape, printer, plotter, and NC punch. Only one of each type of these peripherals can be attached to the HOST and thus accessed by the REMOTE stations on the network.

The following peripherals cannot be shared: editing terminal (VT1ØØ or equivalent), floppy disc, hard disc, digitizer, and modem.

FILE TRANSER VS FILE SERVER - Many networking users are confused about the difference between the file transfer feature and a file server. They are two very different products providing very different functions. **File transfer** requires the user to specify the location of the file to be moved as well as the node the file is being transferred to. A **file server** automatically finds a specified file on any node on the network and allows access of the file data without physically copying it onto the disc where the request was made. The basic difference is that file transfer requires the user to be explicit in naming and locating files, and the files must exist on the workstation that needs access to them. File servers can automatically locate files on any node of the network and allow access of the file data across the network.

DATA COMPATIBILITY - Some users believe that once a file has been transferred to another system, it can be accessed as any other file on the system. This is both true and false. When dealing with "like" systems, such as the EDA-3ØØ and EDA-7ØØ, all files can be accessed in the identical manner on either system. However, when dealing with "unlike" systems, this is only partially true.

For example, once a file has been moved from an EDA-7ØØ to a VAX, that file can be treated as any other file when using file management functions such as COPY, RENAME, DELETE and ARCHIVE. If the file is a standard ASCII text file (i.e. log files, pin files, ASCII data-base out files, etc.) it may be printed as any other VAX text files. Drawing, symbol and net-data-base files cannot be printed or plotted using standard VAX commands. These files are called BINARY files and their contents are written in a special format understood only by Telesis commands.

If one of the purposes of using VAX file transfer is to access drawing information for input into third party packages on the VAX, a translation or drawing conversion may be necessary. The ASCII DATABASE OUT option is intended for just such use.

FUTURE NETWORKING ENHANCEMENTS

The following is a list of features recognized by Telesis as being beneficial to our customers. While they are not currently scheduled for a future release, they are under investigation:

- Allow networking connection to IBM-PCs
- Allow VAX/UNIX connection
- Increase networking capabilities to the EDA-62Ø.

CONCLUSION

Networking provides a means for sharing data and expensive peripherals between many users on different workstations. There is a range of networking products that provide these features in varying degrees, with differing restrictions and limitations. Prior to investing in any networking products, the desired goals should be analyzed to ensure that the final product will best meet the needs of the user.

FILE MANAGEMENT USING WILD CARD

AND ALL FILE OPTIONS

Debra Carrier
Application Instructor
Telesis Systems Corp
Two Omni Way
Chelmsford, MA 01824

FILE MANAGEMENT USING WILD CARD

AND ALL FILES OPTIONS

When using File Management on the Telesis system, you can copy or delete drawings, text files, symbols etc., one at a time. This can be tedious if you have many files to manipulate. With the release of EDA 3000 software, you can now use the ALL FILES or WILD CARD options to access a selected group of files.

To use the ALL FILES option within File Management, simply pick ENTER when the system prompts you for the file name after you have selected a copy or delete command. The system will list all the files of the type selected. You then have the option of invoking or cancelling the command. For example, if you wanted to copy all symbols from your current project to another project, you would use the following sequence:

1. OLD PROJECT NAME PROJECT1

2. FILE MANAGEMENT

3. COPY SYMBOL FROM <ENTER> TO PROJECT2, a list would then appear showing all symbols within your project that will be copied to PROJECT2 along with a message that says:

   "PICK PAGE -> TO REVIEW LIST. PICK CANCEL TO BYPASS LIST".

   Picking cancel at this point doesn′t cancel you from the command; it lets you bypass the additional pages of the copy list.

4. PAGE -> (or CANCEL) Another message comes up and asks you:

   "Are you sure you want to copy all of the listed items?
   Enter CANCEL for NO
   Enter PAGE -> for YES"

5. PAGE -> (or CANCEL) If you pick CANCEL it will cancel the command. If you pick PAGE -> it will copy those files listed into PROJECT2. (It will also create a project called PROJECT2 if none exists on the system.)

The WILD CARD option within File Management allows you to be more selective about which files will be copied or deleted. By using a question mark (?) in place of either the file name or the revision label, you can key off of files with the same name or rev. In order to maximize the use of this capability, you would give certain types of files the same revision label when you create them. With text files for example, you could give device files a revision of DF, pin files a revision of PIN and so on. As with ALL FILES, once the command to copy or delete is picked, the system will show you a list, and will give you the opportunity to cancel out of the command. For example, if you wanted to copy all of your device files at rev DF from your current project into another project, you would use the following sequence:

1. OLD PROJECT NAME PROJECT1

2. FILE MANAGEMENT

3. COPY TEXT FILE FROM ? DF TO PROJECT2
   The copy list appears along with the message:

   "PICK PAGE -> TO REVIEW LIST. PICK CANCEL TO BYPASS LIST"

4. PAGE -> (or CANCEL)

   "Are you sure you want to copy all of the listed items?
   Enter CANCEL for NO
   Enter PAGE -> for YES"

5. PAGE -> (or CANCEL)

   PAGE -> starts copying, CANCEL cancels the command.

You could also use WILD CARDS if you wanted to delete all revisions of your board drawing from your current project. (You wouldn´t use ALL FILES as that would delete all drawing files, including any schematic drawings and symbol drawings.) Do the following:

1. OLD PROJECT NAME PROJECT1

2. FILE MANAGEMENT

3. DELETE DRAWING FROM PCB ?
   A list would appear showing all drawings with a file name of PCB (and associated NET-DATA-BASE files since this is a board drawing) to be deleted along with the message:

   "PICK PAGE -> TO REVIEW LIST. PICK CANCEL TO BYPASS LIST"

4. PAGE -> (or CANCEL)

   "Are you sure that you want to delete all of the listed items?
   Enter CANCEL for NO
   Enter PAGE -> for YES"

5. PAGE -> (or CANCEL)

   PAGE -> starts deleting, CANCEL cancels the command.

NOTES

1. The ALL FILES option works on all menu commands on the File Management menu page except COPY NET-DATA-BASE and DELETE NET-DATA-BASE. If you pick <ENTER> for either of these commands, the system searches for the last NET-DATA-BASE listed on the current index and copies or deletes only that one file.

2. The ALL FILES and WILD CARD options only work with files. You cannot copy or delete projects using them.

3. The ALL FILES and WILD CARD options also work in the same way when archiving to floppy disk or mag tape. Some examples of the format would be:

   ```
   COPY SYMBOL  FROM  <ENTER>  TO F:
   ```
   Copies all symbols from current project to floppy.

   ```
   COPY TEXT FILE  FROM  ? DF  TO F:
   ```
   Copies all text files at rev DF in current project to floppy.

   ```
   DELETE DRAWING  FROM  F: PCB ?
   ```
   Deletes all revisions of drawing PCB from the floppy.

   ```
   COPY SYMBOL  FROM  F: PROJECT,  TO PROJECT
   ```
   Copies all symbols in specified project on floppy to specified project on disk.

4. The ALL FILES and WILD CARD options work at both system level or project level. Specify the FROM project name when working at system level. Example:

   ```
   COPY SYMBOL  FROM  PROJECT1,? 3  TO PROJECT2,

   DELETE SYMBOL  FROM  PROJECT1,
   ```

ARCHIVAL PROCEDURE

USED BY A FELLOW

TELESIS USER COMPANY

ARN D. BUCK

TRANSCOM ELECTRONICS, INC.

PORTSMOUTH, RHODE ISLAND 02871

The following pages are copies of the Transcom Engineering Computer Systems Archival Procedure, the associated forms, and log book instructional sheets.

ENGINEERING COMPUTER SYSTEMS

ARCHIVAL PROCEDURE

	APPROVED	DATE
ENGR	AG	6-18-85
PROD DES		6-18-85
MANAGER	CTI	6/18/85

ECO/REVISION HISTORY

REV	ECO #	SECTION REVISED	DATE

TABLE OF CONTENTS

1.0 SCOPE

This document defines a process which ensures that data files which are stored on magnetic media will be available when needed for back up or modification.

2.0 RESPONSIBILITY

The effectiveness of this data back up system relies on the support of all users at all levels. However, the main responsibility is assigned to the Archive Controller. The duties of the archive controller include:

1. Controlling the storage of all "off system" archival files which are created and used by the engineering department.

2. Ensuring that all log book entries are complete and accurate.

3. Ensuring that labels on all archival files associated with this back up system are filled out and attached properly.

4. Maintaining and distributing indexes of all existing data files that are contained within this back up system.

3.0 PERTINENT FORMS

Several forms have been created for use in conjunction with this procedure. They are as follows:

DESCRIPTION	TRANSCOM PART NUMBER
Project File Log Sheet	107724
Back Up File Log Sheet	107722
Vendor Software File Log Sheet	107720
Project File Label	107725
Back Up File Label	107723
Vendor Software File Label	107721
Magnetic Media Control Sheet	107726

4.0 SYSTEM IDENTIFIER PREFIX

A two digit code is to be used to identify the various computer systems that are used within the Engineering Department. The appropriate code is to be used as a prefix for archival file numbers. These prefixes are listed below.

PREFIX	COMPUTER OR SYSTEM
AP	APPLE (general)
A+	APPLE II+
Ae	APPLE IIe
DR	DEC RAINBOW
DV	DEC VAX
DY	DAISY CAE SYSTEM
IA	IBM AT
IP	IBM PC (general)
IX	IBM XT
MD	TEKTRONIXS MICRO DEVELOPMENT SYSTEM
RA	ROBO GRAPHICS (Apple IIe)
TP	TELESIS (PCB application)
UC	UNICAD CAE SYSTEM

5.0 CLASSIFICATION OF FILES

Magnetic media files to which this procedure is applicable are to be classified under one of the following categories.

1. Project files.
2. Back up files.
3. Vendor supplied software files.
4. Library files.

6.0 PROJECT FILES

6.1 DESCRIPTION

A project file is defined as a copy of a completed project that has been written onto magnetic media (other than a system's primary hard disk).

A completed project is one that has been created at, updated to or released at a specific revision level.

6.2 PURPOSE

A project file serves as a functional, off system record making the data associated with a project available for future review or revision. It also serves as protection against the loss of a project due to a natural catastrophe or accidental (or malicious) deletion of the file from a system.

6.3 FILE DEFINITION

A project file is to contain all the files which are required to successfully complete any future revisions.

6.4 ORIGINATION

Once a given project has been updated to or released at a specific revision level a project file is to be created. Two copies are to be made.

6.5 FILE NUMBER ASSIGNMENT

The file number is to be assigned by taking the next available number from the appropriate project file log book. The number is to consist of a two letter system identifier prefix followed by a four digit number.

6.6 FILE NAME ASSIGNMENT

The project name should comply with the naming convention as specified by the appropriate system manager.

6.7 LABELS

A label will be attached to all project files. Each label is to contain the following information:

1. The project name
2. The file number
3. The project revision level

6.8 STORAGE

One copy of the project file is to be kept in Document Control. The other is to be kept at a remote storage facility.

6.9 INDEXING

The archive controller is to maintain an index of all existing project files. This index is to be kept as a computer file and made accessible to all engineering personnel for reference. Printed copies of this index will be made available if requested.

Within the index there are two listings. One listing is to be sorted alpha-numerically by project name. The other listing is to be sorted by file number.

Each listing is to contain the following information:

1. The project name
2. The file number
3. The project revision level
4. The file location (local or remote)
5. A brief description of the file

6.10 MEDIA RECYCLING

Project files will not be retained after the project has been superseded by two revision levels. Exceptions can be specified by the appropriate system manager. The magnetic media on which the file has been copied will be recycled by removing the attached label and crossing out the appropriate log book entry with a single line.

7.0 BACK UP FILES

7.1 DESCRIPTION

A back up file is a current copy of an active project.

7.2 PURPOSE

The purpose of a back up file is to protect against the loss of data that is associated with an active project due to system hardware malfunctions, natural catastrophe, or the accidental (or malicious) deletion of the file from a system.

7.3 ORIGINATION AND USAGE

A back up file is to be created after approximately eight hours of work on a given system. The title of the file is to consist of the project name with the appropriate system identifier prefix followed by the sequence number as taken from the Back Up File Log Book. A dash is to be placed between the system identifier prefix and the project name, as well as between the project name and sequence number.

At no time should more than two back up copies of a given project exist simultaneously (with the exception of "off site" copies). All but the two most recent copies are to be recycled. This is accomplished by removing the label from the magnetic media and crossing out the entry from the appropriate Back Up Log Book with a single line.

System managers may instruct users to perform daily back up by transferring their files to another computer system. If this procedure is used, it is the responsibility of both system managers to control such files.

7.4 LABELS

A label will be attached to all back up files. Each label is to contain the following information:

1. The file name
2. The project revision level
3. The date that the file was sent to the remote storage facility.

7.5 STORAGE

The location of "on site" back up files is to be designated by the appropriate system manager. Weekly, each system manager is to ensure that the oldest "on site" back up file for any project file worked on that week is given to Document Control. The archive controller is to ensure that back up files are sent to and returned from the remote storage facility at the appropriate time.

7.6 INDEXING

The archive controller is to maintain an index of all existing back up files. This index is to be kept as a computer file and made accessible to all engineering personnel for reference. Printed copies of this index will be made available if requested.

The index is to contain the following information:

1. The back up file name (18 characters max.).
2. The project revision level.
3. The date that the file was sent to the remote storage facility (if sent).
4. The current location of the file (local or remote).
5. A brief description of the file.

7.7 MEDIA RECYCLING

Back up files are to be recycled as soon as two project files have been created for a given project. They are also to be recycled upon returning from the remote storage facility if a more current copy exists. The media used will be recycled by removing the label and crossing out the log book entry with a single line.

8.0 VENDOR SUPPLIED SOFTWARE FILES

8.1 DESCRIPTION

A vendor supplied software file is a copy of software that has been provided by an outside vendor.

8.2 PURPOSE

The purpose of these files is to ensure the availability of vendor supplied software for future use.

8.3 ORIGINATION

These files are created by outside vendors. Typically such software is copy protected to prevent the user from generating unauthorized copies. If copies can be made, the number to be made is to be specified by the appropriate system manager.

8.4 FILE NUMBER ASSIGNMENT

The file number is to consist of the letter "V" as a prefix preceding the appropriate two letter system identification code. This is to be followed by a four digit number which is obtained by using the next available number in the appropriate vendor software log book.

8.5 FILE NAME ASSIGNMENT

The file name is to be specified by the proper system manager.

8.6 LABELS

A label will be attached to all vendor software files. Each label will contain the following information:

1. The file name
2. A brief description of the software.

8.7 STORAGE

A storage area will be provided in Document Control and at the remote storage facility. The utilization of these storage locations is at the discretion of the system manager.

8.8 INDEXING

The archive controller is to maintain an index of all existing vendor software files. This index is to be kept as a computer file and made accessible to all engineering personnel for reference. Hard copies of of this index are to be made available if requested.

The index is to include the following information:

1. The software category (Engineering, Communications, System Utilities, Languages, Math, Business, Word Processing, CAE, CAD, Spread Sheets)
2. The file name
3. The name of the vendor
4. The publication date
5. Date registration card sent in (if applicable)
6. The revision level
7. The type of computer required
8. The operating system
9. A description of the function.

8.9 MEDIA RECYCLING

The magnetic media used for vendor supplied software may be recycled by removing the attached labels and crossing out the appropriate log book entry. This can only be done with the approval of the appropriate system manager.

9.0 LIBRARY FILES

9.1 DESCRIPTION

A library file consists of elements such as symbol files, text files, or macros which are either advantageous or necessary to keep separate from project related files.

9.2 USAGE

The appropriate system manager has the option of using either the project or back up file format for their system libraries.

The project file format is recommended if the library is defined and revision controlled.

The back up file format is recommended if the library is modified frequently as symbols are added, deleted and modified.

10.0 LOG BOOKS

Log books will be used to enable the archive controller to keep track of the archival files which are generated by the various computer systems within the engineering department.

The locations of the various log books are to be determined by the appropriate system managers. A master copy of all logs will be maintained in Document Control by the archive controller.

Each log book is to contain the following:

1. A title sheet.
2. A copy of the Engineering Computer Systems Archival Procedure (drawing number GEN002.PRC).
3. A divider labeled "project files".
4. Instructions for project file labels.
5. Instructions for project log sheet entries.
6. Project file log sheets.
7. A divider labeled "back up files".
8. Instructions for back up file labels.
9. Instructions for back up log sheet entries.
10. Back up file log sheets.
11. A divider labeled "vendor software files".
12. Instructions for vendor software file labels.
13. Instructions for vendor software log entries.
14. Vendor software log sheets.

11.0 CONTROL OF ARCHIVE FILES TO AND FROM REMOTE FACILITY

A Magnetic Media Control Sheet is to be used to record all transactions between the Engineering Department and any remote storage facility. The purpose of this form is to identify the archive files which are being shipped to a remote site and to identify the files which are to be returned.

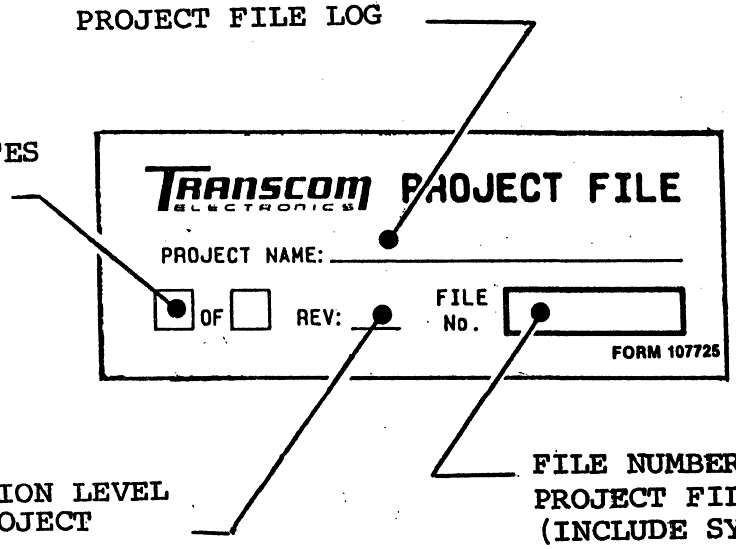
A copy of the form is to accompany all shipments of archive files . A duplicate copy is to be kept in a file maintained by the archive controller.

7/31/85

PROJECT FILE LABELS

MUST MATCH NAME IN
PROJECT FILE LOG

NUMBER OF DISKETTES
OR TAPES IN SET



TRANSCOM ELECTRONICS PROJECT FILE

PROJECT NAME: ______________________

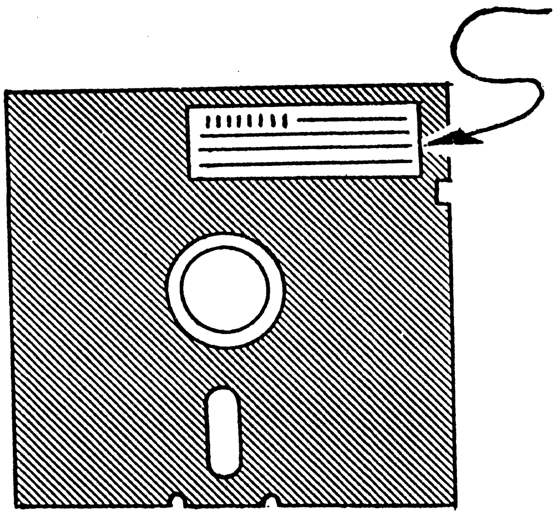
□ OF □ REV: ____ FILE No. □

FORM 107725

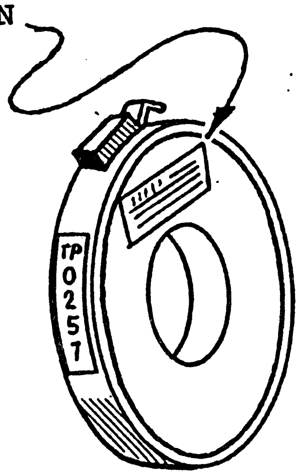
REVISION LEVEL
OF PROJECT

FILE NUMBER TAKEN FROM
PROJECT FILE LOG BOOK
(INCLUDE SYSTEM IDENTIFIER PREFIX)

LABEL LOCATION



FLOPPY DISKETTE



MAGNETIC TAPE

7-31-85

PROJECT FILE LOG BOOK ENTRIES

(SEE SECTION 6.0 OF THE ARCHIVAL PROCEDURE FOR FURTHER DETAILS)

FILE NUMBER: Enter the next number counting consecutively from the previous entry. Make sure that the "System Identifier Prefix" is included.

EXAMPLE:

$$\overbrace{\textbf{DY}}^{\text{SYSTEM IDENTIFIER PREFIX}}\overbrace{\textbf{0157}}^{\text{NUMBER FROM LOG BOOK}}$$

PROJECT NAME: Enter the name that is used to identify the project on the system used. A maximum of 18 characters may be used.

REV: Enter the current project revision level. A maximum of 2 characters may be used.

DESCRIPTION: Enter a brief description of the file. A maximum of 30 characters may be used.

MEDIA: Check the appropriate column to indicate the type media used (floppy diskette or magnetic tape).

SIZE: Enter the size of the media used. If floppy diskettes are used, specify the diameter. If magnetic tape is used, specify the length in feet.

DATE: Enter the date that the file was created.

THE FOLLOWING ENTRIES ARE TO BE COMPLETED BY THE ARCHIVE CONTROLLER ONLY

TO REMOTE: Enter the date that the project file was sent to the remote storage facility.

FROM REMOTE: Enter the date that the project file was returned from the remote storage facility.

FILE DELETED: Enter the date that the media was recycled.

INDEX (ADD): Check this column to indicate that the file has been added to the project file index.

INDEX (DELETE): Check this column to indicate that the file has been deleted from the project file index.

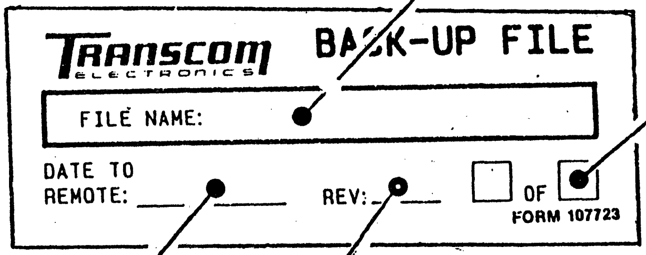
TRANSCOM ELECTRONICS

PROJECT FILE LOG

								ARCHIVE CONTROL USE ONLY				
FILE NUMBER	PROJECT NAME -18 CHARACTERS MAX-	REV	DESCRIPTION -30 CHARACTERS MAX-	MEDIA		SIZE	DATE	TO REMOTE	TO REMOTE	FILE DELETED	INDEX	
				DISK	TAPE						ADD	DEL

BACK UP FILE LABELS

MUST MATCH NAME IN
BACK-UP FILE LOG
INCLUDING SYSTEM IDENTIFIER
PREFIX AND SEQUENCE
NUMBER



TRANSCOM ELECTRONICS BACK-UP FILE

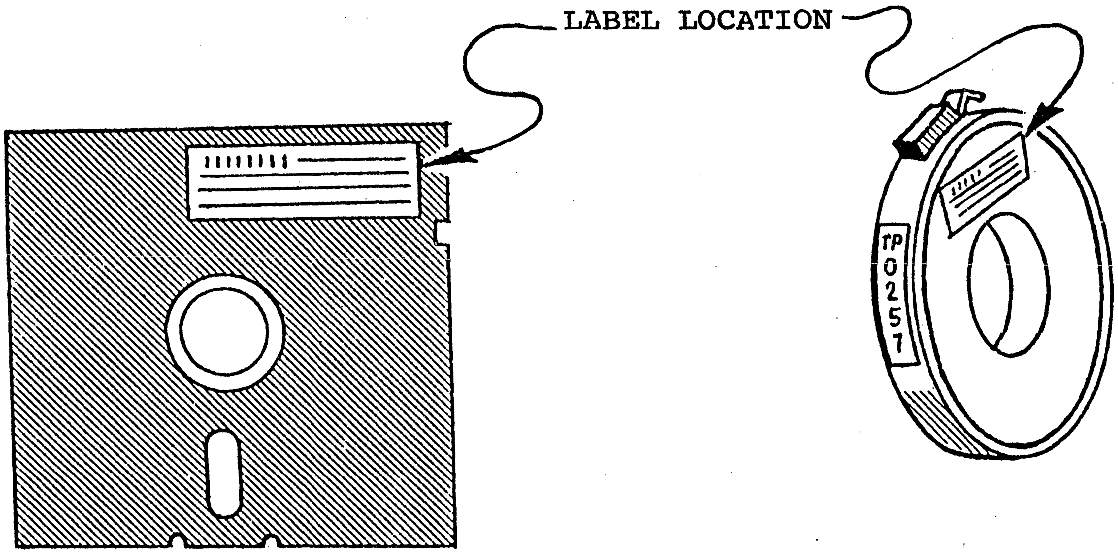
FILE NAME: ●

DATE TO
REMOTE: ___●___ REV:_○_ □ OF ● FORM 107723

NUMBER OF DISKETTES
OR TAPES IN SET

TO BE COMPLETED
BY CONTROLLER

PROJECT REVISION LEVEL



FLOPPY DISKETTE

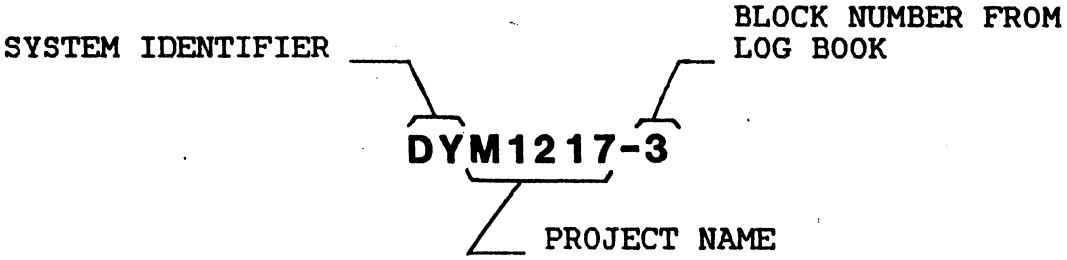
MAGNETIC TAPE

BACK UP FILE LOG BOOK ENTRIES

(SEE SECTION 7.0 OF THE ARCHIVAL PROCEDURE FOR FURTHER DETAILS)

PROJECT NAME: Enter the name that is used to identify the project on the system used. Include the "System Identifier Prefix" and the "Block Number" from the back up file log book. A maximum of 18 characters may be used.

EXAMPLE:



REV: Enter the current project revision level. A maximumof to characters may be used.

BLOCKS 1 THRU 28: Enter the date that the back up file was created in the appropriate block. Begin with block 1 and use each block consecutively up to block 28. Start a new section if additional blocks are required.

ARCHIVE CONTROLLER MARKINGS: The archive controller shall keep track of which file have been sent to the remote storage facility by placing an "X" in the box following the date created block. Recycled media will be identified by drawing a single line through the appropriate date created block.

