Third Annual International Computervision User Conference

Meeting Program and Papers

DALLAS HILTON
DALLAS, TEXAS

September 14-17, 1981

Productivity is Our Product

COMPUTERVISION

Dear CV User:

Welcome to Dallas! We are glad you could come to our Third Annual International CV User Conference.

It is very gratifying to see our attendance grow year by year. Many of you are returning for your second or third Conference.

A closely knit, actively involved CV user family is a very healthy sign indeed. The company and its customers both have a lot to gain through a better understanding of our common hopes and mutual concerns, as well as the state of the art in this exciting technology of ours.

This is what the Conference is all about. You will have the chance to actively participate at the workshops and have the opportunity for one-on-one discussions with CV technical and managerial people.

If you attended our earlier Conferences, you know that each one is a unique and rewarding experience. Every year at these Conferences, there are new CAD/CAM colleagues you will want to exchange ideas with representing companies and industry worldwide.

In addition to the planned receptions and group dinner at the Hilton, Dallas offers a variety of pleasant activities.

We hope you enjoy this fine city and find our Conference most productive.

Cordially,

Martin Allen
President
Computervision Corporation

Third Annual International Computervision User Conference

DALLAS HILTON HOTEL
September 14–17, 1981

CONTENTS

NC & MANUFACTURING ENGINEERING 143

PC CIRCUIT DESIGN 161

ELECTRICAL DIAGRAMS 225

COMMUNICATIONS

CONFERENCE AT A GLANCE

TUESDAY, SEPTEMBER 15th

7:30AM-Authors' and Facilitators' Breakfast—Embassy East
8:30AM

9:00AM-Opening Session—Grand Ballroom
9:30AM

9:30AM-BREAK
9:45AM

PAPER PRESENTATIONS

	Mechanical Design & Drafting Junior Ballroom	Printed Circuit Design Embassy Complex	Communications Ballroom Assembly
9:45AM- Noon	Auto-Draft for Tool Design *Jessie Harrison & Robert David* *General Dynamics* Interoptics *Stanley Haskell* *Polaroid* Applications of Nodal Tabulated Drawings *Bill Higgins & Chris Zimmer* *General Electric* The Shock-Absorber—An Example of Variant Design with VARPRO at Volkswagen *Egbert Elsholz* *Volkswagenwerk, AG*	Microwave Printed Circuit Design Procedure *Stuart Horwitz* *Westinghouse* Electro-Mechanical Packaging of Electronic Products *Dr. Itaru Aramaki* *Sumitomo Electric Industries, Ltd.* Wire-Wrap Board Design Using a Computervision CADDS 3 System *Anthony Minniti* *Xerox Corporation* Introduction of a CAD System for Printed Circuit Board Design Based on Computervision Systems *Hagen Rohrbacher and G. Martin* *Standard Elektrik Lorenz AG* CVLCAP—Computervision Logic Circuit Analysis Program *Allen Brown* *Computervision Centre*	An Overview of CVNet *Ken Johnson* *Westinghouse Electric* CV HASP *Al Killinger* *Westinghouse Electric* CAD/CAM Data Management in a Host-based Network *Raphael McBain* *General Dynamics* CADDS 2/VLSI Programming Language Course Court This course will cover language definition and examples for VLSI design usage.

Noon- LUNCH—Grand Ballroom
1:00PM

	Mechanical Design & Drafting Junior Ballroom	Electrical Diagrams Embassy Complex	Plant & Architectural Ballroom Assembly
1:00PM- 3:00PM	Implementation of Forging Load and Stress Analysis on a Computervision CADDS 3 System *Taylan Altan, Ally Badawy, Carl Billhardt* *Battelle Columbus Laboratories* Three Dimensional Data Analysis Using Computervision CADDS 4 *David Terrill* *Dresser Magcobar* A Customized Procedure for Assembly Drawing & List of Parts *John Painter* *Computervision Centre*	CAD Generated Wirelists *Larry Shuhy* *Harris Corporation* Application of Data Extract to Aerospace Vehicle Wiring *Robert Cook* *Vought Corporation*	Automated Design/Analysis of Pressure Vessels Using the Computervision System *Ruth Hudale and Ramu Mysore* *Westinghouse Electric* Piping Design System *Dave Wilkinson* *H.A. Simons International Ltd.* Plant & Facilities Layout Using Interactive Computer Graphics *Gary Kahler and Harvey Marx* *Western Electric* Special Characteristics of Productivity Systems for Architects *Peter Selby* *Computervision Centre*

3:00PM- BREAK
3:15PM

	NC & Manufacturing Engineering Ballroom Assembly	Integrated Circuits Embassy Complex	Operations Management Junior Ballroom
3:15PM- 5:15PM	The Use of CAD/CAM in a Short Lead-time Prototype Application *Nick Morgan* *Smiths Industries Limited* Use of the CV Flat Pattern Software & Interface with Manufacturing *Collin McMullin* *General Electric*	Discussion of future developments of Integrated Circuits from Computervision.	Initial Graphics Exchange Specification (IGES); A Perspective *Albert Gibbons* *Westinghouse Electric* A Guide to CAD Facilities Organization for the First Time User *Fred Sukow* *General Electric* Productivity & Profitability with CAD/CAM *Chuck Cummings* *Babcock & Wilcox* CADDS 4 Rev 2—A User's Review Evaluation *William Beazley, Ph.D.* *Brown & Root* Applications for CADDS in the Design Process *Jorge Perez* *Vought Corporation*

6:30PM- DINNER—Grand Ballroom
8:30PM

WEDNESDAY, SEPTEMBER 16th

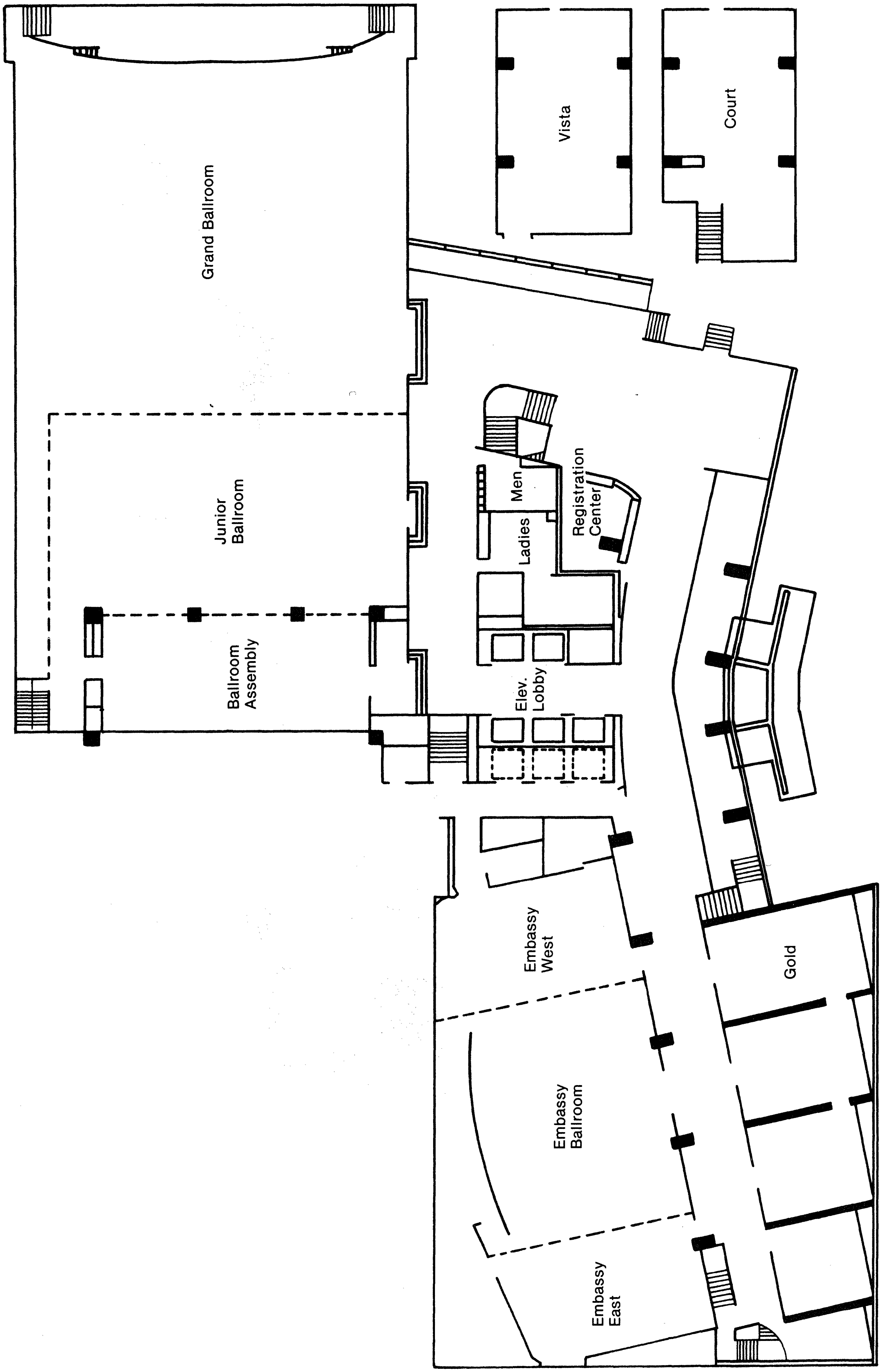
	Dimensioning & Drafting Grand Ballroom	3D Design & Surfacing Embassy & West	The Human Engineered PC Design System Court	Architecture Gold
8:30AM- 10:15AM	The two subjects that will be covered are: CADDS 3 dimensioning and detailing-problems, solutions, work around; and CADDS 4 Rev. 2 dimensioning enhancements. Associative and dual dimensions, centerlines, angular dimensioning, and JIS/ISO standards will also be discussed.	This session will begin with a presentation on CADDS 4 Rev. 2 features related to 3D Design and Surfacing. Each User Group representative will give an informal presentation on 3D Design and Surfacing.	The development of hardware and software for the PC Design product with a major focus on CADDS 4 will be the major topic. Also covered is the development and purpose of PC Design aids: libraries, training files and user guides.	We will consider the creation of a user group, clarify goals and organization of the building design product in the future, discuss problems and enhancements to the existing product and demonstrate new product capabilities. CADDS 2 Lab This lab will be a supplement to the CADDS 2/VLSI Programming Language Course that was on Tuesday.

10:15AM- 10:30AM	BREAK			
	Graphic Languages Embassy East	Tool Design & Fixturing Embassy & West	Engineering Interface to PC Design Court	Mapping Gold
10:30AM- Noon	This session will review present capabilities of PEP/PAREX/EXEC files for the benefit of new participants. Also, an overview VARPRO for CADDS 3 and 4 will be presented outlining present and future capabilities. There will be an interactive discussion on additional capabilities for future development.	Tool Design/Fixturing Workshop will review various user experiences in the tool, mold, die and fixture design areas. A tool library for mold bases and various fixture components will be presented by CV.	This workshop is intended to introduce new capabilities developed in the CADDS 4 product. Discussion will focus on new features for schematic generation and documentation. Topics included are Autolayout, Auto-Annotation, and automatic generation of schematic element reference designators, and the CV approach to logic simulation.	Discussion will focus on the present product and enhancements but also talk about the future capabilities. We will clarify the goals and organization of the mapping product, and consider the creation of a user group. CADDS 2 Lab (continued)
Noon- 1:00PM	LUNCH—Grand Ballroom			
	Engineering Analysis Embassy East	Interfacing CAD with CAM Court	New Capabilities for PC Designers Embassy & West	Integrated Circuits Gold
1:00PM- 3:00PM	A short paper will be presented on the current capabilities of the CV FEM package. The latest FEM features will then be described by Computervision. The remaining time will be devoted to discussions on further improving engineering analysis applications.	This session will describe a user's version of a neutral database implementation, the role of GNA for interfacing CAD with CAM, and Computervision's implementation of the ANSI/IGES standard.	This session is a basic introduction to the new CADDS 4 capabilities of AUTOPLACEMENT, AUTOROUTE, Geometric Design Rules Checking Editing commands and Dynamics. Emphasis in the workshop will be placed on describing the benefits available through these new capabilities which include increased system flexibility and faster design turnaround time.	This session will cover the use of the new CADDS/VLSI features available in Revisions 2.01A and 2.02.
3:00PM- 3:15PM	BREAK			
	Mass Properties Embassy East	Group Technology Court	Interface to PC Manufacturing & Testers Embassy & West	Integrated Circuits Gold
3:15PM- 5:15PM	This session will review present capabilities and CADDS 3 and the new CADDS 4 capabilities. Detailed interactive discussions on additional capabilities for future development will follow the presentations. A paper will also be presented on an enhanced CADDS 3 Mass Properties package developed by a user.	A representative from the Organization for Industrial Research will give a presentation on specific computer assisted process planning on a CAD/CAM system. We will also cover the computer approach to Group Technology.	This workshop is planned to give a review of the manufacturing tools supported in CADDS 4. Those interfaces discussed will include NC/Drill, NC Insertion, various types of plotters and automatic test equipment. Conclusion of the workshop will be highlighted by a presentation previewing PC product direction.	(continued)
6:30PM- 8:30PM	"TEXAN" RECEPTION—Grand Ballroom			

THURSDAY, SEPTEMBER 17th

	Dimensioning & Drafting Embassy & West	Education Junior Ballroom	Plant Design Embassy East	Documentation Ballroom Assembly
8:30AM– 10:15AM	The two subjects that will be covered are: CADDS 3 dimensioning and detailing-problems, solutions, work around; and CADDS 4 Rev. 2 dimensioning enhancements. Associative and dual dimensions, centerlines, angular dimensioning, and JIS/ISO standards will also be discussed.	Topics will include customer site versus educational center training; CADDS 4 training philosophy (2 weeks), specialty seminars, and in-house customer training.	Discussions will focus on 3D plant modeling with emphasis on plant equipment construction, pipe routing and fitting insertion. Time will be available for general discussion of customer concerns.	There will be a discussion of the different types of user documentation, how to use them, and their benefits.
10:15AM– 10:30AM	BREAK			
	3D Design & Surfacing Embassy & West	System Management Junior Ballroom	Numerical Control Embassy East	Software Ballroom Assembly
10:30AM– Noon	This session will begin with a presentation on CADDS 4 Rev. 2 features related to 3D Design and Surfacing. Each User Group representative will give an informal presentation on 3D Design and Surfacing.	Archiving CADDS Parts and Automatic Updating of On-Line Textfile Documentation *Ronald Smith* *M & M MARS* An Alternate Accounting Package *Curt Strobel* *Southern California Edison* System Utilization Tracking by Job Number *Brent Graham &* *Bob Rutherford* *Boeing*	Members of the User Groups will be giving the following presentations: Flat Pattern Routing and NC Progress, Rockwell; Progress in Assembly Tooling, Fairchild; Productive NC Effort and Standard Cutter Library, General Dynamics; and others.	Programming methods will be discussed in the environment of CGOS 200/10 Rev. 2, Graphics Network Architecture and CADDS 4, Rev. 2. Topics that will be discussed are: languages, program callable interfaces, soft architecture, code swapping, and data buffering.
Noon– 12:45PM	LUNCH—Grand Ballroom			
12:45PM– 2:45PM	CLOSING SESSION—Grand Ballroom			

Robert Gothie	— Sales & Services Together
Richard Hessdorfer & Mat Peterson	— Future Direction
Martin Allen	— Closing Remarks
Robert Gothie	— Adjourn



Mezzanine Level

Facilitators

Time	A Mechanical Design & Drafting	B Printed Circuit Design	C Communications	D CADDS 2/VLSI Programming Language Course
9:45 AM- Noon	CV: Steven Weiner E: John Boertjens MW: Gale Saxon W: John Martin Allen Alley	CV: Tim Ristine E: Tom Davis MW: Jerry Hegg W: Paul Van Ast	CV: Paul Hamburger MW: Bill Fagen	CV: Gary Robson
	Mechanical Design & Drafting	Electrical Diagrams	Plant & Architectural	
1:00 PM- 3:00 PM	Continued	CV: Mike Barry E: Richard Skanron MW: Gary Cox W: Doug Conley	CV: Ernie Heimberg	
	NC & Manufacturing Engineering	Integrated Circuits	Operations Management	
3:15 PM- 5:15 PM	CV: Richard Simon E: Warren Gengler MW: Gary Kahler W: Charles Smith	CV: Gerry Shields Discussion of future developments of Integrated Circuits from Computervision.	CV: Ralph Parker E: Chester Markiewicz MW: John Soose W: Robert Houk	

Wednesday, September 16 Workshop Session

Time	A Dimensioning & Drafting	B 3D Design & Surfacing	C The Human Engineered PC Design System	D Architecture
8:30 AM- 10:15 AM	CV: Bob Cronin W: Richard Cunningham	CV: Ashok Bhide E: John Boertjens MW: Darcy Sety W: Allen Alley	CV: John Felahi E: Tom Davis MW: Jeffrey Bytes W: Paul Van Ast	CV: Richard Hessdorfer CADDS 2 Lab CV: Gary Robson
	Graphics Languages	Tool Design & Fixturing	Engineering Interface To PC Design	Mapping
10:30 AM- Noon	CV: Steven Woodbridge E: Ken Johnson MW: Don Crockett W: Allen Alley	CV: Gary Hodgson MW: Elton Ables W: Richard Sauerbrey	CV: Anne Kiburis MW: David Ross W: Paul Van Ast	CV: Richard Holekamp E: Val Gannotti W: Jack Loveland CADDS 2 Lab CV: Gary Robson
	Engineering Analysis	Interfacing CAD with CAM	New Capabilities For PC Designers	Integrated Circuites
1:00 PM- 3:00 PM	CV: Steve Weiner E: John Boertjens W: Gary Brandt Tom Judd	CV: Ed Fournier E: Warren Gengler MW: Don Crockett W: Charles Smith	CV: Elizabeth Barbone MW: Jeffrey Bytes W: Paul Van Ast	CV: Mike Reed
	Mass Properties	Group Technology	Interface To PC Manufacturing and Testers	Integrated Circuits (continued)
3:15 PM- 5:15 PM	CV: Steven Woodbridge E: Tom Heinz MW: Dave Schaller W: Tom Judd	CV: Dan Socha MW: John Tahaney	CV: Ed Kennedy MW: Jerry Hegg W: Allen Alley	

Thursday, September 17 Workshop Session

	A	B	C	D
Time	Dimensioning & Drafting	Education	Plant Design	Documentation
8:30 AM-	CV: Bob Cronin	CV: Jack Rousseau	CV: Ernie Heimberg	CV: **Jesse Kafel**
10:15 AM		MW: Darcy Sety	MW: Robert Weiss	
	W: Richard Cunningham	W: Ron Dubois		
	3D Design & Surfacing	**System Management**	**Numerical Control**	**Software**
10:30 AM-			CV: Don Lowke/	CV: Norm Siegel
Noon	CV: Ashok Bhide	CV: Ralph Parker	C. Gouldson	
	E: Paul Davis	E: Chester B. Markiewicz	E: Warren Gengler	
	MW: Robert Tennison		MW: Gary Kahler	
	W: Allen Alley	W: Robert Houk	W: Charles Smith	

12:45 PM-
2:45 PM

Closing Session

Robert Gothie	Sales & Service—Together
Richard Hessdorfer & Mat Peterson	Future Direction
Martin Allen	Closing Remarks
Robert Gothie	Adjourn

GENERAL INFORMATION

Registration & Information Center

Hours	Sunday, September 13	7:00PM-9:00PM
	Monday, September 14	7:00AM-8:00PM
	Tuesday, September 15	7:00AM-7:00PM
	Wednesday, September 16	8:00AM-7:00PM
	Thursday, September 17	8:00AM-3:00PM

Messages

A message board is located near the Registration & Information Center. Any calls, messages, or last minute program changes will be posted.

Media Services/Publications/Training Booth

New products from Computervision's Media Services and Publications Department will be displayed on the Mezzanine Level. You can examine new training packages, look over the most current publications, and take the literature on new training course offerings.

Hours: Tuesday, September 15th—Noon-5:30PM
Wednesday, September 16th—8:00AM-5:30PM
Thursday, September 17th—8:00AM-Noon

A representative from Computervision's Customer Training Center will be available to discuss training options and strategies on Wednesday afternoon.

Manager's Information Suite

Computervision Managers will be available to meet with you during the conference.

Hours: Tuesday, September 15th—9:00AM-6:00PM
Wednesday, September 16th—9:00AM-6:00PM
Thursday, September 17th—9:00AM-3:00PM

Location will be posted on the message board.

Hotel Check-Out Time

Hilton Hotel check-out time is 1:00PM. Your luggage can be stored with the Bell Captain on the Lobby Level, or you can request a late check-out.

Transportation

Surtran Bus	—50 minutes one way/$6.00 (5 past hour & 25 to hour)
Taxi	—30 minutes one way/$20.00

Badges

Attendees should wear their name badges at all times during the Conference.

WHITE BADGES	—USERS
BLUE BADGES	—COMPUTERVISION PERSONNEL

Meal Tickets

Coupons for all your Conference meals are included in the Conference program. Please present the appropriate coupon to the convention personnel at the door.

Function	Location
Computervision Reception	Grand Ballroom
Authors' & Facilitators' Breakfast	Embassy East (by Invitation)
Luncheons	Grand Ballroom
Coffee Service	Mezzanine Level (Meal Ticket Not Required)
Dinner	Grand Ballroom
"Texan" Reception	Grand Ballroom
Asian & European Hospitality	Check Message Board for location (by Invitation)

COMPUTERVISION USER GROUP OFFICERS

	EAST	MIDWEST	WEST
Chairman	Joe Delvac Sperry Gyroscope	John Tahaney Vought Corporation	Alex Rawling Boeing Company
Vice-Chairman	Paul Davis Pratt & Whitney Aircraft	Charles Palermo General Dynamics	Jack Stahl General Dynamics
Secretary	Tom Davis ITT Telecommunications	Richard Watson ADC Telecommunications	Ed Shaffer McDonnell Douglas
Treasurer	Val Gannotti Design Graphics Incorporated		

CREDITS AND DISCLAIMERS ON PUBLISHED PAPERS

Permission to publish these papers and proceedings in part or in full with credit to the author and references may be obtained by contacting Computervision Corporation, 100 Commerce Way, Woburn, MA 01801.

Computervision Corporation is not responsible for the statements or discussions made relative to these papers published and presented at this conference.

Mechanical Design & Drafting

AUTO-DRAFT FOR TOOL DESIGN

by
Jessie A. Harrison
and
Robert D. David
General Dynamics
Convair Division
Lindberg Field Plant
P.O. Box 80877—MZ/83-4150
San Diego, CA 92138
714-692-8320

TABLE OF CONTENTS

7. LIST OF FIGURES

ACKNOWLEDGEMENTS

We wish to express our thanks to John Harstell, General Dynamics/Convair Division, Tool Design, for his technical input to this project; to Scott Ruberl, General Dynamics/Convair Division, Computer Aided Manufacturing, for his help and guidance in outlining the direction of this project and his support during its development; and to Andrea Cohen for her excellent skills and perseverance, without which this paper would not have been possible.

ABSTRACT

Auto-Draft, a system of software subroutines and text files, was developed by General Dynamics/Convair Division for use on the Computervision CADDS 3 interactive graphics system. Graphical representations of components, with non-graphic properties, are quickly and accurately inserted into Tool Design computer-generated drawings. Bill of Materials reports are automatically generated by Auto-Draft when it searches the drawing for these properties.

1. BACKGROUND

Traditionally, when drafting a tool design on Computervision (CV), tooling components are either drawn as needed or pre-drawn and stored as parts on a disk.

Drawing the component "from scratch" is both time consuming and possibly inaccurate. The tool designer is required to know all of the dimensions and calculations needed to draw the component entity-by-entity. For example, a strap clamp has eight dimensional values and is graphically represented by eight lines and two arcs. See Figure 1a.

Pre-drawn components stored as parts on a disk can be retrieved and inserted many times. When inserted into drawings, a component part can be scaled and/or rotated to best represent its size and orientation. Modification is almost always needed making it necessary to use more than one command. Furthermore, some component groups do not vary dimensionally as a fixed ratio; thus, insertion with a scale factor will make only a portion of the component dimensionally correct.

Rotation of the component in the X-Y plane abouts it origin results in different front and side views as illustrated in Figure 1a. Each pre-drawn and stored part uses a minimum of twelve sectors (3,072 words of disk storage space). A prohibitive amount of storage space would be required to maintain all possible views of the components.

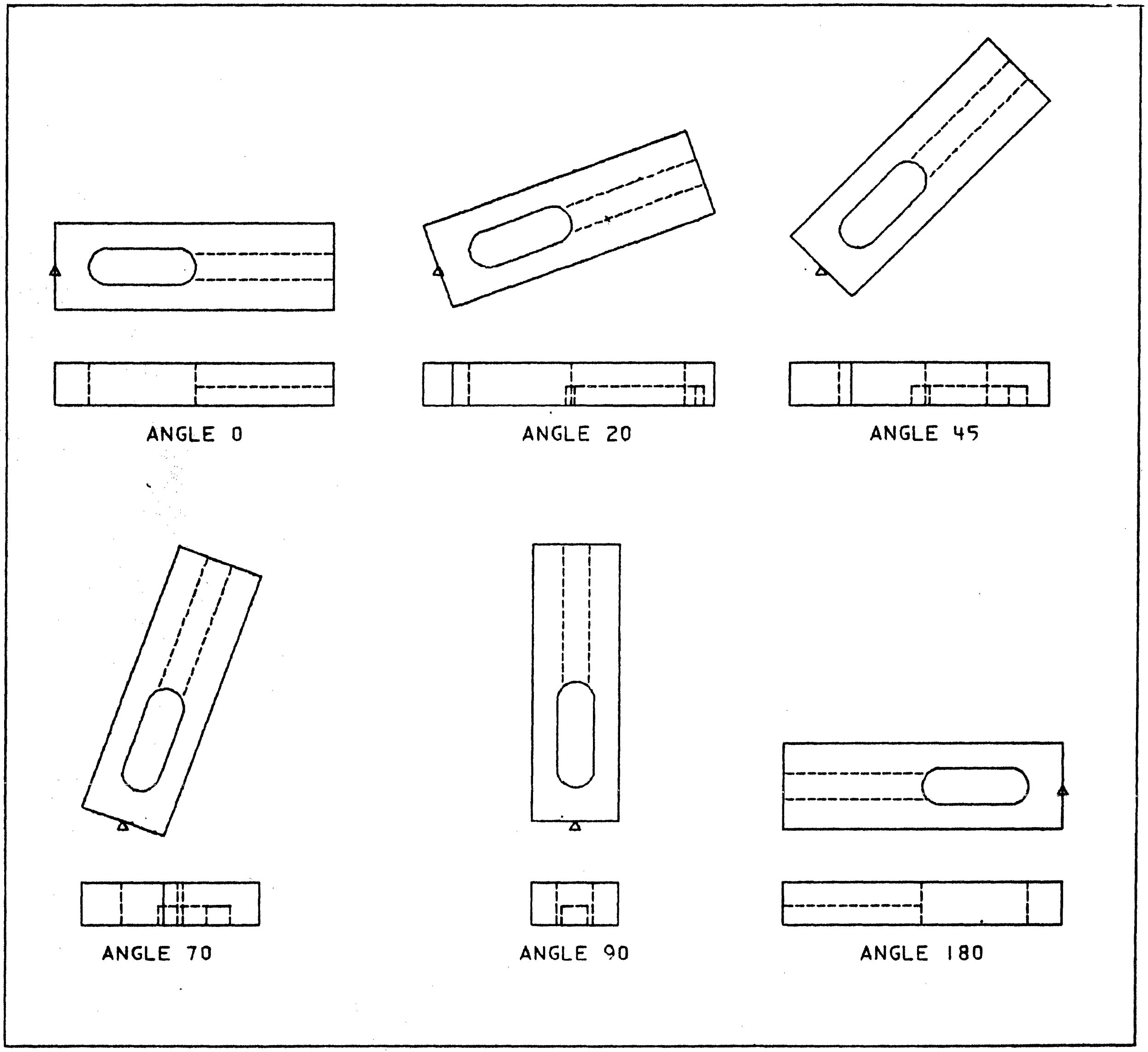


Figure 1a: Strap clamp top and front views at various angles of rotation in the X-Y plane

**Table I:
Pre-drawn part storage**

# PARTS	# SECTORS	# WORDS
1	12	3,072
100	1,200	307,200
400	4,800	1,228,800

It is readily apparent, then, that creating components from "scratch" greatly increases total drafting time; on the other hand, hundreds of tooling components stored as top, front, and side views would require a large amount of disk storage space. An entirely new approach, Auto-Draft, was designed to better meet the needs of tool design.

2. AUTO-DRAFT

The Auto-Draft system was developed to draw tooling components (with associated non-graphic properties attached to them) which can be used in generating Bill of Materials (BOM) reports upon completion of the design task. A single graphics command will quickly draw accurate tool components which require no further modification. The non-graphic properties associated with the components are extracted from the database, tallied, merged, formatted, and output to a text file when a single command, BOM, is invoked. This file is then inserted into a standard BOM form and/or printed to an output device.

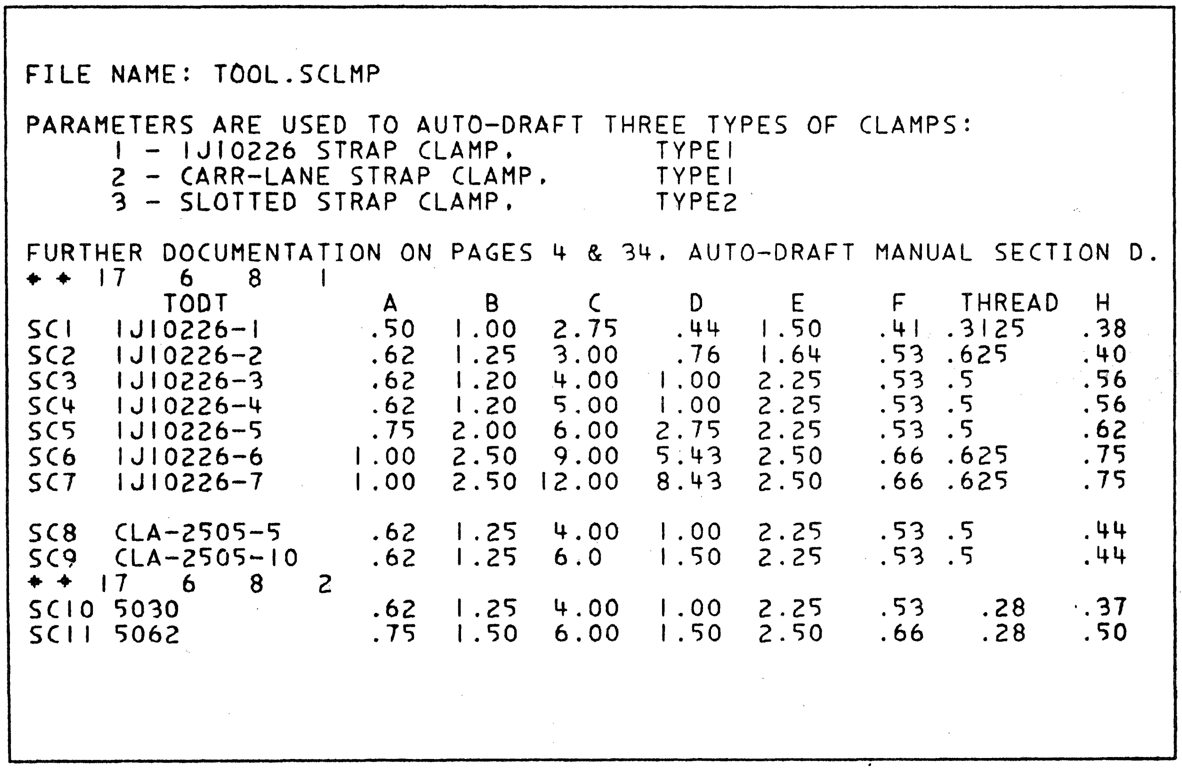
The newly developed Auto-Draft commands, executed in the CADDS3 system, are "DRAW," the graphics command and "RUN BOM," the bill of materials command.

2.1 GRAPHICS COMMAND — DRAW

2.1.1 Parameter File

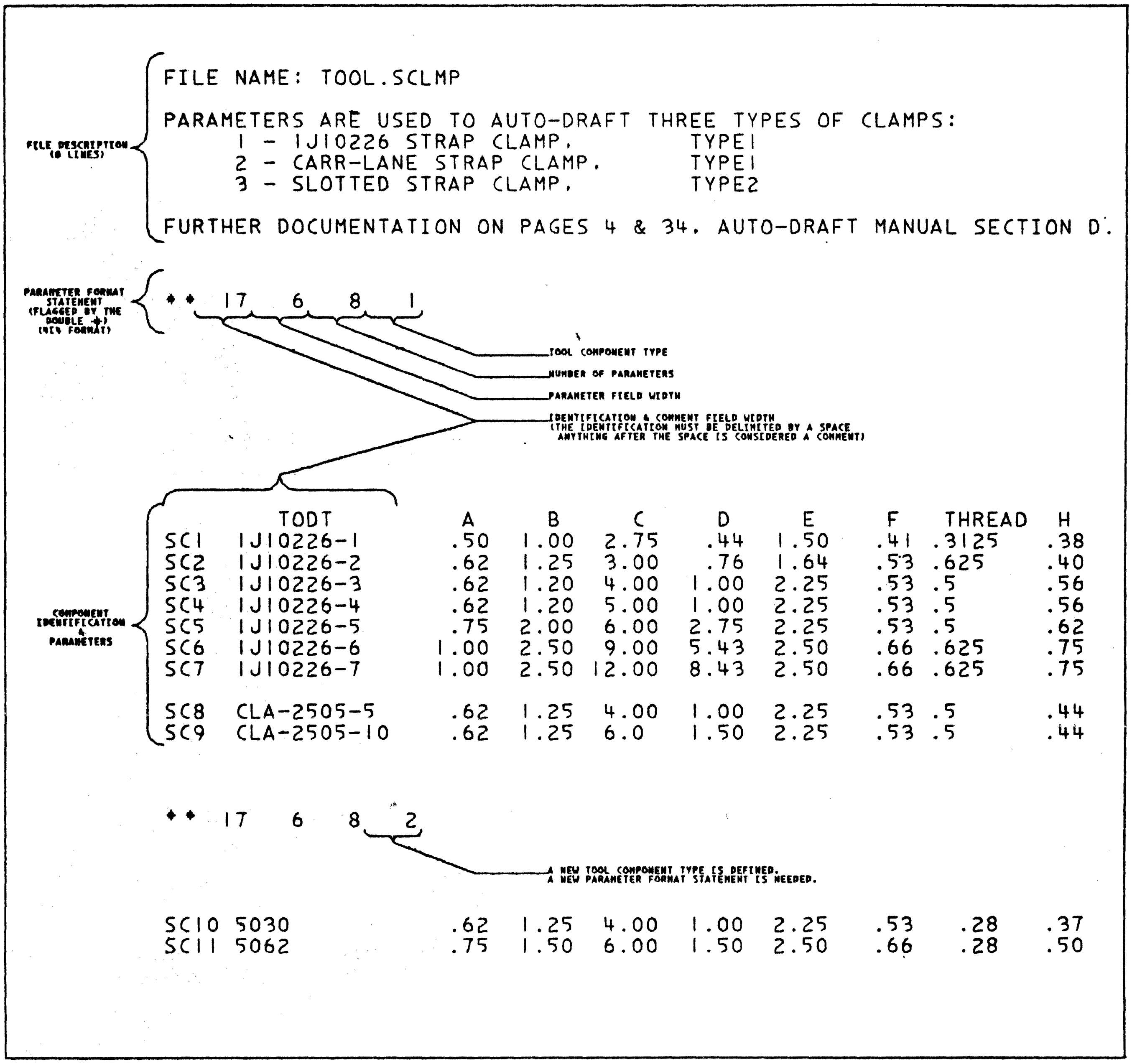
In Auto-Draft, each component is described parametrically. Similarly shaped components are grouped together under common component names (e.g., strap clamp, bolt, washer, etc.). Each group of components is stored in separate text files; each component in a group has a unique identification label, or tool-ID, consisting of two or three mnemonic letters followed by a number.

Additions, deletions, or modifications to these files are easily accomplished through file-editing outside of the Auto-Draft system. Figure 2.1a shows an example of a parameter file. Figure 2.1b describes the format of parameter files. Each tool component has a pre-defined origin and orientation, an example of which is shown in Figure 2.1c.



```
FILE NAME: TOOL.SCLMP

PARAMETERS ARE USED TO AUTO-DRAFT THREE TYPES OF CLAMPS:
     1 - 1J10226 STRAP CLAMP.         TYPE1
     2 - CARR-LANE STRAP CLAMP.       TYPE1
     3 - SLOTTED STRAP CLAMP.         TYPE2

FURTHER DOCUMENTATION ON PAGES 4 & 34. AUTO-DRAFT MANUAL SECTION D.
✦ ✦ 17   6   8   1
        TODT            A     B     C      D     E      F   THREAD  H
SC1  1J10226-1        .50  1.00  2.75    .44  1.50    .41 .3125   .38
SC2  1J10226-2        .62  1.25  3.00    .76  1.64    .53 .625    .40
SC3  1J10226-3        .62  1.20  4.00   1.00  2.25    .53 .5      .56
SC4  1J10226-4        .62  1.20  5.00   1.00  2.25    .53 .5      .56
SC5  1J10226-5        .75  2.00  6.00   2.75  2.25    .53 .5      .62
SC6  1J10226-6       1.00  2.50  9.00   5.43  2.50    .66 .625    .75
SC7  1J10226-7       1.00  2.50 12.00   8.43  2.50    .66 .625    .75

SC8  CLA-2505-5       .62  1.25  4.00   1.00  2.25    .53 .5      .44
SC9  CLA-2505-10      .62  1.25  6.0    1.50  2.25    .53 .5      .44
✦ ✦  17   6   8   2
SC10 5030             .62  1.25  4.00   1.00  2.25    .53    .28  .37
SC11 5062             .75  1.50  6.00   1.50  2.50    .66    .28  .50
```

Figure 2.1a: Parameter file for STRAP CLAMPS



```
FILE NAME: TOOL.SCLMP

PARAMETERS ARE USED TO AUTO-DRAFT THREE TYPES OF CLAMPS:
     1 - 1J10226 STRAP CLAMP,          TYPE1
     2 - CARR-LANE STRAP CLAMP,        TYPE1
     3 - SLOTTED STRAP CLAMP,          TYPE2

FURTHER DOCUMENTATION ON PAGES 4 & 34, AUTO-DRAFT MANUAL SECTION D.

** 17   6   8   1

        TODT              A      B      C      D      E      F  THREAD   H
SC1  1J10226-1          .50   1.00   2.75    .44   1.50    .41 .3125    .38
SC2  1J10226-2          .62   1.25   3.00    .76   1.64    .53 .625     .40
SC3  1J10226-3          .62   1.20   4.00   1.00   2.25    .53 .5       .56
SC4  1J10226-4          .62   1.20   5.00   1.00   2.25    .53 .5       .56
SC5  1J10226-5          .75   2.00   6.00   2.75   2.25    .53 .5       .62
SC6  1J10226-6         1.00   2.50   9.00   5.43   2.50    .66 .625     .75
SC7  1J10226-7         1.00   2.50  12.00   8.43   2.50    .66 .625     .75

SC8  CLA-2505-5         .62   1.25   4.00   1.00   2.25    .53 .5       .44
SC9  CLA-2505-10        .62   1.25   6.0    1.50   2.25    .53 .5       .44


** 17   6   8   2


SC10 5030               .62   1.25   4.00   1.00   2.25    .53    .28   .37
SC11 5062               .75   1.50   6.00   1.50   2.50    .66    .28   .50
```

FILE DESCRIPTION (8 LINES)
PARAMETER FORMAT STATEMENT (FLAGGED BY THE DOUBLE *) (4I4 FORMAT)
TOOL COMPONENT TYPE
NUMBER OF PARAMETERS
PARAMETER FIELD WIDTH
IDENTIFICATION & COMMENT FIELD WIDTH (THE IDENTIFICATION MUST BE DELIMITED BY A SPACE ANYTHING AFTER THE SPACE IS CONSIDERED A COMMENT)
COMPONENT IDENTIFICATION & PARAMETERS
A NEW TOOL COMPONENT TYPE IS DEFINED.
A NEW PARAMETER FORMAT STATEMENT IS NEEDED.

Figure 2.1b: Parameter file format for STRAP CLAMPS

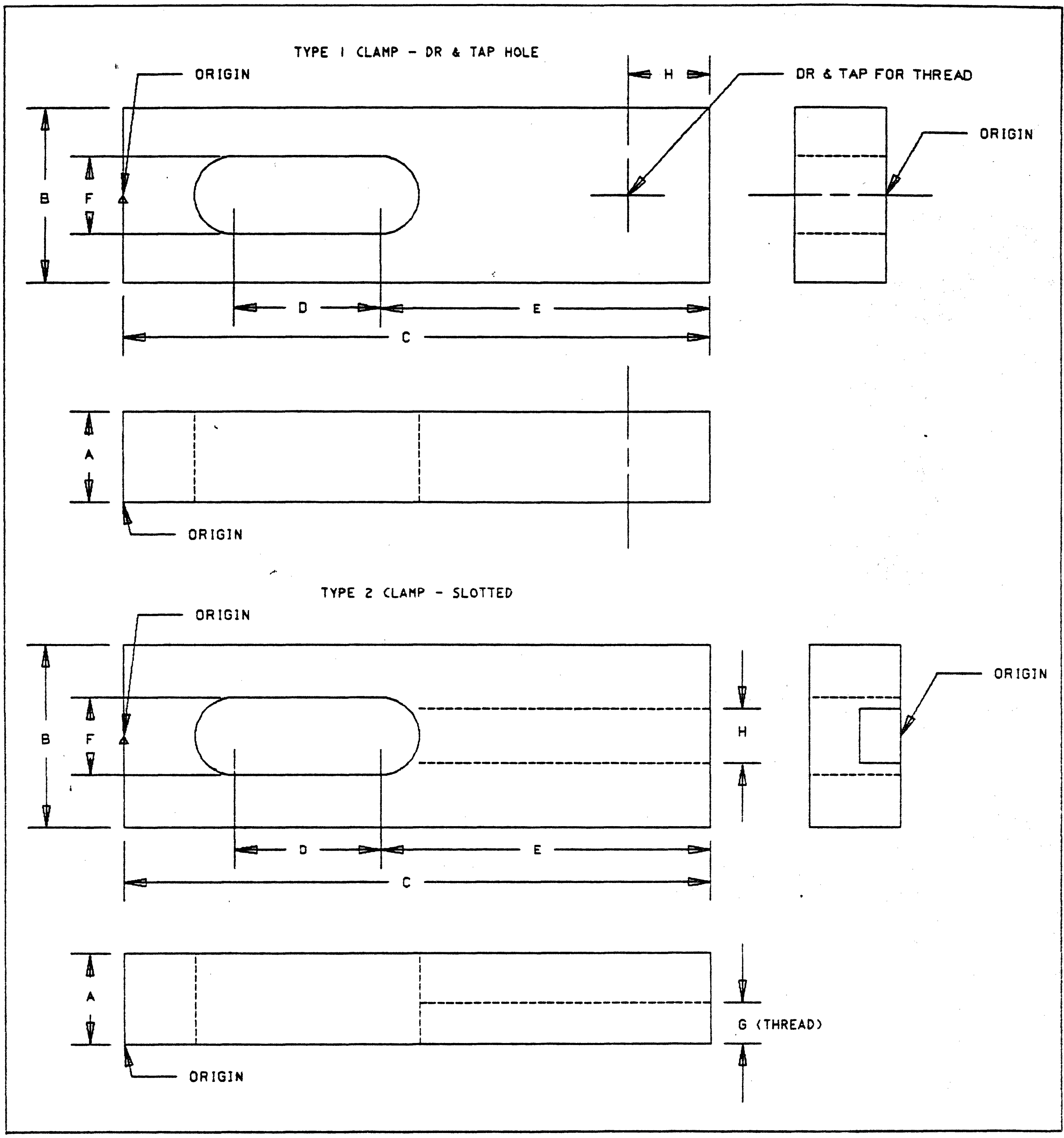
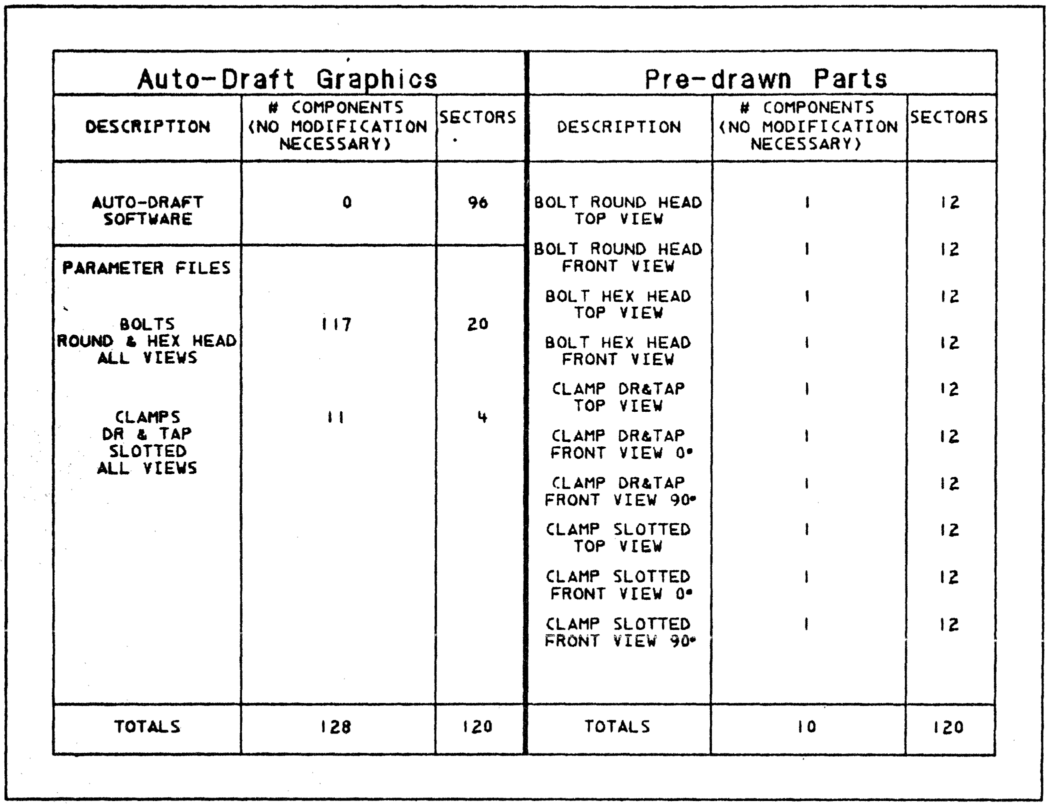


Figure 2.1c: Parameter definitions for STRAP CLAMPS

Table II provides a comparison between the amount of space required for storage of Auto-Draft component parameters and storage of pre-drawn components. Note that the amount of space used for the Auto-Draft software is also included in the comparison.