TRANSCOM ELECTRONICS

BACK-UP FILE LOG

PROJECT NAME:	1		5		9		13		17		21		25	
	2		6		10		14		18		22		26	
	3		7		11		15		19		23		27	
REV:	4		8		12		16		20		24		28	

PROJECT NAME:	1		5		9		13		17		21		25	
	2		6		10		14		18		22		26	
	3		7		11		15		19		23		27	
REV:	4		8		12		16		20		24		28	

PROJECT NAME:	1		5		9		13		17		21		25	
	2		6		10		14		18		22		26	
	3		7		11		15		19		23		27	
REV:	4		8		12		16		20		24		28	

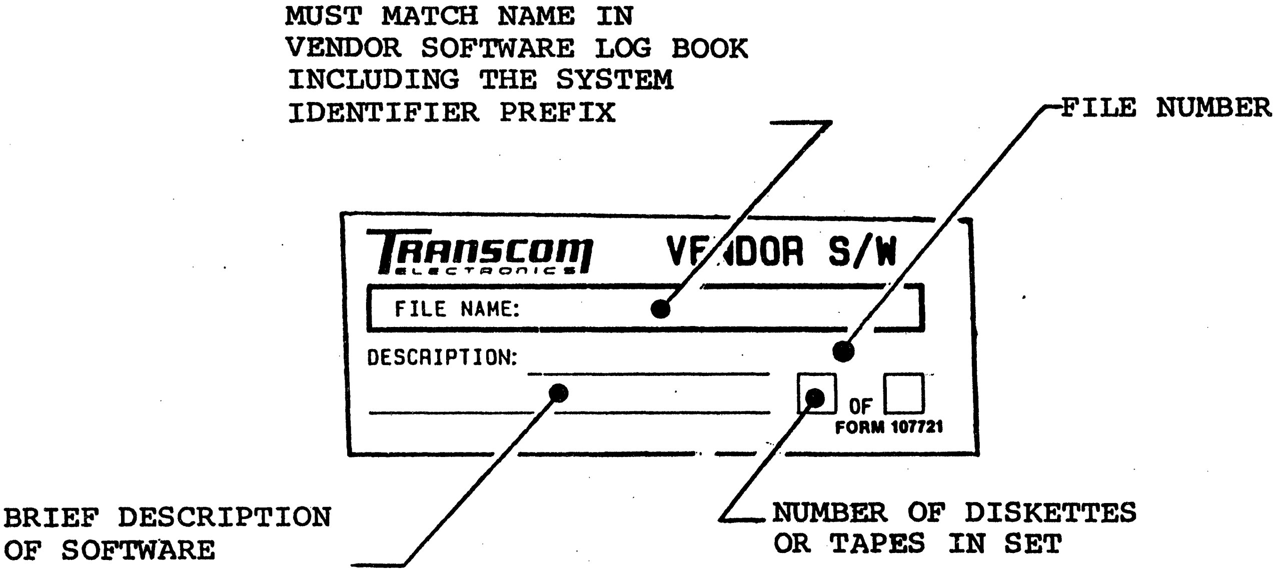
PROJECT NAME:	1		5		9		13		17		21		25	
	2		6		10		14		18		22		26	
	3		7		11		15		19		23		27	
REV:	4		8		12		16		20		24		28	

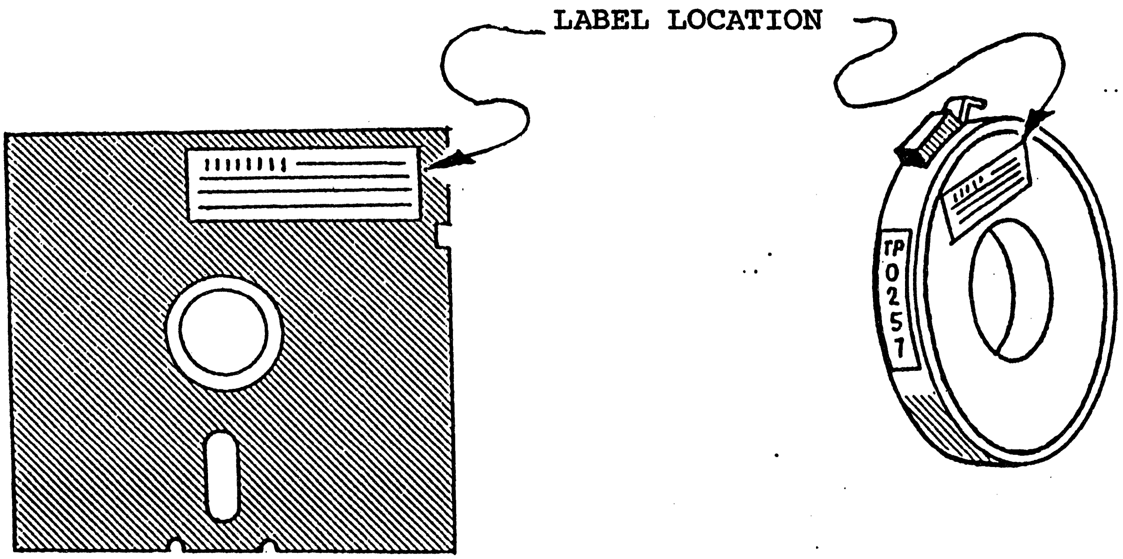
PROJECT NAME:	1		5		9		13		17		21		25	
	2		6		10		14		18		22		26	
	3		7		11		15		19		23		27	
REV:	4		8		12		16		20		24		28	

PROJECT NAME:	1		5		9		13		17		21		25	
	2		6		10		14		18		22		26	
	3		7		11		15		19		23		27	
REV:	4		8		12		16		20		24		28	

PROJECT NAME:	1		5		9		13		17		21		25	
	2		6		10		14		18		22		26	
	3		7		11		15		19		23		27	
REV:	4		8		12		16		20		24		28	

VENDOR SOFTWARE FILE LABELS


MUST MATCH NAME IN VENDOR SOFTWARE LOG BOOK INCLUDING THE SYSTEM IDENTIFIER PREFIX
FILE NUMBER
TRANSCOM ELECTRONICS
VENDOR S/W
FILE NAME:
DESCRIPTION:
OF
FORM 107721
BRIEF DESCRIPTION OF SOFTWARE
NUMBER OF DISKETTES OR TAPES IN SET



FLOPPY DISKETTE

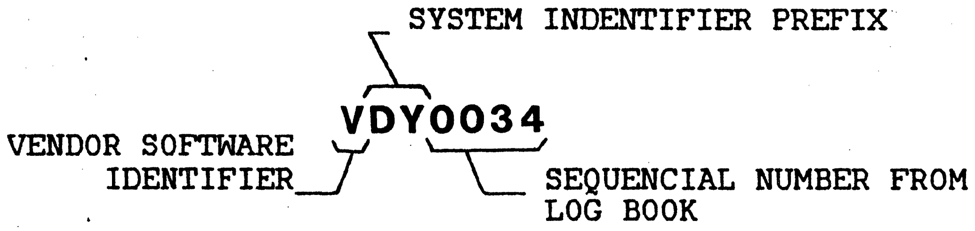
MAGNETIC TAPE

VENDOR SOFTWARE LOG BOOK ENTRIES

(SEE SECTION 8.0 OF THE ARCHIVAL PROCEDURE FOR FURTHER DETAILS)

FILE NUMBER: The number shall begin with the letter "V" to identify it as vendor software. This is to be followed by the appropriate system identifier prefix (see section 4.0 of the archival procedure). Finally the number is to include the a four digit sequencial number. This is to be obtained by using the next available number from the vendor software log book.

EXAMPLE:


SYSTEM INDENTIFIER PREFIX
VDY0034
VENDOR SOFTWARE IDENTIFIER
SEQUENCIAL NUMBER FROM LOG BOOK

TITLE: Enter the name that is used to identify the software. A maximum of 18 characters may be used.

VENDOR: Enter the name of the company that produced the software.

OPERATING SYSTEM: Enter the name of the operating system that the software is designed to run on. This entry is not required if the software has been written to run on a specialized system.

MEDIA (TYPE): Specify the nature of the media on which the software is copied.

MEDIA (SIZE): Enter the size of the media used. If floppy diskettes are used, specify the diameter. If magnetic tape is used, specify the length in feet.

DATE: Enter the date that appears on the vendor supplied label or instruction manual. Use the date that the entry is made into the log book if neither of these dates can be found.

REMOTE: Indicate if the software is to be sent to the remote storage facility by checking this block. This decision is to be made by the appropriate system manager.

THE FOLLOWING ENTRIES ARE TO BE COMPLETED BY THE ARCHIVE CONTROLLER ONLY

TO REMOTE: Enter the date that the software was sent to the remote storage facility.

FROM REMOTE: Enter the date that the software was returned from the remote storage facility.

FILE DELETED: Enter the date that the media was recycled.

INDEX (ADD): Check this column to indicate that the file has been added to the vendor software file index.

INDEX (DELETE): Check this column to indicate that the file has been deleted from the vendor software file index.

TRANSCOM ELECTRONICS

VENDOR SOFTWARE LOG

									ARCHIVE CONTROL USE ONLY				
FILE NUMBER *	TITLE -18 CHARACTERS MAX-	VENDOR -24 CHARACTERS MAX-	OPER SYSTEM	MEDIA		DATE	REMOTE		TO REMOTE	FROM REMOTE	FILE DELETED	INDEX	
				TYPE	SIZE		YES	NO				ADD	DEL

* INCLUDE SYSTEM IDENTIFIER PREFIX

TRANSCOM ELECTRONICS

MAGNETIC MEDIA CONTROL SHEET

TO REMOTE STORAGE FACILITY

LOCATION OF REMOTE SITE: ______________________

DATE SENT TO REMOTE SITE: ________ SENT BY: ______

FILE NUMBER/NAME	MEDIA		DATE RECD BY REMOTE	RECD BY
	TYPE	SIZE		

FROM REMOTE STORAGE FACILITY

DATE REQUESTED FROM REMOTE LOCATION: ______________

REQUESTED BY: __________

FILE NUMBER/NAME	MEDIA		DATE RECD ON SITE	RECD BY
	TYPE	SIZE		

INCREASING PRODUCTIVITY THROUGH BETTER FILE MANAGEMENT

Bonnie Persinger

Annapolis Micro Systems, Inc.

Annapolis, Maryland

ABSTRACT

The following paper provides helpful guidelines for increasing productivity levels through improved file management and pre-processing techniques

INTRODUCTION

The Telesis Design station is an excellent tool for generation of PCB designs. Many of its auto-routing, checking, and artwork generation routines require multiple text files to describe each part. These text files allow the designer a great deal of flexibility during the design process.

As flexible as the design station is, however, it is less than efficient as a text editor. This paper addresses file management procedures using your personal computer. Pin and device file creation, as well as checking will be discussed. A checklist of possible pin and device file syntax errors is included along with a description of programs that may be used for checking device files and netlists. The programs also generate summary reports that list the required packages, drill sizes used, and apertures used. A batch transfer utility that works with the Telesis TRM utility will also be described.

PRODUCTIVITY THROUGH IMPROVED FILE MANAGEMENT

The T-station was designed as a PCB design workstation, not a word processor. Basic word processing functions such as search and replace are not available to the T-station operator. Storage of text file libraries on the T-station is an inefficient use of valuable memory space. Files stored in the SYSTEM-LIBRARY still need to be transferred to the project, otherwise, post-processing is slowed to a crawl as the system searches through a vast SYSTEM-LIBRARY for the required files. Transferring these files with limited wildcards and batch commands is a tedious process, however, these shortcomings can be overcome by switching all text file creation and management to a personal computer. Text files can then be transferred to the T-station using the Telesis TRM transfer utility. Libraries can be stored on diskettes and clerical personnel can update text files using editors that they are already familiar with. Economically, it also makes more sense to use the T-station for the tasks it was designed for, as long as it can be supported on the front end by less expensive personal computers. Incidentally, I also believe that for maximum system productivity, schematic generation should be performed on the EDA-1ØØØ.

Files can be more easily organized and accessed on a PC than on a T-station and they can be gatherered into subdirectories. For example, I keep pin files in the PIN directory, and device files in the DEV directory. Control files are in a third directory, and checking programs in a fourth. Default paths can be set up so that checking programs are automatically called from text file directories. Any time a file needs to be found, it can be located using wildcard directory searches or a LOCATE utility.

Although it is not absolutely necessary, I like to set up a project directory. This directory has subdirectories called PIN, DEV, and DWG. The DWG directory contains the EDA-1ØØØ schematic drawings, netlist and report summaries. Control files could also be gathered in anticipation of a file transfer to the T-station. The entire project can be backed up on floppies, using one floppy to archive and another to travel with other project documentation. We use a subroutine to create a batch print file of both the DEV and PIN directories. These printouts go with the project documentation, allowing the designer to consult them during layout.

PRE-PROCESSING FILE CHECKING

The only possible advantage of inputting text files on the T-station is that it automatically checks syntax and semantics during the input process. To overcome the loss of this feature during PC text generation, the operator must either know every syntax rule (punctuation, acceptable file length, line order, etc.) or must use a checking program.

FIGURE 1 diagrams a pin file and indicates some syntax rules for each line. The relationship of certain fields to control files and package fields is also indicated.

FIGURE 2 diagrams a device file and indicates some syntax rules for each line. The relationship of certain fields to control files and package symbols is shown as well.

Device files are required to not only be semantically correct, but match the netlist also, regardless of what method was used for netlist generation. At AMS we generate netlists exclusively using the EDA-1000 schematic capture system. As you are aware, every device file and the netlist must be perfect in order for LOAD-TXT-NETLIST to run successfully. Since it may only take ten hours to find out if your netlist will load successfully, we wrote a device file and netlist checker. Invoking this checker, and correcting any problems prior to using LOAD-TXT-NETLIST has generally resulted in us only having to LOAD-TXT-NETLIST once. Our checker only checks EDA-1000 netlists, however if there is enough interest for a hand generated netlist checker, we will write that also.

USEFUL PROJECT SUMMARY REPORTS

A typical project at Annapolis Micro Systems proceeds in the following manner. First, an engineer begins the design of his circuit using an EDA-1000. As new schematic symbols are created, the engineer fills out a form that will be consulted during the device file creation. Normal EDA-1000 checks are performed and the netlist created. Old device files are transferred from the device library and placed into the project along with new device files.

Next, AMSNET is invoked. AMSNET checks the syntax of each device file called out in the netlist, and then verifies that it matches the netlist. AMSPIN is then invoked. This generates a report listing of the required pin files (and packages) and creates a batch file to be used to transfer all device files and the netlist to the T-station.

The Telesis operator then takes the package list and gathers and creates the appropriate package symbols. As new package symbols are created, a form is filled out listing the information that will be used to create new pin files. By the end of the year we hope to have a pin file syntax checker that would check for syntax and compatibility with the LAYERSTD. For now, we check them manually, and they are checked again by the first PDRC.

We then invoke PINSUM which checks that all the required pinfiles are in the directory, and creates a batch transfer file. The PINSUM report lists the apertures used so that you can check your APERTURE-TAB. It also lists the drill sizes used so that you can reduce the number of hole sizes and easily create your NCDRILL-FIG file. This report includes only those components that are called out in the netlist. If there are additional mounting holes, card ejectors, heatsinks, etc., that are not called out in the netlist, their pinfiles must also be consulted for creation of your Telesis control files.

BATCH TRANSFER OF FILES TO THE TELESIS

When you no longer wish to create your text files directly on the T-station but would prefer to transfer files using a personal computer, you must purchase a Telesis utility called TRM. TRM is a terminal emulator and transfer utility. Some software goes on your T-station, some on your personal computer. It is a wonderful, easy-to-use program that is set up in a simple question and answer format. However, there is a long pause between some of the questions and it is therefore tedious and time-consuming.

To overcome this, AMS wrote a batch transfer utility that replaces the operator during use of TRM. AMSXFR operates in two modes; if you only want to transfer one file, AMSFXR asks you the same questions that TRM would and then invokes TRM and answers the questions, leaving the operator free to get a cup of coffee. If several files need to be transferred, you can write (or have PINSUM create) a batch file. That way, during lunch or overnight, the PC can upload files to the T-station. This not only requires less operator intervention than using TRM alone, it also requires considerably less operator intervention than if the files are being transferred out of the SYSTEM-LIBRARY using Telesis file management techniques.

CONCLUSION

By performing all text file creation and management on a personal computer, and using preprocessing checkers and a batch transfer utility, Annapolis Micro Systems has dramatically reduced design time based on the following items:

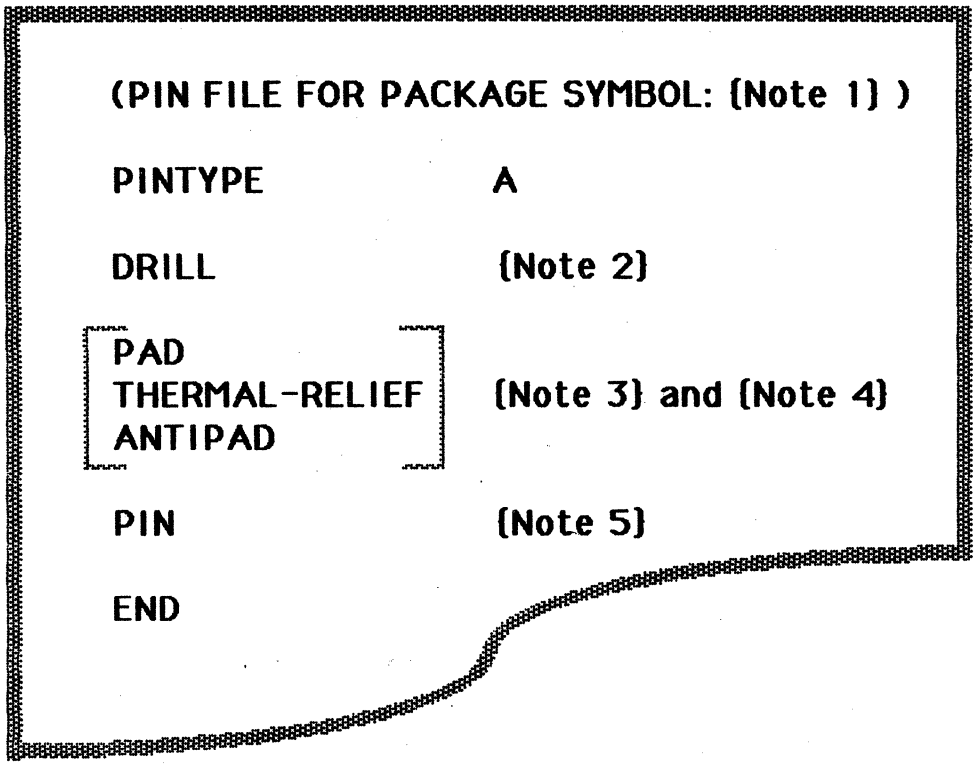
1. System memory is not wasted on pin and device files.

2. Transferring of files uses only the machine's time -- not the operator's.

3. Only one LOAD-TXT-NETLIST is required as all files are checked and corrected prior to being sent to the T-station.

4. You can use convenient text file editors to copy a file and search and replace certain fields.

5. Most offices have multiple PCs so that if the usual station is in use, editing can be off-loaded to another PC.



```
(PIN FILE FOR PACKAGE SYMBOL: {Note 1} )

PINTYPE              A

DRILL                {Note 2}

┌PAD             ┐
 THERMAL-RELIEF     {Note 3} and {Note 4}
└ANTIPAD         ┘

PIN                  {Note 5}

END
```

Note 1: Text file name must match package symbol name, (plus "-PIN").

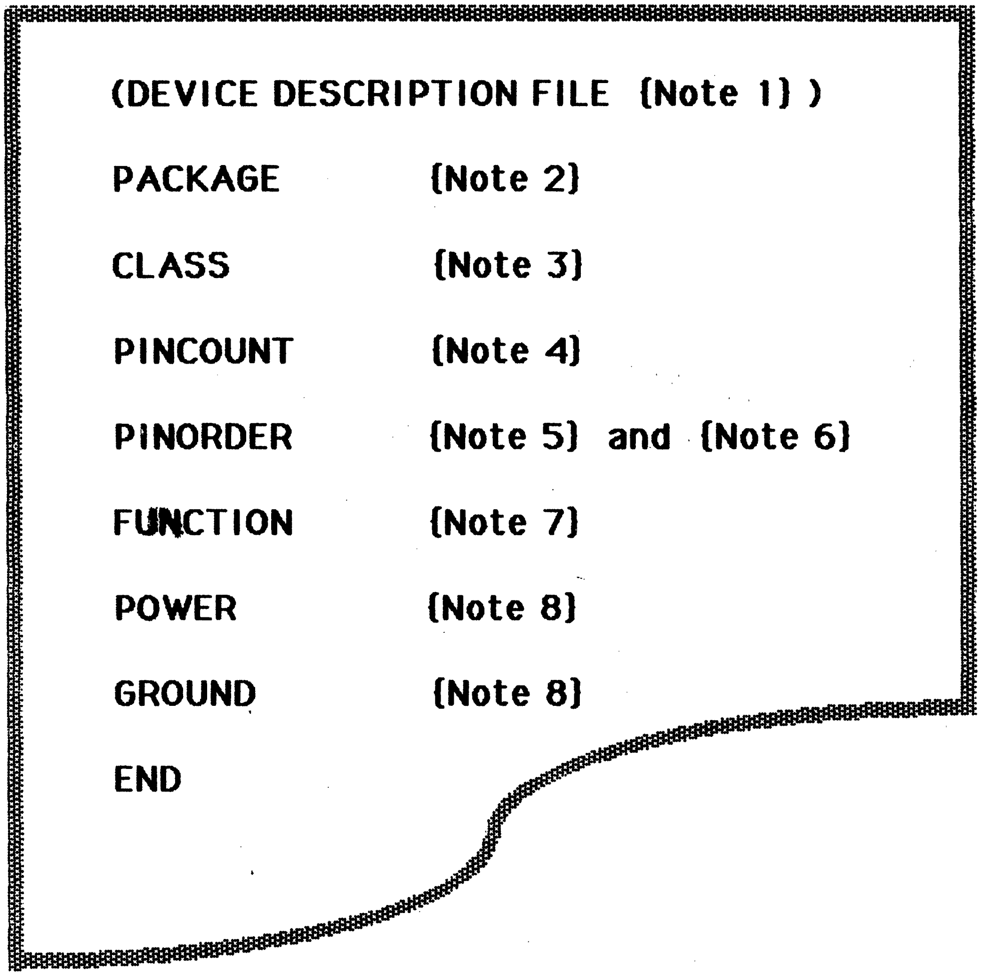
Note 2: Must be a decimal number (Don't use letter "o" for zero!).

Note 3: Field must be in your APERTURE-TAB file.

Note 4: Field must be in your LAYERSTD file.

Note 5: Must match Telesis package symbol pin numbers.
All pins must be included in some PIN line.
Pin numbers need not be sequential.

FIGURE 1	
1986	ANNAPOLIS MICRO



```
(DEVICE DESCRIPTION FILE  {Note 1} )

PACKAGE            {Note 2}

CLASS              {Note 3}

PINCOUNT           {Note 4}

PINORDER           {Note 5}  and  {Note 6}

FUNCTION           {Note 7}

POWER              {Note 8}

GROUND             {Note 8}

END
```

Note 1: Name of text file must match name of symbol or device type.
Note 2: Name of package must match name of package symbol
and the -PIN file name (less "-PIN").
Note 3: Important for interactive component placement.
Note 4: Too large a number will cause conflict when placing components.
Note 5: Function type must match device name or function name.
Note 6: Pin names must match pin names in schematic.
Note 7: Function type must match PINORDER function type.
Note 8: POWER and GROUND lines must be in device file
or connections made explicitly for each instance
in schematic.

FIGURE 2	
1986	ANNAPOLIS MICRO

CAD-TO-CAM FOR TELESIS USERS

Joe Prang

Product Marketing Manager

Telesis Systems

Chelmsford, Massachusetts

ABSTRACT

There are many requirements for data extraction from design databases (such as the Telesis) for downstream systems and databases. The amount of data, as well as the format, vary with the application. Tools may be designed as general purpose, or application-specific. The purpose of this paper is to inform Telesis customers of the tools available to support CAD/CAM interfacing.

INTRODUCTION

Before we delve into the specifics of the various packages, let's take a look at them from the top level:

NAME	DATA CONTENT	DATA FORMAT	APPLICATIONS
1. ASCII DATABASE OPTION	GRAPHICS DB	ONE FORMAT ONLY	WHEN FULL GRAPHICAL DATA REQUIRED
2. CAM INTERFACE	SELECTED NET DATA	FIELD/ORDER SELECTION	WHEN ONLY NON-GRAPHICAL DATA REQUIRED
3. ATE FIXTURE INTERFACE	PROBE HOLE LOCATIONS	TAPE AND/OR PLOT	DATA TO BUILD/ DOCUMENT BED-OF-NAILS FIXTURE FOR ANY TESTER
4. GENRAD/FLUKE FACTRON TESTER INTERFACE	NET/PARTS LIST DATA	TESTER FORMAT ONLY	NET/PARTS LIST IN TESTER-SPECIFIC FORMAT

As you can see, the data content ranges from a complete dump of the database in ASCII form to net/parts list data. Data format can be user-selectable, or made available as a file in the format specified. Thus, a package like ASCII database contains the required content to meet almost any application, although it may need to be reformatted. The CAM Interface serves a narrower range of applications, but may be formatted closely to the end application. Fixture/tester interfaces are intended for one application only and are provided in the specific format. In this paper we will be covering packages 2, 3, and 4 to focus on CAM applications.

CAM INTERFACE

A variety of manufacturing equipment exists that requires CAD database information for set-up. Time-to-market is reduced not only by using design automation equipment, but by implementing automatic set-up for manufacturing as well. Some examples of systems and equipment that may require data for set-up are:

- Insertion Systems
- AXIAL
- DIP
- RADIAL
- Automatic or semiautomatic
- SMD pick and place systems
- Wire wrap systems
- Bill of materials

To get a better understanding of the need, we'll look at examples of the data requirements for insertion systems and wire wrap.

Generally, insertion systems require a program that tells the system the location for component insertions, what order they should be inserted in, what types of components should be inserted, etc., (**FIGURE A**). If data is not extracted from a CAD database, the operator/programmer must digitize the component locations from plots and bareboards and then manually edit in the component data -- a time-consuming, error prone process. For automated set-up, the required data content might be:

X/Y DATA: This is placement data necessary for determining the location of the component insertion. The reference point on the component may be a pin (For example, PIN 1 on a digital device) or, the center of the component. On the Telesis system, it is easier to extract PIN 1 coordinates than to use the body center. From a format standpoint, since the movement of the insertion table has the most negative impact on throughput, it is preferable to order the X/Y data to "minimize" distance from one component to the next (**FIGURE B**).

SYMBOL NAME: Symbol names can be thought of as the "package type" in reference to CAM interfaces. Insertion systems work optimally if similar package types are grouped together in an insertion program. This is especially true for AXIAL leaded components.

DEVICE TYPE: Device type describes the functional contents of the package (e.g. 74ØØ). This data is relevant because devices of the same type may be delivered to the inserter together. In DIP inserters, the ICs come in tubes with the same device type. And, as with the package type, device types may be grouped together in the insertion program to minimize movement of the insertion head from tube to tube -- the less movement, the better the throughput (**FIGURE B**).

ROTATION: Insertion systems operate best when all components are oriented in the same direction. In fact, it is typically an option to get a rotary table on an inserter. Once again, throughput is impacted by switching back and forth between horizontally and vertically oriented components, and constantly rotating the insertion table (**FIGURE B**). Thus, components of like orientation are grouped together.

Now that we've seen some of the typical data content requirements for insertion systems, let's look at the CAM interface package and how it works for insertion systems.

CAM INTERFACE SOFTWARE

The CAM Interface software operates off of the net-database by extracting one or more user-selectable fields and then listing them in report form. The interface runs from the text terminal and requires an active drawing. **TABLE A** shows all of the data that can possibly be extracted. In the Insertion System example, the required data is contained in the fields numbered 1,2,4,7,8 and 11. The appropriate data is selected in a simple manner. First a menu (**FIGURE C**) is presented to the user and the user selects either <cr> for yes, or "n" for no. This causes the correct data to be included in the file. Note that the inserter works off of the PIN 1 location, so we've chosen only that pin in the "pin number range" field.

The next consideration is how the data will be "ordered". The CAM Interface allows a field to be selected that will also be the field the data is sorted by. In **FIGURE C**, we've selected the X coordinate field for ordering, since insertion table movement from component to component should be minimized in order to maximize throughput. The sorter will order the entries from top to bottom, and from left to right, creating a smooth "pattern" for insertion. The data sorted will be in accordance with the standard ASCII chart; numerics first, alphabet second, and special characters third. Upper case takes priority over lower (**FIGURE D**). The resulting output is shown in **FIGURE E**. The structure is kept simple (with only spaces used between fields, and no unusual delimiters such as , ; ! etc.) to facilitate any further manipulation of the file. The output report is a standard ASCII file that can be manipulated by the editor on the Telesis or other systems. Using the same data content required, and changing the format slightly by selecting the "symbol name" field for sorting, would result in the report shown in **FIGURE F**.

WIREWRAP

Our second application example is a wire wrap system that might be used for manufacturing prototype boards while production etched boards are being done. The data requirements are more simple than those for insertion since all that is needed is to define the end points (pins) that will be wirewrapped together to create each net. Returning again to the CAM Interface menu, the appropriate fields for this application are selected (**FIGURE G**) including netnumber, all pins (by selecting <cr> for pin range), and x,y locations. Sorting is ordered by the netnumber so that all pins on a particular net will be listed sequentially. The resulting report is shown in **FIGURE H**. This data is suitable for driving the wire wrap system. If additional documentation is needed, (i.e, for debugging), another run could be performed to create the same file but with, for example, reference designators and netnames, (**FIGURE I**). Creating a separate file avoids unnecessary programming to "strip" the documentation out of the file driving the machine.

These two examples show the targeted applications for the CAM Interface and how the interface can best be utilized. Any of the data shown in **TABLE A** can be included; fields may be included or omitted, and one field may be selected for ordering. There are some additional fields not shown in the examples. Let's take a look at these and how they may be used:

LAYERS: SMD through holes - zero

PINUSE CODES: Pinuse codes describe the electrical functionality of each pin on a device. Example codes are shown in **TABLE B**.

These codes could be used to generate reports and for making decisions for:

1. ECL/High frequency digital connections
2. ATE Fixture design
3. Logic design checking
4. Other applications that depend on the "functionality" of each pin

It's important to remember that these codes are entered into the device file and then included in the net-database (the source of all data extracted by the CAM Interface software).

MIRROR INFO: If the target CAM system is an SMD pick & place system, the mirror information will identify those components that should be mounted on the top versus the bottom. Thus, separate lists are easily made for each side.

ATE FIXTURE INTERFACE

The second package that is very useful for CAM applications is the ATE Fixture Interface. The purpose of this package is also to reduce manufacturing set-up time by generating the necessary data for vacuum bed-of-nails fixtures used to test boards. These fixtures are used with in-circuit, functional, and even in-house developed testers to eliminate manual probing of a circuit board when testing or diagnosing faults. Typically, one such fixture is required for each board type. Fixturing is both costly ($3,000+ each), and prohibitive to meeting schedules, as no testing occurs until the fixture is complete. **FIGURE J** shows the usual costs for developing these ATE fixtures, and what cost areas can be reduced by using the ATE Fixture software.

One of the highest cost areas involves assigning locations and creating drill tapes to indicate where the nails (probes) are to be located. The nails are spring loaded, and as a PCB is placed on a fixture and the vacuum applied, the PCB is compressed against the nails. The nails are physically located at component pins to contact the lead protruding through the board.

In many cases, plots (or bareboards) are manually marked to show where the probes will be placed (multiple probes per net for bareboard; typically one per net for loaded board). Then, similar to the Insertion System set-up, these manually derived locations are digitized to create the actual drill tape. This method is error prone and creates no useable documentation. The ATE Fixture software affects these and other costs as shown in **FIGURE K**. The software automates both the assignment and tape generation process. **FIGURE L** diagrams how the process works, which is very similar to the CAM Interface.

An active drawing (and, hence a net-data-base) are up on the EDA-300/700 and the ATE Fixture Interface is accessed via the text terminal. The user interface is again very simple (**FIGURE M**) allowing the user to create logfiles, plots, or tapes as outputs. The log file shows the list of locations selected by the probe target selection algorithm. The selection algorithm operates off of the "pinuse" codes previously described, and "class" codes that are in the device files. The program will select probe targets even if no pinuse or class code is specified in the device file library. However, there is a significant reduction in the quality of the resulting probe "map".

If the pinuse and class codes are used, the priority for selection works according to the list in **TABLE C.** The purpose behind this selection order is to minimize the impact of ECOs on an existing fixture. Often when track is cut and rerouted as part of an ECO (even if placement remains fixed) a net may lose its corresponding probe because the track was contacted at a component pin on the "other side" of the track from where the cut occurs. **FIGURE N** shows that by placing probes at the "sources" of signals, output, pins of ICs, edge connectors for incoming signals, etc., you can be certain of having a probe anyplace that you can route a track to.

Once the probe targets are selected by the algorithm, it is critical to create the following to fabricate, document, and debug the fixture:

1. A punch tape created in Excellon #2 format to drive the drill machine. It drills the plate that holds the bed-of-nails probes in place at the correct component pin locations. A sample of the tape is shown in **FIGURE O.** The format of the tape is specified in a control file called NC DRILL-PAR. This is the same file used to create standard drill tapes for board fabrication. The punch tape is an option available with Telesis systems.

2. A log file that documents the contents of the drill tape and plots in list form. This is used for documentation and debugging. **(FIGURE P).**

3. A number of plots are available to graphically show the fixture layout:

   - **FIGURE Q** - A plot of all component pins is sometimes used to drill all locations and load only the selected holes with probes.

   - **FIGURE R** - A plot of only the selected probe targets that can be used for drilling or debugging.

   - **FIGURE S** - A plot showing both component pins and probe targets that is useful for cross referencing.

- **FIGURE T** - A plot showing probe targets and cross reference numbers from the tester or CAD database (for example, net number). This is a highly sought after documentation/debug tool for test engineers that will be available in our next release.

- **FIGURE U** - A plot of the "keep-out" areas as they apply to fixturing. It is important to note that the keep out area rules for fixturing may differ from those required for insertion systems, card guides, etc.

From these tapes, plots, and reports, the time and cost of both designing and documenting a fixture is substantially reduced. In addition, assessing the impact of ECOs on existing fixtures and creating new up-to-date documentation is significantly improved.

ATE TESTER INTERFACES

The Automatic Tester Program generation process is similar to the PCB CAD process. Like CAD, it begins with a netlist, but since the tester will typically make measurements of each part, it is also necessary to provide a parts-list. This is especially critical for analog parts, where value and tolerance information is required. The importance of an accurate netlist for ATG is the same as for PCB design -- it's got to be right! It isn't hard to imagine that if the tester netlist/parts list is not automatically extracted from the CAD system, it must be manually re-keyed in on the tester. For medium to large boards, this process can consume up to 2Ø% of the total programming time (usually measured in weeks). If the data is re-keyed manually, it's also difficult to justify running the ATG early on in the design process for testability checking, since the netlist/parts list changes so frequently. New netlist/parts lists are also required for ECOs in order to assess the impact that design changes have on existing programs.

The ATE Tester Interfaces can automate the transfer of netlist/parts list data from Telesis to GenRad, Schlumberger, FACTRON, and other testers. The operation is very similar to the CAM Data Generator, and ATE Fixture Interface. A drawing is activated on the EDA-3ØØ/7ØØ and the ATE Tester Interface menu is accessed via the text terminal. A block diagram of the process is shown in **FIGURE V.** Once run, a tester-specific netlist is created in the format required by the tester. **FIGURE W** shows a GenRad file called a CKT file.

One of the key features of the ATE Tester Interface is the ability to adapt to the format requirements of the tester. This is an area where device files on the Telesis system play a key role. **TABLE D** shows the format requirements for GenRad in regards to device type definitions, pin naming conventions, etc. To automatically generate data in this format, device files are created that meet these three restrictions. **FIGURE X** shows an example device file for an analog device ; **FIGURE Y** shows one for a digital device. The fields marked with a "V" contain data that is specified to meet tester format requirements. The closer the device files meet the requirements, the better the quality of the output. It is even possible to support two different testers with corresponding device files.

With the ATE Tester Interface, input files for testers can be created that provide well over 9Ø% of the data required. There will typically be some additional input required to describe fixture wiring or test strategy/set-up requirements.

CONCLUSION

Three tools have been described that will automate a large portion of the manufacturing set up. These tools can significantly leverage the investment in Telesis CAD equipment. The impact on time-to-market (to first shipments) is great, as is the quality of data and documentation produced. The tools are:

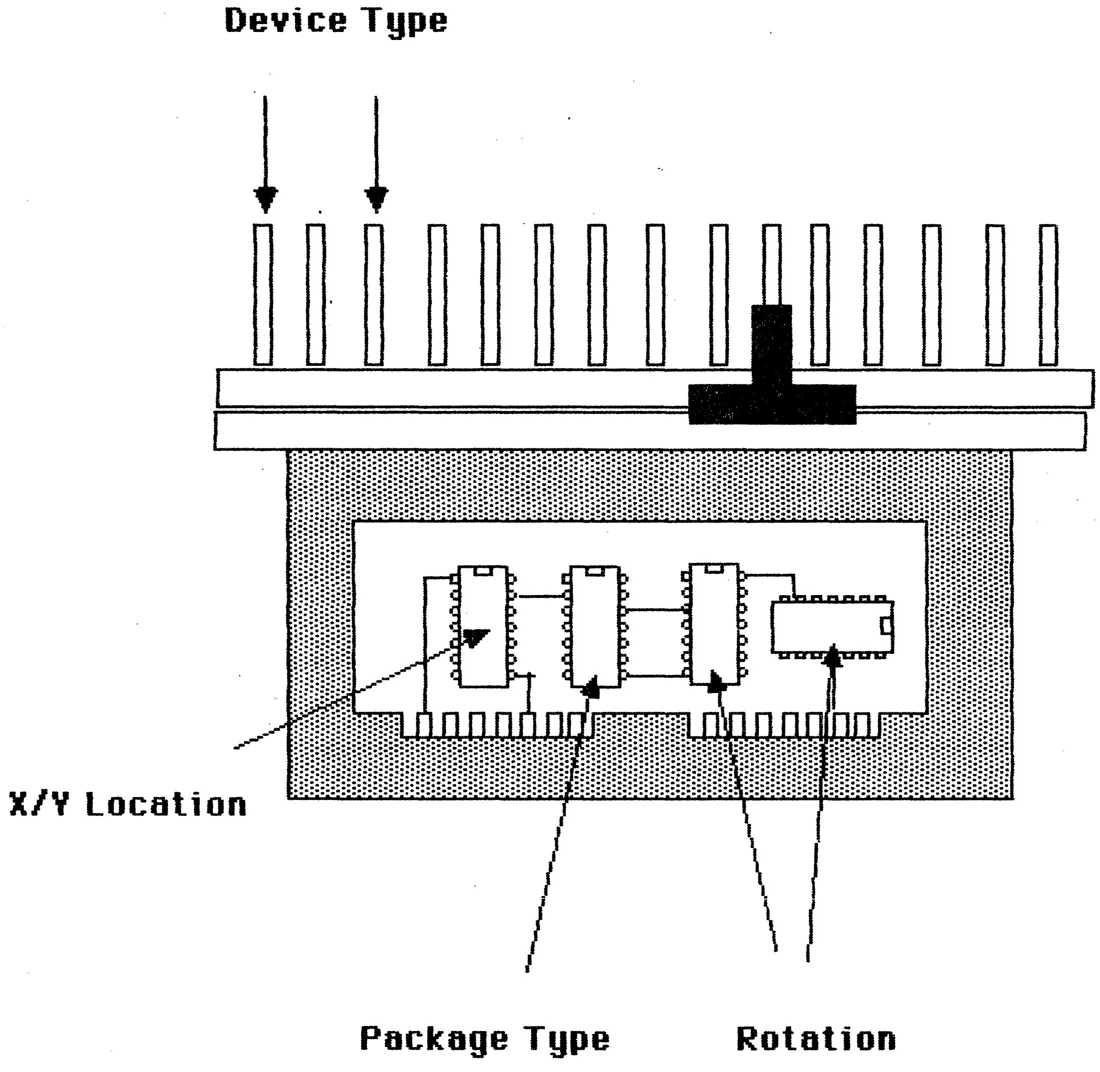
CAM Data Generator for extracting data for insertion systems, wirewrap, etc.

ATE Fixture Interface for designing, and documenting bed-of-nails fixtures.

ATE Tester Interface for the generation of tester specific netlist/parts lists necessary to drive automatic test program generation.

Figure A

Typical DIP Insertion Data Requirements



Device Type

X/Y Location

Package Type

Rotation

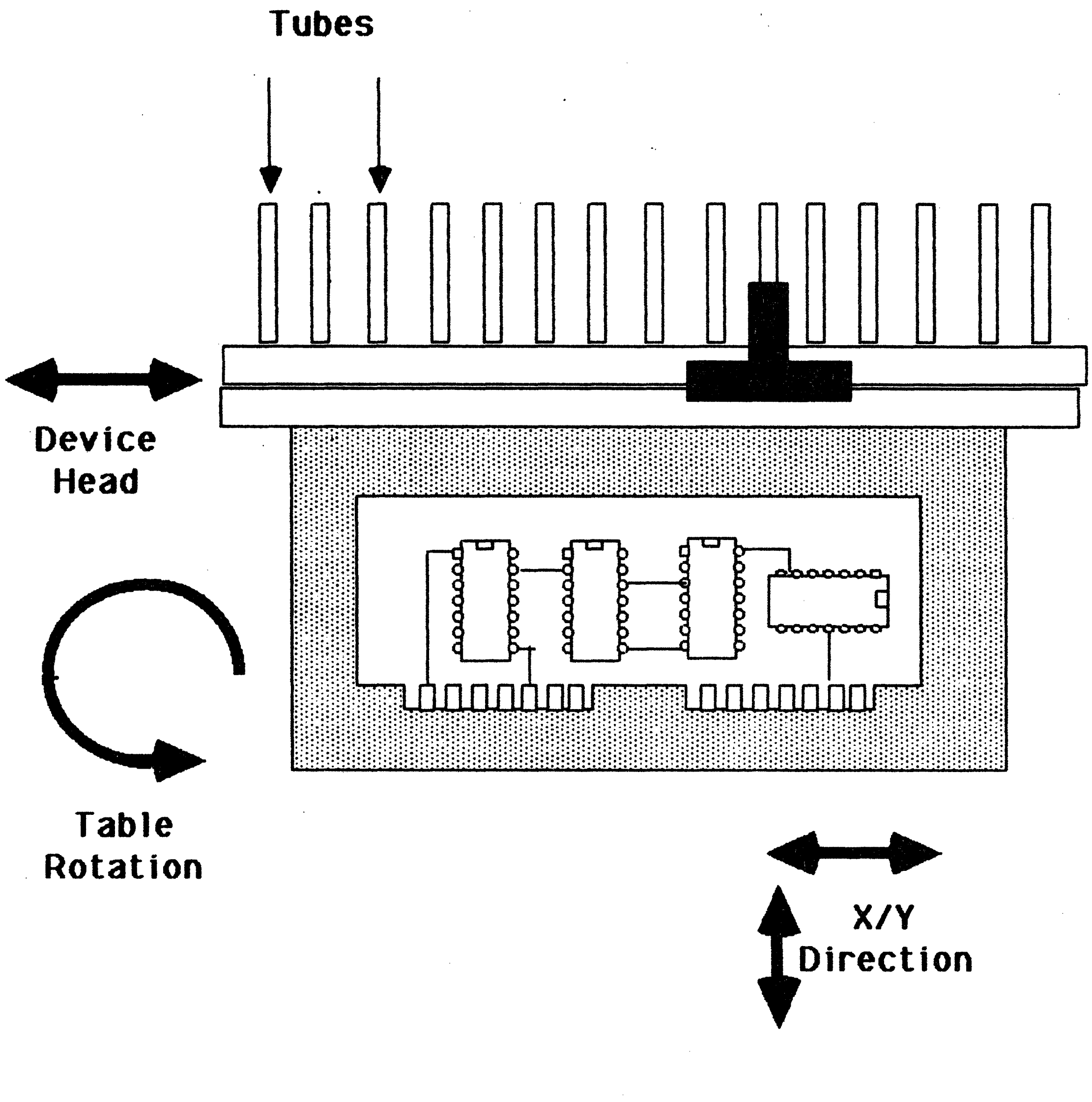
CAM/CAT Toolbox

1986

Telesis

Figure B

Typical DIP Insertion System Movements



CAM/CAT Toolbox

1986 Telesis

FIGURE C

```
>RUN [103.,50]PINLST

For the following prompts, type <CR> for yes, 'N' for no.
When a range is requested, delimit single numbers
with a comma or space, ranges with a dash.

Include symbol names?
Include device types?
Include ref des? N
Enter pin number range 1
Include net numbers? N
Include net names? N
Include X locations?
Include Y locations?
Include layers? N
Include pin use? N
Include rotations?
Include mirror info? N

Selected fields

1) Symbol name
2) Device type
3) Pin number
4) X location
5) Y location
6) Rotation

Field you would like to sort by: 1

DELETING OLD PIN-LISTING FILE

READING GRAPHICAL DATA BASE INFORMATION

READING NET DATA BASE INFORMATION

CONVERTING DATA FILE TO TEXT FILE

SORTING TEXT FILE

ADDING PIN-LISTING TO DRAWING MANAGER

ADDING HEADER

PIN-LISTING GENERATION DONE.
```

FIGURE D

ASCII CHARACTER SET

Octal Code	Character	Octal Code	Character	Octal Code	Character	Octal Code	Character
000	NUL	040	SP	100	@	140	
001	SOH	041	!	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BEL	047	'	107	G	147	g
010	BS	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FF	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	[	173	{
034	FS	074	<	134	\	174	|
035	GS	075	=	135	]	175	}
036	RS	076	>	136	^	176	~
037	US	077	?	137	_	177	DEL

* Equivalent to the Radix-50 character set

Figure E

```
(SYMBOL NAME          DEVICE TYPE               PIN#     X       Y     ROT)
 CAPRAD300             CAP,01                      1   12000    8700    3
 DIP14                 74S04                       1   12000    9800    0
 CAPRAD300             CAP,01                      1   12000   10100    3
 DIP16                 HM7621A                     1   12000   11200    0
 CAPRAD300             CAP.01                      1   12000   11500    3
 DIP16                 HM7621A                     1   12000   12600    0
 CAPRAD300             CAP,01                      1   12000   12900    3
 DIP14                 74LS11                      1   12000   14000    0
 CAPRAD300             CAP,01                      1   12000   14300    3
 CAPRAD300             CAP.01                      1   12000   15700    3
 CONN/E/50             ECONN                       1   12400    7950    0
 RES400                RES430                      1   12900    9200    0
 RES400                RES430                      1   12900   14800    0
 CAPRAD300             CAP.01                      1   13000    8700    3
 DIP16                 74S163                      1   13000    7800    0
 CAPRAD300             CAP.01                      1   13000   10100    3
 DIP16                 74LS193                     1   13000   11200    0
 CAPRAD300             CAP.01                      1   13000   11500    3
 DIP14                 74LS196                     1   13000   12600    0
 CAPRAD300             CAP.01                      1   13000   12900    3
 DIP16                 74LS193                     1   13000   14000    0
 CAPRAD300             CAP.01                      1   13000   14300    3
 DIP16                 74LS112                     1   13000   15400    0
 CAPRAD300             CAP.01                      1   13000   15700    3
 RES400                RES820                      1   13400    9300    0
 RES400                RES820                      1   13400   15300    0
 SIP8                  TERMINATOR1                 1   13700   12650    0
 CAPAX750              CAP6.8                      1   13800    8400    3
 CAPRAD300             CAP.01                      1   14000    8700    3
 DIP16                 74LS283                     1   14000    9800    0
 CAPRAD300             CAP,01                      1   14000   10100    3
 DIP16                 74LS283                     1   14000   11200    0
 CAPRAD300             CAP,01                      1   14000   11500    3
 DIP14                 74S86                       1   14000   12600    0
 CAPRAD300             CAP,01                      1   14000   12900    3
 DIP16                 74LS193                     1   14000   14000    0
 CAPRAD300             CAP,01                      1   14000   14300    3
 DIP14                 74S00                       1   14000   15400    0
 CAPRAD300             CAP,01                      1   14000   15700    3
 CAPRAD300             CAP.01                      1   15000    8700    3
 DIP16                 74LS367                     1   15000    9800    0
 CAPRAD300             CAP.01                      1   15000   10100    3
 DIP16                 74LS367                     1   15000   11200    0
 CAPRAD300             CAP.01                      1   15000   11500    3
 DIP14                 74LS04                      1   15000   12600    0
 CAPRAD300             CAP.01                      1   15000   12900    3
 DIP16                 74LS193                     1   15000   14000    0
 CAPRAD300             CAP.01                      1   15000   14300    3
 DIP14                 74LS74                      1   15000   15400    0
 CAPRAD300             CAP.01                      1   15000   15700    3
 RES400                RES430                      1   15900   12100    0
 RES400                RES430                      1   15900   13400    0
 CAPRAD300             CAP,01                      1   16000    8700    3
 DIP16                 74LS283                     1   16000    9800    0
 CAPRAD300             CAP,01                      1   16000   10100    3
 DIP16                 74LS378                     1   16000   11200    0
 CAPRAD300             CAP,01                      1   16000   11500    3
 DIP14                 74S74                       1   16000   12600    0
 CAPRAD300             CAP,01                      1   16000   12900    3
 DIP16                 74LS160                     1   16000   14000    0
 CAPRAD300             CAP,01                      1   16000   14300    3
 DIP14                 74S74                       1   16000   15400    0
 CAPRAD300             CAP,01                      1   16000   15700    3
 RES400                RES820                      1   16400   12500    0
 RES400                RES820                      1   16400   13900    0
>
>
>
>
>
>
>
>ER
```

Figure F

```
(SYMBOL_NAME          DEVICE TYPE             PIN1    X      Y     ROT)
 CAPAX750             CAP6.8                    1   13800   8400   3
 CAPRAD300            CAP.01                    1   12000   8700   3
 CAPRAD300            CAP.01                    1   12000  10100   3
 CAPRAD300            CAP.01                    1   12000  11500   3
 CAPRAD300            CAP.01                    1   12000  12900   3
 CAPRAD300            CAP.01                    1   12000  14300   3
 CAPRAD300            CAP.01                    1   12000  15700   3
 CAPRAD300            CAP.01                    1   13000   8700   3
 CAPRAD300            CAP.01                    1   13000  10100   3
 CAPRAD300            CAP.01                    1   13000  11500   3
 CAPRAD300            CAP.01                    1   13000  12900   3
 CAPRAD300            CAP.01                    1   13000  14300   3
 CAPRAD300            CAP.01                    1   13000  15700   3
 CAPRAD300            CAP.01                    1   14000   8700   3
 CAPRAD300            CAP.01                    1   14000  10100   3
 CAPRAD300            CAP.01                    1   14000  11500   3
 CAPRAD300            CAP.01                    1   14000  12900   3
 CAPRAD300            CAP.01                    1   14000  14300   3
 CAPRAD300            CAP.01                    1   14000  15700   3
 CAPRAD300            CAP.01                    1   15000   8700   3
 CAPRAD300            CAP.01                    1   15000  10100   3
 CAPRAD300            CAP.01                    1   15000  11500   3
 CAPRAD300            CAP.01                    1   15000  12900   3
 CAPRAD300            CAP.01                    1   15000  14300   3
 CAPRAD300            CAP.01                    1   15000  15700   3
 CAPRAD300            CAP.01                    1   16000   8700   3
 CAPRAD300            CAP.01                    1   16000  10100   3
 CAPRAD300            CAP.01                    1   16000  11500   3
 CAPRAD300            CAP.01                    1   16000  12900   3
 CAPRAD300            CAP.01                    1   16000  14300   3
 CAPRAD300            CAP.01                    1   16000  15700   3
 CONN/E/50            ECONN                     1   12400   7950   0
 DIP14                74LS04                    1   15000  12600   0
 DIP14                74LS11                    1   12000  14000   0
 DIP14                74LS196                   1   13000  12600   0
 DIP14                74LS74                    1   15000  15400   0
 DIP14                74S00                     1   14000  15400   0
 DIP14                74S04                     1   12000   9800   0
 DIP14                74S74                     1   16000  12600   0
 DIP14                74S74                     1   16000  15400   0
 DIP14                74S86                     1   14000  12600   0
 DIP16                74LS112                   1   13000  15400   0
 DIP16                74LS160                   1   16000  14000   0
 DIP16                74LS193                   1   13000  11200   0
 DIP16                74LS193                   1   13000  14000   0
 DIP16                74LS193                   1   14000  14000   0
 DIP16                74LS193                   1   15000  14000   0
 DIP16                74LS283                   1   14000   9800   0
 DIP16                74LS283                   1   14000  11200   0
 DIP16                74LS283                   1   16000   9800   0
 DIP16                74LS367                   1   15000   9800   0
 DIP16                74LS367                   1   15000  11200   0
 DIP16                74LS378                   1   16000  11200   0
 DIP16                74S163                    1   13000   9800   0
 DIP16                HM7621A                   1   12000  11200   0
 DIP16                HM7621A                   1   12000  12600   0
 RES400               RES430                    1   12900   9200   0
 RES400               RES430                    1   12900  14800   0
 RES400               RES430                    1   15900  12100   0
 RES400               RES430                    1   15900  13400   0
 RES400               RES820                    1   13400   9800   0
 RES400               RES820                    1   13400  13300   0
 RES400               RES820                    1   16900  12500   0
 RES400               RES820                    1   16400  13700   0
 SIP8                 TERMINATOR1               1   13700  12650   0
>
```

FIGURE G

```
For the following prompts, type <CR> for yes, 'N' for no.
When a range is requested, delimit single numbers
with a comma or space, range with a dash.

Include symbol names? N
Include device types? N
Include ref des? N
Enter pin number range
Include net numbers?
Include net names? N
Include X locations?
Include Y locations?
Include layers? N
Include pin use? N
Include rotations? N
Include mirror info? N

Selected fields

1) Pin number
2) Net number
3) X location
4) Y location

Field you would like to sort by : 2


DELETING OLD PIN-LISTING FILE

READING GRAPHICAL DATA BASE INFORMATION

READING NET DATA BASE INFORMATION

CONVERTING DATA FILE TO TEXT FILE

SORTING TEXT FILE

ADDING PIN-LISTING TO DRAWING MANAGER

ADDING HEADER

PIN-LISTING GENERATION DONE.
```

Figure H

```
( PIN#   NET#     X       Y    )
   6      0     12000    9300
   5      0     12000    9400
   8      0     12300    9200
   9      0     12300    9300
  10      0     12300    9400
  11      0     12300    9500
  12      0     12300    9600
  13      0     12300    9700
   2      0     12400    7950
   1      0     12400    7950
   3      0     12550    7750
   4      0     12550    7950
  10      0     13000    7950
   9      0     13000    7750
   5      0     13000   12200
   6      0     13000   14900
  11      0     13300    9300
  12      0     13300    9400
  13      0     13300    9500
  14      0     13300    9600
  13      0     13300   10900
  13      0     13300   13700
   8      0     13700   11950
   7      0     13700   12050
   6      0     13700   12150
   3      0     13700   12450
   2      0     13700   12550
   4      0     14000   10900
   8      0     14300   12000
   9      0     14300   12100
  10      0     14300   12200
  11      0     14300   12300
  12      0     14300   12400
  13      0     14300   12500
  13      0     14300   13700
  11      0     14300   15100
  12      0     14300   15200
  13      0     14300   15300
  27      0     14350    7750
  28      0     14350    7950
   7      0     15000   13400
```

```
   6      0     15000   13500
   2      0     15000   13900
   6      0     15000   14900
   5      0     15000   15000
  13      0     15300    9500
  14      0     15300    9600
   9      0     15300   10500
  13      0     15300   10900
  14      0     15300   11000
  12      0     15300   13600
  13      0     15300   13700
   8      0     15300   14800
   9      0     15300   14900
  10      0     15300   15000
  11      0     15300   15100
  12      0     15300   15200
  13      0     15300   15300
  50      0     16000    7750
  49      0     16000    7950
   6      0     16000   12100
   6      0     16000   14900
   9      0     16300    9100
  10      0     16300    9200
  13      0     16300    9500
  14      0     16300   11000
  15      0     16300   11100
   9      0     16300   12100
  14      0     16300   13800
  13      0     16300   13900
   8      0     16300   14800
   9      0     16300   14900
  10      0     16300   15000
  11      0     16300   15100
  12      0     16300   15200
  13      0     16300   15300
   5      1     12700    7750
   3      1     13000    9600
   6      2     12700    7750
   4      2     13000    9500
   7      3     12850    7750
   5      3     13000    9400
   8      4     12850    7950
```

Figure J

LOADED BOARD ATE FIXTURING COSTS

	Typical Costs[1]
1. Fixture Kit (~$500)	$ 500.00
2. Probe/Socket (~$1/pair)	$ 700.00
3. Probe/Socket Assembly Costs (~$2/pair)	$1400.00
↓ -Assigning Holes	
↓ -Drilling Holes	
-Loading Sockets	
-Wiring Sockets	
4. Finishing Costs (~$500)	$ 500.00
-Final Assembly	
-Checkout	
↓ -Documentation	
↓ 5. Fixture Debug (~$200)	$ 200.00
↓ 6. Fixture Maintenance (~$50/checkout)	$ 50.00
TOTAL	$~3300.00 (TYPICAL)

[1] AVERAGE 700 PROBES PER FIXTURE

↓ COSTS REDUCED BY TELESIS' ATE FIXTURE SOFTWARE

ATE FIXTURE INTERFACE	
1986	Telesis

Figure K

REDUCED ATE FIXTURING COSTS

1. Probe Hole Assignments
   - Software selected algorithmically rather than manually marking prints.

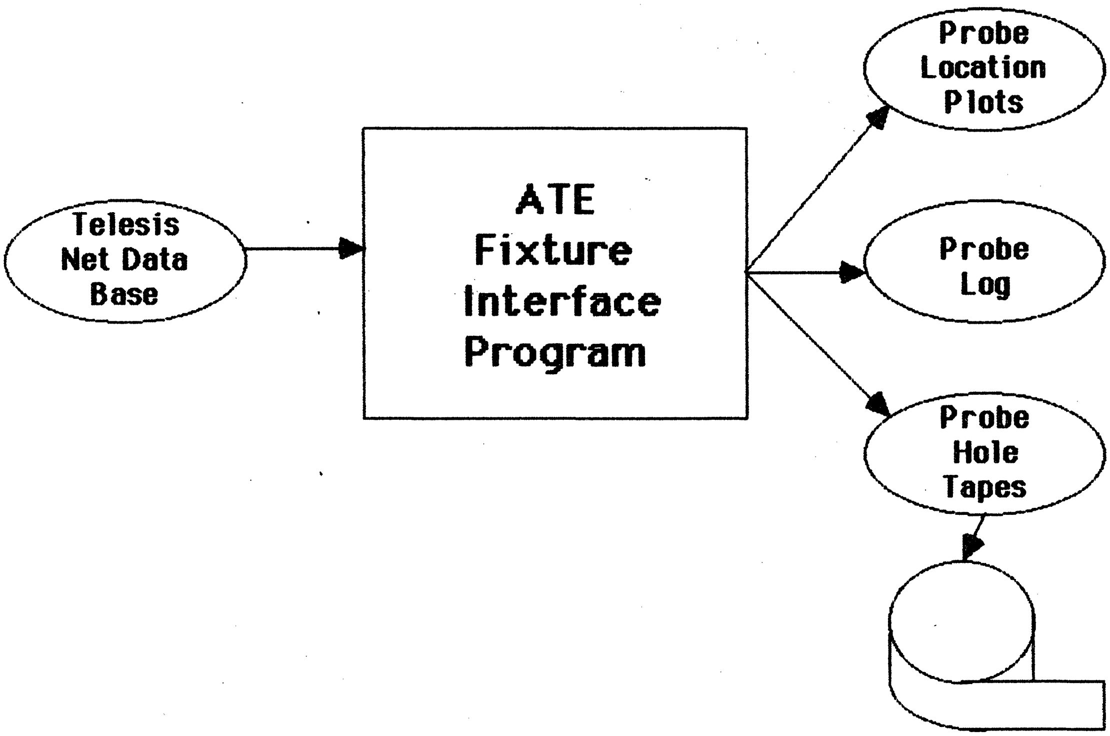
2. Probe Hole Drilling
   - Automatic generation of drill tape rather than manually digitizing.

3. Fixture Documentation
   - Drill plot shows all hole locations. Probe map indicates all probe locations. Component outlines and reference designators help locate probe hole sites.

4. Fixture Debug/Maintenance
   - Annotated fixture maps provide cross reference numbers to Telesis net name or tester pin number. This provides correlation between fixture documentation and test program/fixture checkout program. Never available previously.