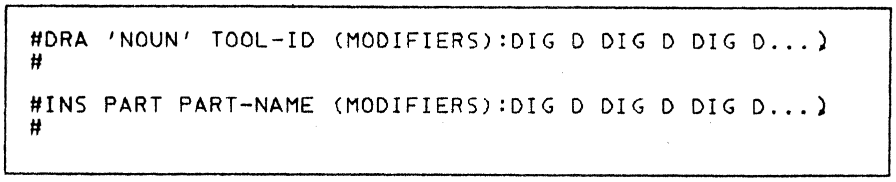
Table II:
Sector size comparison between Auto-Draft components and Pre-drawn parts



Auto-Draft Graphics			Pre-drawn Parts		
DESCRIPTION	# COMPONENTS (NO MODIFICATION NECESSARY)	SECTORS	DESCRIPTION	# COMPONENTS (NO MODIFICATION NECESSARY)	SECTORS
AUTO-DRAFT SOFTWARE	0	96	BOLT ROUND HEAD TOP VIEW	1	12
PARAMETER FILES			BOLT ROUND HEAD FRONT VIEW	1	12
BOLTS ROUND & HEX HEAD ALL VIEWS	117	20	BOLT HEX HEAD TOP VIEW	1	12
			BOLT HEX HEAD FRONT VIEW	1	12
CLAMPS DR & TAP SLOTTED ALL VIEWS	11	4	CLAMP DR&TAP TOP VIEW	1	12
			CLAMP DR&TAP FRONT VIEW 0°	1	12
			CLAMP DR&TAP FRONT VIEW 90°	1	12
			CLAMP SLOTTED TOP VIEW	1	12
			CLAMP SLOTTED FRONT VIEW 0°	1	12
			CLAMP SLOTTED FRONT VIEW 90°	1	12
TOTALS	128	120	TOTALS	10	120

2.1.2 Auto-Draft Graphics Software

The verb DRAW and its alternate spelling DRA are added to the CADDS 3 graphics command tables (verb-noun tables). This verb is used to activate the Auto-Draft system. Once the verb is added, no further updating of the tables is required regardless of any additions, modifications, or deletions to the Auto-Draft system. The component group name, within Auto-Draft, is used to simulate the 'noun' in the command syntax. The format of a standard CADDS 3 command has been maintained as shown in Figure 2.1d.



```
#DRA 'NOUN' TOOL-ID (MODIFIERS):DIG D DIG D DIG D...)
#

#INS PART PART-NAME (MODIFIERS):DIG D DIG D DIG D...)
#
```

Figure 2.1d: Comparison of Auto-Draft command syntax to CADDS 3 command syntax

The structure of the Auto-Draft graphics software is comprised of both newly-developed subroutines and CADDS 3 utility subroutines. The newly-developed software was written to interpret the data input by the user and to calculate the graphic output, or entities, required for a given component. The system utility subroutines are used as interfaces between the user and Auto-Draft and between Auto-Draft and the graphics database. See Appendix B for a description of the Auto-Draft and system subroutines used. Auto-Draft subroutines are written in FORTRAN-S, an adaptation of standard FORTRAN*. FORTRAN-S was developed by Computervision for the CADDS 3 graphics system.

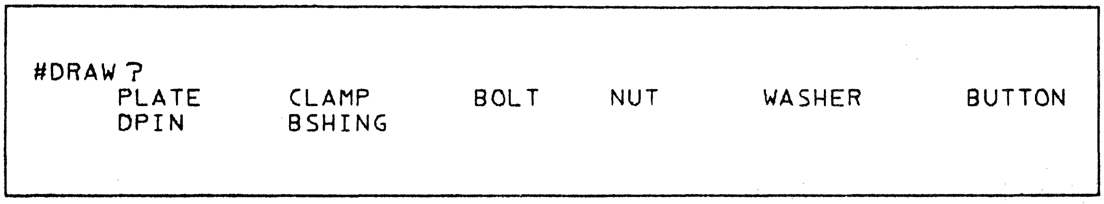
The Auto-Draft subroutines fall into two categories: the input command interpreter and the graphics output.

2.1.2.1 Auto-Draft Input Command Interpreter

This section processes the user input. It is divided into four subsections necessary for the user to identify a specific component:

a. component name or "noun" (Auto-Draft "noun" processor);
b. component identification (tool-ID);
c. component orientation and/or view (modifier processor);
d. component-origin coordinate location (GETDATA).

Component names are stored in the Component Name Table. System subroutines are used to return the table index corresponding to the user-input name. This index is then used to identify the group parameter file and geometry subroutines that will calculate and insert the resultant geometry into the graphics database. Inputting a question mark (?) instead of the component name directs the system to list the table of available Auto-Draft components on the command device. The user can input the name at this point and continue in the Auto-Draft mode, or exit with a carriage return. Figure 2.1e describes a typical input sequence that illustrates this feature.



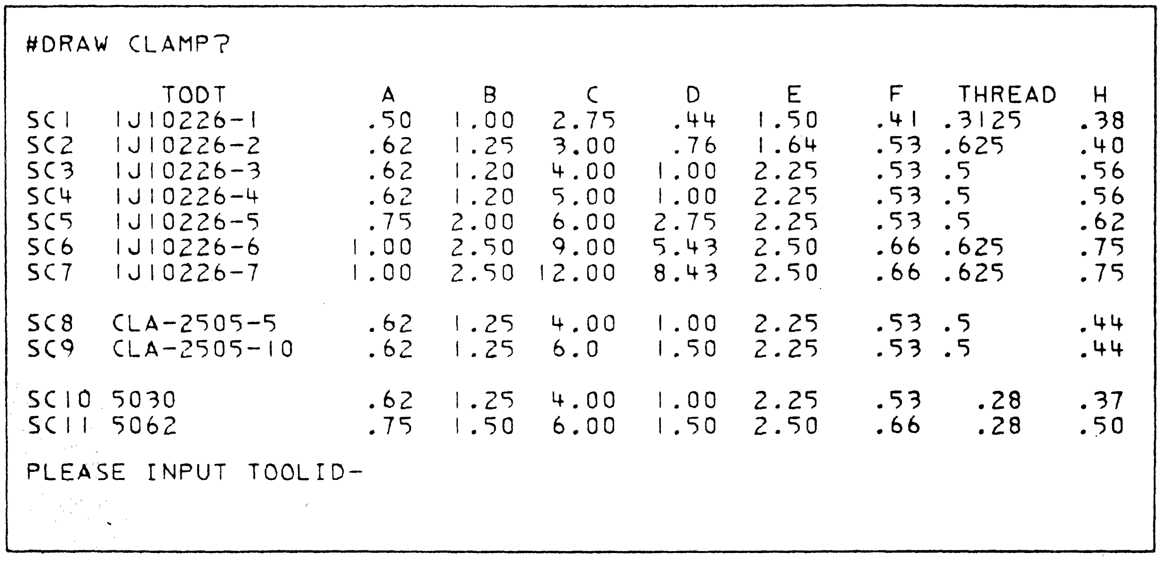
```
#DRAW ?
      PLATE      CLAMP      BOLT     NUT        WASHER        BUTTON
      DPIN       BSHING
```

Figure 2.1e: Component name listing to command device

NOTE: A carriage return or Control E entered at any point within the command sequence will terminate the command.

Component identification, or tool-ID's, are input after the appropriate component group file has been opened up. The file is searched until a match is found; then the corresponding parameters are loaded into a parameter array to be used by the geometry section of the software. If no match is found, then the system informs the user of this fact and Auto-Draft is exited. A question mark (?) input in place of the tool-ID serves the same purpose as described above, except that the list would contain the tool-ID's and parameters found in the active group file. Auto-Draft then prompts the user for the desired tool-ID, as shown in Figure 2.1f.

*Standard FORTRAN, as described in American Standard FORTRAN, approved March 7, 1966, American Standards Association, Inc. Document x3-9-1966.

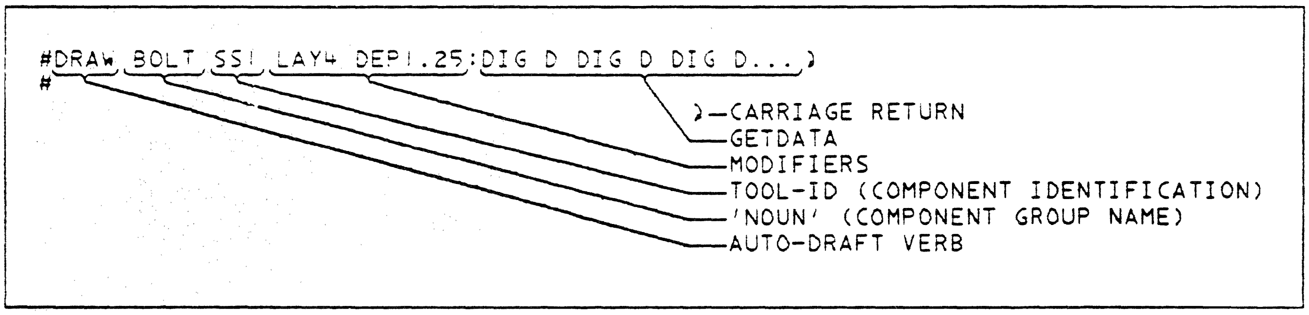


```
#DRAW CLAMP?

       TODT             A      B       C       D      E       F   THREAD   H
SC1    1J10226-1       .50   1.00    2.75     .44   1.50    .41 .3125    .38
SC2    1J10226-2       .62   1.25    3.00     .76   1.64    .53 .625     .40
SC3    1J10226-3       .62   1.20    4.00    1.00   2.25    .53 .5       .56
SC4    1J10226-4       .62   1.20    5.00    1.00   2.25    .53 .5       .56
SC5    1J10226-5       .75   2.00    6.00    2.75   2.25    .53 .5       .62
SC6    1J10226-6      1.00   2.50    9.00    5.43   2.50    .66 .625     .75
SC7    1J10226-7      1.00   2.50   12.00    8.43   2.50    .66 .625     .75

SC8    CLA-2505-5      .62   1.25    4.00    1.00   2.25    .53 .5       .44
SC9    CLA-2505-10     .62   1.25    6.0     1.50   2.25    .53 .5       .44

SC10 5030              .62   1.25    4.00    1.00   2.25    .53    .28   .37
SC11 5062              .75   1.50    6.00    1.50   2.50    .66    .28   .50

PLEASE INPUT TOOLID-
```

Figure 2.1f: Tool component listing to command device

The listing can be interrupted by using the escape key (ESC); various control characters are then input (e.g., Q, N, K). Entering a Q will direct the system to jump to the end of the file and output the user-prompt to the command device.

Component view and orientation modifiers are input after the desired component has been identified with the tool-ID (see Figure 2.1g.)



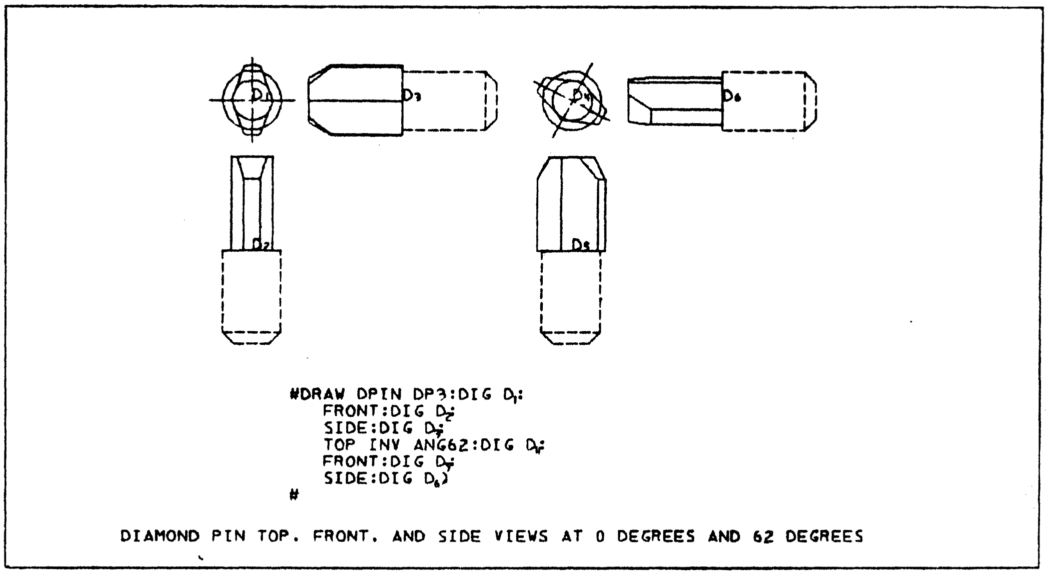
```
#DRAW BOLT SS1 LAY4 DEP1.25:DIG D DIG D DIG D...)
#
```

)—CARRIAGE RETURN
GETDATA
MODIFIERS
TOOL-ID (COMPONENT IDENTIFICATION)
'NOUN' (COMPONENT GROUP NAME)
AUTO-DRAFT VERB

Figure 2.1g: Auto-Draft command syntax

A series of default values exist and will be used if no modifiers are input. Auto-Draft allows for nine modifiers: view — TOP, FRONT, SIDE; orientation — ANG, PRP, VEC; LAY; DEP; INV.

TOP — Draw the top view of the component (default)
FRONT — Draw the front view
SIDE — Draw the side view



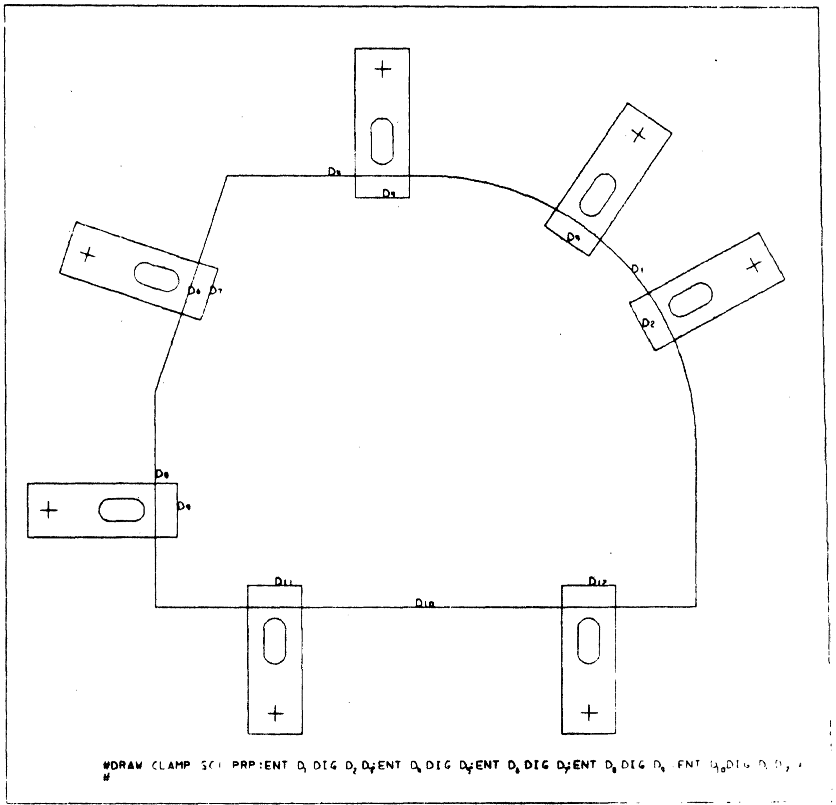
```
#DRAW DPIN DP3:DIG D1:
    FRONT:DIG D2;
    SIDE:DIG D3;
    TOP INV ANG62:DIG D4;
    FRONT:DIG D5;
    SIDE:DIG D6)
#
```

DIAMOND PIN TOP. FRONT. AND SIDE VIEWS AT 0 DEGREES AND 62 DEGREES

Figure 2.1h: Example of Auto-Draft graphic command structure using the ANG and View modifiers

ANG — Draw the component such that its major axis froms the specified angle with the positive X-axis, rotated in the X-Y plane about the component-origin (default is 0.0 degrees). This mode, ANG 0, is default when no view modifiers are input.

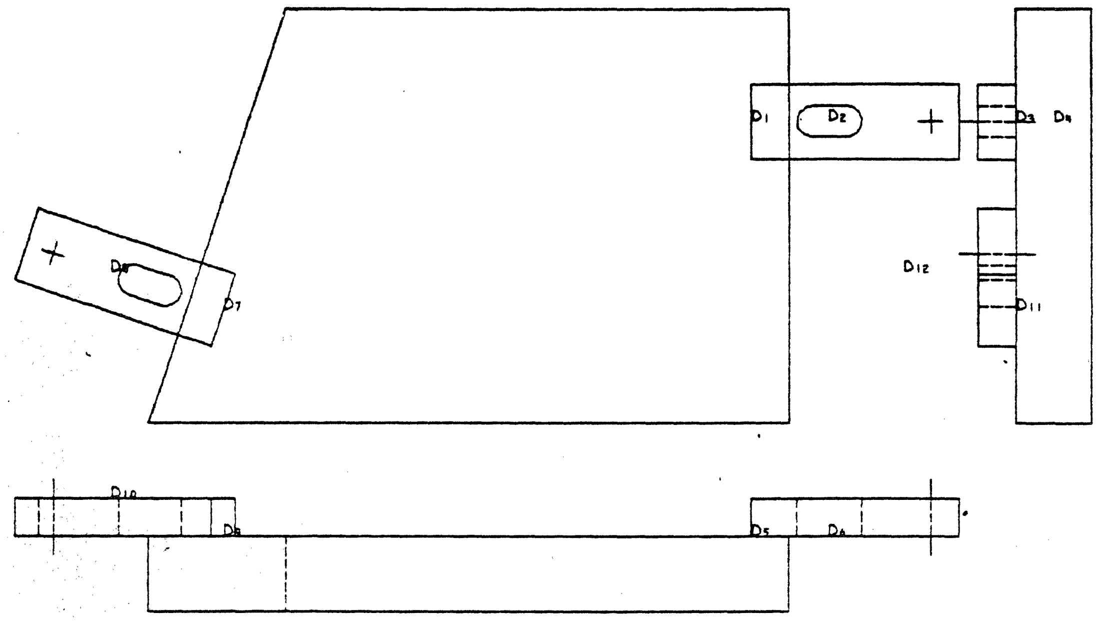
PRP — Draw the component perpendicular to a selected entity (line or circle/arc). The coordinate-locations of the origins are then digitized.



```
#DRAW CLAMP SC1 PRP:ENT D1 DIG D2 D3;ENT D4 DIG D5;ENT D6 DIG D7;ENT D8 DIG D9 ENT D10DIG D11 D12,
#
```

Figure 2.1i: Example of Auto-Draft graphic command structure using the PRP modifier

VEC — Draw the component along the vector defined by two input locations or digitizes, the first of which locates the origin of the component.



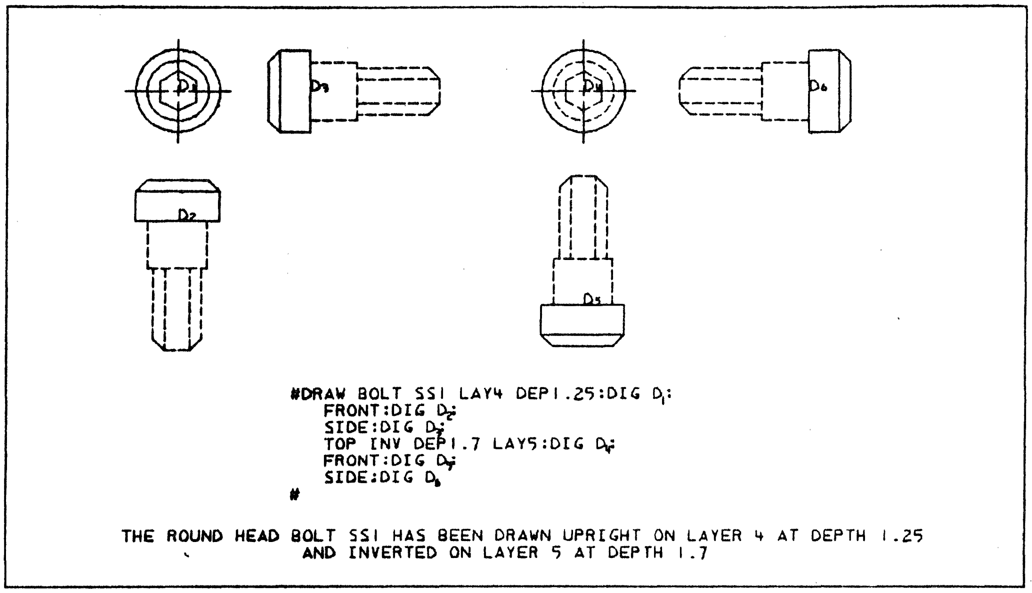
```
#DRA CLAMP SC1 VEC:DIG D1 D2:
    FRONT:DIG D3 D4:
    SIDE:DIG D5 D6:
    TOP:DIG D7 D8:
    FRONT:DIG D9 D10:
    SIDE:DIG D11 D12:
#
```

Figure 2.1j: Example of Auto-Draft graphic command structure using the VEC and View modifiers

LAY — Draw the component on the specified layer (default is current construction layer)

DEP — Draw the component at the specified Z-depth (default is current construction depth)

INV — Draw the component in an inverted (up-side down) position (default is non-inverted)



```
#DRAW BOLT SS1 LAY4 DEP1.25:DIG D1:
   FRONT:DIG D2:
   SIDE:DIG D3:
   TOP INV DEP1.7 LAY5:DIG D4:
   FRONT:DIG D5:
   SIDE:DIG D6
#

THE ROUND HEAD BOLT SS1 HAS BEEN DRAWN UPRIGHT ON LAYER 4 AT DEPTH 1.25
            AND INVERTED ON LAYER 5 AT DEPTH 1.7
```

Figure 2.1k: Example of Auto-Draft graphic command structure using the INV, DEP, LAY and View modifiers

Not all modifiers apply to all components. For example, strap clamps are not drawn inverted; orientation modifiers are meaningless for nuts, bolts, and washers. A colon (:) terminates the modifier section and initiates the component coordinate location, or GETDATA, processor.

The user inputs the coordinate location information via the GETDATA processor. When Auto-Draft is in the ANG mode the system prompt — DIG — is output to the command device. Subsequent digitizes select the origin locations for the active tool component. In the VEC mode two digitizes per component are required after the DIG prompt. The component is drawn such that its major axis is placed along the vector described by the two digitized points. In the PRP mode the system prompt — ENT — prompts the user to select the line, circle, or arc that he wants the component to be perpendicular to. Upon selection, the system prompts with DIG, after which the user digitizes the origin(s) of the component(s) to be drawn. A semicolon (;) entered in the PRP GETDATA mode will return to the ENT prompt. A colon returns the user to the modifier processor with all previously specified modifier states (chosen and default) still in effect.

2.1.2.2 Auto-Draft Graphics Output

This section outputs the entities that comprise the graphics representation of the component to the graphics database. It is divided into two subsections:

a. geometric graphics description;
b. non-graphic-related data.

Geometric graphics description subroutines are of two types: general geometric shapes and component-specific geometry. Auto-Draft is designed to build geometry by first defining basic entities (lines, circles, etc.), and then combining these to make simple shapes (rectangles, hexagons, etc.). These shapes are then combined into figures that represent parts of the component. Figure 2.1l shows the combination of three arcs, three lines, and a rectangle (open-topped) into a hex-head figure.

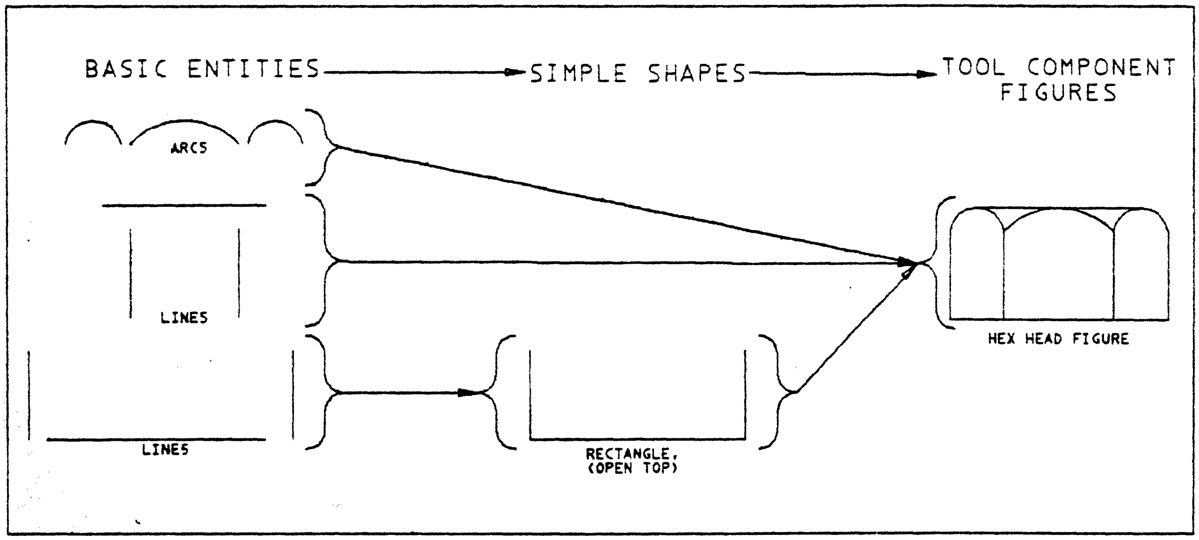


Figure 2.1l: Side view of the hex head figure

These figures are further combined with other figures to form the tooling component. Each component requires one or two subroutines to define which combination of figures, shapes, and/or entities should be used to represent front, side, or top views of the component. This is illustrated in Figure 2.1m.

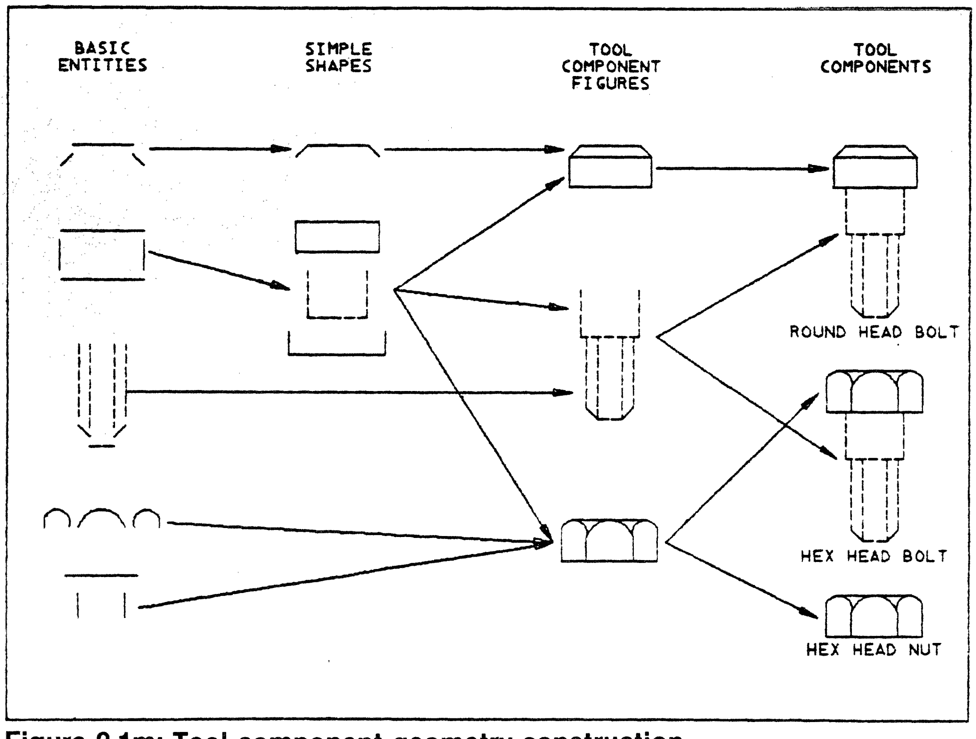


Figure 2.1m: Tool component geometry construction

Non-graphic-related data is associated with each component. The entities comprising a component are related to each other via a CV device called a "group." This allows the user and certain CADDS 3 utilities to treat the component as if it were a single entity. Movement or erasure of the component, for example, can be effected by selecting it as a group with a single digitize.

Each component has its identification (tool-ID) stored with it in the form of a "tag." The tag is text (six characters or less) that can be temporarily displayed via the CADDS command — DIS TAG.

Each component also has four "properties" associated with it. These are: component name; component identification (tool-ID); angle of orientation; and a flag indicating whether the component is inverted or not. These properties are stored on a "node," a CADDS 3 entity, included in the database for each component. These properties can be listed to the command device with the use of the CADDS command — LST PROP. The component identification is used by the BOM portion of Auto-Draft. The others are used in another application currently under development at General Dynamics.

Once all graphic and non-graphic data has been inserted into the database, the graphic representation of the component is inserted into the display, or REGEN, file for output to the display device.

2.2 Automatic Bill of Materials (BOM) Command

The command used to invoke the Bill of Materials feature of Auto-Draft is — RUN BOM. The verb RUN is a utility command provided by the CADDS 3 graphics system; therefore, no modification to the system's command (verb-noun) tables is required.

The top view of each component drawn with the Auto-Draft system contains a node with the tool-ID attached to it. Auto-Draft searches the drawing database for these nodes. When found, the tool-ID is extracted from the node and entered into a table that contains a count of each component's occurrence in the drawing. Upon completing the database search and building the tool-ID/count table, the user is prompted to input the file name that the BOM data will be written into. He is also prompted for the part number, tool symbol, tool number, number of setups, and T/D change (revision status). Auto-Draft then formats a header which is output to the BOM data file and the command device.

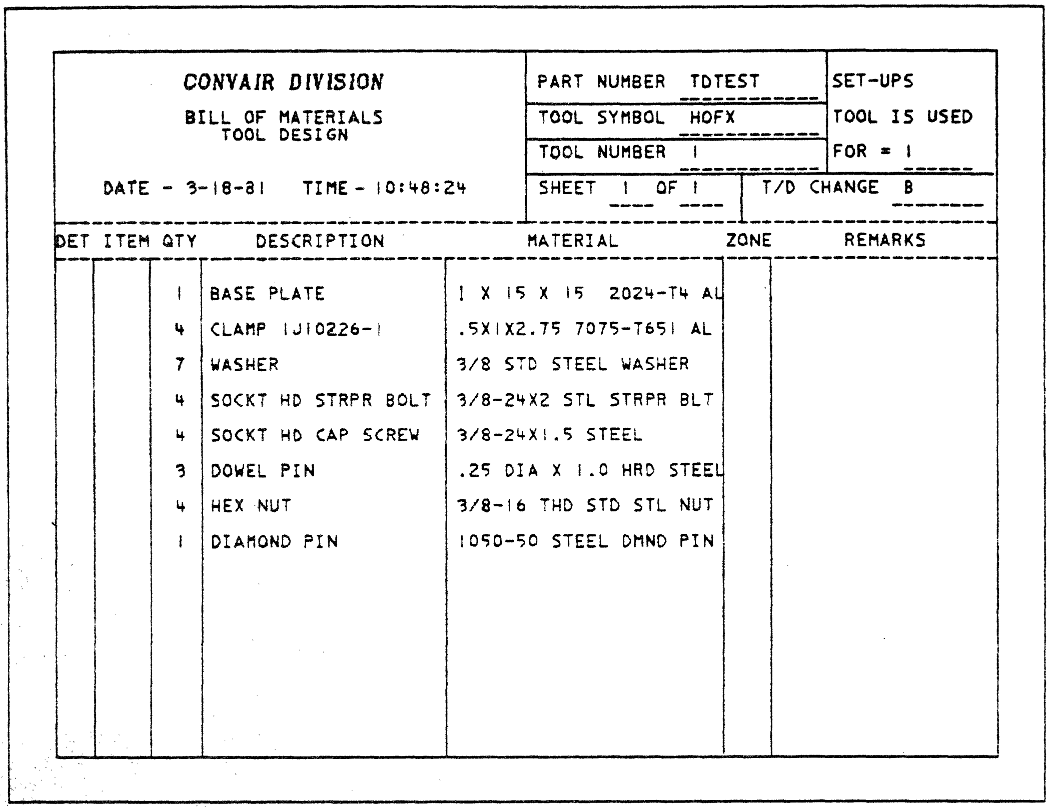
Attributes and material information for all components in the system are contained in another file which is then searched for a match of tool-ID's. If a match is found, then the information is copied to the BOM data file, along with the component count, and output to the command device. If no match is found, then the tool-ID, component count, and a message stating that no information was found for the component is output to the BOM data file and the command device.

The Auto-Draft system tracks the building of the BOM data file so that when thirteen components have been written to the file (enough to fill a page) a new header is output. The new header contains all of the same data previously entered by the user and, additionally, an automatic count of the pages. This count is entered into the 'SHEET ____ OF ____' section of the header.

If no Auto-Draft components were used in the drawing, then a message is output to the command device informing the user of this fact. The RUN BOM command is then exited.

The BOM form used in Convair's Tool Design department is stored as a CADDS 3 part. While in CADDS, an execute file* is used to insert the form and the data in the BOM data file into the drawing (on a layer reserved for this use). Figure 2.2a is an example of the BOM format.

*Execute files are a series of commands, and answers to prompts that may be invoked with the use of the command — EXECUTE.



CONVAIR DIVISION BILL OF MATERIALS TOOL DESIGN	PART NUMBER TDTEST	SET-UPS
	TOOL SYMBOL HOFX	TOOL IS USED
	TOOL NUMBER 1	FOR = 1
DATE - 3-18-81 TIME - 10:48:24	SHEET 1 OF 1	T/D CHANGE B

DET	ITEM	QTY	DESCRIPTION	MATERIAL	ZONE	REMARKS
		1	BASE PLATE	1 X 15 X 15 2024-T4 AL		
		4	CLAMP 1J10226-1	.5X1X2.75 7075-T651 AL		
		7	WASHER	3/8 STD STEEL WASHER		
		4	SOCKT HD STRPR BOLT	3/8-24X2 STL STRPR BLT		
		4	SOCKT HD CAP SCREW	3/8-24X1.5 STEEL		
		3	DOWEL PIN	.25 DIA X 1.0 HRD STEEL		
		4	HEX NUT	3/8-16 THD STD STL NUT		
		1	DIAMOND PIN	1050-50 STEEL DMND PIN		

Figure 2.2a: Auto-Draft Bill of Materials

3. SUMMARY

The Auto-Draft system is used to quickly and accurately draw tooling components, such as diamond pins, clamps, and base plates, in tool design drawings produced on Computervision. All figures of components contained in this paper were drawn using Auto-Draft. The Bill of Materials report (Figure 2.2a) was generated from a "tool design" that included Auto-Draft components.

Modifications to existing component parameter files are easily effected outside of Auto-Draft. Changes are made while in the CV operating system, using the text editor.

New component groups may be added to Auto-Draft. This requires inserting the new name into the Component Name Table and adding the appropriate geometry-construction subroutines to the Auto-Draft software.

The command DRAW, which invokes the graphics portion of Auto-Draft, requires a one-time modification made to the standard CADDS 3 verb-noun tables. This is so regardless of any future modifications to or additions of components.

The Auto-Draft command interpreter was designed as a general purpose device. This will allow the application of Auto-Draft to *ANY* design task that uses a library of standard components. Different Component Name Tables and their corresponding geometric graphics description subroutines are loaded independent of the Auto-Draft interpreter.

Likewise, data contained in component attribute-and material-information files are easily built to reflect new applications. Multiple forms, for each application, can be stored on disk; easily created or updated as a CADDS 3 part.

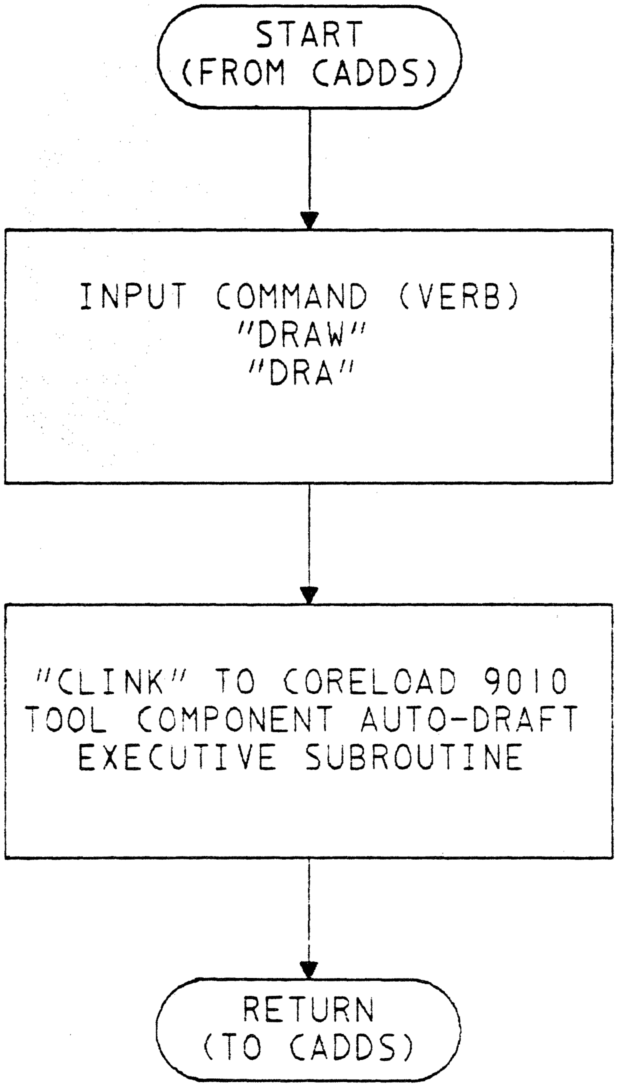
REFERENCES

1. "Computervision Programming Languages (FORTRAN-S/TPL)"
   Copyright August 1978, Computervision Corporation USA,
   order no. 001-00001
2. "CADDS 3 Graphics Operator Manual"
   Copyright June 1979, Computervision Corporation,
   order no. 001-00201
3. "CADDS 3 Database Reference Manual"
   Copyright March 1978, Computervision Corporation,
   order no. 001-00007
4. "CGOS 100/10 Operator Manual"
   Copyright 1979, Computervision Corporation,
   order no. 001-00010-000
5. "Tool Design Manual"
   Tool Engineering, Tool Design Department 433-0
   General Dynamics/Convair Division