Figure L

ATE Fixture Interface



o Accurate Test Fixture Data

o Accurate Documentation

o Reduced Labor Costs & Errors

CAM / CAT Toolbox

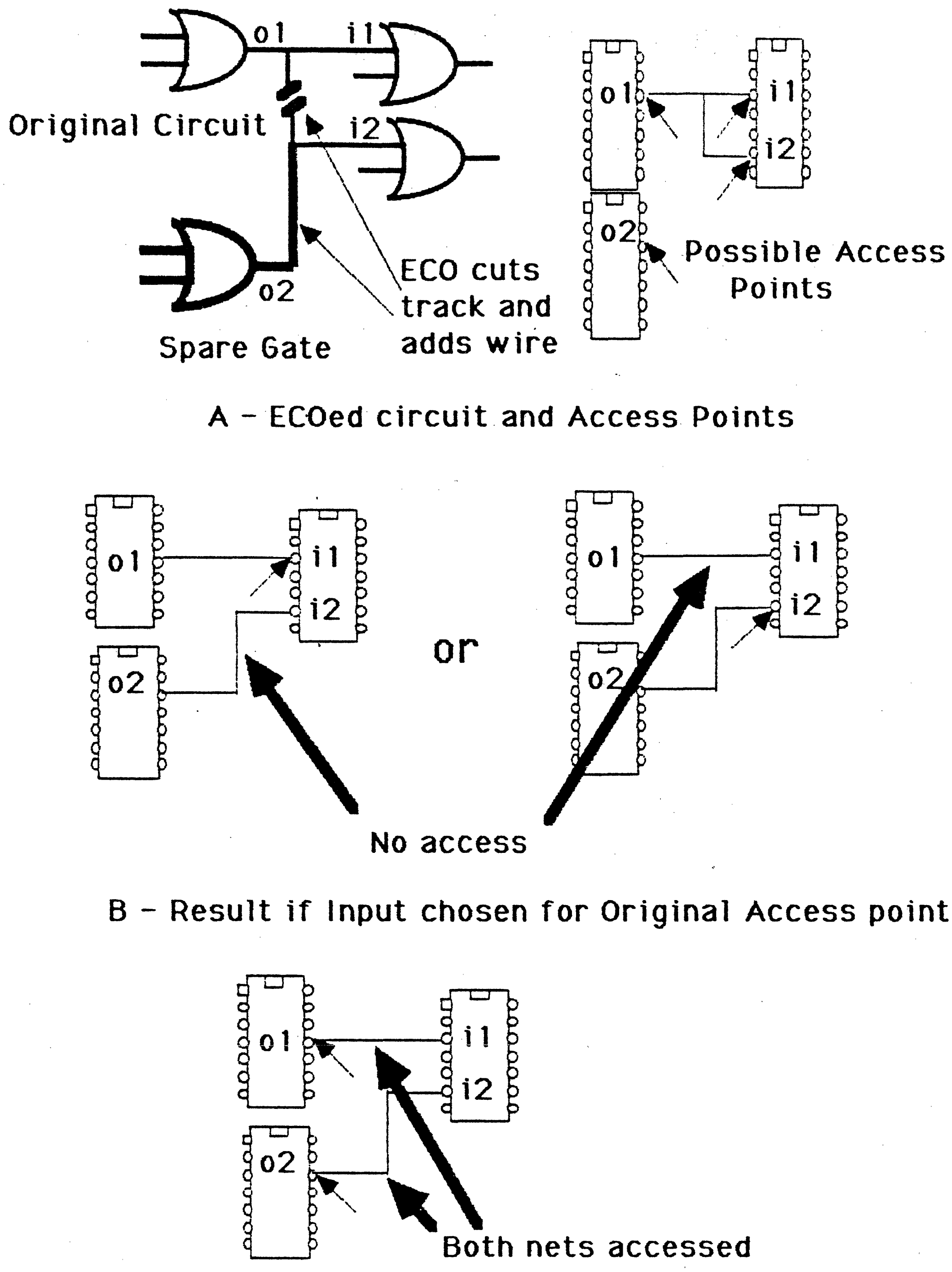
1986

Telesis

FIGURE M

```
                TELESIS ATE FIXTURE PROGRAM



CHOOSE ONE OF THE FOLLOWING OPTIONS:

1 ... RUN THE PROGRAM that creates the PROBE-TAPE and PROBE-LOG files.

2 ... RUN THE PROGRAM that punches the paper tape according to the data
      in PROBE-TAPE.

3 ... HELP.

4 ... EXIT.

ENTER YOUR CHOICE  :  3
```

Figure N



A - ECOed circuit and Access Points

B - Result if Input chosen for Original Access point

C - Result if Output chosen for Original Access point

Figure O

Sample PROBE-TAPE file.

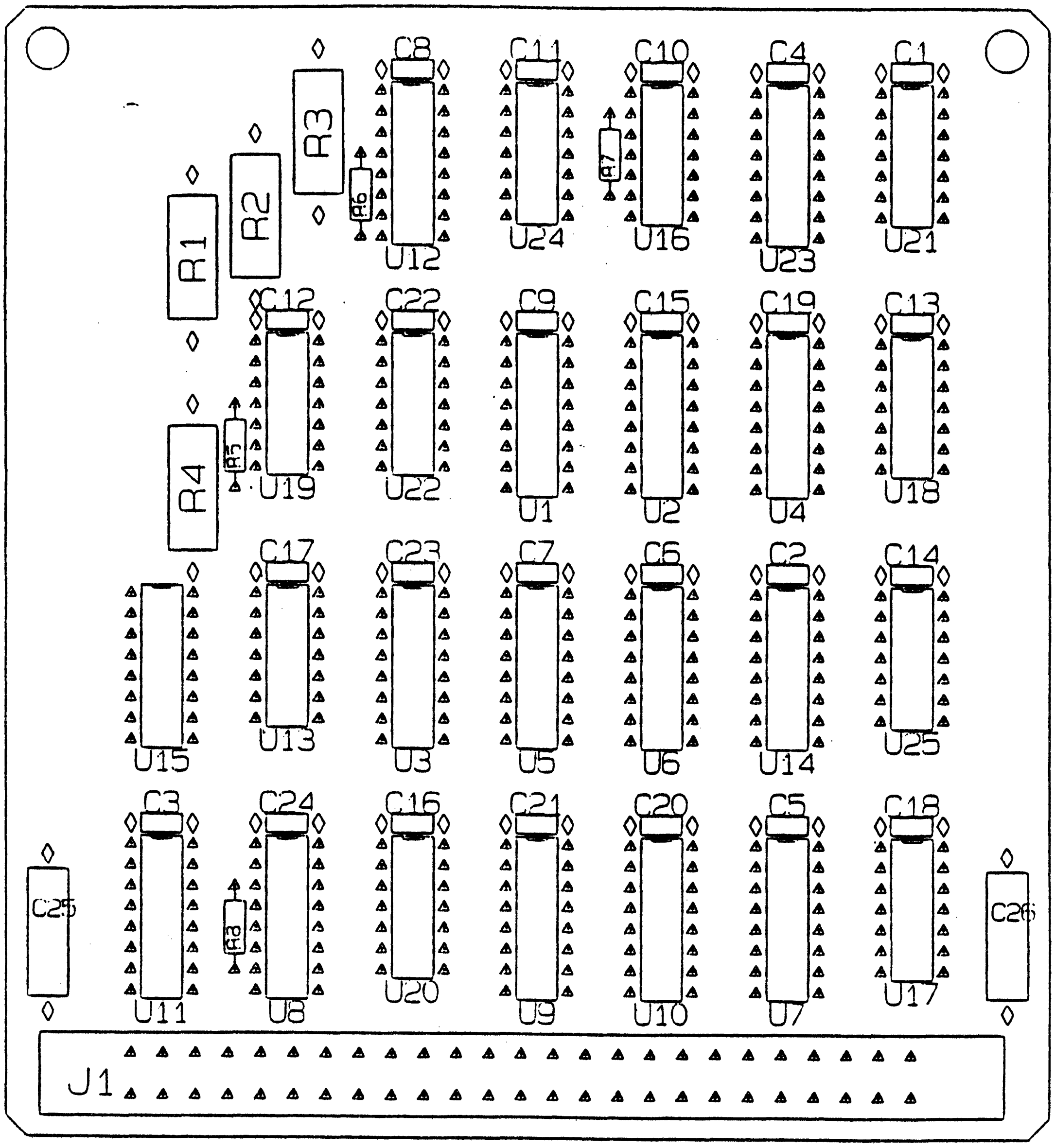
```
X03500Y03150     (Drill at location x,y)
X03650Y03300
X03800Y03450
X03900Y03150
X03950Y03600
X04050Y03300
X04100Y03750
X04200Y03450
X04300Y03900
X04350Y03600
X04450Y04050
X04500Y03750
X04600Y04200
X04700Y03900
X04850Y04050
X05000Y04200
X04750Y06400
R07Y-00100       (Decrement y seven times, drilling each time)
X04450Y05700
R06Y00100
X03050Y06300
X04450Y05200
R06Y-00100
X04750Y04600
R06Y00100
X03050Y05200
R06Y-00100
X03350Y04600
R06Y00100
M00              (End of block)
```

Figure P

SAMPLE PROBE-LOG FILE

```
-------------------------------------------------
PROJECT : ACC-TEST
DRAWING : PCB
REV     : 1

DATE    : 10-JAN-85
TIME    : 13:43:12
-------------------------------------------------
                PROBE HOLES LIST
                              Reference
Net Number      Pin Number    Designator
-------------------------------------------------
   14               10            U2
    2                9            U2
   12               12            U2
    1               13            U2
    0               11            U1
    4                8            U1
    3                6            U1
    5                3            U1
    0               15            J1
    0               14            J1
    0               13            J1
   15                1            J1
   16               12            J1
   15               11            J1
   16               10            J1
   13                9            J1
    5                8            J1
    6                7            J1
    7                6            J1
    8                5            J1
    9                4            J1
   10                3            J1
   11                2            J1
-------------------------------------------------
PARAMETERS USED DURING CREATION OF PROBE-TAPE

CODE EIA
ENGLISH
ABSOLUTE
FORMAT    2.4
TRAILING-ZEROS SUPPRESS
LEADER    10
MACHINE-OFFSET   0,0
HEADER ACCEPTANCE TEST
-----------------------------------------------------
```

Figure Q - STANDARD DRILL DRAWING

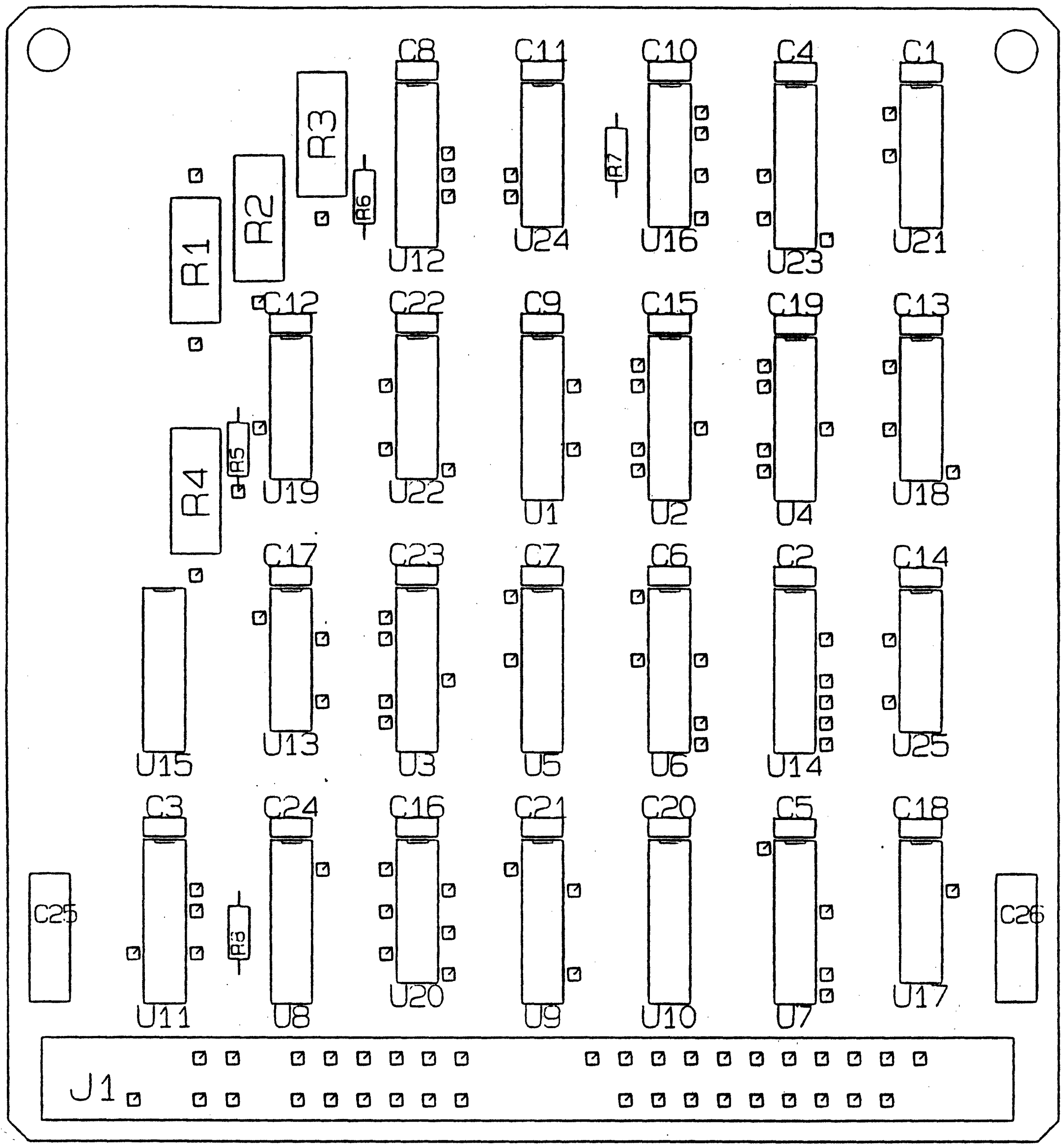


- o Component Outlines
- o Reference Designators
- o Component Hole Locations

ATE FIXTURING DATA

1986 Telesis

Figure R - FIXTURE PROBE MAP



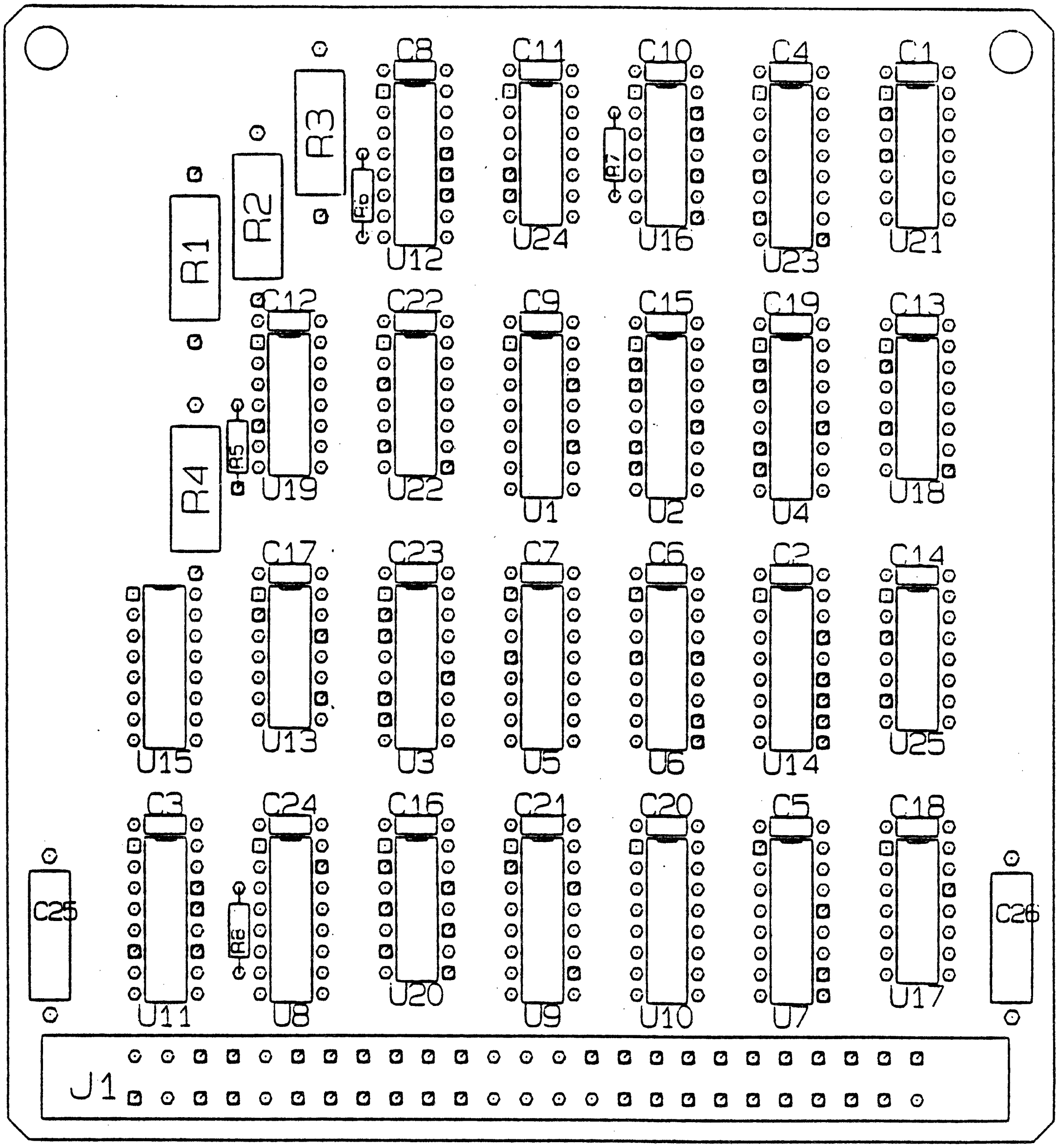
- o Component Outlines
- o Reference Designators
- o Probe Hole Locations

ATE FIXTURING DATA

1986

Telesis

Figure S - FIXTURE PROBE MAP AND COMPONENT HOLES

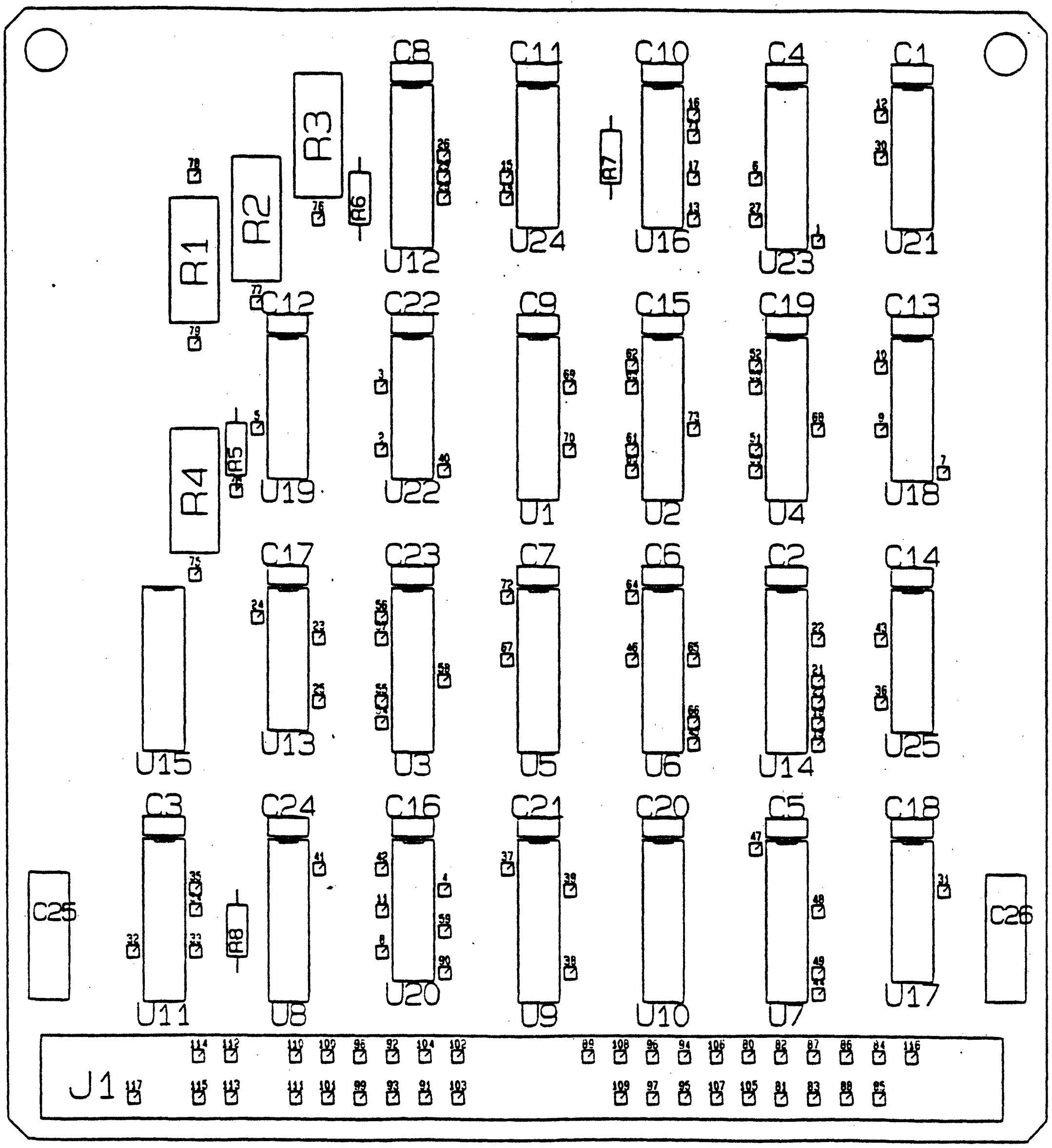


- Component Outlines
- Reference Designators
- Probe Hole □ and Component Hole ○ Locations

ATE FIXTURING DATA

1986 Telesis

Figure T - FIXTURE PROBE MAP WITH REFERENCE NUMBERS

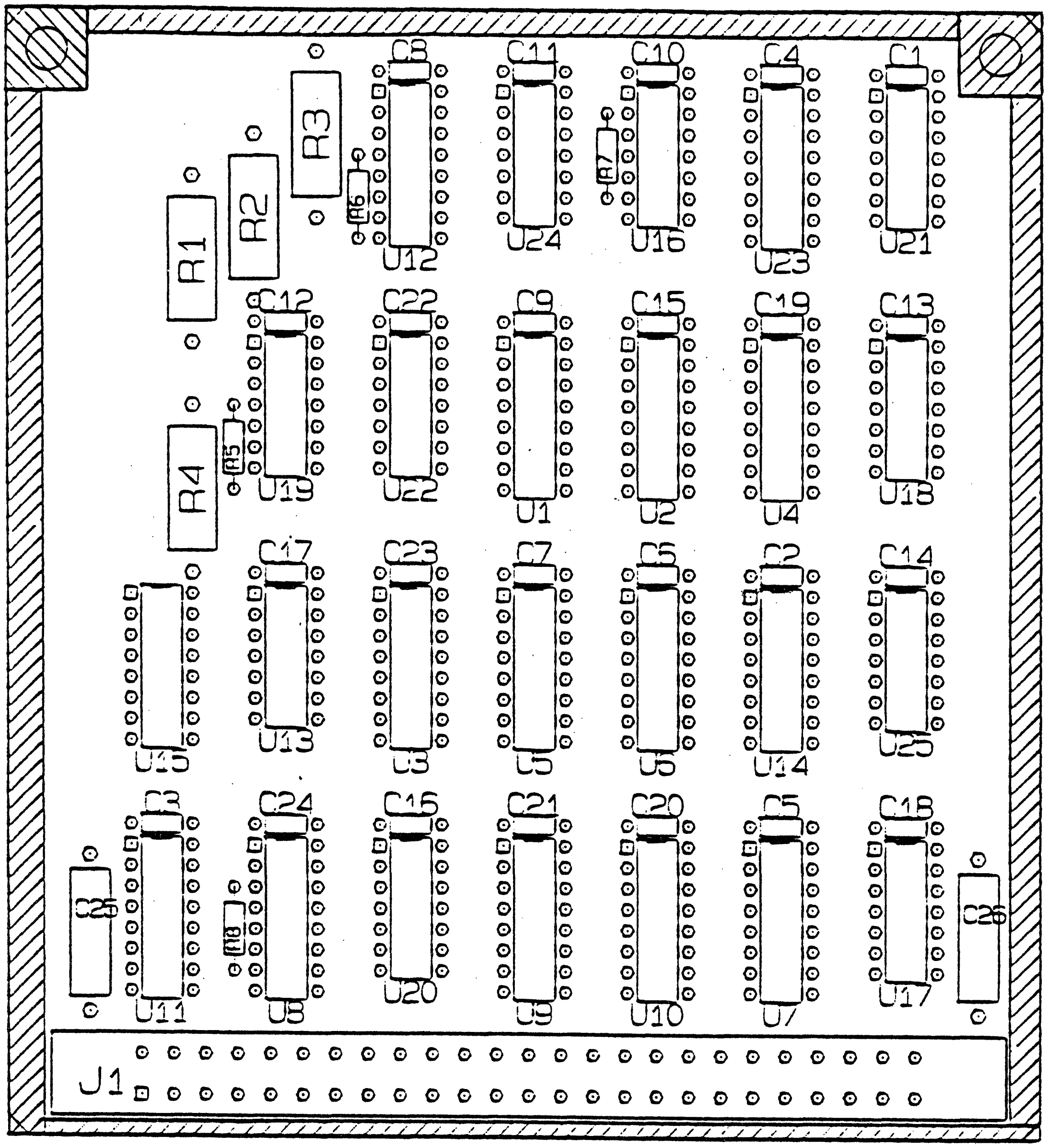


- o Component Outlines
- o Reference Designators
- o Tester/Telesis Net Number

ATE FIXTURING DATA

1986 Telesis

Figure U - "KEEP OUT" RECTANGLES



- Allows room for Fixture Seal
  - Board Edge
  - Cutouts

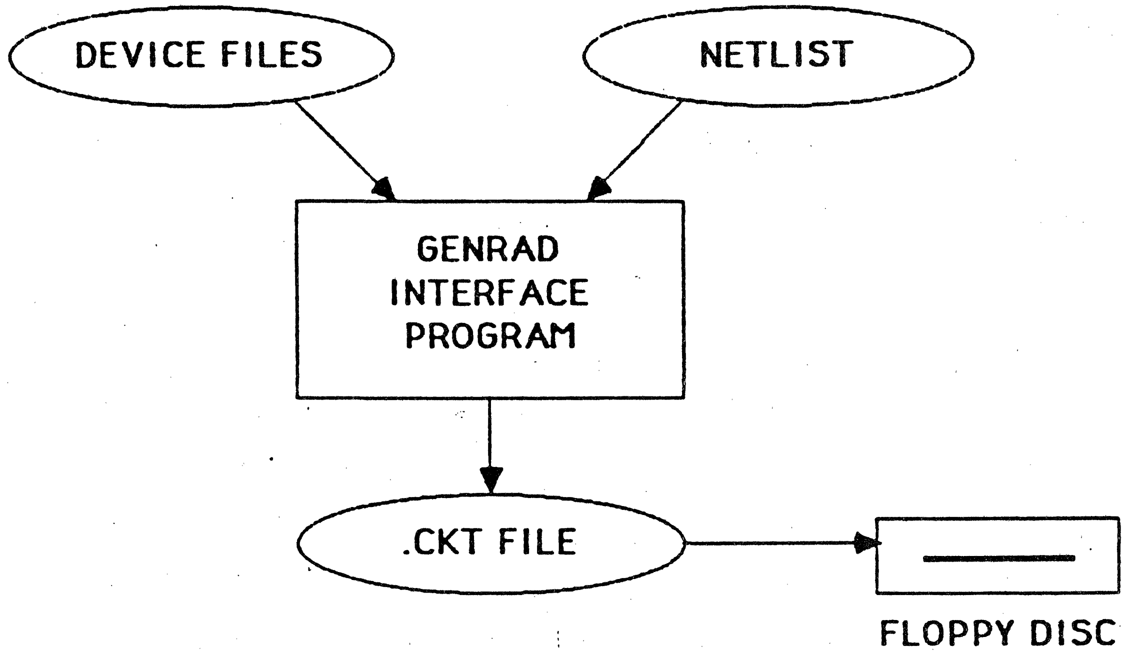
- Defines placement/routing area

ATE FIXTURING DATA

1986 Telesis

Figure V

TELESIS TO GENRAD TESTER INTERFACE



WHY ?