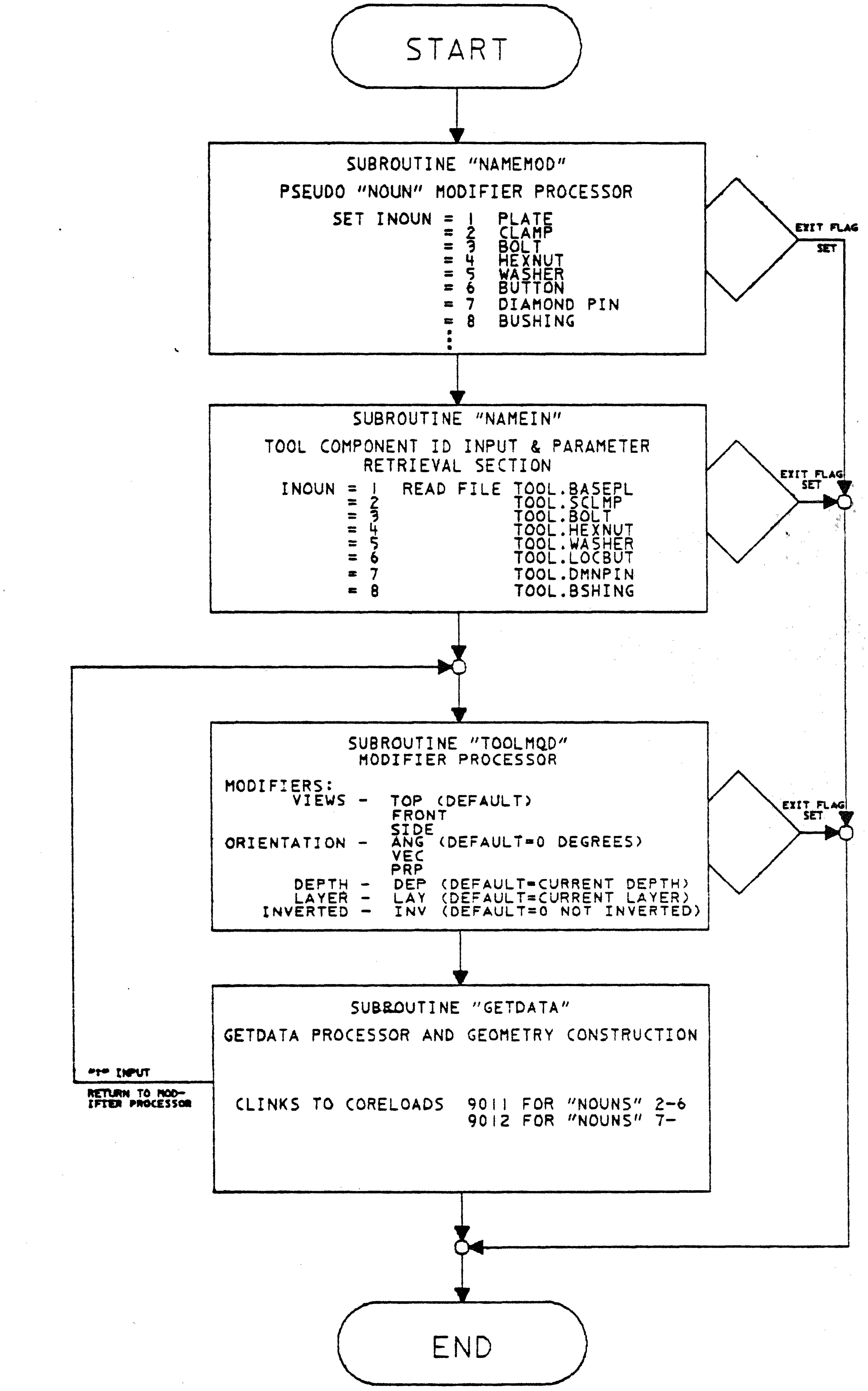
Appendix A

Auto-Draft System Flowcharts

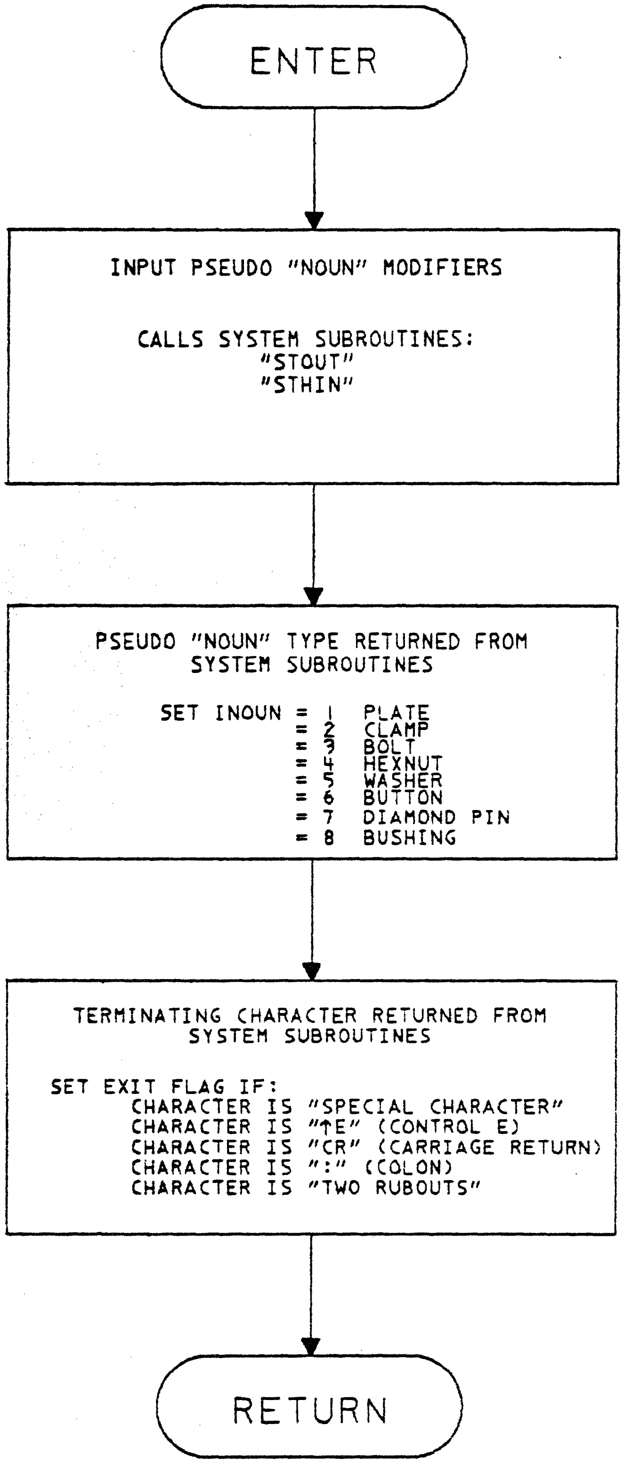
GENERAL DYNAMICS/CONVAIR DIVISION
TOOL COMPONENT AUTO-DRAFT SYSTEM
DEVELOPED FOR IMPLEMENTATION ON COMPUTERVISION



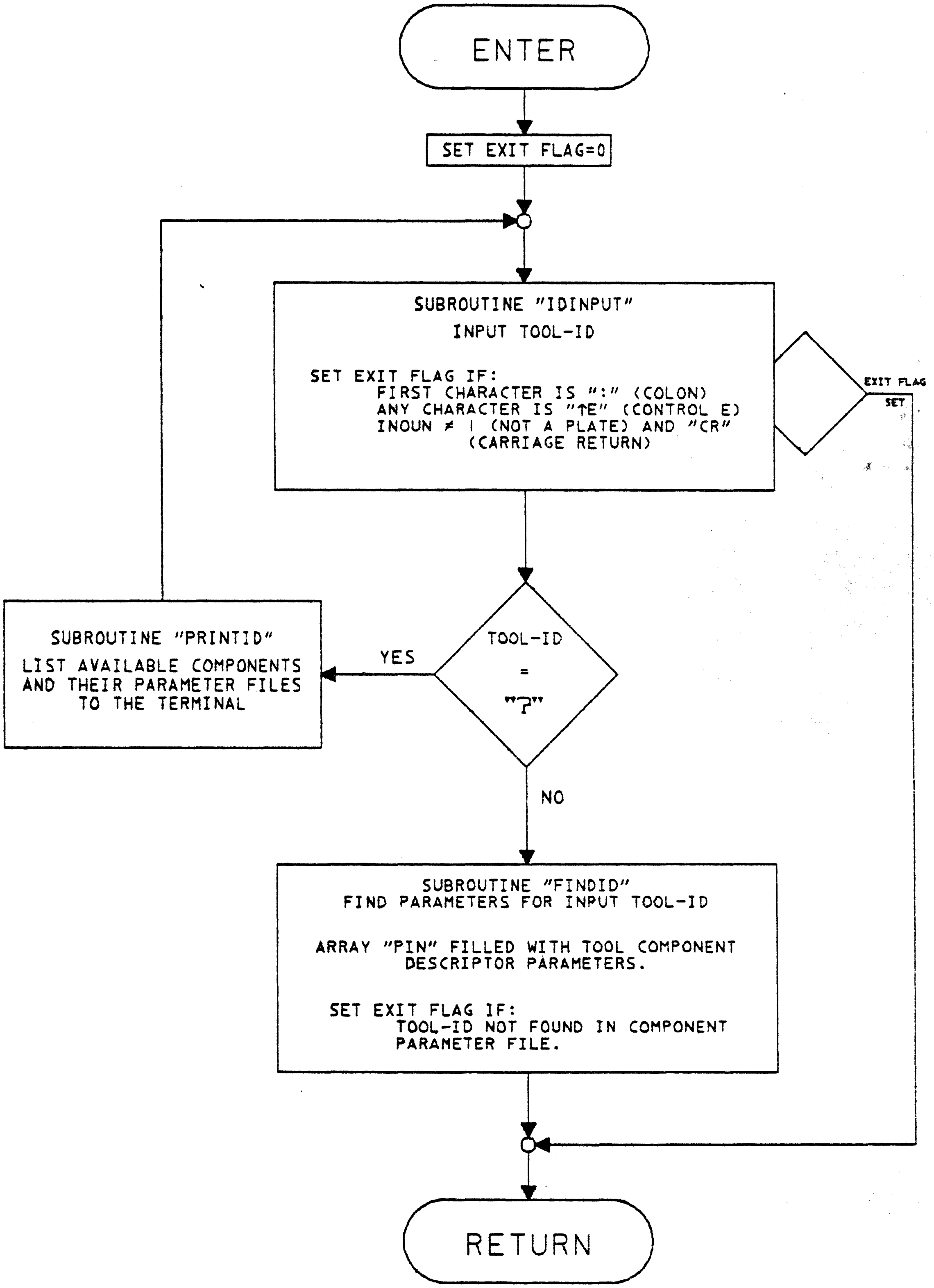
SUBROUTINE "MAIN9010"
EXECUTIVE SUBROUTINE OF CORELOAD 9010



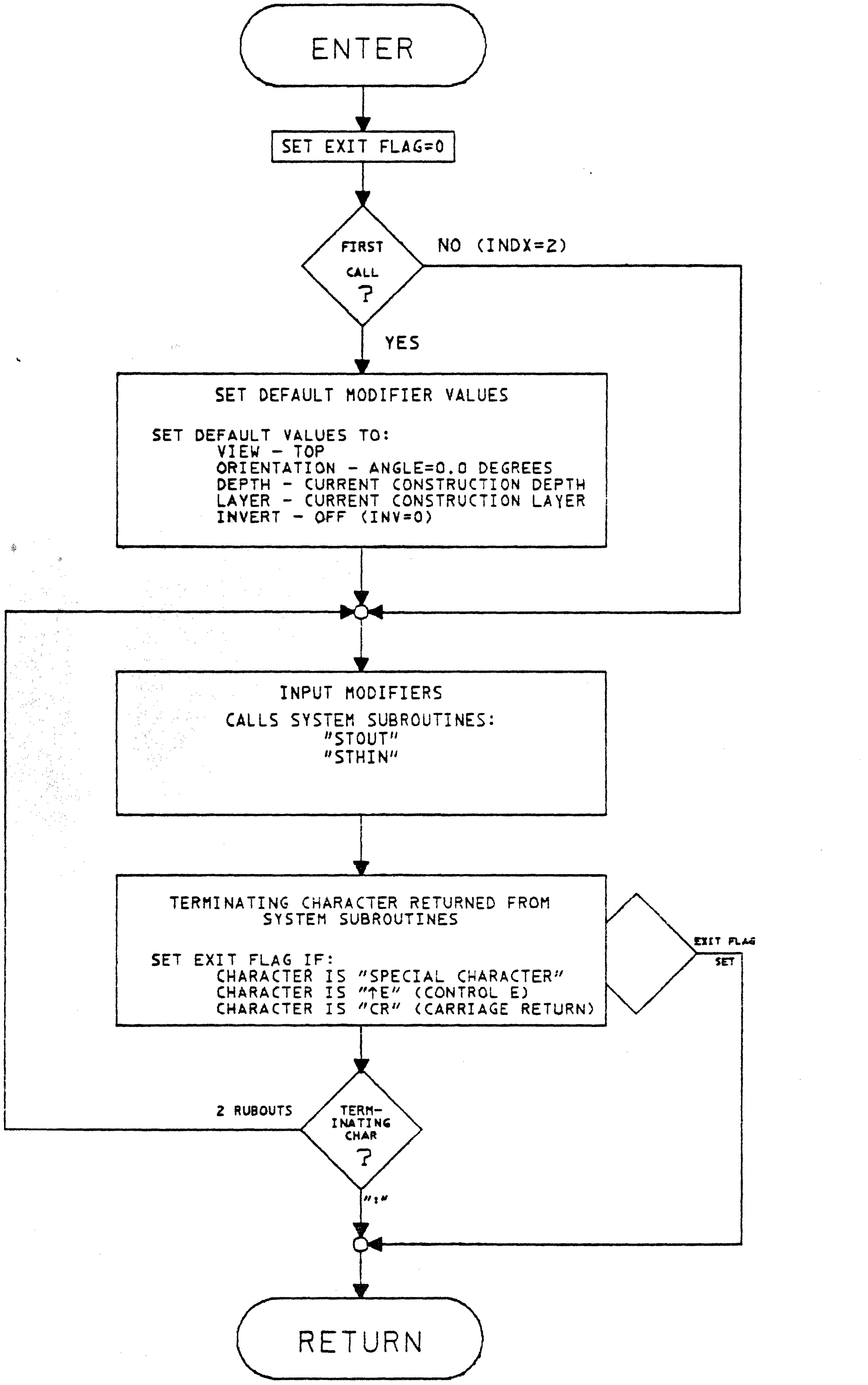
SUBROUTINE "NAMEMOD"
PSEUDO "NOUN" MODIFIER PROCESSOR


ENTER
INPUT PSEUDO "NOUN" MODIFIERS
CALLS SYSTEM SUBROUTINES:
"STOUT"
"STHIN"
PSEUDO "NOUN" TYPE RETURNED FROM SYSTEM SUBROUTINES
SET INOUN = 1 PLATE
= 2 CLAMP
= 3 BOLT
= 4 HEXNUT
= 5 WASHER
= 6 BUTTON
= 7 DIAMOND PIN
= 8 BUSHING
TERMINATING CHARACTER RETURNED FROM SYSTEM SUBROUTINES
SET EXIT FLAG IF:
CHARACTER IS "SPECIAL CHARACTER"
CHARACTER IS "↑E" (CONTROL E)
CHARACTER IS "CR" (CARRIAGE RETURN)
CHARACTER IS ":" (COLON)
CHARACTER IS "TWO RUBOUTS"
RETURN

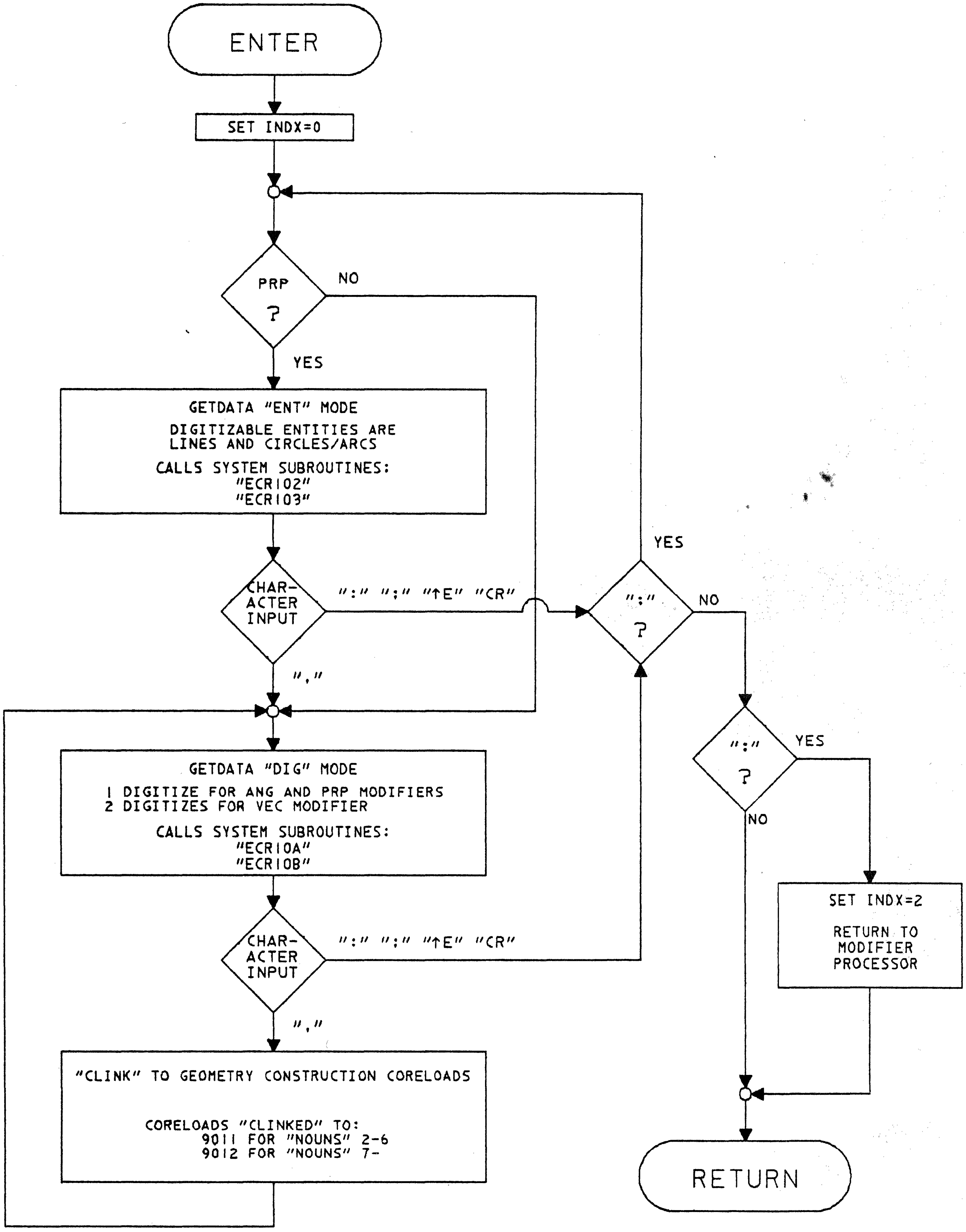
SUBROUTINE "NAMEIN"
TOOL COMPONENT ID INPUT AND RETRIEVAL SECTION


ENTER
SET EXIT FLAG=0
SUBROUTINE "IDINPUT"
INPUT TOOL-ID
SET EXIT FLAG IF:
FIRST CHARACTER IS ":" (COLON)
ANY CHARACTER IS "↑E" (CONTROL E)
INOUN ≠ 1 (NOT A PLATE) AND "CR"
(CARRIAGE RETURN)
EXIT FLAG
SET
TOOL-ID
=
"?"
YES
SUBROUTINE "PRINTID"
LIST AVAILABLE COMPONENTS
AND THEIR PARAMETER FILES
TO THE TERMINAL
NO
SUBROUTINE "FINDID"
FIND PARAMETERS FOR INPUT TOOL-ID
ARRAY "PIN" FILLED WITH TOOL COMPONENT
DESCRIPTOR PARAMETERS.
SET EXIT FLAG IF:
TOOL-ID NOT FOUND IN COMPONENT
PARAMETER FILE.
RETURN

SUBROUTINE "TOOLMOD"
MODIFIER PROCESSOR FOR VIEW AND ORIENTATION

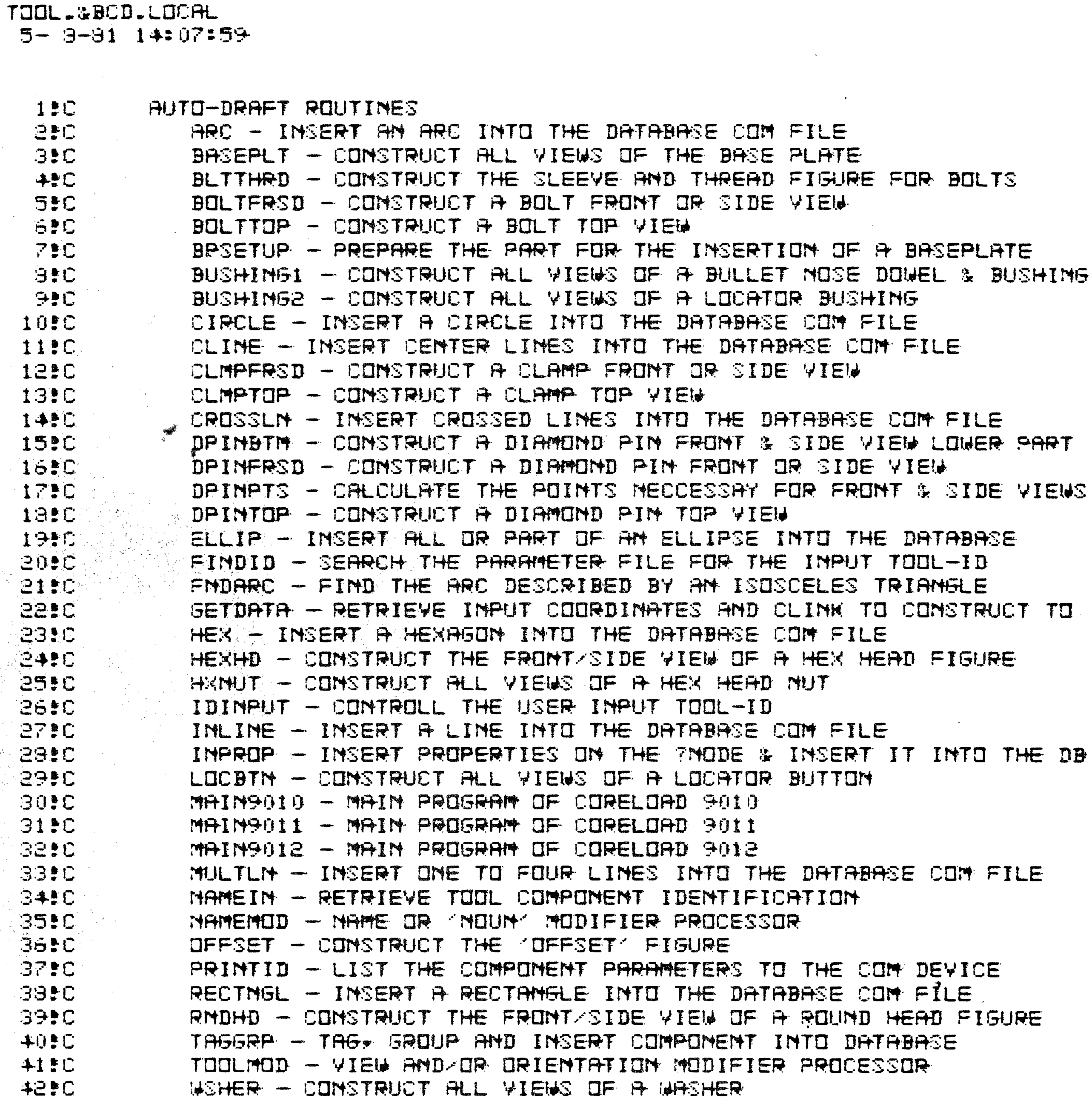

ENTER
SET EXIT FLAG=0
FIRST CALL ?
NO (INDX=2)
YES
SET DEFAULT MODIFIER VALUES
SET DEFAULT VALUES TO:
VIEW - TOP
ORIENTATION - ANGLE=0.0 DEGREES
DEPTH - CURRENT CONSTRUCTION DEPTH
LAYER - CURRENT CONSTRUCTION LAYER
INVERT - OFF (INV=0)
INPUT MODIFIERS
CALLS SYSTEM SUBROUTINES:
"STOUT"
"STHIN"
TERMINATING CHARACTER RETURNED FROM SYSTEM SUBROUTINES
SET EXIT FLAG IF:
CHARACTER IS "SPECIAL CHARACTER"
CHARACTER IS "↑E" (CONTROL E)
CHARACTER IS "CR" (CARRIAGE RETURN)
EXIT FLAG SET
2 RUBOUTS
TERM- INATING CHAR ?
";"
RETURN

SUBROUTINE "GETDATA"
GETDATA PROCESSOR AND GEOMETRY CONSTRUCTION


ENTER
SET INDX=0
PRP ?
NO
YES
GETDATA "ENT" MODE
DIGITIZABLE ENTITIES ARE LINES AND CIRCLES/ARCS
CALLS SYSTEM SUBROUTINES: "ECR102" "ECR103"
CHAR-ACTER INPUT
":" ";" "↑E" "CR"
","
":" ?
YES
NO
GETDATA "DIG" MODE
1 DIGITIZE FOR ANG AND PRP MODIFIERS
2 DIGITIZES FOR VEC MODIFIER
CALLS SYSTEM SUBROUTINES: "ECR10A" "ECR10B"
CHAR-ACTER INPUT
":" ";" "↑E" "CR"
","
";" ?
YES
NO
SET INDX=2
RETURN TO MODIFIER PROCESSOR
"CLINK" TO GEOMETRY CONSTRUCTION CORELOADS
CORELOADS "CLINKED" TO:
9011 FOR "NOUNS" 2-6
9012 FOR "NOUNS" 7-
RETURN

Appendix B

Auto-Draft System Subroutines

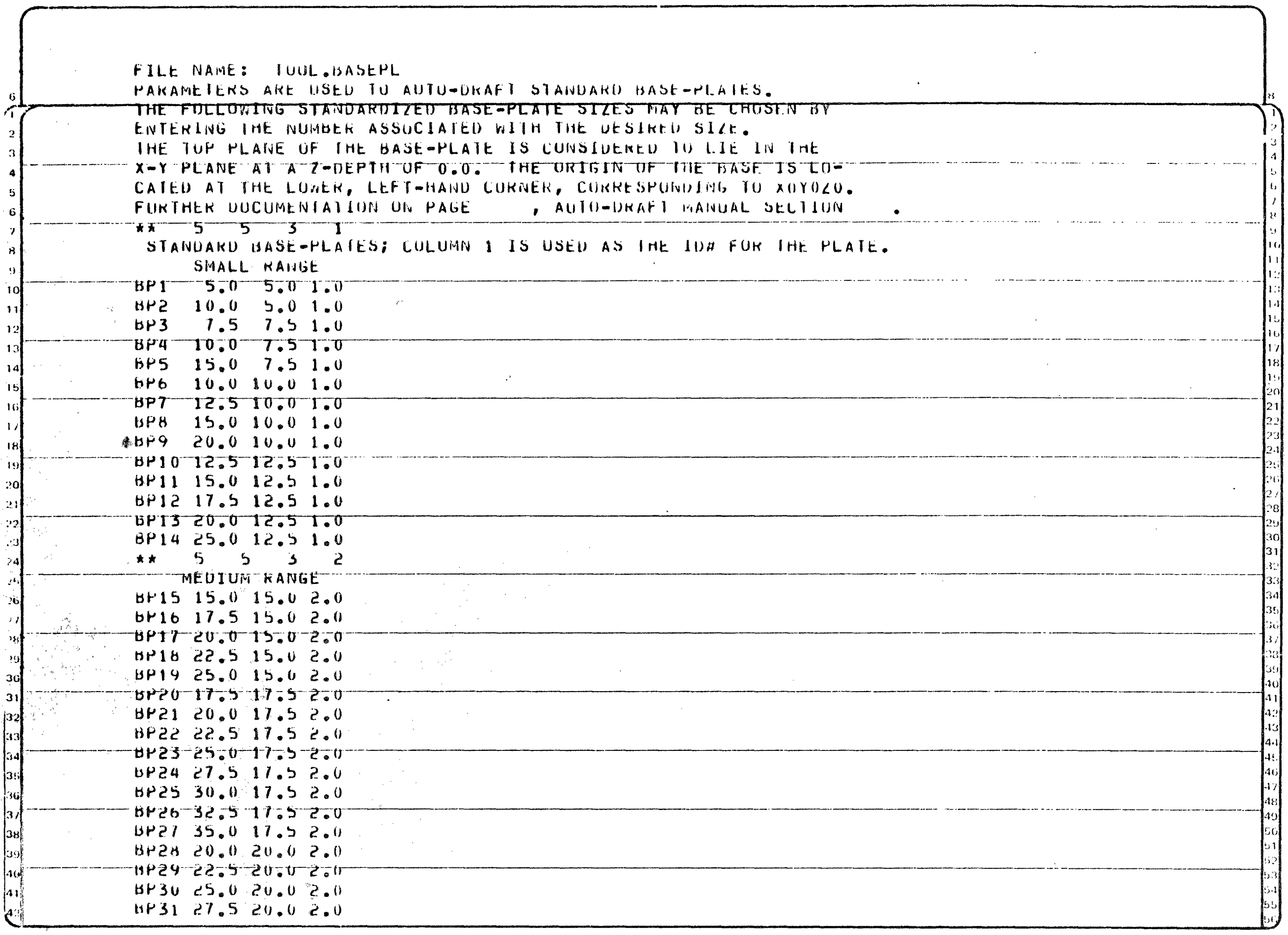


```
TOOL.&BCD.LOCAL
 5- 8-81 14:07:59


   1!C      AUTO-DRAFT ROUTINES
   2!C         ARC - INSERT AN ARC INTO THE DATABASE COM FILE
   3!C         BASEPLT - CONSTRUCT ALL VIEWS OF THE BASE PLATE
   4!C         BLTTHRD - CONSTRUCT THE SLEEVE AND THREAD FIGURE FOR BOLTS
   5!C         BOLTFRSD - CONSTRUCT A BOLT FRONT OR SIDE VIEW
   6!C         BOLTTOP - CONSTRUCT A BOLT TOP VIEW
   7!C         BPSETUP - PREPARE THE PART FOR THE INSERTION OF A BASEPLATE
   8!C         BUSHING1 - CONSTRUCT ALL VIEWS OF A BULLET NOSE DOWEL & BUSHING
   9!C         BUSHING2 - CONSTRUCT ALL VIEWS OF A LOCATOR BUSHING
  10!C         CIRCLE - INSERT A CIRCLE INTO THE DATABASE COM FILE
  11!C         CLINE - INSERT CENTER LINES INTO THE DATABASE COM FILE
  12!C         CLMPFRSD - CONSTRUCT A CLAMP FRONT OR SIDE VIEW
  13!C         CLMPTOP - CONSTRUCT A CLAMP TOP VIEW
  14!C         CROSSLN - INSERT CROSSED LINES INTO THE DATABASE COM FILE
  15!C         DPINBTM - CONSTRUCT A DIAMOND PIN FRONT & SIDE VIEW LOWER PART
  16!C         DPINFRSD - CONSTRUCT A DIAMOND PIN FRONT OR SIDE VIEW
  17!C         DPINPTS - CALCULATE THE POINTS NECCESSAY FOR FRONT & SIDE VIEWS
  18!C         DPINTOP - CONSTRUCT A DIAMOND PIN TOP VIEW
  19!C         ELLIP - INSERT ALL OR PART OF AN ELLIPSE INTO THE DATABASE
  20!C         FINDID - SEARCH THE PARAMETER FILE FOR THE INPUT TOOL-ID
  21!C         FNDARC - FIND THE ARC DESCRIBED BY AN ISOSCELES TRIANGLE
  22!C         GETDATA - RETRIEVE INPUT COORDINATES AND CLINK TO CONSTRUCT TO
  23!C         HEX - INSERT A HEXAGON INTO THE DATABASE COM FILE
  24!C         HEXHD - CONSTRUCT THE FRONT/SIDE VIEW OF A HEX HEAD FIGURE
  25!C         HXNUT - CONSTRUCT ALL VIEWS OF A HEX HEAD NUT
  26!C         IDINPUT - CONTROLL THE USER INPUT TOOL-ID
  27!C         INLINE - INSERT A LINE INTO THE DATABASE COM FILE
  28!C         INPROP - INSERT PROPERTIES ON THE 7NODE & INSERT IT INTO THE DB
  29!C         LOCBTN - CONSTRUCT ALL VIEWS OF A LOCATOR BUTTON
  30!C         MAIN9010 - MAIN PROGRAM OF CORELOAD 9010
  31!C         MAIN9011 - MAIN PROGRAM OF CORELOAD 9011
  32!C         MAIN9012 - MAIN PROGRAM OF CORELOAD 9012
  33!C         MULTLN - INSERT ONE TO FOUR LINES INTO THE DATABASE COM FILE
  34!C         NAMEIN - RETRIEVE TOOL COMPONENT IDENTIFICATION
  35!C         NAMEMOD - NAME OR 'NOUN' MODIFIER PROCESSOR
  36!C         OFFSET - CONSTRUCT THE 'OFFSET' FIGURE
  37!C         PRINTID - LIST THE COMPONENT PARAMETERS TO THE COM DEVICE
  38!C         RECTNGL - INSERT A RECTANGLE INTO THE DATABASE COM FILE
  39!C         RNDHD - CONSTRUCT THE FRONT/SIDE VIEW OF A ROUND HEAD FIGURE
  40!C         TAGGRP - TAG, GROUP AND INSERT COMPONENT INTO DATABASE
  41!C         TOOLMOD - VIEW AND/OR ORIENTATION MODIFIER PROCESSOR
  42!C         WSHER - CONSTRUCT ALL VIEWS OF A WASHER
```

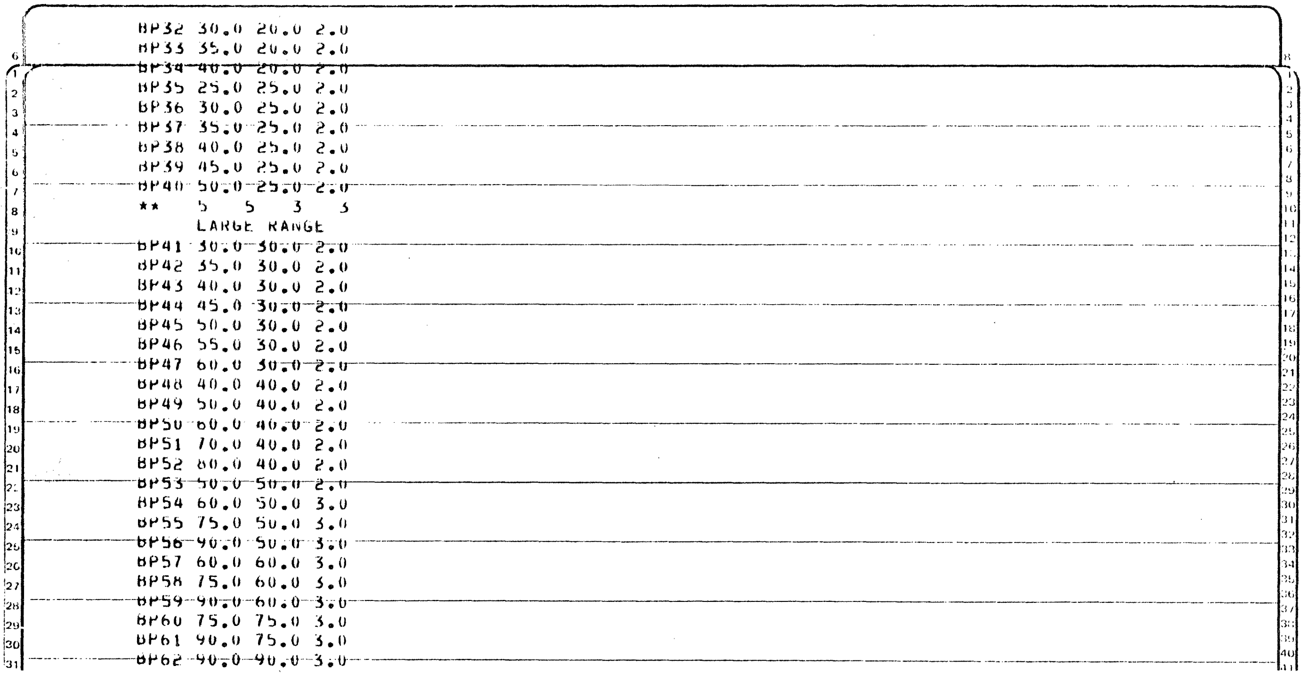
```
TOOL.&BCD.SYSTEM
 5-22-81 10:55:30


  1!C      SYSTEM ROUTINES
  2!C         ABS - ABSOLUTE VALUE
  3!C         ACOS - CALCULATE THE ARCCOSINE OF AN ANGLE
  4!C         ATAN - CALCULATE THE ARCTANGENT OF AN ANGLE
  5!C         ADLIST - ADD A LIST OF ENTITIES TO THE GENERATION SYSTEM
  6!C         AU02 - MODIFY "MI" DESCRIPTOR WORDS
  7!C         AU03 - UNPACK MASTER INDEX ENTRY
  8!C         BREAK - SEE IF AN INTERRUPT KEY HAS BEEN HIT;
  9!C                 PAUSE, AND/OR PASS CONTROL TO THE CORRECT ROUTINE
 10!C         BRKSET - DESIGNATE THE CORRECT ROUTINE
 11!C         CLINK (65) - REGEN RE-NORMALIZATION CORELOAD
 12!C         CLINK (9010) - RETURN TO TOOL COMPONENT AUTO-DRAFT INTERPERTER
 13!C         CLINK (9011) - TOOL COMPONENT AUTO-DRAFT GEOMETRY CORELOAD #1
 14!C         CLINK (9012) - TOOL COMPONENT AUTO-DRAFT GEOMETRY CORELOAD #2
 15!C         COS - CALCULATE THE COSINE OF AN ANGLE
 16!C         DBTFRE - FREE TASK DATABASE SCRATCH SPACE
 17!C         DBTRD - READ/WRITE DATA TO/FROM THE TASK DATA BASE
 18!C         DM127 - MODIFY THE EXTENTS IN THE PDF
 19!C         DW50 - OPEN A PDF RECORD FOR CREATE
 20!C         DW54 - CLOSE A CREATED PDF RECORD & ACCEPT THE ENTITY IN THE DB
 21!C         DW55 - STORE AN ARRAY AS A SUBRECORD DURING A CREATE SEQUENCE
 22!C         DW122R - INSERT A REAL PROPERTY ON THE ?NODE
 23!C         DW122T - INSERT A TEXT PROPERTY ON THE ?NODE
 24!C         DW203 - CREATES A LIN IN THE DATABASE
 25!C         DW205 - CREATES AN ARC/CIRCLE IN THE DATABASE
 26!C         ECR10A - APPLICATION INTERFACE ROUITNE FOR CALLING GETDATA.
 27!C                  CALLS GETDATA BOTH TO INITIALIZE & RETRIEVE DATA
 28!C         ECR10B - RETURN TO APPLICATION CODE DISK LOCATION OF
 29!C                  REQUESTED DATA
 30!C         ECR102 - RETRIEVE DATA INPUT FROM GETDATA
 31!C         ECR103 - APPLICATION INTERFACE ROUITNE FOR CALLING GETDATA.
 32!C                  CALLS GETDATA BOTH TO INITIALIZE & RETRIEVE DATA
 33!C         ECR106 - CLEAR GETDATA FOR NEW COMMAND
 34!C         GETCLS - CLOSE A FILE
 35!C         GETFIL - OPEN A FILE
 36!C         GETLIN - READ IN A LINE FROM A FILE
 37!C         GETSKP - SKIP LINES IN A FILE
 38!C         GRP13 - TURN ON ATTENTION BIT
 39!C         MU01P - GENERATE A PERPENDICULAR FROM A POINT TO A LINE
 40!C         MU02P - GENERATE A PERPENDICULAR FROM A POINT TO A CIRCLE.
 41!C                 NOTE...THE CLOSEST PERPENDICULAR POINT IS RETURNED.
 42!C         MVP07 - SET THE EXTENTS IN THE PDF
 43!C         SIN - CALCULATE THE SINE OF AN ANGLE
 44!C         TAN - CALAULATE THE TANGENT OF AN ANGLE
 45!C         TAGENT - TAG SPECIFIED ENTITY WITH SIX(6) CHARACTERS
 46!C         TYPE - OUTPUT A MESSAGE TO THE TERMINAL WITH A CARRIAGE RETURN
 47!C                AND A LINE FEED
 48!C         TYPOUT - OUTPUT A MESSAGE TO THE TERMINAL WITHOUT A CARRIAGE
 49!C                  RETURN OR A LINE FEED
```