- NEED ACCURATE AND TIMELY CIRCUIT DESIGN DATA FOR TESTING

RESULTS

- REDUCED SET UP TIME
- REDUCED SET UP COST
- QUICK EVALUATION OF ECO IMPACT

Design for TEST

1986 Telesis

Figure W

SAMPLE .CKT FILE

```
/*CIRCUIT DESCRIPTION FILE MYFILE.CKT FOR GENRAD 227Ø*/

%CIRCUIT;

GND       EXT       ,U1.7 R2.A J.1 C2.P C1.P U2.11;
VCCA      EXT       ,J.12 C2.N C1.N;
C1        CP        P=U1.7 R2.A J.1 C2.P U2.11,N=J.12 C2.N;
C2        CP        P=U1.7 R2.A J.1 C1.P U2.11,N=J.12 C1.N;
J         UK        U1.7 R2.A C2.P C1.P U2.11,,U1.2,U1.3,
                    U2.5 U2.3,U2.7 U2.1,U1.5,U1.6,,
                    U2.8 U2.14 U1.1Ø,,C2.N C1.N,,,;
U1        74LSØØ    U2.2,J.3,J.4,U2.6,J.7,J.8,R2.A
                    J.1 C2.P C1.P U2.11,U2.13,U2.9,U2.8
                    U2.14 J.1Ø,,,,U2.4;
R2        VZ        A=U1.7 J.1 C2.P C1.P U2.11,C=U2.12;
R4        VZ        ;
U2        74L74     U2.7 J.6,U1.1,U2.5 J.5,U1.14,U2.3
                    J.5,U1.4,U2.1 J.6,U2.14 U1.1Ø J.1Ø,
                    U1.9,,U1.7 R2.A J.1 C2.P C1.P,R2.C,
                    U1.8,U2.8 U1.1Ø J.1Ø;

%VALUE:227Ø;

C2        = 1.U,1Ø%;
R4        = 1Ø.MEG,+5%,-5%;
C1        = 2Ø,ØN,+2Ø%,-5%;
R2        = 7.9K,5%;


%ADAPTOR:227Ø;

H      1 = N      1;
H      2 = N      2;
H      3 = N      3;
H      4 = N      4;
H      5 = N      5;
H      6 = N      6;
H      7 = N      7;
H      8 = N      8;
H      9 = N      9;
H     1Ø = N     1Ø;
H     11 = N     11;
H     12 = N     12;
H     13 = N     13;
H     14 = N     14;
H     15 = N     15;
H     16 = N     16;
H     17 = N     17;
H     18 = N     18;
```

FIGURE X

ANALOG DEVICE FILE

```
(DEVICE FILE FOR : CAPEL800)
PACKAGE CAPEL800
CLASS DISCRETE
PINCOUNT2
PINORDER CAPEL800 P N
PINUSE CAPEL80 B1 B1
FUNCTION G1 CAPEL800 1 2
DEVICE-LABEL CO
END
```

FIGURE Y

DIGITAL DEVICE FILE

```
(DEVICE DESCRIPTION FILE: 7400)
PACKAGE DIP14
CLASS IC
DEVICE-LABEL 7400
PINCOUNT 14
PINORDER 7400 A B Y
PINUSE 7400 IN IN OUT
PINSWAP 7400 A B
FUNCTION G1 7400 1 2 3
FUNCTION G2 7400 4 5 6
FUNCTION G3 7400 9 10 8
FUNCTION G4 7400 12 13 11
POWER +5V  ;  14
GROUND GND  '  7
END
```

TABLE A

DATA FIELDS

1. Device type
2. symbol name
3. reference designator
4. pin numbers
5. net numbers
6. net name
7. X position
8. Y position
9. layer
10. pin use
11. rotation
12. mirror

TABLE B

PINUSE CODES

- IN - Input (receiving)
- OUT - Output (sending)
- BI - Bidirectional (sending or receiving)
- TRI - TRISTATE (sending, receiving or held at some state)
- OCA - Output Collector And (used to tie multiple outputs together in TTL circuitry without receiving an error).
- POWER - Pins used as power
- GROUND - Pins used as ground
- NC - Not connected internally

TABLE C

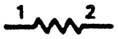
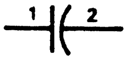
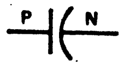
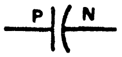
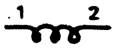
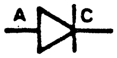
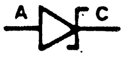
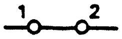
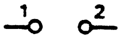
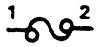
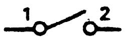
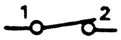
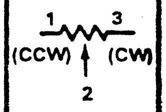
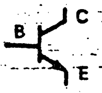
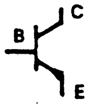
Probe Target

Selection Table

CLASS	PINUSE	PRIORITY
IO	-	2
IC	IN (Input)	3
	OUT (Output)	1
	OCA (Output)	2
	BI (Output)	1
	TRI (Output)	1
	OCL (Output)	1
	POWER (Input)	3
	GND (Input)	3
	NC (Input)	3
DISCRETE	-	4
-	-	4

Table D

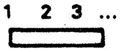
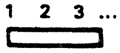
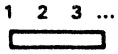
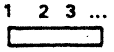
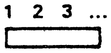
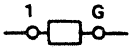
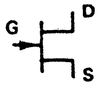
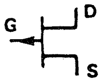
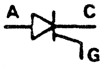
COMPONENT TYPE SYMBOLS USED IN CDL

COMPONENT TYPE	CDL SYMBOL	SCHEMATIC SYMBOL	COMPONENT DESCRIPTION
RESISTOR	R		Any resistor. The leads must be designated as 1 and 2.
CAPACITOR	C		Any non-polarized capacitor. The leads must be designated as 1 and 2.
CAPACITOR POLARIZED	CP		Polarized capacitor. The leads must be designated P for the positive and N for the negative. No reverse bias greater than 1.0 volt is allowed by ATG.
CAPACITOR POLARIZED .	CP1		Same as CP except no reverse bias greater than 0.0 volt is allowed by ATG. Typically for tantalum capacitors.
INDUCTOR	L		Any inductor. The leads must be designated as 1 and 2.
DIODE	CR		Any non-Zener diode. The leads must be designated A for anode and C for cathode.
ZENER DIODE	VZ		Any Zener diode. The leads must be designated A for anode and C for cathode.
JUMPER	J		Any very low resistance connection. The ends must be designated 1 and 2.
NO JUMPER	OJ		Any open circuit. The nodes must be designated 1 and 2.
FUSE	F		Any fuse. The leads must be designated 1 and 2.
OPEN CONTACT	NO		Any normally open contact. The terminals must be designated 1 and 2.
CLOSED CONTACT	NC		Any normally closed contact. The terminals must be designated 1 and 2.
POTENTIOMETER	RV		Any adjustable resistor. The leads must be designated such that the wiper is 2, the terminal that the wiper approaches when turned counter clockwise is 1, and the terminal that the wiper approaches when turned clockwise is 3.
TRANSISTOR NPN	QN		Lead designations are B for base, C for collector, and E for emitter.
TRANSISTOR PNP	QP		Lead designations are B for base, C for collector, and E for emitter.

1552

Table D

COMPONENT TYPE SYMBOLS USED IN CDL

COMPONENT TYPE	CDL SYMBOL	SCHEMATIC SYMBOL	COMPONENT DESCRIPTION
PC BOARD	BOARD		Used when a flagspec that applies to the pc board is to be specified by the user.
DEVICES NOT RECOGNIZED BY ATG LIBRARIES	ICnn		A device for which the ATG libraries do not have a test. Assumed to be a semiconductor device; so, the diode model is used, the VCC and GND guarding is used and the knee voltage VETO is used. The number of pins on the device replaces the nn in the CDL symbol. The number of pins allowable is any integer from 2 through 127. Default is 40 pins.
	ICAnn		Same as ICnn except that the guarding is not done.
	DICnn		A digital IC which has no test in the library but ATG will provide a dummy burst in the test program and nail assignment will reserve pins on MUX tester. The number of pins allowable is 8 or an even number from 14 through 40.
	UKnn		A device that is assumed not to be a semiconductor device, so no guarding of VCC and GND, no knee voltage VETO, and no diode model are used. The impedance between pins is considered to be infinite for ATG circuit analysis. Pin quantity is same as ICnn.
INTEGRATED CIRCUIT	'(type)		An IC for which there is a test in a digital library or component library. Pin numbers are consistent with manufacturer's data and consequently with library specification.
USER SPECIFIED GUARD	GD		Forces a guard at node G regardless of whether the library procedure specifies a guard.
COMPONENT LIBRARY CIRCUIT ELEMENTS			
FIELD EFFECT TRANSISTOR N JUNCTION	NJFET		Lead designations are D for drain, G for gate and S for source. ATG provides a FORM test only for this device. The user must complete the test.
FIELD EFFECT TRANSISTOR P JUNCTION	PJFET		Lead designations are D for drain, G for gate and S for source. ATG provides a FORM test only for this device. The user must complete the test.
SILICON CONTROLLED RECTIFIER	SCR		Lead designations are A for anode, C for cathode, and G for gate.

1553

BOARD LEVEL PHYSICAL DESIGN FOR TESTABILITY

~~by~~

Joseph A. Prang
Telesis Systems Corporation
Chelmsford, MA 01824

INTRODUCTION

Research to date in the area of testability has focused almost exclusively on the area of IC design. Experts in the area of logic simulation have developed a variety of programs that assess the two most predominant measures of testability, "controllability" and "observability." The basic premise is that if the circuit can be controlled to set the logic to the desired state (controllability) and the fault being modeled is visible (observability) the circuit is testable. While these techniques have proved very useful at the IC level they have been less useful at the PCB level. One obvious reason is the complexity factor significantly increases at the board level and also the difficulty of modeling complex commercial VLSI devices. Another less obvious reason is that while the circuit may be controllable and observable from an electrical standpoint, the physical board design may make it impossible to access the circuit with the board test equipment. This access can be accomplished via bed-of-nails fixtures, test connectors (especially for conformally coated boards), or hand held probes. However, many times it may be impractical or at least costly to relayout a board once its been "cast in fiberglass" to gain the necessary access. This accessibility problem will significantly increase as more and more surface mount devices (which may be totally inaccessible) are used. In fact, the constant pressure on PCB CAD companies to support improved blind and buried via capability for auto routing further deteriorates testability at the board level. It is these set of circumstances that lead to the need for new tools for "Design for Accessibility" (DFA) for PCB CAD to improve overall board testability. The concepts, issues, and process of Design for Accessibility at the board level is the subject of this paper.

TEST METHODS VERSUS ACCESSIBILITY

DFA then is the physical problem of providing a test point or "target" on a printed circuit board that is of the right size, shape, and spacing from an adjacent target. The target mustalso be unobstructed by outside layers or component packaging. It's probably intuitively obvious that to fully control and observe a circuit with an ATE system all points (nets) would be physically accessible. Thus, we could set the ideal case as the one where every net can be accessed at least once. Of course the appropriate level of access varies with the test method:

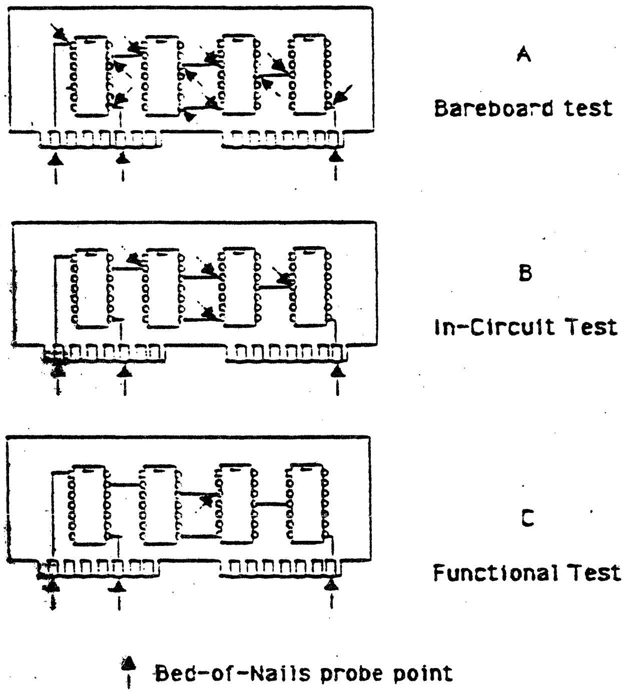
1. Multiple access points/net for bareboard testing to provide shorts and opens test capability. (Figure 1A).

2. One access point per net for in-circuit testing to isolate each component for testing. This is the most predominant commercially available board test method. (Figure 1B).

3. One access point for all significant stimulus and response points for functional testing to minimize programming costs and minimize manual "guided probe" steps for diagnosis. (Figure 1C).

Figure 1

Test Method vs Typical Access Required

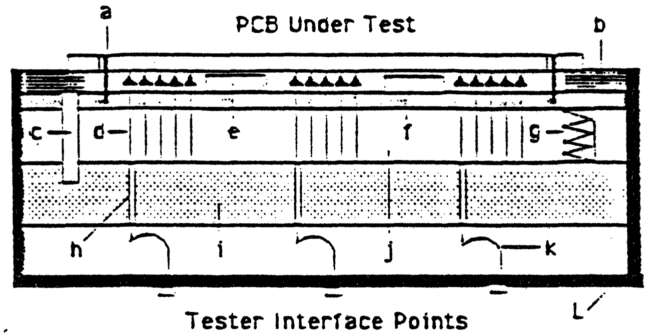


So while the access required varies slightly from method to method, the affect is the same - less access, less test! And, for almost all situations where hundreds of access points are required, the vacuum bed-of-nails (BON) fixture is used (Figure 2). In order to establish a reference point, the following situation will be used as a typical case for the balance of this paper.

- o ACCESS GOAL - one point per net
- o FIXTURING METHOD - Bed-of-nails
- o TEST METHOD - In-Circuit Style

Figure 2

Vacuum Bed-of-Nails Fixture



a	Tooling Hole/Pin
b	Edge Seal Gasket
c	Alignment Pin
d	Bed-of-nails probe
e	Vacuum chamber rubber stop
f	PCB support board
g	Spring
h	Probe socket
i	Socket base board
j	Vacuum chamber
k	Wire wrap connection
L	Fixture frame

DESIGNING FOR BED-OF-NAILS FIXTURES

To begin to develop rules for DFA the BON fixtures and the probes which actually contact the board must be examined. BON fixtures are typically vacuum actuated. That means the edge of the board and any cutouts must be sealed with a foam gasket that protrudes under the PCB under test. Thus, either component pads must not be allowed within a certain distance of the edge and cutouts or it must be insured that other pads are accessible on the net beside those near an edge. Figure 3 shows a very useful tool available on some PCB CAD systems call a "Keep In" or "Keep out" area. These can be used to force either component placement or trace routing/vias in a specified area.

Figure 3

"KEEP OUT" Area for Fixture Sealing

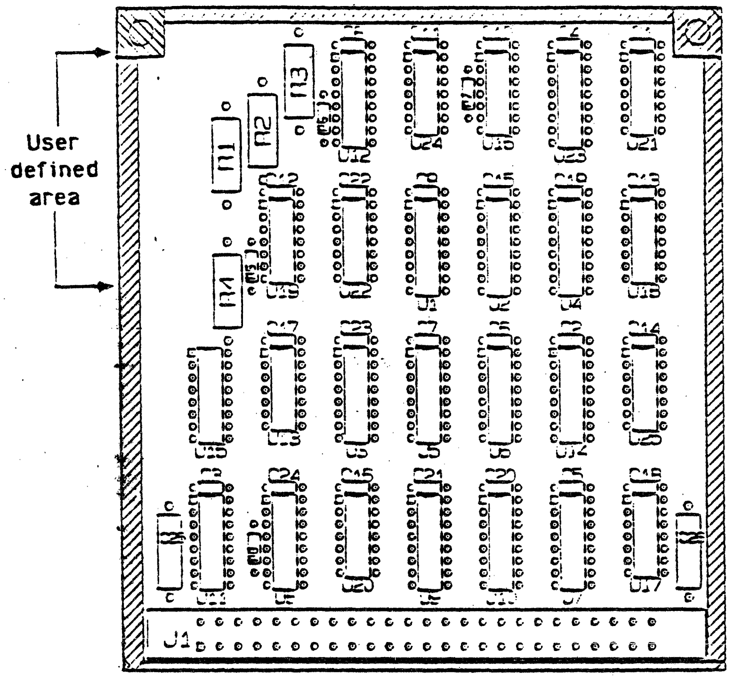


Table 1 lists some of the more important characteristics of the two most predominant probe types, the 100 mil and 50 mil. The 50 mil probe is more fragile due to the decreased barrel diameter and so it is more costly to manufacture and less reliable. The total per point cost includes the probe/socket combination, drilling, loading, and wiring. 100 mil probe sockets may be wire wrapped, while 50 mil are so small that they must be crimped or soldered for connection to the tester interface. Thus, the 100 mil is preferable over the 50 mil. The reduced travel distance of a 50 mil probe means it can accomodate less board warpage and smaller height components protruding from a board. The travel of 50 mil probes is less because their reduced diameter makes it difficult to have a long barrel which doesn't bend under a "side load" condition. The force of the probe determines both how many probes can be clustered in a certain area and still be compressed by the fixture, and how well the probe can break through contaminants to get to the pad or lead. In PCB layout, careful attention must be paid to solder mask design to insure it doesn't cover too much of the target. The target size for each probe type is usually surprising information. The fact that the 50 mil probe may need a larger target than the 100 mil can be attributed to the angle a 50 mil probe can deflect in its socket versus 100 mil, the surface contact area of a 50 mil spear probe versus 100 mil serrated, etc. Lastly, there is the continuing dilemma that 100 mil probes are typically designed for component leads while 50 mil typically require a flat surface like a via or pad to avoid loading. Vias must typically be filled with solder to avoid losing vaccum with the fixture. Recommendations for target size, shape, and spacing can then be derived from this data.

Table 1

Bed-of-Nails Probe Comparision

Characteristic	100 Mil Probe	50 Mil Probe
Price	X	~2X
Reliability	Y	~1/2Y
Probe Travel	250 Mil Typical	125 Mil Typical
Probe Force	8 oz Typ	4 oz Typ
Probe Tip Styles	9/5 point Serrated	Spear or Chisel
Min Size Target	~28 Mil	~35 Mil
Typical Target	Component Lead	Via or Pad

DFA PROCESS AND ACCESS CRITERIA

The process used to select the targets where nails will be placed is critical to the overall effectiveness of DFA. If a schematic is used by the test engineer for picking test points, problems can occur since all spare parts or gates may not be recorded on the schematic. That can create big problems when an ECO comes along that requires that spare part to be used, and the fixture must be modified to access it. The CAD system is where the real physical packaging is done and a more accurate database may exist than at schematic capture (especially if back annotation is not done or incomplete). CAD is the place to do the access planning. Rules can be developed to sort through a database for a board design and select the most appropriate point on the net to access using a variety of criteria. One example criteria would be to assign the access points at the sources of all signals on the board. This point would be the output of all ICs in a digital applications. The reason for this selection criteria is shown in Figure 4. Given the high probability that a design will be ECOed at sometime, with the associated cut and paste of track, an assignment that covers all signal sources virtually assures access at all current and ECOed nets (Figure 4A). Should the assignment be done manually using a random selection or automatically using a simplistic criteria like "the first pin on every net in the database", problems like Figure 4B can occur. Here the ECO has caused a portion of the original net to be without access because an input was randomly selected for the access point. The choice of an available output as the access point insures a net will not be without access due to the ECO (Figure 4C).

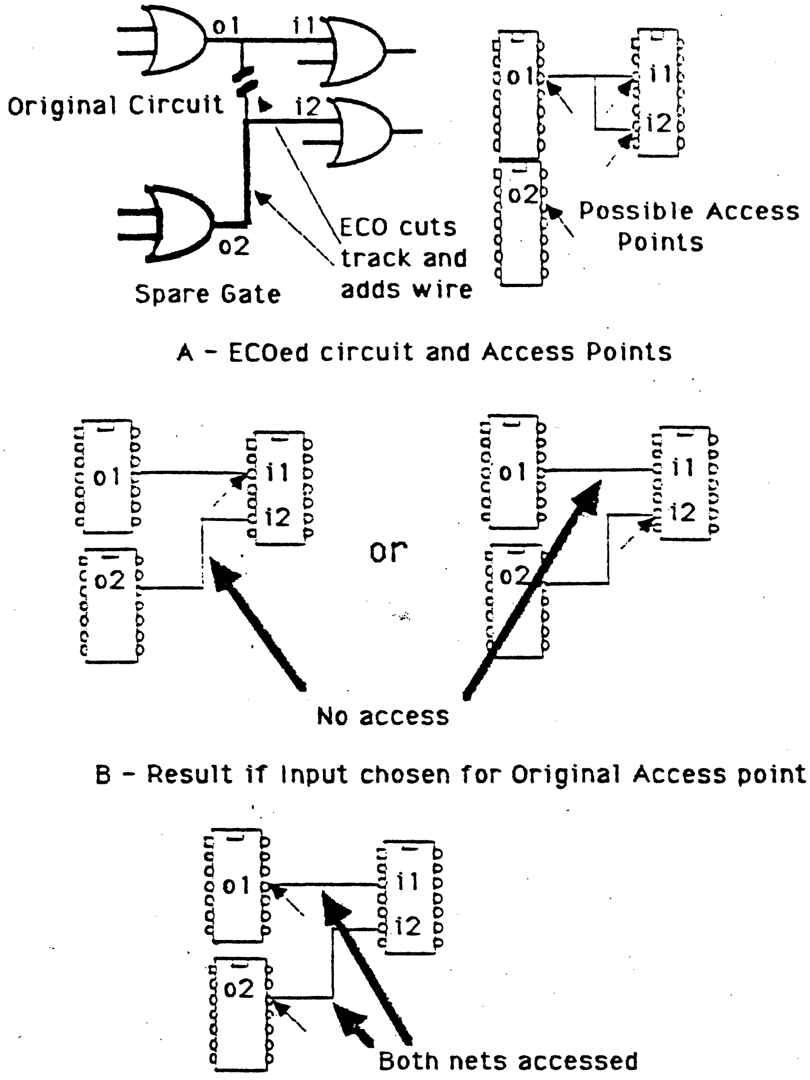


Figure 4

A - ECOed circuit and Access Points

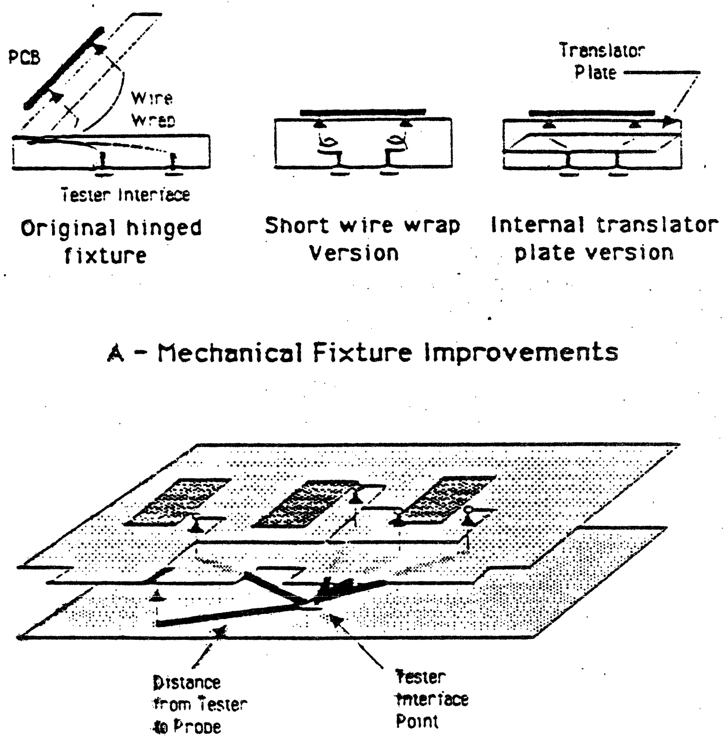
B - Result if Input chosen for Original Access point

C - Result if Output chosen for Original Access point

Another possible criteria could be used in cases where a pin connector is part of a net. If for example, there were a number of input signals to the board and, thus, the "cover all IC outputs" rule stated above didn't apply, we could choose the access point at the input pin of the IC or the connector. The connector may be the preferred location, because it allows the continuity of the track from the connector to the input pin on the IC to be verified at board test rather than possibly waiting for the first time the board is plugged into a system to find an open track. These two examples illustrate how DFA can affect both testability and minimize the effect of an ECO on a test fixture. Dozens of other criteria can be used as appropriate to the combined access goal/fixture method/test method. One criteria that will have increased significance in the future will be DFA that is geographic based. The reason is that the wire length in the fixture is a limiting factor on the speed of the tester. The mechanics of the fixture have been changed (Figure 5A) to improve the wire length somewhat, but the big impact will come from software that selects the access point based on minimum distance to the desired driver/sensor in the tester (Figure 5B).

Figure 5

High Speed Board Test Fixturing



A - Mechanical Fixture Improvements

B - Minimum distance probe assignment

DFA DOCUMENTATION

Accurate, automatically generated documentation may be as important for productivity improvements "downstream" in board test as the appropriate selection criteria. An example is an annotated probe map. This is similar to Figure 3, but with the test point numbers for each probe marked on the plot (Figure 6). There actually may be several tapes/plots used in the fixturing process. The drilling data (Figure 6A) may bed used to drill all holes where a BON probe may ever be needed. Then, using the appropriate selection criteria, a plot which indicates the actual access points chosen and their tester pin assignments can be generated (Figure 6B). This plot can prove extremely useful in three ways:

- Debugging the fixture and verifying its correct drilling/wiring. Remeber there may be several thousand probes and wires in a fixture. Also, when a undebugged program fails, its usually very difficult to decide whether its the tester, the program, the fixture, or the board.

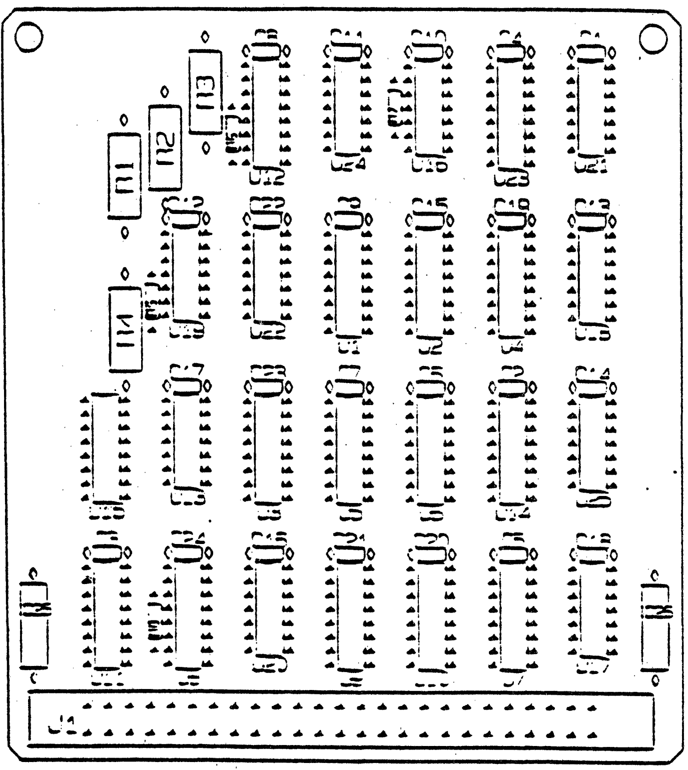
- It provides a cross reference between the fixture's physical connections and the test program generated by and ATG package or simulator. The beginning point for test generation is a netlist and parts list which can also be extracted from the CAD database (insuring good cross referencing between netlist/parts list and fixture data). The automatic netlist/parts list generation for the test programming process can save up to 20% of the overall progra.. prep costs. And, test programming is like PCB layout, an inaccurate/incomplete netlist means the same for the resulting test program.

- As a repair aid, probe maps allow operators to physically locate the tracks identified via tester numbers on diagnostic tickets. Shorts/opens location time is well known to be the most expensive fault locating activity in repair.

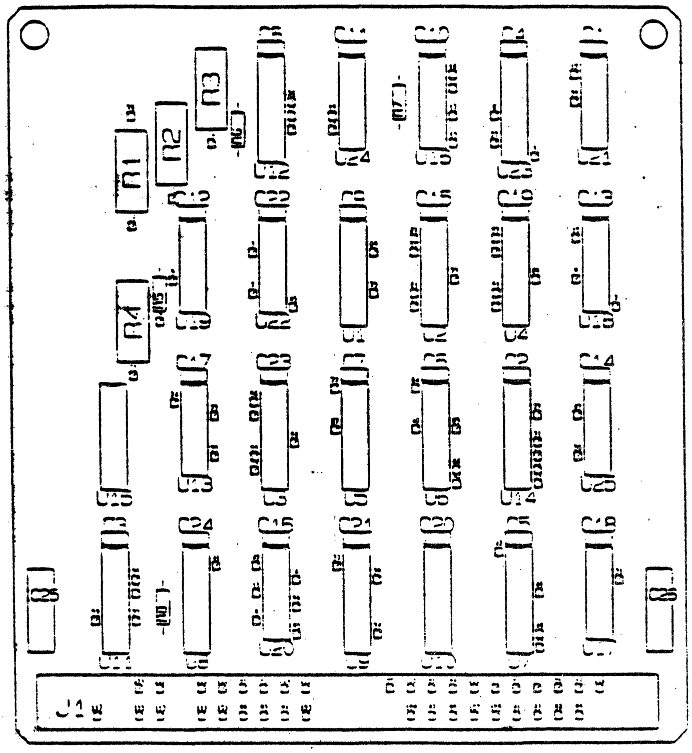
Figure 6

DFA Documentation

A - Drill Drawing



B- Probe Hole Locations/Reference Numbers



DFA FOR SMD DESIGNS

The impact of SMD on our access/fixture/test goals can be dramatic. Some of the physical differences of an SMD design versus their possible impact on accessibility are listed in Table 2. The possibility of having to access a surface mounted board from both the bottom and top side brings with it a myriad of alignment, reliability, and cost problems. Getting two probe plates aligned to opposite sides of a PC board can be very tricky, especially on 50 mil centers. Given a reasonable mix of SMD and some through-hole components, it has been estimated to require about 5-10% more surface area to get every net accessible once from the bottom side of a board with acceptably sized targets. The blind/buried vias, SMD packaging styles, and pad designs all contribute to invisible targets from <u>either</u> side. In through-hole technology, the visibility came for free! The closer spacing means the 50 mil probes with all their idiosyncracies may be needed. The ultimate impacts vary depending on the level of DFA implemented:

Typical Through Hole Access -	Bottom side only/100 mil
Best SMD Access -	Bottom side only /100 mil
Good SMD Access -	Bottom side only/50 mil
Poor SMD Access -	Top/Bottom side access required
-	50 mil
Worst SMD Access -	Limited/no access on top/bottom side
-	50 mil

It's clear that the most important issues are getting the target visible and unobstructed on the outside layers, getting them all on the bottom side if possible, and making them as big as possible.