Appendix C

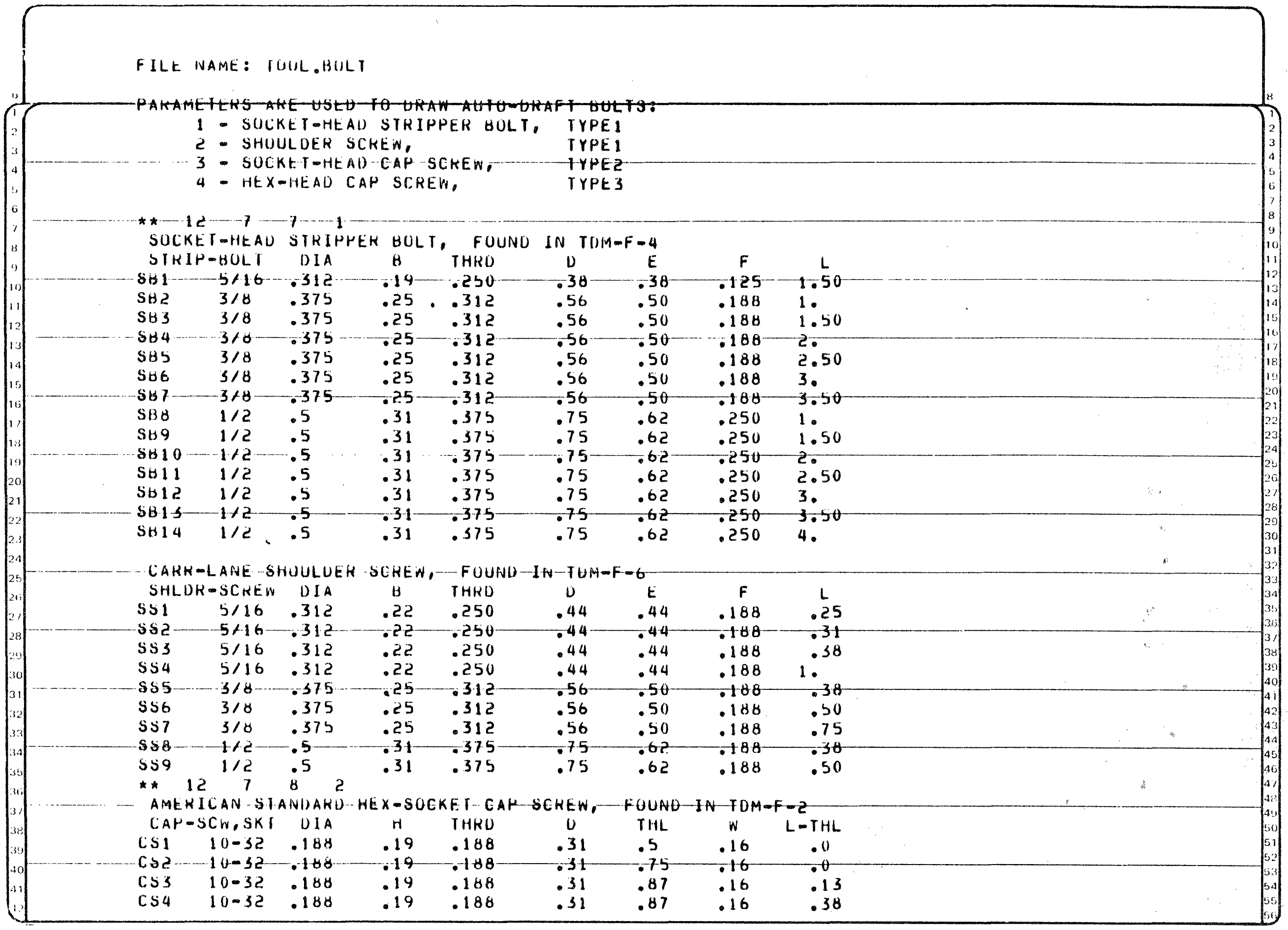
Parameter Files for Existing Components



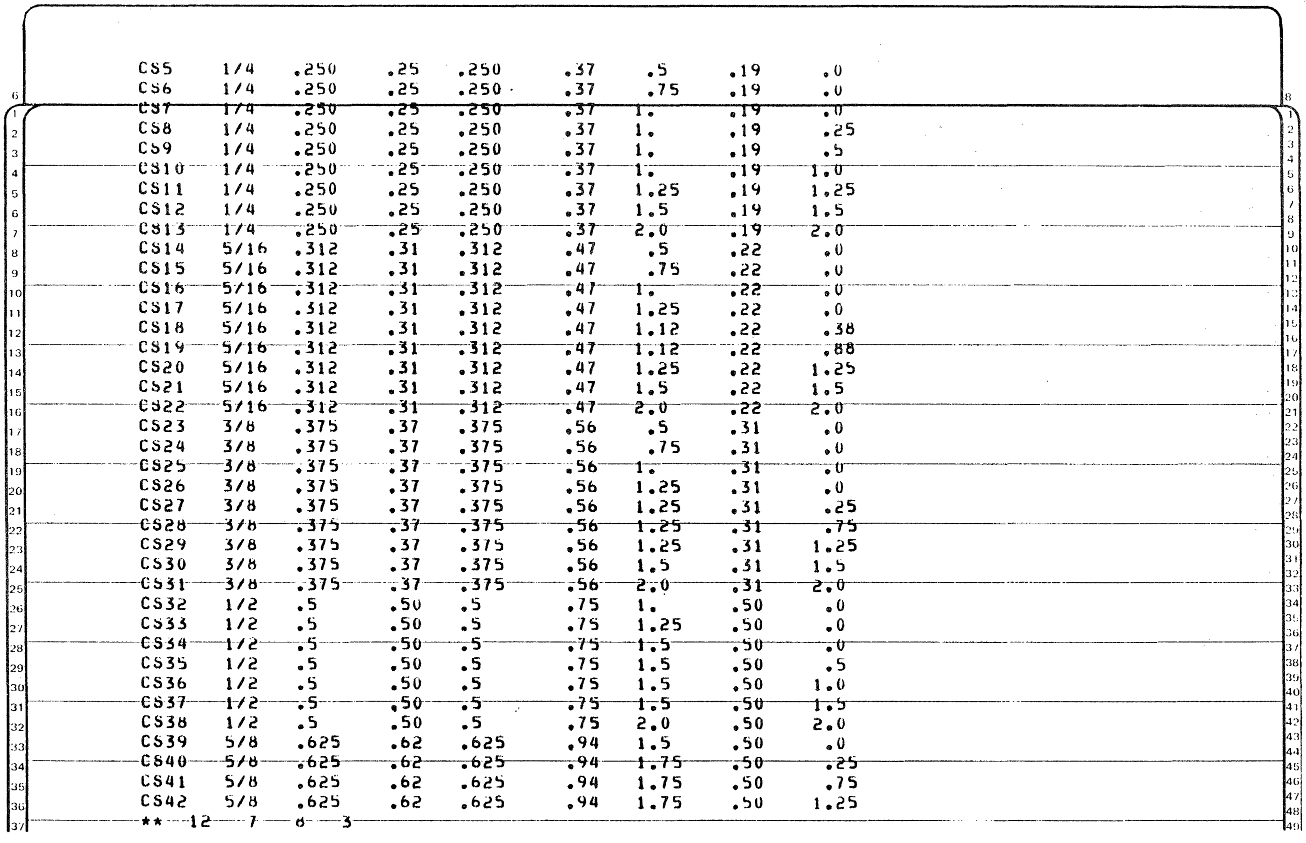
```
FILE NAME:  TOOL.BASEPL
PARAMETERS ARE USED TO AUTO-DRAFT STANDARD BASE-PLATES.
THE FOLLOWING STANDARDIZED BASE-PLATE SIZES MAY BE CHOSEN BY
ENTERING THE NUMBER ASSOCIATED WITH THE DESIRED SIZE.
THE TOP PLANE OF THE BASE-PLATE IS CONSIDERED TO LIE IN THE
X-Y PLANE AT A Z-DEPTH OF 0.0.  THE ORIGIN OF THE BASE IS LO-
CATED AT THE LOWER, LEFT-HAND CORNER, CORRESPONDING TO X0Y0Z0.
FURTHER DOCUMENTATION ON PAGE      , AUTO-DRAFT MANUAL SECTION    .
**   5   5   3   1
 STANDARD BASE-PLATES; COLUMN 1 IS USED AS THE ID# FOR THE PLATE.
      SMALL RANGE
BP1    5.0  5.0 1.0
BP2   10.0  5.0 1.0
BP3    7.5  7.5 1.0
BP4   10.0  7.5 1.0
BP5   15.0  7.5 1.0
BP6   10.0 10.0 1.0
BP7   12.5 10.0 1.0
BP8   15.0 10.0 1.0
BP9   20.0 10.0 1.0
BP10 12.5 12.5 1.0
BP11 15.0 12.5 1.0
BP12 17.5 12.5 1.0
BP13 20.0 12.5 1.0
BP14 25.0 12.5 1.0
**    5    5    3    2
     MEDIUM RANGE
BP15 15.0 15.0 2.0
BP16 17.5 15.0 2.0
BP17 20.0 15.0 2.0
BP18 22.5 15.0 2.0
BP19 25.0 15.0 2.0
BP20 17.5 17.5 2.0
BP21 20.0 17.5 2.0
BP22 22.5 17.5 2.0
BP23 25.0 17.5 2.0
BP24 27.5 17.5 2.0
BP25 30.0 17.5 2.0
BP26 32.5 17.5 2.0
BP27 35.0 17.5 2.0
BP28 20.0 20.0 2.0
BP29 22.5 20.0 2.0
BP30 25.0 20.0 2.0
BP31 27.5 20.0 2.0
```



```
BP32 30.0 20.0 2.0
BP33 35.0 20.0 2.0
BP34 40.0 20.0 2.0
BP35 25.0 25.0 2.0
BP36 30.0 25.0 2.0
BP37 35.0 25.0 2.0
BP38 40.0 25.0 2.0
BP39 45.0 25.0 2.0
BP40 50.0 25.0 2.0
**    5    5    3    3
      LARGE RANGE
BP41 30.0 30.0 2.0
BP42 35.0 30.0 2.0
BP43 40.0 30.0 2.0
BP44 45.0 30.0 2.0
BP45 50.0 30.0 2.0
BP46 55.0 30.0 2.0
BP47 60.0 30.0 2.0
BP48 40.0 40.0 2.0
BP49 50.0 40.0 2.0
BP50 60.0 40.0 2.0
BP51 70.0 40.0 2.0
BP52 80.0 40.0 2.0
BP53 50.0 50.0 2.0
BP54 60.0 50.0 3.0
BP55 75.0 50.0 3.0
BP56 90.0 50.0 3.0
BP57 60.0 60.0 3.0
BP58 75.0 60.0 3.0
BP59 90.0 60.0 3.0
BP60 75.0 75.0 3.0
BP61 90.0 75.0 3.0
BP62 90.0 90.0 3.0
```

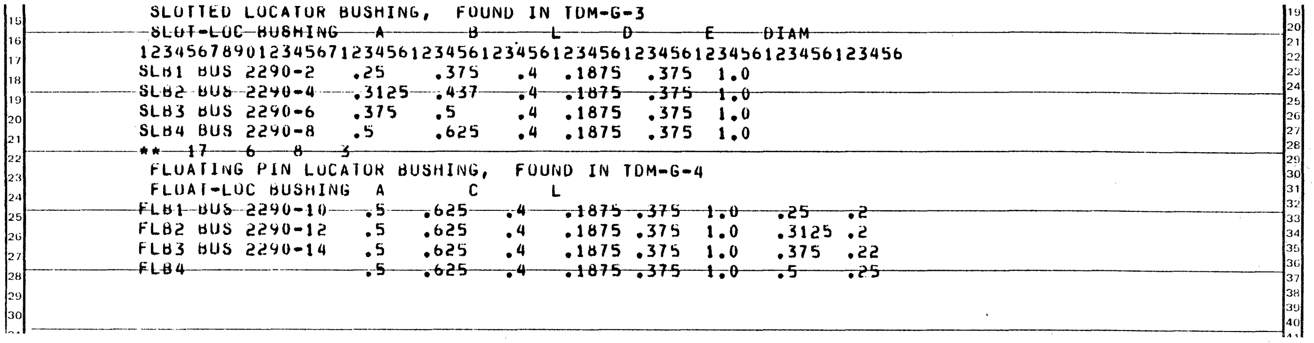


```
FILE NAME: TOOL.BOLT

PARAMETERS ARE USED TO DRAW AUTO-DRAFT BOLTS:
      1 - SOCKET-HEAD STRIPPER BOLT,  TYPE1
      2 - SHOULDER SCREW,             TYPE1
      3 - SOCKET-HEAD CAP SCREW,      TYPE2
      4 - HEX-HEAD CAP SCREW,         TYPE3

** 12  7  7  1
 SOCKET-HEAD STRIPPER BOLT,  FOUND IN TDM-F-4
 STRIP-BOLT   DIA      B     THRD       D       E        F       L
SB1    5/16  .312     .19   .250       .38     .38     .125    1.50
SB2    3/8   .375     .25 . .312       .56     .50     .188    1.
SB3    3/8   .375     .25   .312       .56     .50     .188    1.50
SB4    3/8   .375     .25   .312       .56     .50     .188    2.
SB5    3/8   .375     .25   .312       .56     .50     .188    2.50
SB6    3/8   .375     .25   .312       .56     .50     .188    3.
SB7    3/8   .375     .25   .312       .56     .50     .188    3.50
SB8    1/2   .5       .31   .375       .75     .62     .250    1.
SB9    1/2   .5       .31   .375       .75     .62     .250    1.50
SB10   1/2   .5       .31   .375       .75     .62     .250    2.
SB11   1/2   .5       .31   .375       .75     .62     .250    2.50
SB12   1/2   .5       .31   .375       .75     .62     .250    3.
SB13   1/2   .5       .31   .375       .75     .62     .250    3.50
SB14   1/2   .5       .31   .375       .75     .62     .250    4.

 CARR-LANE SHOULDER SCREW,  FOUND IN TDM-F-6
 SHLDR-SCREW  DIA      B     THRD       D       E        F       L
SS1    5/16  .312     .22   .250       .44     .44     .188     .25
SS2    5/16  .312     .22   .250       .44     .44     .188     .31
SS3    5/16  .312     .22   .250       .44     .44     .188     .38
SS4    5/16  .312     .22   .250       .44     .44     .188    1.
SS5    3/8   .375     .25   .312       .56     .50     .188     .38
SS6    3/8   .375     .25   .312       .56     .50     .188     .50
SS7    3/8   .375     .25   .312       .56     .50     .188     .75
SS8    1/2   .5       .31   .375       .75     .62     .188     .38
SS9    1/2   .5       .31   .375       .75     .62     .188     .50
**  12   7   8   2
 AMERICAN STANDARD HEX-SOCKET CAP SCREW,  FOUND IN TDM-F-2
 CAP-SCW,SKT  DIA      H     THRD       D      THL       W     L-THL
CS1   10-32  .188     .19   .188       .31     .5      .16      .0
CS2   10-32  .188     .19   .188       .31     .75     .16      .0
CS3   10-32  .188     .19   .188       .31     .87     .16      .13
CS4   10-32  .188     .19   .188       .31     .87     .16      .38
```

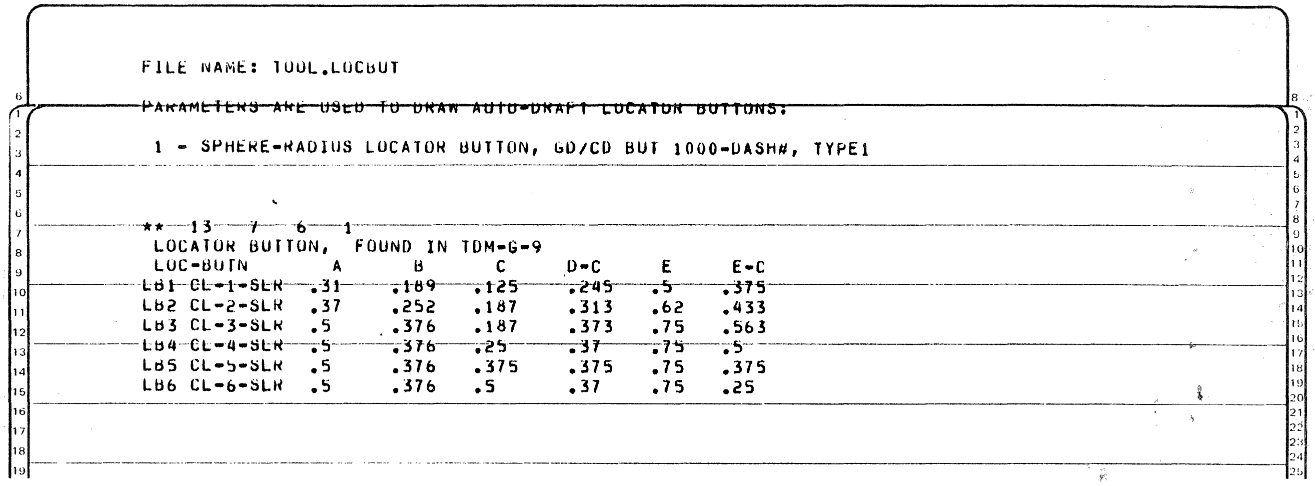


```
CS5    1/4   .250     .25   .250       .37     .5      .19      .0
CS6    1/4   .250     .25   .250 .     .37     .75     .19      .0
CS7    1/4   .250     .25   .250       .37    1.       .19      .0
CS8    1/4   .250     .25   .250       .37    1.       .19      .25
CS9    1/4   .250     .25   .250       .37    1.       .19      .5
CS10   1/4   .250     .25   .250       .37    1.       .19     1.0
CS11   1/4   .250     .25   .250       .37    1.25     .19     1.25
CS12   1/4   .250     .25   .250       .37    1.5      .19     1.5
CS13   1/4   .250     .25   .250       .37    2.0      .19     2.0
CS14   5/16  .312     .31   .312       .47     .5      .22      .0
CS15   5/16  .312     .31   .312       .47     .75     .22      .0
CS16   5/16  .312     .31   .312       .47    1.       .22      .0
CS17   5/16  .312     .31   .312       .47    1.25     .22      .0
CS18   5/16  .312     .31   .312       .47    1.12     .22      .38
CS19   5/16  .312     .31   .312       .47    1.12     .22      .88
CS20   5/16  .312     .31   .312       .47    1.25     .22     1.25
CS21   5/16  .312     .31   .312       .47    1.5      .22     1.5
CS22   5/16  .312     .31   .312       .47    2.0      .22     2.0
CS23   3/8   .375     .37   .375       .56     .5      .31      .0
CS24   3/8   .375     .37   .375       .56     .75     .31      .0
CS25   3/8   .375     .37   .375       .56    1.       .31      .0
CS26   3/8   .375     .37   .375       .56    1.25     .31      .0
CS27   3/8   .375     .37   .375       .56    1.25     .31      .25
CS28   3/8   .375     .37   .375       .56    1.25     .31      .75
CS29   3/8   .375     .37   .375       .56    1.25     .31     1.25
CS30   3/8   .375     .37   .375       .56    1.5      .31     1.5
CS31   3/8   .375     .37   .375       .56    2.0      .31     2.0
CS32   1/2   .5       .50   .5         .75    1.       .50      .0
CS33   1/2   .5       .50   .5         .75    1.25     .50      .0
CS34   1/2   .5       .50   .5         .75    1.5      .50      .0
CS35   1/2   .5       .50   .5         .75    1.5      .50      .5
CS36   1/2   .5       .50   .5         .75    1.5      .50     1.0
CS37   1/2   .5       .50   .5         .75    1.5      .50     1.5
CS38   1/2   .5       .50   .5         .75    2.0      .50     2.0
CS39   5/8   .625     .62   .625       .94    1.5      .50      .0
CS40   5/8   .625     .62   .625       .94    1.75     .50      .25
CS41   5/8   .625     .62   .625       .94    1.75     .50      .75
CS42   5/8   .625     .62   .625       .94    1.75     .50     1.25
** 12   7   8   3
```

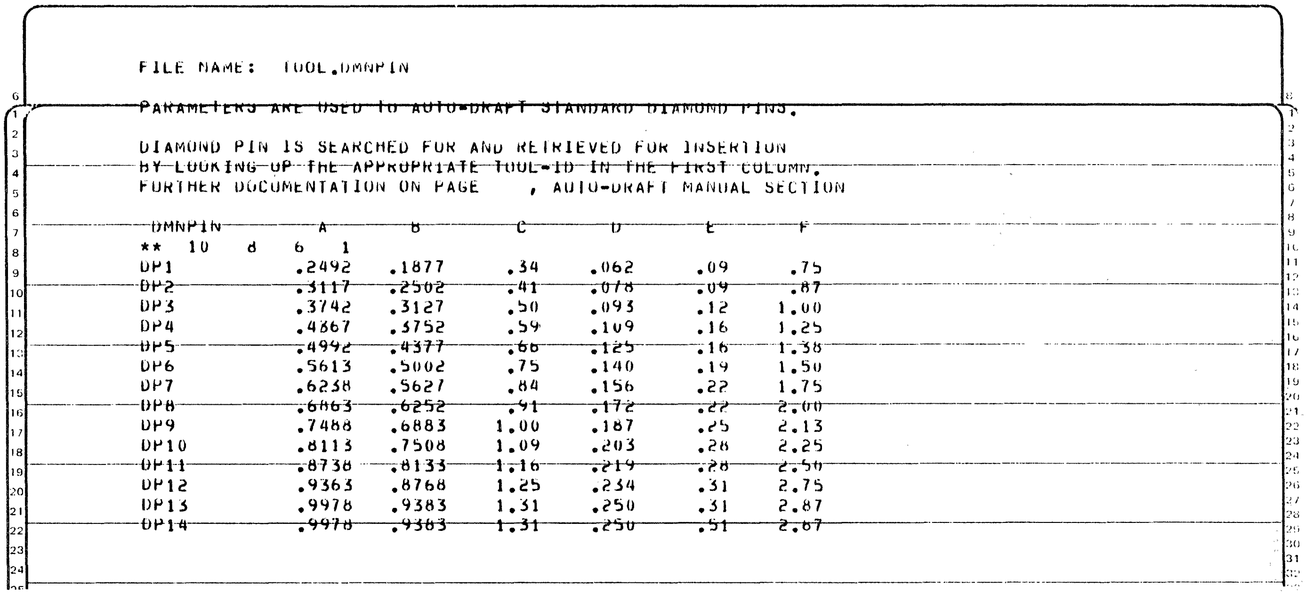
```
 AMERICAN STANDARD HEX-HEAD CAP SCREW,  FOUND IN TDM-F-1
 HEX-HD BOLT  DIA      H     THRD       W       THL       W1    L-THL OFFSET
HH43   1/4   .250     .19   .250       .50     .75      .44       .0   .094
HH44   1/4   .250     .19   .250       .50     .75      .44       .25  .094
HH1    1/4   .250     .19   .250       .50     .75      .44       .75  .094


HH2    1/4   .250     .19   .250       .50     .75      .44     1.25  .094
HH3    1/4   .250     .19   .250       .50     .75      .44     1.75  .094
HH4    1/4   .250     .19   .250       .50     .75      .44     2.25  .094
HH5    1/4   .250     .19   .250       .50     .75      .44     2.75  .094
HH6    1/4   .250     .19   .250       .50     .75      .44     3.25  .094
HH7    1/4   .250     .19   .250       .50     .75      .44     4.25  .094
HH8    5/16  .312     .23   .312       .56     .75      .50       .0  .109
HH9    5/16  .312     .23   .312       .56     .75      .50       .25 .109
HH10   5/16  .312     .23   .312       .56     .87      .50       .63 .109
HH11   5/16  .312     .23   .312       .56     .75      .50     1.13  .109
HH12   5/16  .312     .23   .312       .56     .75      .50     1.63  .109
HH13   5/16  .312     .23   .312       .56     .75      .50     2.13  .109
HH14   5/16  .312     .23   .312       .56     .75      .50     2.63  .109
HH15   5/16  .312     .23   .312       .56     .75      .50     3.13  .109
HH16   3/8   .375     .27   .375       .62    1.0       .56       .0  .125
HH17   3/8   .375     .27   .375       .62    1.0       .56       .5  .125
HH18   3/8   .375     .27   .375       .62    1.0       .56     1.0   .125
HH19   3/8   .375     .27   .375       .62    1.0       .56     1.5   .125
HH20   3/8   .375     .27   .375       .62    1.0       .56     2.0   .125
HH21   3/8   .375     .27   .375       .62    1.0       .56     2.5   .125
HH22   3/8   .375     .27   .375       .62    1.0       .56     3.0   .125
HH23   3/8   .375     .27   .375       .62    1.0       .56     4.0   .125
HH24   1/2   .5       .37   .5         .84    1.0       .75     0.0   .125
HH25   1/2   .5       .37   .5         .84    1.0       .75     0.25  .125
HH26   1/2   .5       .37   .5         .84    1.0       .75     0.75  .125
HH27   1/2   .5       .37   .5         .84    1.0       .75     1.25  .125
HH28   1/2   .5       .37   .5         .84    1.0       .75     1.75  .125
HH29   1/2   .5       .37   .5         .84    1.0       .75     2.25  .125
HH30   1/2   .5       .37   .5         .84    1.0       .75     2.75  .125
HH31   1/2   .5       .37   .5         .86    1.0       .75     2.75  .125
HH32   1/2   .5       .37   .5         .84    1.0       .75     3.75  .125
HH33   1/2   .5       .37   .5         .86    1.0       .75     4.25  .125
HH34   1/2   .5       .37   .5         .86    1.0       .75     4.75  .125
HH35   1/2   .5       .37   .5         .86    1.0       .75     5.5   .125
HH36   1/2   .5       .37   .5         .86    1.0       .75     6.5   .125
HH37   1/2   .5       .37   .5         .86    1.0       .75     7.5   .125
HH38   5/8   .625     .46   .625       .98    1.0       .87     0.0   .125
HH39   5/8   .625     .46   .625       .98    1.5       .87     0.0   .125
HH40   5/8   .625     .46   .625       .98    1.5       .87     1.0   .125
HH41   5/8   .625     .46   .625       .98    1.5       .87     1.5   .125
HH42   5/8   .625     .46   .625       .98    1.5       .87     2.0   .125
HH43   5/8   .625     .46   .625       .98    1.5       .87     2.5   .125
HH44   5/8   .625     .46   .625       .98    1.5       .87     3.5   .125
HH45   3/4   .75      .52   .75       1.38    1.75     1.12       .5   .125


HH46   3/4   .75      .52   .75       1.38    1.75     1.12     1.25  .125
HH47   3/4   .75      .52   .75       1.29    1.75     1.12     2.25  .125
HH48   3/4   .75      .52   .75       1.38    1.75     1.12     2.25  .125
HH49   3/4   .75      .52   .75       1.29    1.75     1.12     3.25  .125
HH50   3/4   .75      .52   .75       1.38    1.75     1.12     3.25  .125


FILE NAME: TOOL.BSHING

PARAMETERS ARE USED TO DRAW AUTO-DRAFT BUSHINGS:
 1 - BULLET-NOSE DOWEL & BUSHING, CARR LANE # CL-X,        TYPE1
 2 - LOCATOR BUSHINGS, SLOTTED, GD/CD BUS 2290-DASH#,      TYPE2
 3 - LOCATOR BUSHINGS, FLOATING PIN, GD/CD BUS 2290-DASH#, TYPE3


**  13   7   7   1
 BULLET-NOSE DOWEL & BUSHING,  FOUND IN TDM-G-2
 BULL-NOSE        A       B       C        D        E      F       L
BN1 CL-1-BND   .25     .251    .280     .5      .125   .907    .125
BN2 CL-2-BND   .3125   .313    .344     .5      .157   .939    .156
BN3 CL-3-BND   .375    .376    .407     .5      .187  1.032    .1875
BN4 CL-4-BND   .5      .501    .532     .5      .187  1.094    .25
**  17   6   6   2
```