Table 2

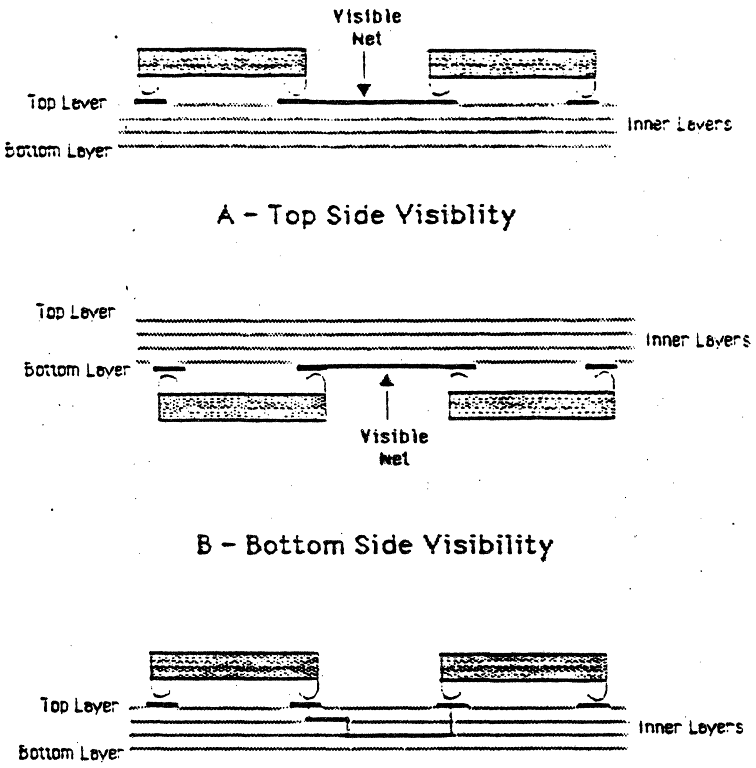
SMD Technology versus Accessability

SMD Attribute	Accessability Impact
Top side surface mount components	Traditional bottom side access not guaranteed. May need to access from top and bottom side.
Blind/Buried Vias	Same as above, also net may not be accessible from either side depending on component packaging and pad design.
J Lead, Leadless, or Gullwing package	Probe target is pad or flat lead (Gullwing) compared to cut/clinched lead for through hole packaging.
Variety of pad designs	More difficult to determine accessability manually compared to through hole design.
Closer Pad Spacing	50 Mil probe now needed or "fanout" pads to 100 Mils. PCB to fixture alignment critical.

The first step is getting the target visible in the first place. If the design uses exclusively SMD components with blind/buried vias, a totally invisible design can easily be created. It may be that the net itself is visible on the top layer (Figure 7A) or bottom layer (Figure 7B), but the pads are hidden by the SMD component packaging. J leaded packages would cause this situation. Another case has no outer layer net visibility because blind vias are used at the ends of the connections to drop to inner layers under the SMD's component pads (Figure 7C).

Figure 7

Net Visibility for SMD



A - Top Side Visiblity

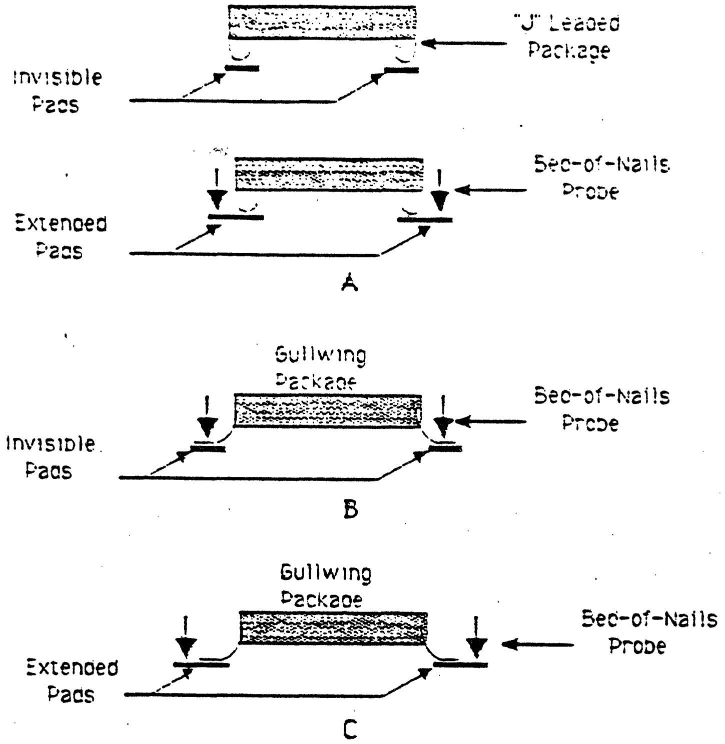
B - Bottom Side Visibility

C - No External Visibility

The solution to this external visibility problem can take several forms. One method is to provide pad designs that extend out from the J lead component packaging the proper distance to accept a probe (Figure 8A). If the package is a gullwing lead design, the lead itself may be the contact (Figure 8B) if probe bending and stress on the lead-to-pad solder joint is deemed acceptable. If not, the pad must be extended beyond the gullwing lead to provide a target pad "extension" (Figure 8C).

Figure 8

Pad Design for Accessability



These "extension" methods are not the only solution. In fact, in the case of a top-sided surface mount component the extended pad gets the net visible on top (and improves access from the worst to poor level) but means a two-sided fixture must still be used. Also, the pad extensions would probably be contained with the package/pad data. Since our access goal is only once per net, it really isn't necessary to extend every pad and suffer the resulting density loss. Another alternative would be to have a target inserted somewhere along a net by DFA software. This could take into account more criteria than simply getting the net visible on an external layer.

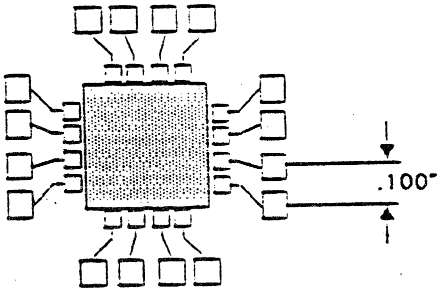
- The DFA software could search for an existing through-hole on a net, assuring access from the bottom and eliminating any pad extensions.

- The DFA software would know if a pad extension were being made on the top or bottom layer to know if only visibility was accomplished or also bottom access.

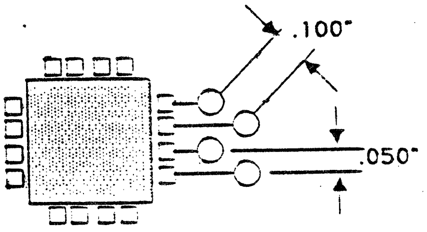
- The DFA software could be cognizant of any routing or design rule impact from the insertion of a probe target. The requirements for size and spacing of component pads, vias, and probe targets may be different. While the physical design rule for line to line/line to pad/and pad to pad spacing may be in the the 6 to 8 mil range, the probe target may be required to have an open area around it that may be in the 18-20 mil range. This is to account for worst-case tolerance buildups that may exist especially in two-sided fixtures that could cause a probe (especially the serrated type) to hit the wrong target.

Figure 9

Pad Fanout Techniques



Direct



Staggered

If, through the use of DFA software, bottom side access can be gained, the remaining way to improve the level of DFA is to provide for 100 mil targets. Pad extensions like those in Figure 8 make a pad accessible, but would typically require a 50 mil probe. Extension techniques to both gain access and achieve 100 mil spacing are shown in Figure 9.

The staggering pad method could provide better use of real estate and have less impact in terms of the number of router channels blocked, if the router has the ability and space to serpentine between the pads. Capabilities like 45° routing and the ability to have the channels between pads preferred for use will minimize the impact of these fanout techniques.

CONCLUSION

DFA or Design for Accessibility is a key component in achieving board level design for testability. All commercially avaialable board test methods are strongly impacted by the accessibility of a design. The large number of factors to consider in physical layout that affect accessibility make it difficult to perform manually. By using the PCB CAD database and appropriate criteria access can be planned, implemented, and documented.

The use of SMD design techniques could have a serious impact on accessibility. To maintain access and not incur significantly increased costs due to the need for both side access or smaller/more expensive probes, DFA for SMD is a must. Pad extensions, fanouts, and inserting probe targets on a net will be the kinds of tools used by DFA software to maintain access.

CV-OUT TELESIS-TO-CADDS/4X COMPATIBILTY

Don Deily

Senior Member, Technical Staff

Telesis Systems Corporation

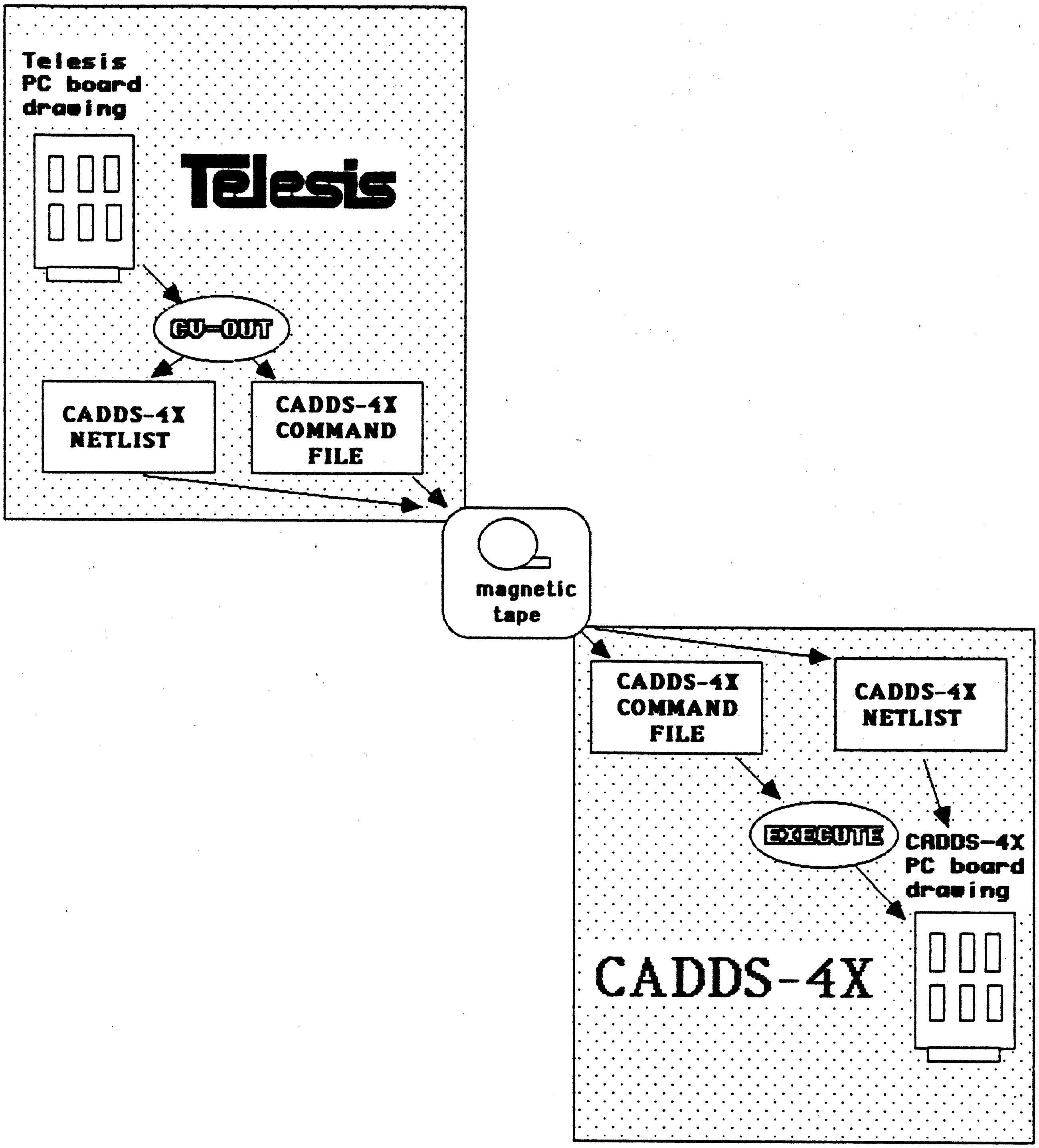
Chelmsford, Massachusetts

ABSTRACT

The following document was reprinted from the Telesis EDA-700 Series Documentation (Part No. 76-00062-001), Telesis Users Manual.

INTRODUCTION

Any Telesis EDA-700 printed circuit board drawing may be converted to a Computervision CADDS-4X drawing by using the Telesis CV-OUT feature. CV-OUT scans the Telesis drawing and writes a text (ASCII) file of CADDS-4X commands that will create a CADDS-4X drawing exactly like the original Telesis drawing. The text file can be transferred to the CADDS-4X system by 9-track tape using standard Telesis and CADDS-4X commands, as described in Section 4 in this document. The data flow is shown in the following diagram:



Telesis to CADDS-4X PCB
drawing conversion
dataflow

CV-OUT

1986

Telesis

HOW TO CREATE CADDS-4X DRAWINGS FROM THE TELESIS EDA-7ØØ

Set Up of Control Files

Before running CV-OUT, there are several things you must first do on the Telesis EDA-7ØØ:

a. When you first build the PC board drawing to be translated to CADDS-4, make sure you use symbols that have exactly the same pin positions and pad geometry on every layer as corresponding CADDS nodal subfigures (symbols). The CADDS command file created for conversion will assume the subfigures are already in CADDS.

b. Create a conversion control text file called CV-CONVERT. This specifies layer correspondence, conversion of Telesis to CADDS characters, net names, symbol-to-subfigure names. See the CV-CONVERT FILE section below.

c. Create (or extract from your CADDS library) a subfigure automation text file CV-SYMDISP. This file tells where the reference designator and device type text nodes are on each library subfigure. See the CV-SYMDISP below. This document also gives a CADDS execute file that you can use to get the required information from your parts library and format it inot a form required by CV-OUT. You can then transfer this file to the Telesis system and it can be available for any number of conversions.

d. Create a preamble CADDS command text file, called CVDRAW-OPEN. CV-OUT automatically puts CVDRAW-OPEN at the beginning of the CADDS command file it builds. CVDRAW-OPEN typically opens a CADDS drawing.

e. Create a postamble CADDS command text file, CV-DRAW CLOSE. CV-OUT automatically puts CV-DRAW CLOSE at the end of the CADDS command file it builds. CV-DRAW typically contains drawing save commands.

INVOKING THE CV-OUT COMMAND

Once you have created these files, you are ready to run CV-OUT.

1. On the Telesis terminal, activate the Telesis drawing that is to be converted. DO NOT operate any commands using the function screen.

2. If your text editing keyboard terminal is attached to a port other than port Ø, you may have to log into the EDA operating system. You can determine what port your keyboard editing terminal is plugged into by reading the number printed to the left of its connector on the Telesis cabinet. Type the following:

   ```
   HEL SYSTEM/SYSTEM
   ```

   The terminal will either respond with a series of log on messages, or "HEL -- Other User Logged On." Either way, you can now run the CADDS conversion program.

3. At the text editing keyboard terminal, type:

   ```
   >@CV
   ```

   (Ignore messages "ABO -- ..., REM -- ... etc.)

4. A menu will appear on the keyboard terminal with the following selections:

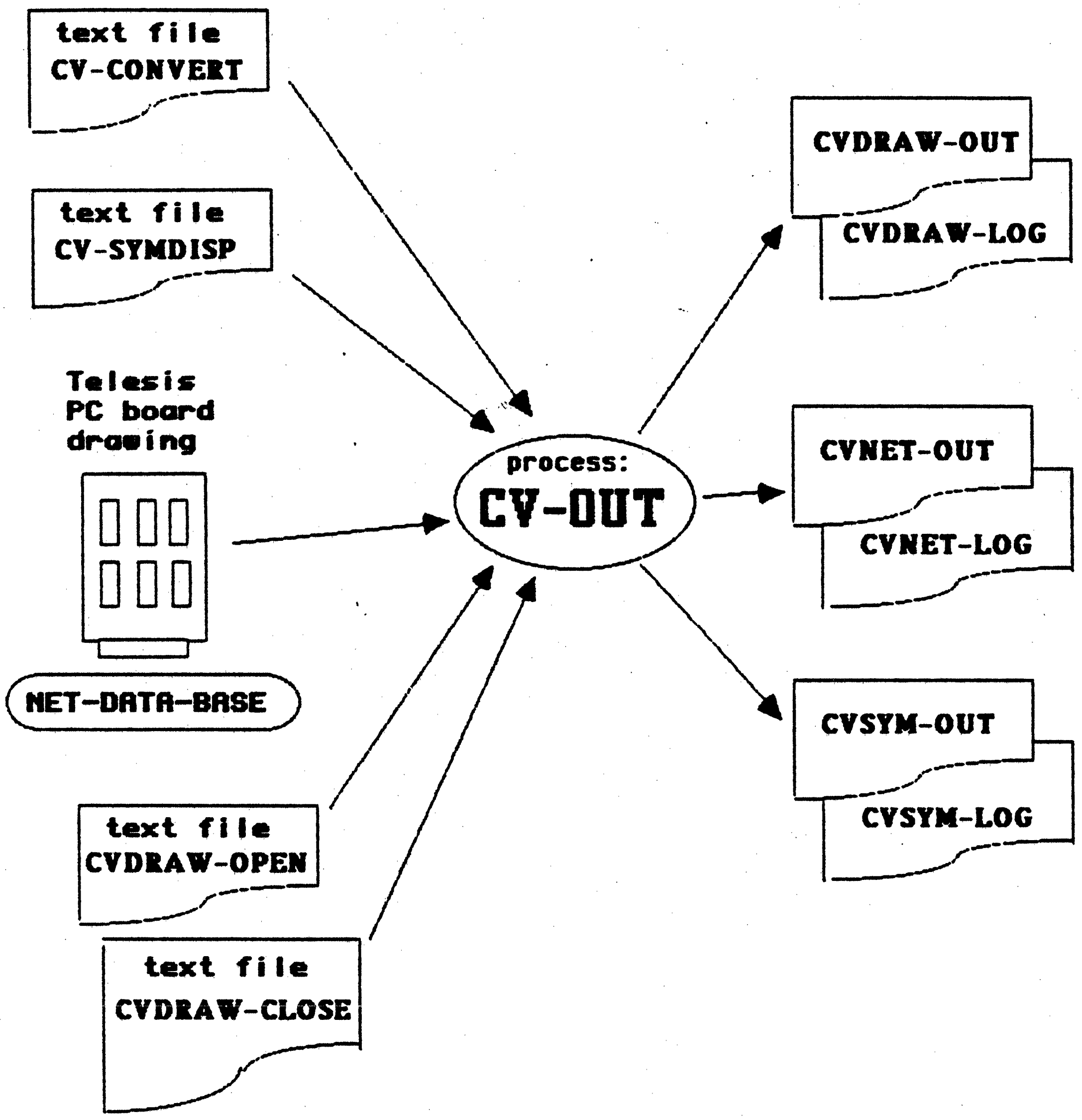
```
Ø. Exit
1. Convert TELESIS to COMPUTERVISION
2. .... (Ignore all other options)
3. .... (Do not use other options)
```

Select 1. This will display the following menu:

```
Ø. Exit
1. Create CV conversion files (netlist/drawing commandfile)
2. Create symbol summary report
```

You should select "Create symbol summary report" first, and print the report created (CVSYM-OUT) to check that the control files are correct, and that they symbol-reference-designator offsets are also correct. Once you have checked the symbol offsets in CV-SYM-OUT, you can run "Create CV conversion files" to create the actual command files for transfer to the CADDS-4X system.

The following diagram shows the data flow of CV-OUT on the Telesis system:



CV-OUT -- DATA FLOW

1986

Telesis

THE CONVERSION PROCESS FLOW

a. The program first reads the text files that determine the header and trailer test for the CADDS-4X execute file (CVDRAW-OPEN, CVDRAW-CLOSE), the conversion of non-CV characters, netnames, and symbol names (CV-CONVERT), and offset of the refdes and device type labels on CV symbols (CV-SYMDISP). Refer to the sections below for a detailed description of these files.

b. The program then writes the file of CADDS-4X commands to create a CADDS drawing equivalent to the Telesis drawing. The first commands place the CV components using INS NFIG (insert nodal figure). Then INS NTXT (insert nodal text) commands annotate the components, inserting reference designators and and device types. The commands place the text at the locations given by the CV-SYMDISP file. The system then writes commands to rotate and mirror the nfigs to their correct positions. Next, it writes INS NLIN commands to add the connecting etch. The program inserts each nlin at the layer specified in the CV-CONVERT file. Finally, the program writes these commands inserting all auxiliary entities: the board outline, drawing format, etc. During the conversion process, the program writes progress and error messages to a file called CVDRAW-LOG. The structure of the CADDS-4X command file is:

```
ASCII file CVDRAW-OUT
|============================================|
| CVDRAW-OPEN (user-defined preamble file)   |
|--------------------------------------------|
| INS NFIG  (using names from CV-CONVERT)     |
| ........................................... |
| INS NTEXT (using offsets from CV-NFIG-TEXT) |
| ........................................... |
| INS STR   (for Telesis shapes and rectangles)|
| ........................................... |
| INS NLIN WDT N    (connecting lines)        |
| ........................................... |
| INS STR  (board outline, drawing format, etc.)|
| ........................................... |
| INS TEXT  (drawing notes, etc.)             |
| ........................................... |
| CVDRAW-CLOSE (user-defined postamble file)  |
|============================================|
```

c. The program then writes the board netlist in CADDS-4X format to the file CVNET-OUT. Since the CV system requires a non-blank name for every net, the program creates CV-compatible net names for all nets with blank net names (on the Telesis side). The CADDS-4X operator reads in this file and merges with the CADDS-4X drawing using the MERGE NET command.

CONVERSION CONTROL FILES

CV-CONVERT File

The CV-CONVERT text file allows you to specify the mapping of the following, from Telesis to CADDS:

a. Individual ASCII characters, since some characters allowed in Telesis symbol names and net names are not allowed in CADDS. (Keyword = CVCHAR).

b. Layers (Keyword = CVLAY).

c. Net names (Keyword = CVNET).

d. Symbol names (Keyword = CVSYM).

e. Universal discrimination layers ofr INS NLIN (Apply to connections on all layers-- Keyword = CVULY).

f. Particular discrimination layers for INS NLIN (Apply to connections on a particular layer -- Keyword = CVCLY).

g. Layer for INS NFIG of the VIA-type symbol in CADDS. (Keyword = CVVIA).

The general grammar of CV-CONVERT is:

```
------------------------------------------------------
| <keyword> <Telesis-value> <CADDS-value> <option>   |
------------------------------------------------------
```

The grammar for each keyword follows. Each of the three (or optionally four) fields -- contents of the angle brackets <> -- is separated from the one before by one or more blanks or tabs.

COMMENTS: Any characters inside paranthesis are ignored as comments.

Keyword CVCHR:

```
----------------------------------------------------
| CVCHR  <Telesis-character> <CADDS-character>     |
----------------------------------------------------
```

This keyword specifies conversion of Telesis characters that are not allowed in CADDS-4X. For example, the slash

(/) is allowed in Telesis symbol names, but not in CADDS-4X symbol names.

Keyword CVLAY:

```
 -----------------------------------------------------
 | CVLAY  <Telesis-layer-number>  <CADDS-layer-number>|
 -----------------------------------------------------
```

This keyword gives the layer correspondence between Telesis and CADDS. Any element found on the given Telesis layer will be inserted into the CADDS database on the corresponding CADDS layer. To suppress the output of a particular Telesis layer to the CADDS-4X file, put a -1 as the CADDS-layer-number.

Keyword CVNET:

```
------------------------------------------------------
| CVNET  "<Telesis-net-name>"  <CADDS-net-name>      |
------------------------------------------------------
```

The Telesis net name must be in quotes if the net name includes any spaces. Otherwise, the program would not be able to parse the CVNET command. If there is not a pair of quotes, the program takes the net name to be the field up to the next space.

Keyword CVSYM:

```
------------------------------------------------------
| CVSYM <Telesis-sym-name>  <CADDS-sym-name>  <code> |
------------------------------------------------------
```

The CVSYM keyword defines a CADDS-4X nfig name for every Telesis symbol name, and optionally, for each allowed rotation (Ø, 9Ø, 18Ø, 27Ø) and mirroring of the Telesis symbol. The codes are shown in the following table:

ROTATION	MIRRORED	CODE
0 degrees	NO	1
90 degrees	NO	2
180 degrees	NO	3
270 degrees	NO	4
0 degrees	YES	5
90 degrees	YES	6
180 degrees	YES	7
270 degrees	YES	8

NOTE 1: If the code field is blank for a symbol name, then the program will use the given CADDS-4X symbol name for all instances, and generate commands to rotate and/or mirror the symbol in the CADDS-4x drawing to the same orientation as the original Telesis symbol. Also, if no entry exists for a symbol name, then the Telesis symbol name will be used in CADDS-4X, and it will be rotated and mirrored as required. NOTE 2: THE CVSYM ENTRIES MUST BE IN ALPHABETICAL ORDER BY TELESIS-SYMBOL-NAME. This allows the look-up process to be very fast. If the names are not in alphabetic order, some of them will not be found during translation. NOTE 3: To suppress conversion of a Telesis symbol, put a "-" as the CADDS-4X name.

Keyword CVULY:

```
------------------------------------------------------------
| CVULY <list of universal connecting CV layers>           |
------------------------------------------------------------
```

This layer will be included in the ECHO lAY <layer-list> layer discrimination command beginning each INS NLIN block. It will be concatenated with the CVCLY layer list of cnodes to be recognized only for this layer. These two layer lists combined will ensure that connections will be made only to cnodes associated with the current etch layer, and any universal layers. This will overcome the fact that the CADDS INS NLIN command will connect to any cnode at any layer, unless the undesired layers have been made invisible.

Keyword CVCLY:

```
----------------------------------------------------------
| CVCLY <CADDS-layer-number>  <CADDS-cnode-layer-list>    |
----------------------------------------------------------
```

This layer list will be included in the ECHO LAY <layer-list> layer discrimination command beginning each INS NLIN block. It will be concatenated with the CVULY layer list of cnodes to be recognized only for this layer. These two layer lists combined ensure that every connection will be made only to cnodes associated with the current etch layer, and any universal layers. This overcomes the fact that the CADDS INS NLIN command will connect to any cnode at any layer, unless the undesired layers have been made invisible.

Keyword CVVIA:

```
-------------------------
| CVVIA <CV-via-layer>  |
-------------------------
```

This specifies the layer to which the Telesis symbol via will be added in CADDS. This command operates independently from CVSYM, which may also cause a translation to a name different from VIA in CADDS.

SAMPLE CV-CONVERT FILE

A sample CV-CONVERT file is shown in the diagram below:

```
|-------------------------------------------------------|
| CVCHR     /          .                                |
|                                                       |
| CVSYM   DIP14     PC.SYM.DIP14.Ø     1                |
| CVSYM   DIP14     PC.SYM.DIP14.9Ø    2                |
| CVSYM   DIP14     PC.SYM.DIP14.18Ø   3                |
| CVSYM   DIP14     PC.SYM.DIP14.27Ø   4                |
| CVSYM   EL-56     -  (SUPPRESS THIS SYMBOL CONV.)     |
|                                                       |
| CVNET  "+5 VOLTS"  SIGØØ5V                            |
| CVNET   GND        SIGGND                             |
|                                                       |
| CVLAY     Ø          1    (CNODES)                    |
| CVLAY     1         12    (COMPONENT-SIDE ETCH)       |
| CVLAY     2         13    (INNER-LAYER-2 ETCH)        |
| CVLAY     3         14    (INNER-LAYER-3 ETCH)        |
| CVLAY     4         19    (SOLDER-SIDE ETCH)          |
| CVLAY    12         42    (COMPONENT-SOLDER-MASK)     |
| CVLAY    13         49    (SOLDER-SOLDER-MASK)        |
| CVLAY    7Ø         8Ø    (ROUTER KEEP-OUTS)          |
| CVLAY   1Ø1         -1    (SUPPRESS RATNEST)          |
| CVLAY   1Ø2         -1    (SUPPRESS DENSTY HISTOGRAM |
|                                                       |
| CVULY  1 2Ø               (UNIVERSAL CONNECT LAYERS) |
|                                                       |
| CVCLY    12          22   (CNODE LAYER FOR COMP-SIDE |
| CVCLY    19          29   (CNODE LAYER FOR SLDR-SIDE |
|                                                       |
| CVVIA    2Ø               (LAYER FOR VIAS)            |
|-------------------------------------------------------|
```

CV-SYMDISP FILE

The general grammar of CV-SYMDISP is:

```
-------------------------------------
| <CV-nodal-figure-name>            |
| <x-offset> <y-offset>  COMPNAME   |
| <x-offset> <y-offset>  DEVICE     |
-------------------------------------
```

a. THE NODAL FIGURE NAMES MUST BE IN ALPHABETICAL ORDER. This allows the conversion program to run faster. If the names are not in alphabetical order, some of them will not be found, and incorrect offsets will be used.

b. The data for each nodal figure must be on three lines, as shown.

c. The COMPNAME and DEVICE lines may be in either order.

d. All fields on one line must be separated by one or more blanks or tabs.

e. The x and y offsets are signed numbers in drawing coordinates giving the offset of that text node from the library symbol origin. The first number pair is taken as the REFDES offset, and the next as the DEVICE -TYPE.

f. If the offsets are missing for any figure name, the conversion program will insert that nodal text at the nfig origin.

SAMPLE CV-SYMDISP FILE

```
--------------------------
| PC.LIB.DIP14           |
|.250 - .050   COMPNAME  |
|.150 - .350   DEVICE    |
| PC.LIB.RES 400         |
| .050   .085  DEVICE    |
|-.075 - .250  COMPNAME  |
--------------------------
```

HOW TO TRANSFER CADDS-4X DATA FROM TELESIS TO CADDS-4X

CADDS FILES TO TAPE (FROM TELESIS) - After you have created the correct CADDS-4X command file and netlist file on the EDA-700, mount a tape with a write ring onto your Telesis tape drive. Pick ARCHIVE_TAPE, OLD_TAPE, ASCII_TO_TAPE. Then pick TEXT_FILE_TO_TAPE twice, entering the names CVDRAW-OUT, and CVNET-OUT. Next, pick ENTER and the system will write the two files to tape in standard ASCII (80 byte record) format.

TAPE TO CADDS-4X (ON CADDS SYSTEM) - Mount the tape on the CADDS-4X system and type the following commands:

```
0>ATTACH MT,TAPE

0>CONVERT :MT TELE.EXEC//CONVERT=A-C,FORMAT=(RECCNT=1,RECSIZE=80)

0>CONVERT :MT TELE.NET//CONVERT=A-C,FORMAT=(RECCNT=1,RECSIZE=80)

0>REWIND

0>DETACH MT
```

EXECUTE THE CADDS-4X COMMAND FILE (ON CADDS-4X)

```
Ø>EXECUTE TELE.EXEC
```

The EXECUTE command builds the Telesis board as described in this document, by adding components (INS_NFIG) and connecting them with the INS_NLIN commands, and adding drawing format an notes, etc., with INS_LINE and INS_TEXT commands. After the creation is complete, the operator can complete the transfer using MERGE_NET, and check it with the COMPARE_NET command.

HOW TO EXTRACT CADDS-4X SYMBOL DATA FOR CV-SYMDISP

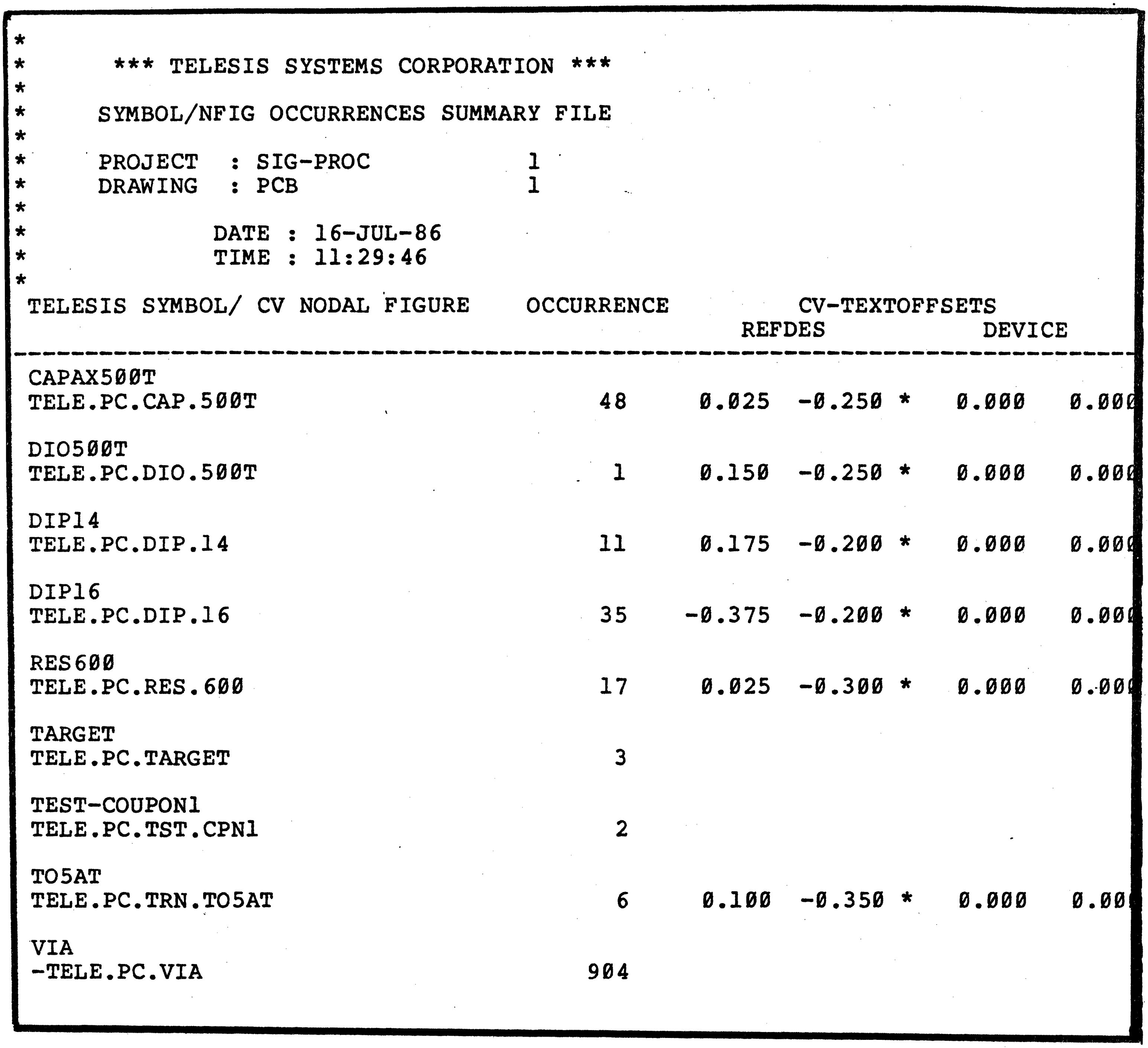
The CADDS-4X execute file (following) will get the X and Y offsets of the refdes and device text nodes by using VERIFY entity, and then create a condensed text file, using TECO commands, for transfer to Telesis. The data extract file can be set up to extract any number of CADDS-4X figures by using CADDS GENCOM macros.

As part of the CV-OUT release package, Telesis supplies a tape of the following CADDS execute file:

```
MOVE TELE.TABLE.&BCD.CONV-LIST TRASH.IT//NLEV,CHECKS=N,REPL=Y
MOVE TELE.TEMP.&BCD.CONV-LIST TRASH.IT//NLEV,CHECKS=N,REPL=Y
GENCOM TELE.TEMP.CONV-LIST<CTRL-U>
HARDFILE TELE.TABLE.CONV-LIST
OK
CADDS<<CR><CR>
ACT PAR $1,-2$
ACT DRA 1
ECH LAY ALL
MARK ENT ON COND
<*
<*    The following line is to be the property name used
<*       on tnod for the REFERENCE DESIGNATOR.
<*
COMPNAME
MARK ENT ON COND
<*
<*    The following line is to be the property name used
<*       on tnod for DEVICE TYPE.
<*    If DEVICE tnod non-existent, insert an asterisk at start of
<*       line numbers 17 and 24
<*
DEVICE
LIST PART STATUS
VER ENT :MARK VWIN NAME 1<<CR><<CR>
EXIT PAR Q<CR>
<*
<*    The following line can be a PART NAME or CATALOG NAME
<*       followed by "/&PD/NLEV"
<*
TELE.PC/&PD/NLEV
<CR>EXIT CADDS
HARDFILE
EDIT TELE.TABLE.CONV-LIST
A TEMP.FILE
OK
Q
TECO TELE.TABLE.CONV-LIST
J<<ESC><<<<S#<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<ESC><<<<S STA<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SX,Y,Z<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<RPART NAME =<<ESC> <<ESC>><<ESC><<ESC>
J<<ESC><<<<SDrawing file not found<<ESC>ØL<<ESC>K<<ESC>><<*
<<ESC><<ESC>
J<<ESC><<<<SCONVERT SOLID<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SAUTOANN<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SENTERING<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<S:<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<STEXT<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SSEQU<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SMARK<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
```

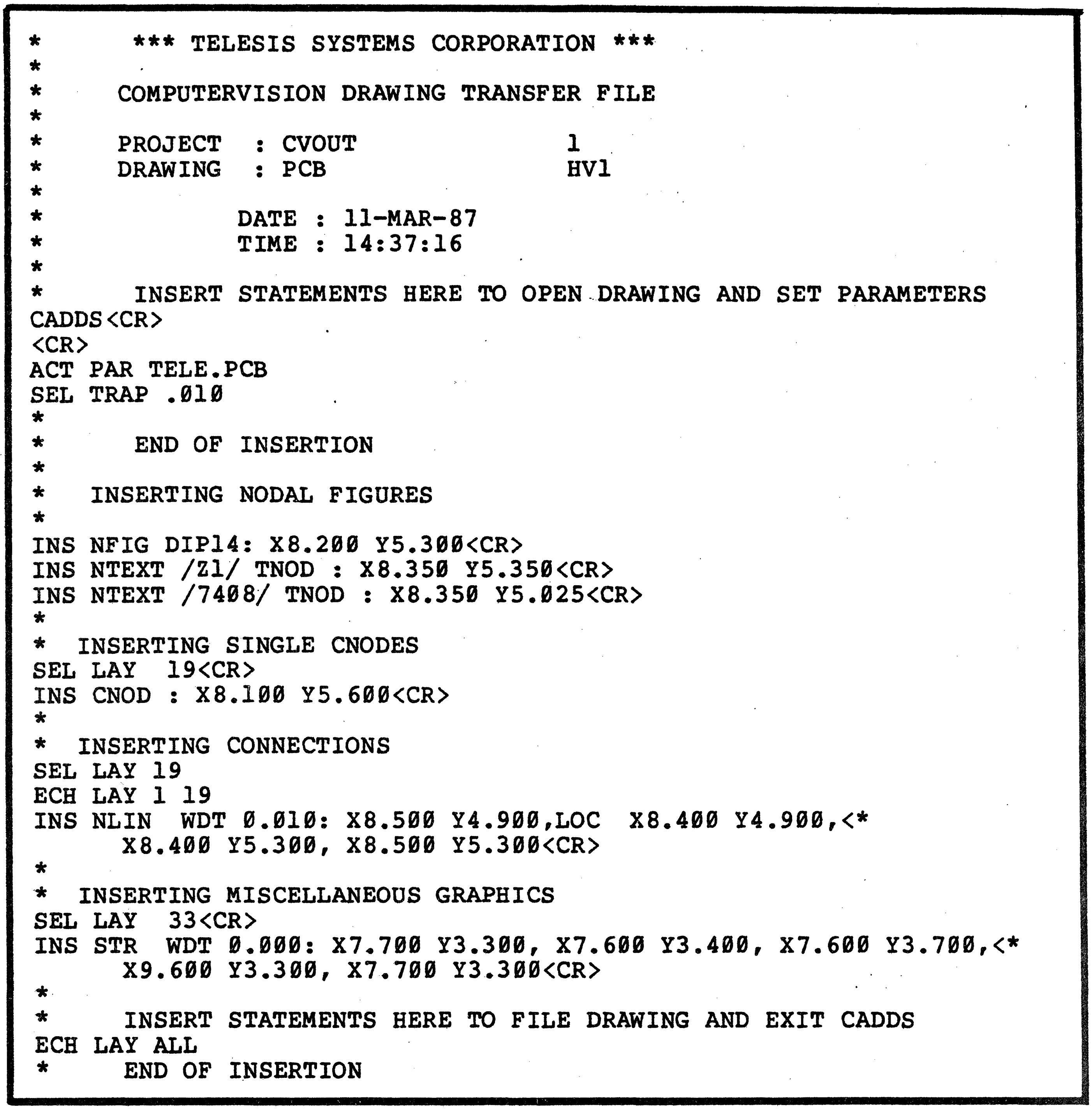
```
J<<ESC><<<<S(<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SENTITIES<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<STELE.TABLE<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SINPUT DEVICE<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SON CON<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<SLAYER<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<R. <<ESC>  <<ESC>><<ESC><<ESC>
J<<ESC><<<<S?<<ESC>ØL<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<S<<CR><<CR><<ESC>-1D<<ESC>-1L<<ESC>><<ESC><<ESC>
J<<ESC><<<<RX =<<ESC> <<ESC>RY =<<ESC> <<ESC><<*
SZ =<<ESC>-3C<<ESC>K<<ESC>><<ESC><<ESC>
J<<ESC><<<<R  <<ESC> <<ESC>-1L<<ESC>><<ESC><<ESC>
EX<<ESC><<ESC><CR>
X
```

SAMPLE CVSYM-OUT (SYMBOL SUMMARY REPORT)



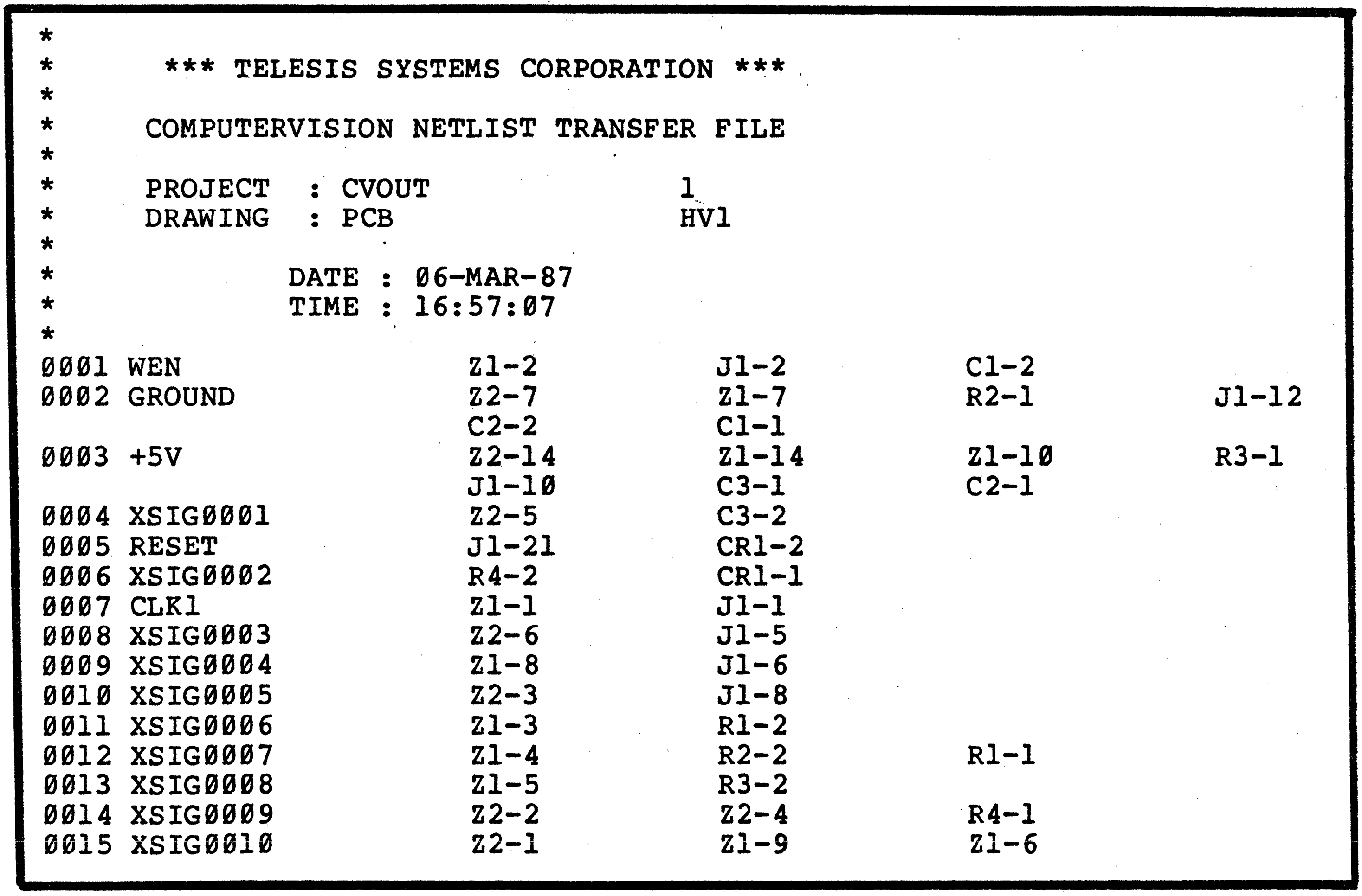
```
*
*      *** TELESIS SYSTEMS CORPORATION ***
*
*     SYMBOL/NFIG OCCURRENCES SUMMARY FILE
*
*     PROJECT  : SIG-PROC           1
*     DRAWING  : PCB                1
*
*             DATE : 16-JUL-86
*             TIME : 11:29:46
*
 TELESIS SYMBOL/ CV NODAL FIGURE    OCCURRENCE          CV-TEXTOFFSETS
                                                    REFDES            DEVICE
--------------------------------------------------------------------------------
 CAPAX5ØØT
 TELE.PC.CAP.5ØØT                            48     Ø.Ø25   -Ø.25Ø *   Ø.ØØØ    Ø.ØØØ

 DIO5ØØT
 TELE.PC.DIO.5ØØT                             1     Ø.15Ø   -Ø.25Ø *   Ø.ØØØ    Ø.ØØØ

 DIP14
 TELE.PC.DIP.14                              11     Ø.175   -Ø.2ØØ *   Ø.ØØØ    Ø.ØØØ

 DIP16
 TELE.PC.DIP.16                              35    -Ø.375   -Ø.2ØØ *   Ø.ØØØ    Ø.ØØØ

 RES6ØØ
 TELE.PC.RES.6ØØ                             17     Ø.Ø25   -Ø.3ØØ *   Ø.ØØØ    Ø.ØØØ

 TARGET
 TELE.PC.TARGET                               3

 TEST-COUPON1
 TELE.PC.TST.CPN1                             2

 TO5AT
 TELE.PC.TRN.TO5AT                            6     Ø.1ØØ   -Ø.35Ø *   Ø.ØØØ    Ø.ØØ

 VIA
-TELE.PC.VIA                                9Ø4
```

SAMPLE CVDRAW-OUT (CADDS-4X COMMAND FILE)



```
*      *** TELESIS SYSTEMS CORPORATION ***
*
*     COMPUTERVISION DRAWING TRANSFER FILE
*
*     PROJECT  : CVOUT               1
*     DRAWING  : PCB                 HV1
*
*             DATE : 11-MAR-87
*             TIME : 14:37:16
*
*      INSERT STATEMENTS HERE TO OPEN DRAWING AND SET PARAMETERS
CADDS<CR>
<CR>
ACT PAR TELE.PCB
SEL TRAP .Ø1Ø
*
*      END OF INSERTION
*
*   INSERTING NODAL FIGURES
*
INS NFIG DIP14: X8.2ØØ Y5.3ØØ<CR>
INS NTEXT /Z1/ TNOD : X8.35Ø Y5.35Ø<CR>
INS NTEXT /74Ø8/ TNOD : X8.35Ø Y5.Ø25<CR>
*
*  INSERTING SINGLE CNODES
SEL LAY  19<CR>
INS CNOD : X8.1ØØ Y5.6ØØ<CR>
*
*  INSERTING CONNECTIONS
SEL LAY 19
ECH LAY 1 19
INS NLIN  WDT Ø.Ø1Ø: X8.5ØØ Y4.9ØØ,LOC  X8.4ØØ Y4.9ØØ,<*
      X8.4ØØ Y5.3ØØ, X8.5ØØ Y5.3ØØ<CR>
*
*  INSERTING MISCELLANEOUS GRAPHICS
SEL LAY  33<CR>
INS STR  WDT Ø.ØØØ: X7.7ØØ Y3.3ØØ, X7.6ØØ Y3.4ØØ, X7.6ØØ Y3.7ØØ,<*
      X9.6ØØ Y3.3ØØ, X7.7ØØ Y3.3ØØ<CR>
*
*     INSERT STATEMENTS HERE TO FILE DRAWING AND EXIT CADDS
ECH LAY ALL
*     END OF INSERTION
```

SAMPLE CVNET-OUT (CADDS-4X NETLIST FILE)



```
*
*      *** TELESIS SYSTEMS CORPORATION ***
*
*     COMPUTERVISION NETLIST TRANSFER FILE
*
*     PROJECT  : CVOUT               1
*     DRAWING  : PCB                 HV1
*
*             DATE : Ø6-MAR-87
*             TIME : 16:57:Ø7
*
ØØØ1 WEN                Z1-2         J1-2         C1-2
ØØØ2 GROUND             Z2-7         Z1-7         R2-1         J1-12
                        C2-2         C1-1
ØØØ3 +5V                Z2-14        Z1-14        Z1-1Ø        R3-1
                        J1-1Ø        C3-1         C2-1
ØØØ4 XSIGØØØ1           Z2-5         C3-2
ØØØ5 RESET              J1-21        CR1-2
ØØØ6 XSIGØØØ2           R4-2         CR1-1
ØØØ7 CLK1               Z1-1         J1-1
ØØØ8 XSIGØØØ3           Z2-6         J1-5
ØØØ9 XSIGØØØ4           Z1-8         J1-6
ØØ1Ø XSIGØØØ5           Z2-3         J1-8
ØØ11 XSIGØØØ6           Z1-3         R1-2
ØØ12 XSIGØØØ7           Z1-4         R2-2         R1-1
ØØ13 XSIGØØØ8           Z1-5         R3-2
ØØ14 XSIGØØØ9           Z2-2         Z2-4         R4-1
ØØ15 XSIGØØ1Ø           Z2-1         Z1-9         Z1-6
```

TELESIS<--->CADDS-4X EQUIVALENCIES

The table below gives a general correspondence between Telesis and CV CADDS-4X. However, particular functions may be accomplished in different ways on the two systems, or, have no counterpart.

TELESIS ELEMENT	CADDS-4X ENTITY
SYMBOL (with connect points)	NODAL FIGURE
SYMBOL (without connect pts)	SUBFIGURE
CONNECT POINT	CONNECT NODE
TEXT POINT	TEXT NODE
TEXT ON A TEXT POINT	NODAL TEXT (if part of nfig)
TEXT ON A TEXT POINT	TEXT (if stand-alone)
CONNECT LINE	NODAL LINE
LINE	STRING
RECTANGLE	STRING (rectangular, closed)
ARC/CIRCLE	ARC/CIRCLE
SHAPE	STRING (closed)
SHAPE-VOID	(no corresponding entity)

GLOSSARY OF CONVERSION RULES

The following table lists how the various Telesis features convert to CADDS-4X. The list is in alphabetical order.

BURIED VIAS: are handled correctly with layer discrimination in INS NLIN.

CARD OUTLINE, PLATING BAR: are lines and arcs at a width in Telesis. They will translate as strings and arcs in CADDS.

COMMAND LINE COMPLETION: CV-OUT adds the four characters <CR> at the end of each CADDS command. This ensures that every command will reach completion, since in some cases (for example, INS NTEXT) it may still be expecting user input.

CONNECT LAYER DISCRIMINATION: CADDS INS NLIN (add connection) doesn't automatically discriminate cnodes by layer. That is, it may connect to ANY cnode at a location, regardless of the cline layer. This causes a problem as nodal lines that are directly over each other (for example, going to connector fingers) may connect to an incorrect cnode on another layer. The equivalent Telesis command will only connect to cnodes on the layer of the cline, or to a through-hole if one exists.

Telesis CV-OUT overcomes this CADDS limitation by preceding INS NLIN commands for a particular layer with the command ECHO LAY <layer-list> where the layers in the list are only the layers allowed for connection to etch on that layer. All other layers will then be invisible, assuring reliable connection to the desired cnodes.

DEVICE TYPES: are text points attached to symbols in Telesis. The text converts to CADDS as INS NTXT onto the corresponding tnode with property DEVICE.

DRAWING FORMAT: will convert as strings, arcs, and standalone text nodes.

DRC ERROR MARKERS: do not convert these. Instead mark layers 131-144 as no-convert. CADDS has its own DRC methods and markers.

IMPLICIT POWER AND GROUND PLANES: Telesis has no explicit database representation of thermal-relief/anti-pads on component pins, since our CREATE PHOTOPLOT writes the correct flash automatically, according to the net name. CADDS has no way to automatically set the correct pad. CV users may either create the photoplot output using Telesis' automatic feature, or use their normal method on CV.

NO-ROUTE NETS: will be reported in CVDRAW-LOG. There is no equivalent CADDS data.

PADS (OFFSET): if they are used in Telesis, they must be reflected in the equivalent CADDS library padstacks. (Apparently, this is a recent CADDS feature).

PIN NUMBERS: are non-graphic properties of component pins in Telesis. They do not convert, but there must be a corresponding cnode with the same pin number in the CADDS nfig.

PLACEMENT, PINSWAP, AND FUNCTION SWAP DATA: CADDS handles placement and swapping in a special pre-routing process, using its own library information. Therefore, the Telesis on-line swapping information will not convert.

RATSNEST (LINES AND HISTOGRAM): CADDS has its own ratsnest methods, so don't convert the ratsnest lines. Simply mark the Telesis layers 1Ø1 (lines), and 1Ø2 (histogram) as no-convert.

RECTANGLES: Telesis and CADDS generally treat rectangles in the same manner; if the photoplot program sees them, it fills them; if the router sees them, it considers them as keepins or keepouts, according to their layer; if pen-plot sees them, it plots their outline. All rectangles will convert to INS STR and should behave correctly, as long as they are mapped to the correct CADDS layer.

REFERENCE DESIGNATORS: are text points attached to symbols in Telesis. The text converts to CADDS as INS NTXT onto the corresponding tnode with the property COMPNAME.

ROUTE WIDTH FOR NETS: will be reported in CVDRAW-LOG. There is no equivalent CADDS data.

ROUTER AND PLACEMENT KEEPINS & KEEPOUTS: (in board and as part of the symbols) convert as INS STR in corresponding layer in CADDS. CADDS' router will respond correctly, as long as they are on the correct layer. **SHAPES:** Telesis shapes allow voids -- CADDS does not. CADDS will photoplot-fill any closed string, however, it will also overwrite voids. The solution is to convert the boundary of a shape as INS STR. If there are any voids, write an error message to CVDRAW-LOG, and as a comment to CVDRAW-OUT. Then continue conversion.

SILKSCREEN: consists of the users' choice of lines, text, arcs, etc. These convert to the corresponding CADDS strings, text, and arcs at the designated layer(s).

SOLDER MASK: The solder mask pads are simply another set of padsizes in both Telesis and CADDS. Other solder mask graphics will convert to the corresponding CADDS layer as strings, arcs, etc.

SURFACE-MOUNTS: CADDS has no special provisions for surface mounts. Our output will build CADDS boards correctly when surface mounts are present, since we add explicit layer discrimination before INS NLIN.

T-CONNECTIONS: automatically convert to the corresponding CV layer. No user input should be required.

VIA KEEPOUTS: are rectangles in Telesis and will convert to closed strings in CADDS.

CAD

POST PROCESSING

PROCEDURE

FINCOR

INCOM INTERNATIONAL INC

INCOM

PREPARED BY D. KUHN DATE: 4-18-86

APPROVED BY R. L. Zeller DATE: 4/23/86

The following is a procedure to be used during the post processing stages of a printed wiring board. That is, the stage when the board has been approved by both the engineer in charge and the UL coordinator. Once the approval has been given, the board can be processed in one of two ways. The artworks can be made by using the FILLED PENPLOT command or the film-arts can be copied to tape using the GERBER TO TAPE command and sent out to a photoplotter.

1. Using FILLED PENPLOT command
This command simply allows the filling of various line widths and pad shapes in the PWB design. The filled penplots are to be inked on a sheet of frosted mylar of an appropriate size at a scale of 2:1. If the board is too large to be plotted at this scale and twelve (12) mil lines are used, then the board needs to be sent out to the photoplotter. On the other hand, if the board is to large to be plotted at 2:1 scale and the smallest line width is twenty-five (25) mils, it is acceptable to plot the board at 1:1 scale.

After picking FILLED PENPLOT, the system prompts you for an artwork file name (-ART) and the scale you wish to plot it at. Next, a pen width must be chosen for each pen used in the plotter. A list of pen sizes and widths which can be used are as follows:

PEN SIZE	WIDTH (MILS)
0	12
1	19
2	23
2½	27
4	46

The pen sizes needed for the fill penplot depends on the scale used. If a board is being plotted at 2:1 scale using 12 mil lines, then the smallest pen needed would be a number two because a 12 mil line plotted at 2:1 scale would be a 24 mil line. Normally, for a board plotted at 2:1 scale the only pens needed for the fill penplots would be pen size two and four.

When penplotting use Rapidraw 3084-F type of ink. This ink is 98% black and works well on mylar.

A recommended pen to use when pen plotting is the Kor-i-noors VTB series. The difference between the VTB and the TB is the two notches that are cut in a crossing pattern on the writing tip of the VTB. These notches give the pen the ability to travel at a higher rate of speed while maintaining an accurate line width. The nature of the grooves causes sharp edges which pick up paper fibers, thus the VTB shouldn't be used on paper. The pen life of the VTB is slightly less than the TB series because of the grooves. It is not recommended to use on fill penplot anything larger than #2 1/2 VTB. So when using a #4, use the 4TB, not 4VTB. The use of pens smaller than a #1 in 2:1 drawings is not recommended. When plotting at 1:1 scale, the use of smaller pens seems to cause less problems.

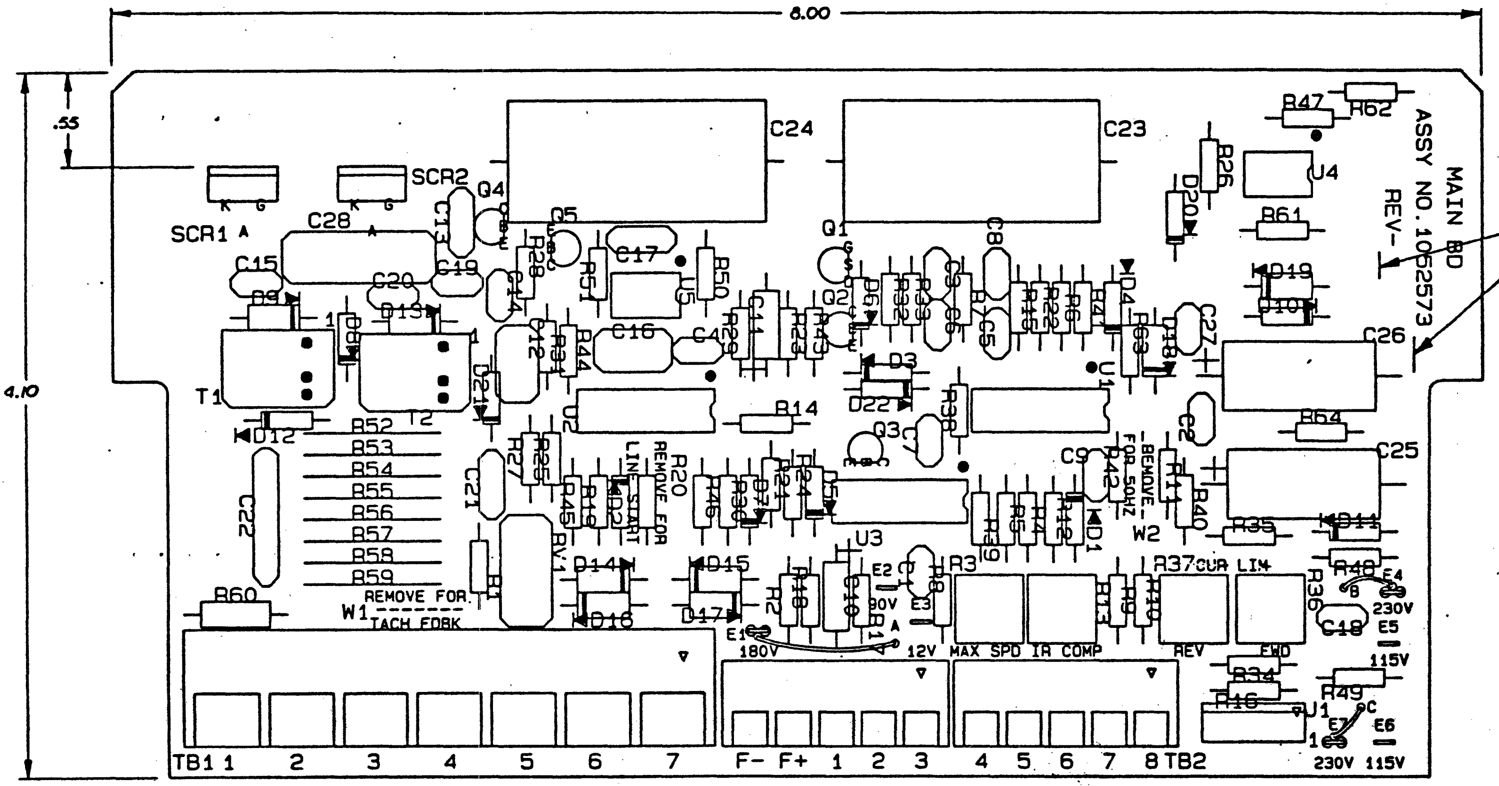
Once the fill penplots are complete, they are burned onto Fincor format using the NuArc plate burner. The documentation of the raw board is similar to that described in the Fincor PRINTED WIRING BOARD DRAFTING STANDARD (A1030927 REV A) with one exception. Instead of burning sheet 1 (drill & trim) onto mylar, the CAD generated NC DRILL (layers 98 & 99) will be plotted using the HP plotter onto Fincor formatted vellum using the .3mm disposable ink pen at 1:1 scale. The assembly drawing will vary slightly in that all CAD generated boards will be a two sheet drawing as opposed to one. The first sheet will show everything as listed in the Fincor PRINTED WIRING BOARD DRAFTING STANDARD (A1030927 Rev. A) with the exception of the item numbers (See Fig. 1). The second sheet will simply show the board outline (layer 33), the package symbols, all item numbers and the title of the board (layer 89). No board dimensions or reference designations are necessary on sheet 2 (See Fig. 2). Both sheets of the assembly drawing will be plotted on the HP plotter using Fincor formatted vellum and the .3 mm disposable ink pen.

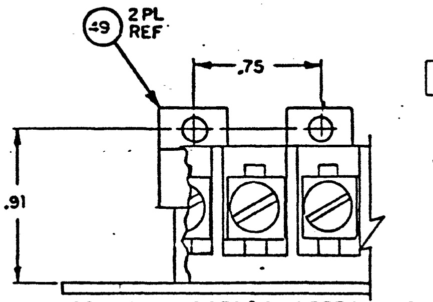
For more information on FILLED PENPLOT please reference the Telesis User Manual Volume 1, Post Processing, pages 55-58.

2. Using GERBER TO TAPE command
This command allows the transferring of artwork files, which are created through the CREATE PHOTOPLOT command, to tape which is used as a direct input to a photoplotter. Simply pick GERBER TO TAPE, ARTFILE TO TAPE, and then input the artfile to be copied. Repeat this procedure for all files needed. Once the artwork files are copied to tape, it is recommended that the files be transferred back to the system so they can be checked by using the PENPLOT ARTWORK command. The system copies the file or files back to the current project and gives each file a name based on file's sequence on the tape. For example, the first file on the tape will be named "TAPE1-ART" in the current project, the second file will be named "TAPE2-ART", etc.

When a tape is sent to a photoplotter, a copy of the APERATURE-TAB file and a copy of the PHOTOPLOT-LOG must accompany it. The PHOTOPLOT-LOG is needed so the vendor has a list of all the used aperatures. Also a photoplot spec sheet must be sent along with the tape (See Fig. 3). Simply take the tape and files to the shipping area and give to the supervisor. He will decide how to package the material and how it should be sent. Once the photoplot process is complete, the vendor will return photo positives. These positives are used in the documentation of the board. Sheet 1 (drill & trim) will be plotted on the HP plotter using Fincor formatted vellum and the .3mm disposable ink pen at 1:1 scale (See Fig. 4). The remaining sheets will be produced by burning the photo positives onto mylar by using the NuArc plate burner (See Fig. 5 & 6).

For additional information on the GERBER TO TAPE process reference the Telesis User Manual Volume 1, Post Processing, pages 53-54.





POSITION SCR1&2 APPROX AS SHOWN PRIOR TO SOLDERING

NOTES :

1. NUMBERS INSIDE OR ADJACENT TO COMPONENTS INDICATE ITEM NUMBERS.
2. MARK GROUP NUMBER AND REVISION LEVEL IN AREA SHOWN, DATE CODE. USE NON-CONDUCTIVE EPOXY INK.
3. ALL DIMENSIONS ARE FOR REF. ONLY.
4. THIS ASSEMBLY CONTAINS STATIC SENSITIVE "MOS" COMPONENTS. THESE COMPONENTS CAN BE IDENTIFIED BY "MOS" SUFFIX ON IID NUMBER. THEY MUST BE INSERTED BY A GROUNDED OPERATOR AT A GROUNDED ANTI-STATIC WORKSTATION. AFTER INSERTION OF "MOS" COMPONENTS, ASSEMBLY MUST BE PUT IN AN ANTI-STATIC CONTAINER FOR TRANSPORT BETWEEN OPERATIONS OR FOR FINAL SHIPMENT.

SHOP FILE □04 ■40 □59 ☒34 □51 □

REV.	DESCRIPTION	REVISED BY	DATE
B	ECN EC14855 (ML ONLY)	DEK	4-18-86
A	ECN EC14838	PLM	2-14-86

REVISIONS

THIS PRINT AND ITS CONTENTS ARE THE PROPERTY OF INCOM INTERNATIONAL INC. AND THEY ARE LOANED TO YOU SUBJECT TO RETURN UPON INCOM INTERNATIONAL INC'S DEMAND AND ARE TO BE MAINTAINED IN CONFIDENCE BY YOU AND ARE NOT TO BE USED IN ANY WAY DIRECTLY OR INDIRECTLY DETRIMENTAL TO INCOM INTERNATIONAL INC'S INTERESTS

UNLESS OTHERWISE SPECIFIED — ANGLES ± 1/2° 2 PLACE DEC. ± .03. 3 PLACE DEC. ± .016 COMMERCIAL TOLERANCES SHALL APPLY TO SIZES OF BAR, ROD, WIRE, SHEET, TUBE, ETC. PLATED PARTS MUST FIT GAUGES AND MEET SPECIFIED TOLERANCES AFTER PLATING.

MATERIAL SEE ML

DRAWN BY / DATE	RMH 9-18-85
CHECKED BY / DATE	DEK 9-19-85
APPROVED BY / DATE	JRP 9-20-85
MODEL NO.	2255
NEXT ASSY.	1052571
SCALE	2/1

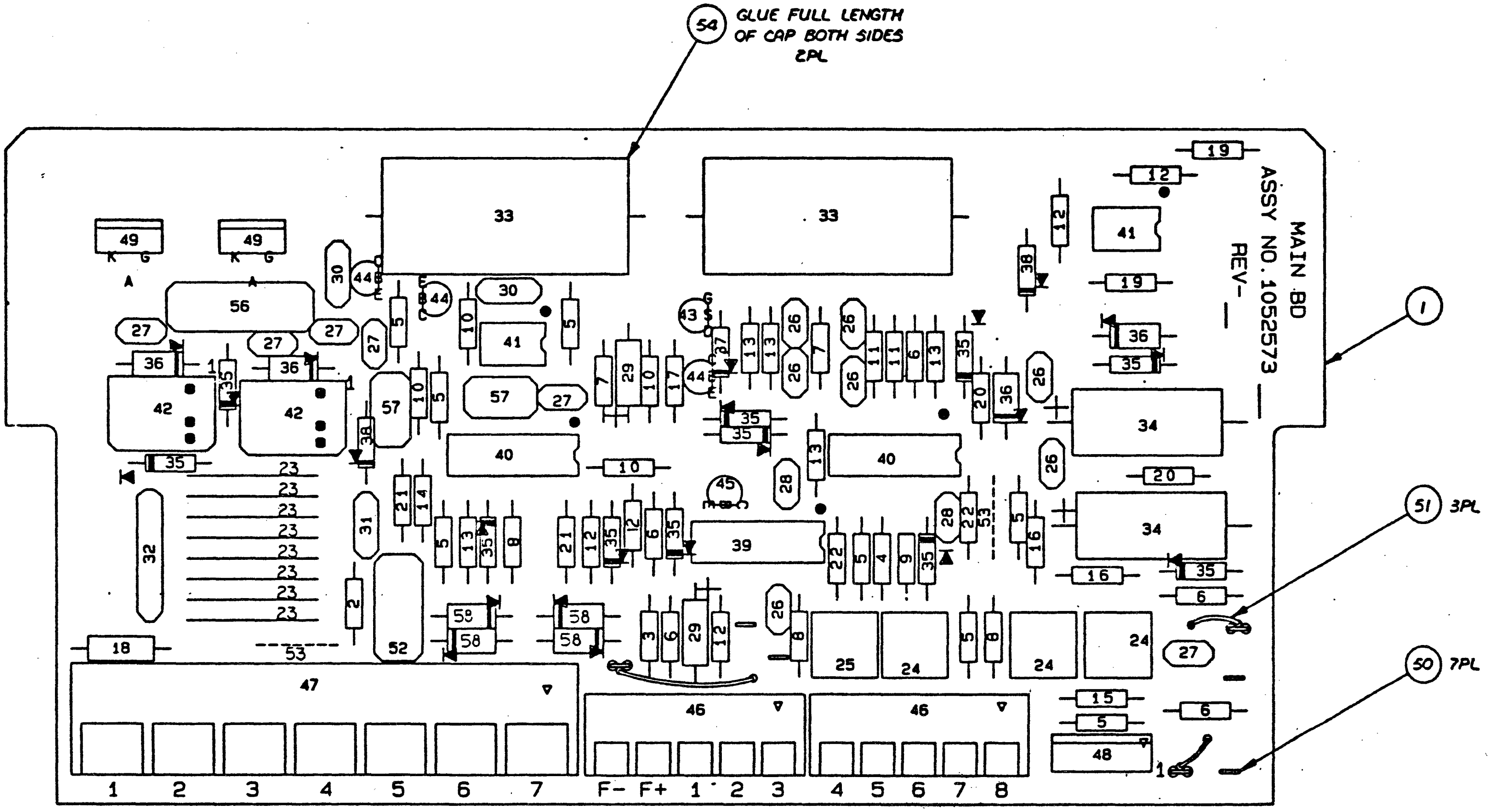
FINCOR
INCOM INTERNATIONAL INC
INCOM

3750 East Market Street
York, Pennsylvania 17402
(717) 757-4641

PRINTED WIRING ASSY
MAIN BD

SHT. 1 OF 2 DWG SIZE C 1052573 REV. B

FIG. 1



SHOP FILE □04 ☒40 □39 ☒34 □51 □

3750 East Market Street
York, Pennsylvania 17402
(717) 757-4641

FINCOR
INCOM INTERNATIONAL INC
INCOM

PRINTED WIRING ASSY
MAIN BD

SHT. 2 OF 2 | DWG. SIZE C | 1052573 | REV. B

FIG. 2

TRIANGLE CIRCUITS
of PITTSBURGH, INC.
931 Third Street
Oakmont, PA 15139
(412) 828-5322

FIG. 3

PHOTOPLOT SPEC SHEET

INPUT CODE: ☑ASCII ☐ISOASCII ☐BCD
☐EBCIDIC ☐EIA

DENSITY(BPI) ☐800 ☑1600

ZERO SUPPRESSION:☐ NONE ☐ LEADING ZEROS☐ TRAILING ZEROS

FORMAT: INTEGER DIGITS =_______ FRACTIONAL DIGITS =_______

DATA: ☐ ABSOLUTE ☐ INCREMENTAL

UNITS: ☐ ENGLISH ☐ METRIC

SCALE FACTOR: _______:_________

BOARD SIZE: X =_______Y =_________

OFFSETS(if any)X =_______Y =_________

TOTAL NUMBER OF FILES ON TAPE: 5

FILES TO BE MIRRORED: F#___ F#___ F#___ F#___

☐ NONE F#___ F#___ F#___ F#___

ADDITIONAL COPIES REQUIRED: ☐ ______POSITIVES

☐ ______NEGATIVES

DELIVERY: ☐ NORMAL ☐ EXPRESS

N/C TAPE MADE: ☐ YES ☐ NO

ADDITIONAL NOTES:_______________________________

APERTURE DESIRED

STN #	SIZE	TYPE	"D" CODE
1			10
2			11
3			12
4			13
5			14
6			15
7			16
8			17
9			18
10			19
11			70
12			71
13			20
14			21
15			22
16			23
17			24
18			25
19			26
20			27
21			28
22			29
23			72
24			73

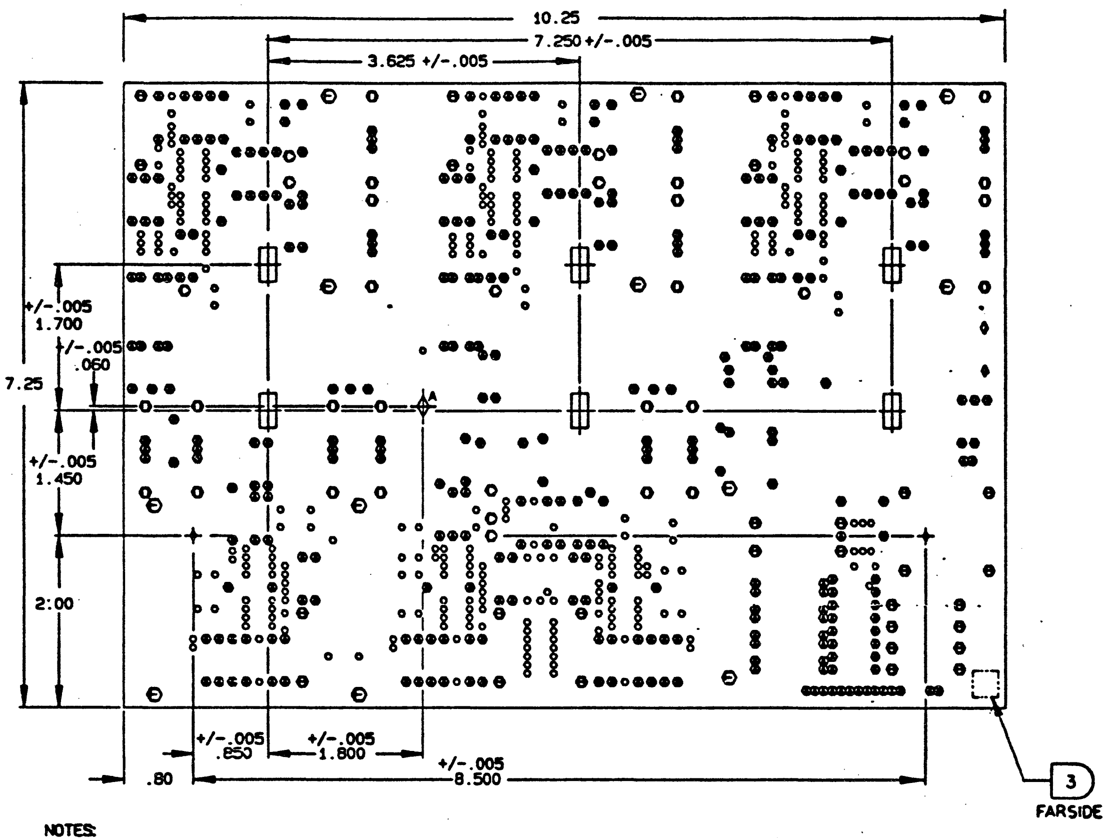
CUSTOMER FINCOR

ADDRESS 3750 E. MARKET ST
YORK, PA 17402

CONTACT BOB RUTH

PHONE (717) 757-4641

DATE 4-13-86



NOTES:

1. MATERIAL-LAMINATED FIBER-GLASS FLAME RETARDENT EPCXY RESIN FR-4 .062 +/-.005 THK 1OZ .0014THK COPPER CLAD TWO SIDES COLOR NATURAL MATERIAL TO CONFORM TO MIL-P-13949 FL-GFN .062C2/2A2A OR EQUAL .

2. UNLESS OTHERWISE SPECIFIED ON THIS DRAWING FABRICATION OF THIS PRINTING WIRING BOARD TO BE IN ACCORDANCE WITH FINCOR SPECIFICATION A1038301, TYPE II, CLASS 2.

3. THIS BOARD IS TO BE SUBMITTED TO UNDERWRITERS LABORATORIES FOR RECOGNITION AND MUST BE MANUFACTURED BY A UL APPROVED VENDOR. VENDOR TO MARK BOARD APPROX WHERE SHOWN WITH HIS UL TYPE/CODE DESIGNATION.

4. APPLY SOLDER-MASK HYSOL PC401 OR EQUAL BOTH SIDES .

/FIGURE/	HOLESIZE/	QTY/
◊A	.218-P	1
◊	.156-N	4
▯	* .218X.375P	6
⊖	.096-P	12
◐	.073-P	12
⊙	.086-P	24
⊖	.055-P	28
○	.033-P	276
⊕	.046-P	351

ALL SIZES AFTER PLATING
* ROUND END SLOT

REV.	DESCRIPTION	REVISED BY	DATE
A	ECN EC14788	RMH	7-8-85

REVISIONS

THIS PRINT AND ITS CONTENTS ARE THE PROPERTY OF INCOM INTERNATIONAL INC. AND THEY ARE LOANED TO YOU SUBJECT TO RETURN UPON INCOM INTERNATIONAL INC.'S DEMAND AND ARE TO BE MAINTAINED IN CONFIDENCE BY YOU AND ARE NOT TO BE USED IN ANY WAY DIRECTLY OR INDIRECTLY DETRIMENTAL TO INCOM INTERNATIONAL INC.'S INTERESTS.

UNLESS OTHERWISE SPECIFIED ANGLES ± 1/2° 2 PLACE DEC. ± .01. 3 PLACE DEC. ± .016 COMMERCIAL TOLERANCES SHALL APPLY TO SIZES OF BAR, ROD, WIRE, SHEET, TUBE, ETC. PLATED PARTS MUST FIT GAUGES AND MEET SPECIFIED TOLERANCES AFTER PLATING.

MATERIAL 1

DRAWN BY N.RUTH DATE 5/8/85
CHECKED BY DATE 5-9-85
APPROVED BY DATE 5/9/85
MODEL NO 6130
NEXT ASSY 1051825
SCALE 1/1

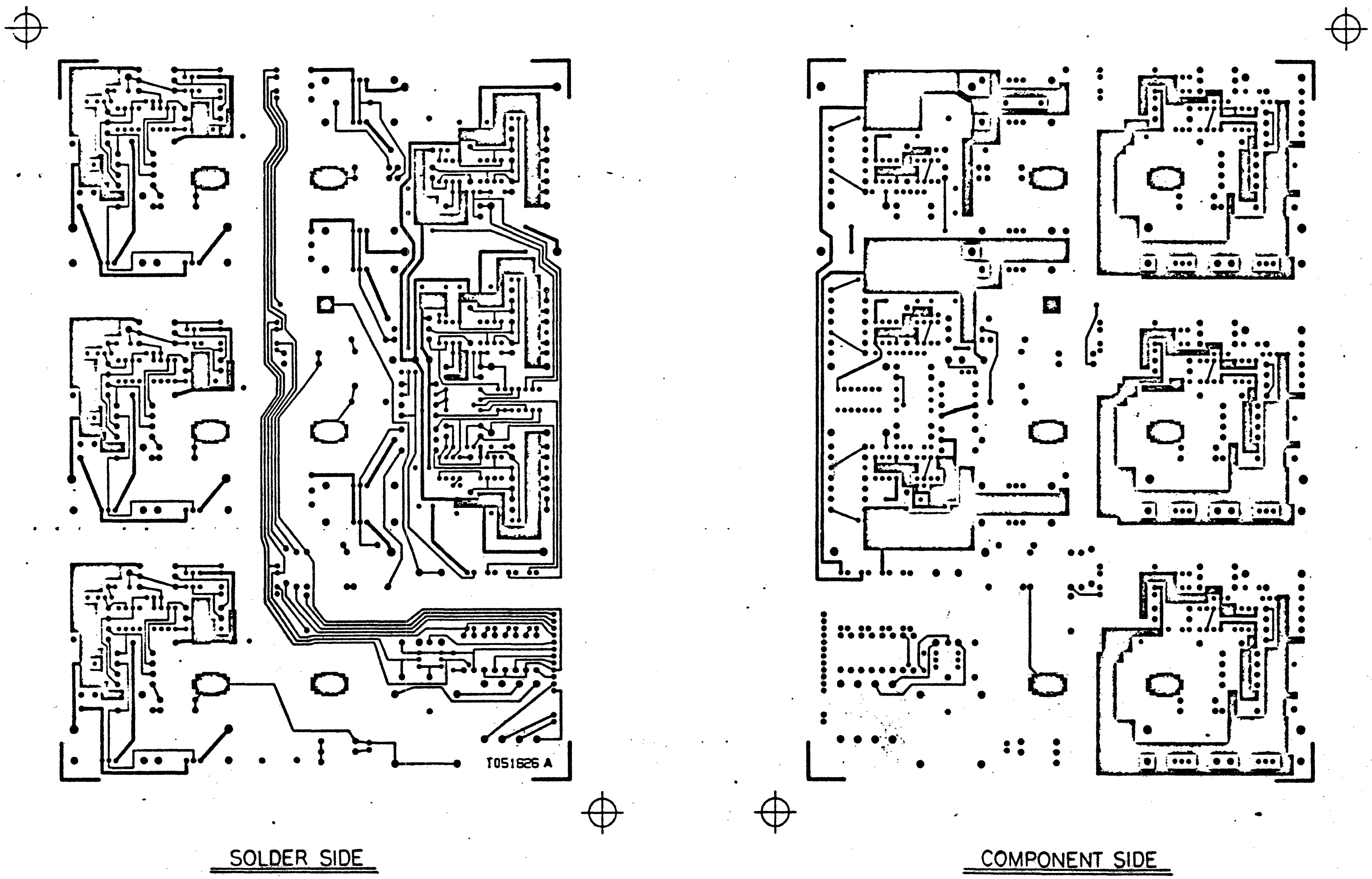
3750 East Market Street
York, Pennsylvania 17402
(717) 757-4641

FINCOR
INCOM INTERNATIONAL INC
INCOM

PRINTED WIRING BD
BASE DRIVER
DRILL & TRIM

SHT. 1 OF 3 DWG SIZE C 1051826 REV. A

FIG. 4



FINCOR

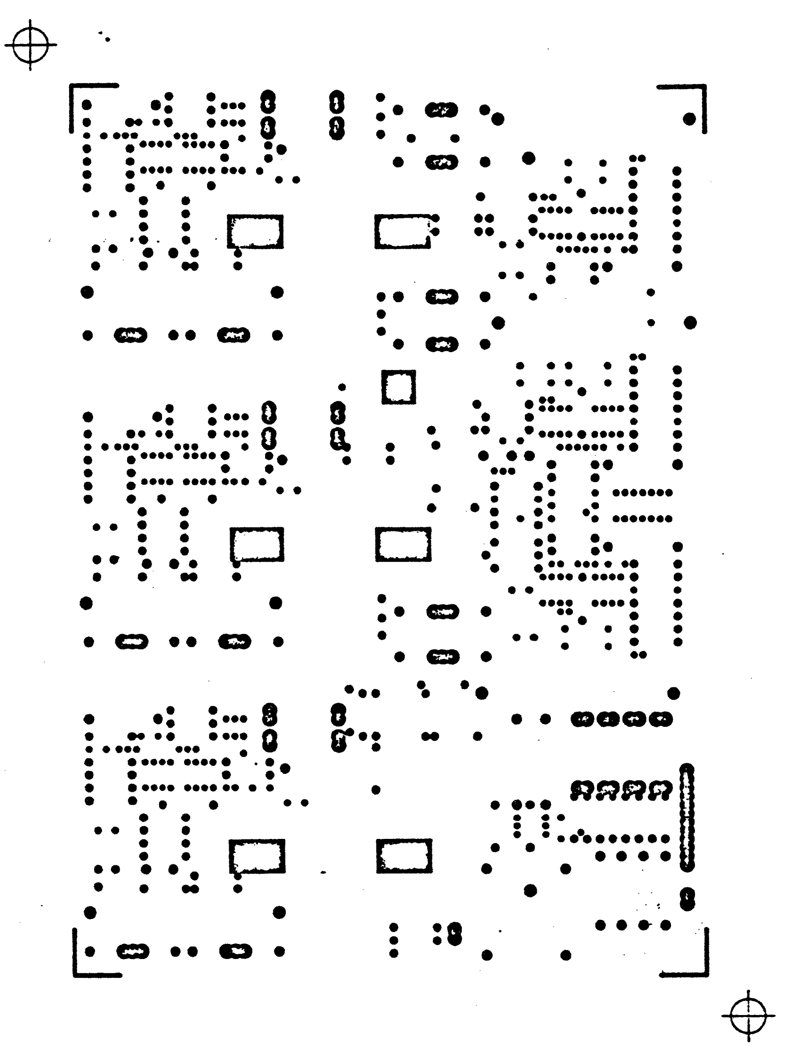
3750 East Market Street
York, Pennsylvania 17402
[illegible]

INCOM INTERNATIONAL INC
INCOM

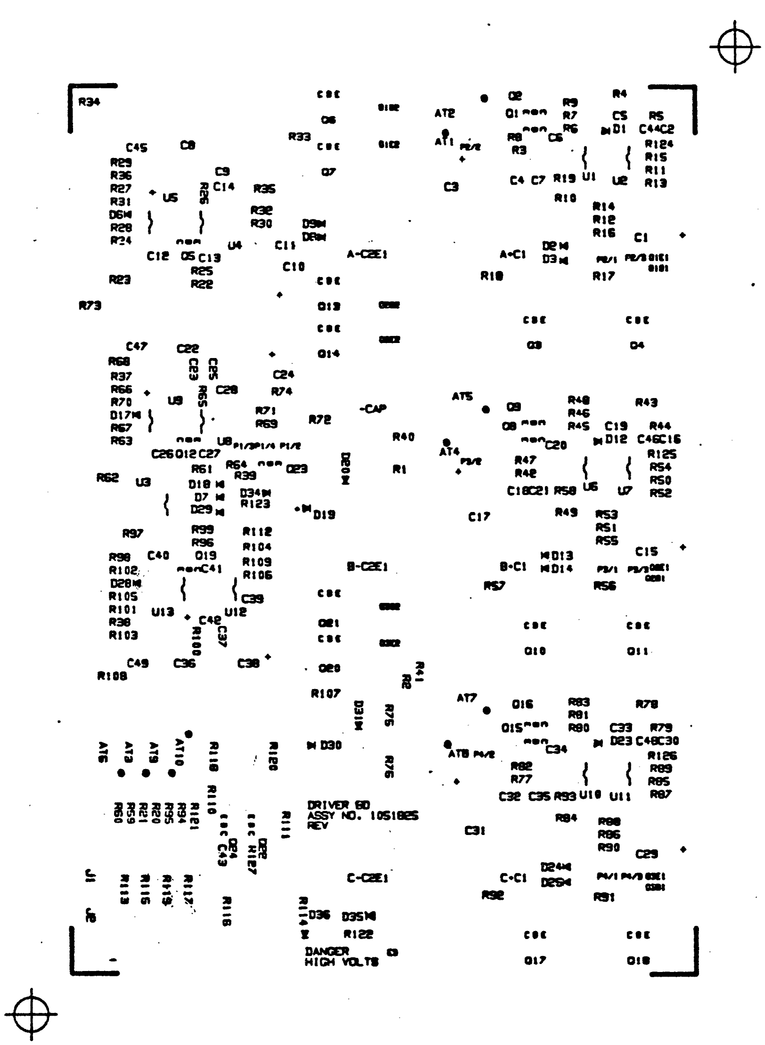
PRINTED \/IRING BD
BASE DRIVER

SHT 2 OF 3 | DWG SIZE C | 1051826 | REV. A

FIG. 5



SOLDER MASK
TYP BOTH SIDES



SILKSCREEN

FINCOR
INCOM INTERNATIONAL INC
3750 East Market Street
York, Pennsylvania 17402

PRINTED WIRING BD
BASE DRIVER

SHEET 3 OF 3 | DWG SIZE C | 1051826 | REV A

FIG. 6