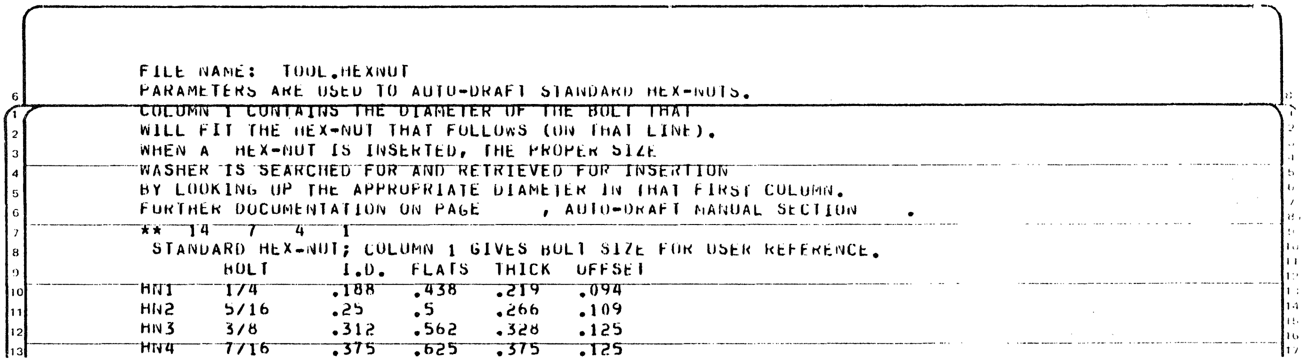
```
 SLOTTED LOCATOR BUSHING,  FOUND IN TDM-G-3
 SLOT-LOC BUSHING    A        B      L     D       E    DIAM
123456789012345671234561234561234561234561234561234561234561234561234 56
SLB1 BUS 2290-2    .25     .375    .4  .1875  .375  1.0
SLB2 BUS 2290-4    .3125   .437    .4  .1875  .375  1.0
SLB3 BUS 2290-6    .375    .5      .4  .1875  .375  1.0
SLB4 BUS 2290-8    .5      .625    .4  .1875  .375  1.0
**  17   6   8   3
 FLOATING PIN LOCATOR BUSHING,  FOUND IN TDM-G-4
 FLOAT-LOC BUSHING  A        C       L
FLB1 BUS 2290-10    .5    .625    .4    .1875 .375  1.0    .25    .2
FLB2 BUS 2290-12    .5    .625    .4    .1875 .375  1.0    .3125 .2
FLB3 BUS 2290-14    .5    .625    .4    .1875 .375  1.0    .375   .22
FLB4                .5    .625    .4    .1875 .375  1.0    .5     .25
```



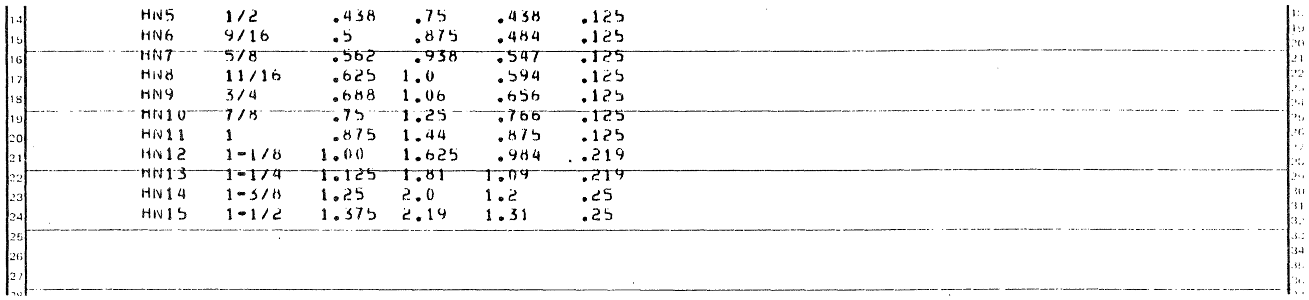
```
FILE NAME: TOOL.LOCBUT

PARAMETERS ARE USED TO DRAW AUTO-DRAFT LOCATOR BUTTONS:

 1 - SPHERE-RADIUS LOCATOR BUTTON, GD/CD BUT 1000-DASH#, TYPE1



**  13   7   6   1
 LOCATOR BUTTON,  FOUND IN TDM-G-9
 LOC-BUTN         A        B       C        D-C      E       E-C
LB1 CL-1-SLR  .31      .189    .125     .245     .5      .375
LB2 CL-2-SLR  .37      .252    .187     .313     .62     .433
LB3 CL-3-SLR  .5       .376    .187     .373     .75     .563
LB4 CL-4-SLR  .5       .376    .25      .37      .75     .5
LB5 CL-5-SLR  .5       .376    .375     .375     .75     .375
LB6 CL-6-SLR  .5       .376    .5       .37      .75     .25
```



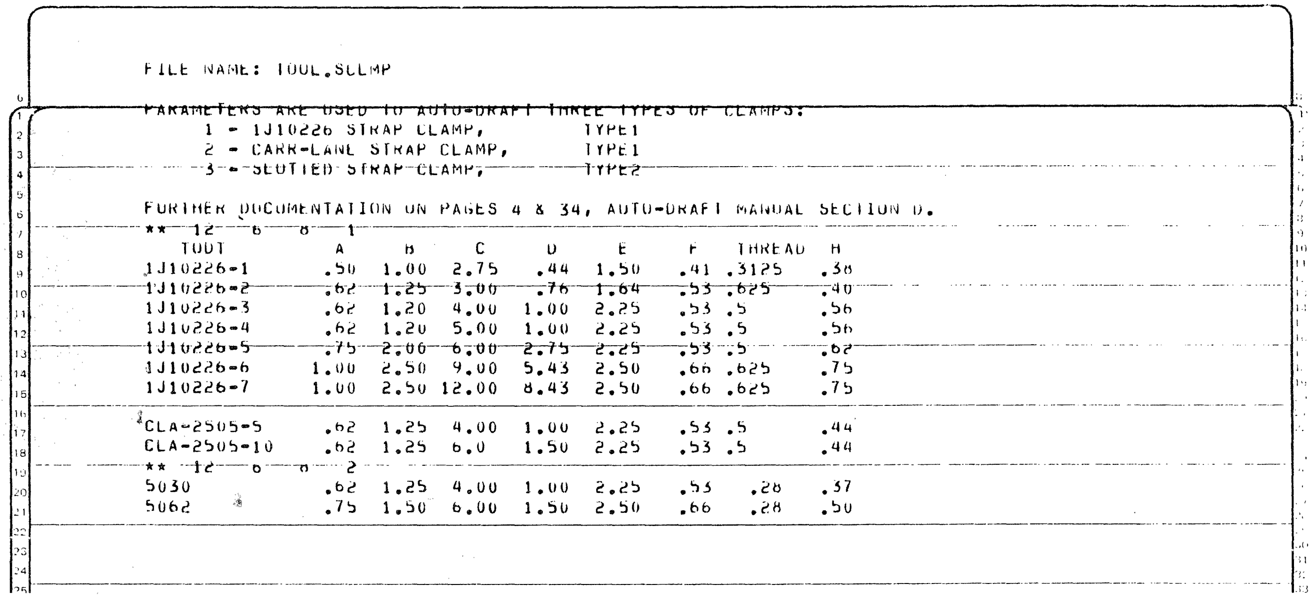
```
FILE NAME:  TOOL.DMNPIN

PARAMETERS ARE USED TO AUTO-DRAFT STANDARD DIAMOND PINS.

DIAMOND PIN IS SEARCHED FOR AND RETRIEVED FOR INSERTION
BY LOOKING UP THE APPROPRIATE TOOL-ID IN THE FIRST COLUMN.
FURTHER DOCUMENTATION ON PAGE     , AUTO-DRAFT MANUAL SECTION

 DMNPIN           A        B          C          D          E         F
**  10   8   6   1
DP1             .2492    .1877       .34       .062       .09       .75
DP2             .3117    .2502       .41       .078       .09       .87
DP3             .3742    .3127       .50       .093       .12      1.00
DP4             .4867    .3752       .59       .109       .16      1.25
DP5             .4992    .4377       .66       .125       .16      1.38
DP6             .5613    .5002       .75       .140       .19      1.50
DP7             .6238    .5627       .84       .156       .22      1.75
DP8             .6863    .6252       .91       .172       .22      2.00
DP9             .7488    .6883      1.00       .187       .25      2.13
DP10            .8113    .7508      1.09       .203       .28      2.25
DP11            .8738    .8133      1.16       .219       .28      2.50
DP12            .9363    .8768      1.25       .234       .31      2.75
DP13            .9978    .9383      1.31       .250       .31      2.87
DP14            .9978    .9383      1.31       .250       .51      2.67
```



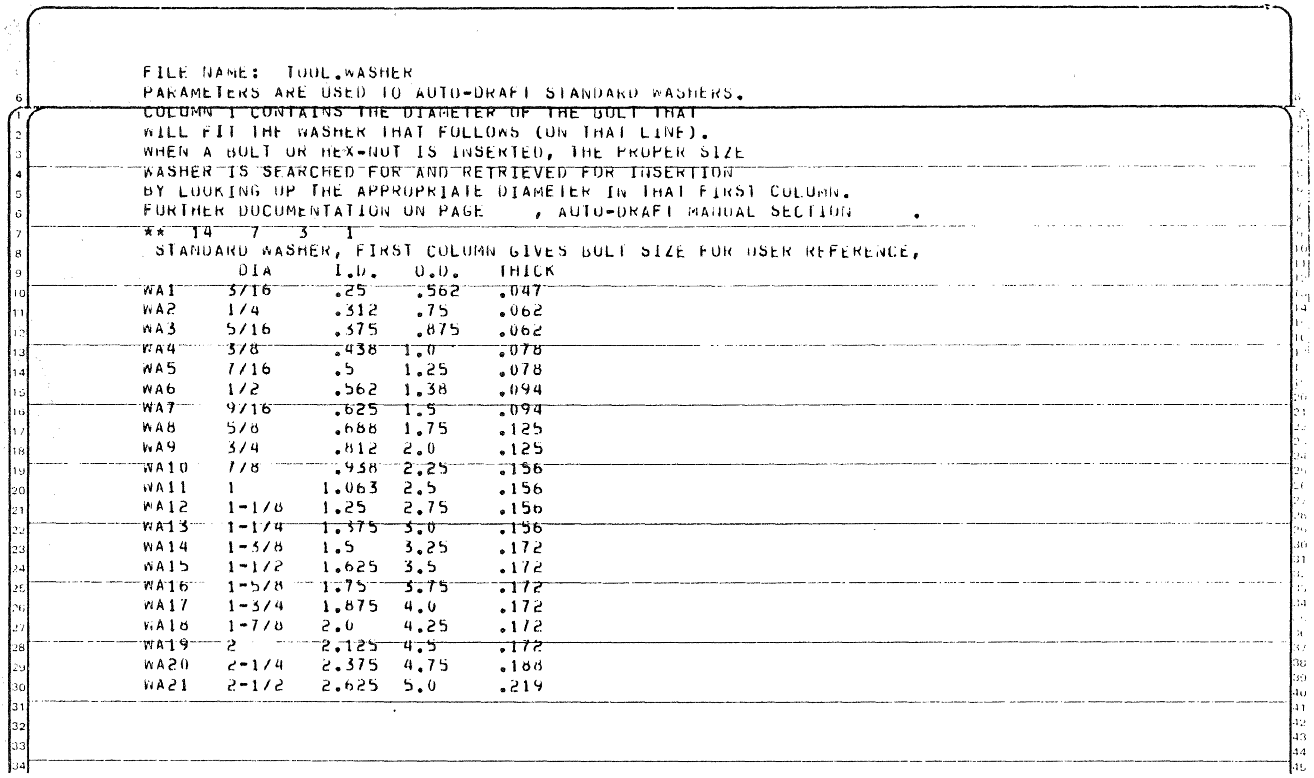
```
FILE NAME:  TOOL.HEXNUT
PARAMETERS ARE USED TO AUTO-DRAFT STANDARD HEX-NUTS.
COLUMN 1 CONTAINS THE DIAMETER OF THE BOLT THAT
WILL FIT THE HEX-NUT THAT FOLLOWS (ON THAT LINE).
WHEN A  HEX-NUT IS INSERTED, THE PROPER SIZE
WASHER IS SEARCHED FOR AND RETRIEVED FOR INSERTION
BY LOOKING UP THE APPROPRIATE DIAMETER IN THAT FIRST COLUMN.
FURTHER DOCUMENTATION ON PAGE     , AUTO-DRAFT MANUAL SECTION    .
**  14   7   4   1
 STANDARD HEX-NUT; COLUMN 1 GIVES BOLT SIZE FOR USER REFERENCE.
        BOLT       I.D.  FLATS  THICK  OFFSET
HN1     1/4        .188   .438   .219   .094
HN2     5/16       .25    .5     .266   .109
HN3     3/8        .312   .562   .328   .125
HN4     7/16       .375   .625   .375   .125
```



```
HN5     1/2       .438   .75     .438    .125
HN6     9/16      .5      .875   .484    .125
HN7     5/8       .562    .938   .547    .125
HN8     11/16     .625   1.0     .594    .125
HN9     3/4       .688   1.06    .656    .125
HN10    7/8       .75    1.25    .766    .125
HN11    1         .875   1.44    .875    .125
HN12    1-1/8    1.00    1.625   .984   ..219
HN13    1-1/4    1.125   1.81   1.09     .219
HN14    1-3/8    1.25    2.0    1.2      .25
HN15    1-1/2    1.375   2.19   1.31     .25
```



```
FILE NAME: TOOL.SCLMP

PARAMETERS ARE USED TO AUTO-DRAFT THREE TYPES OF CLAMPS:
     1 - 1J10226 STRAP CLAMP,          TYPE1
     2 - CARR-LANE STRAP CLAMP,        TYPE1
     3 - SLOTTED STRAP CLAMP,          TYPE2

FURTHER DOCUMENTATION ON PAGES 4 & 34, AUTO-DRAFT MANUAL SECTION D.
**  12   6   8   1
   TOOT              A     B     C      D     E      F   THREAD  H
1J10226-1          .50  1.00  2.75    .44  1.50    .41 .3125   .38
1J10226-2          .62  1.25  3.00    .76  1.64    .53 .625    .40
1J10226-3          .62  1.20  4.00   1.00  2.25    .53 .5      .56
1J10226-4          .62  1.20  5.00   1.00  2.25    .53 .5      .56
1J10226-5          .75  2.00  6.00   2.75  2.25    .53 .5      .62
1J10226-6         1.00  2.50  9.00   5.43  2.50    .66 .625    .75
1J10226-7         1.00  2.50 12.00   8.43  2.50    .66 .625    .75

CLA-2505-5         .62  1.25  4.00   1.00  2.25    .53 .5      .44
CLA-2505-10        .62  1.25  6.0    1.50  2.25    .53 .5      .44
**  12   6   8   2
5030               .62  1.25  4.00   1.00  2.25    .53   .28   .37
5062               .75  1.50  6.00   1.50  2.50    .66   .28   .50
```



```
FILE NAME:  TOOL.WASHER
PARAMETERS ARE USED TO AUTO-DRAFT STANDARD WASHERS.
COLUMN 1 CONTAINS THE DIAMETER OF THE BOLT THAT
WILL FIT THE WASHER THAT FOLLOWS (ON THAT LINE).
WHEN A BOLT OR HEX-NUT IS INSERTED, THE PROPER SIZE
WASHER IS SEARCHED FOR AND RETRIEVED FOR INSERTION
BY LOOKING UP THE APPROPRIATE DIAMETER IN THAT FIRST COLUMN.
FURTHER DOCUMENTATION ON PAGE    , AUTO-DRAFT MANUAL SECTION      .
**  14   7   3   1
 STANDARD WASHER, FIRST COLUMN GIVES BOLT SIZE FOR USER REFERENCE,
        DIA       I.D.    O.D.    THICK
WA1     3/16      .25     .562    .047
WA2     1/4       .312    .75     .062
WA3     5/16      .375    .875    .062
WA4     3/8       .438   1.0      .078
WA5     7/16      .5     1.25     .078
WA6     1/2       .562   1.38     .094
WA7     9/16      .625   1.5      .094
WA8     5/8       .688   1.75     .125
WA9     3/4       .812   2.0      .125
WA10    7/8       .938   2.25     .156
WA11    1        1.063   2.5      .156
WA12    1-1/8    1.25    2.75     .156
WA13    1-1/4    1.375   3.0      .156
WA14    1-3/8    1.5     3.25     .172
WA15    1-1/2    1.625   3.5      .172
WA16    1-5/8    1.75    3.75     .172
WA17    1-3/4    1.875   4.0      .172
WA18    1-7/8    2.0     4.25     .172
WA19    2        2.125   4.5      .172
WA20    2-1/4    2.375   4.75     .188
WA21    2-1/2    2.625   5.0      .219
```

INTEROPTICS

Stanley Haskell
Polaroid
38 Henry Street
M/S 2A
Cambridge, MA 02139
617-577-3901

ABSTRACT

INTEROPTICS is an interactive optical systems package designed to permit users at various levels of experience to do graphical analysis of an optical system or to integrate optical systems into a mechanical system.

The design package contains routines for constructing the standard optical cross-sectional drawing of the lens system from the lens formula parameters, for mathematically tracing and plotting meridional rays through the system interactively by the user, and for tracing bundles, fans, or single rays interactively. An important feature of the system is the capability of building-in constraints such that a user without extensive optical experience can develop valid optical information without violating the specifications prescribed by the optical system designer.

A database of designs has been created on the system for use by engineering personnel in other disciplines. A directory of these optical systems is available by a command from the menu. The menu also contains commands to initiate ray traces and drawings of new systems.

Interoptics is a software system to permit an interactive interface between the user and an optical systems data base. One of the main goals of the project was to provide a tool wherein a mechanical designer could be given substantial freedom to develop the mechanical aspects of his project and still stay within the constraints of the optical system requirements. This can be achieved by the optical system designer providing an ''active'' data base for his optical system such that the mechanical designer can make certain modifications to the system without violating the optical constraints.

DESIGN GOALS

The capabilities of the system include:

1. An ''active'' interface between the operator-mechanical designer. ''Active'' interface means that the CAD system operator can develop the characteristics of an optical system as he needs them by requests to the CAD system through the keyboard or a menu. The results of these requests, graphically and numerically displayed, can be reviewed as they appear on the displays. This information can be retained as part of the database or rejected and new information requested based on the previous results.

2. A data base of optical designs for use by various engineering groups. A single set of parameters (radii, thicknesses, separations, and indices of refraction) for any given lens system are entered into the database. This data is used for all subsequent programs (lens drawings, ray traces, etc.) thus insuring that all information is consistent and compatible.
3. A means of doing certain types of optical analysis utilizing the specific advantages of CAD presentation. By the very nature of the CAD graphical presentation, a form of optical analysis is implied. One can watch the ray traces develop and make judgments or interpretations of the lens system performance. It is even possible to observe the effects of the typical aberrations (spherical aberration, coma, astigmatism, curvature of field, and distortion) on certain classes of systems. Spot diagrams of image characteristics can also be demonstrated. Of course the extent to which these effects can be seen are a function of the correction of the lens system and the inherent resolution of the CAD system.
4. An integrated process to permit the optical, mechanical, and fabrication aspects of a project to be coordinated within a CAD system. Instead of having separately created drawings of lens elements, lens assemblies, mechanical assemblies, etc. as is traditionally done, all of the above can now be created using a uniform database. The integrated optical-mechanical system resulting from this approach should be a more consistent and error free design.
5. As an effective learning device for optical concepts. We have observed that the process of interactively tracing rays through a system graphically has enhanced the understanding of how optical systems work. This increased awareness of the sometimes difficult concepts of optics strengthens our premise that such a system as this should result in sounder, more functional optical-mechanical designs.

It was not the intention to provide an optical design capability in the traditional sense but to provide a facility for non-optical engineering people to more effectively use the optical designs supplied to them. All aspects of the system can utilize the full dimensional capabilities of the system.

IMPLEMENTATION

Through a directory provided on the system, the user can select the optical system to be studied.

The optical designer provides PEP or FORTRAN programs that supply a number of options:

1. A standard optical cross section diagram of the lens including focal surface and aperture designation. Figure 2. The drawing is created from the constructional data for the lens system and a PEP program rather than by digitizing the entities. This tends to insure a consistency and accuracy for programs that follow.
2. A meridional ray trace program in which the operator can select appropriate rays for his mechanical requirements. Figure 3. The optical designer has built-in limits that prevent the operator from tracing invalid rays.

   For example, the maximum ray height through the aperture can be programmed such that an attempt to trace rays greater than the designed aperture will result in an error message. Similarly, lens element clear apertures intended to introduce controlled vignetting can be programmed to reject rays beyond the clear aperture.

   A similar restriction could be placed on the selection of field angles if this is appropriate to the design at hand. In some cases, such as the one described in the next section, it is more realistic to leave the field angle unrestricted. In this case, one of the operator's options would be to examine unwanted rays that do get through the optical system and need to be blocked by a mechanical baffle.

   Where appropriate, e.g. photographic systems, unit focussing or front element focussing can be included and operator input of subject distance provided. The graphic presentation can show the displacement of the lens unit or front element for the refocussed lens.

   Intermediate or final numerical results of the optical calculations can be shown on the operator's terminal if desired.

3. The optical designer will also provide whatever additional optical components are required, such as mirrors, prisms, etc. As an example of the interaction aspect of the program consider a diagonal plane mirror introduced into the optical path to fold the system. Figure 3. To give the mechanical designer the greatest freedom in his work, he is permitted to select the position of that mirror and its angle relative to the optical axis. The reflected and unreflected rays traced through the system will guide him to a workable configuration wherein all desired rays get to the image surface and unwanted rays are properly baffled or excluded in some way.
4. A skew ray trace is also provided. A skew ray is one which can pass through the optical system at any orientation in three dimensional space. With this capability the mechanical designer can define the maximum cone of light required by the optical system and most efficiently use adjacent space for his application. Again the optical designer can provide constraints for the ray trace. Figures 4 and 5 show eight skew rays traced through a corrected three element optical system to an off-axis image point. All rays pass through a common image point within the resolution of the drawing. Figure 6 shows four views of skew rays traced through a simple single element lens to three off-axis image points. The effects of spherical aberration and curvature of field can readily be seen. Figure 7 is an enlarged view from Figure 6 showing a typical spot diagram and illustrating the aberration known as coma.
5. A special software subsystem written in FORTRAN using matrix methods has been developed to permit analysis and design of prism and mirror systems. Intended for plane reflecting surfaces, the program graphically presents the orientation and position of images and rays in multiple mirror systems in three dimensions. The technique can be very useful in determining in a graphic way the effect of the plane mirror system and its perturbations on the image of a system. For example, the effect of reversals and inversions due to the mirror system and the tilts and rotations of the image due to errors in the mirror or prism angles are easily seen.

   Figure 8 shows the well known penta prism with the letter "F" at its entrance face. Views 1, 2, 3, and 7 are shown. The position and orientation of the image of the letter is then shown as it would be seen through the viewing face. For comparison, Figure 9 shows the similar effect for the standard right-angle prism. Note the reversal of the "F" and the changed location of the image. Figure 10 illustrates the effect of tilting the hypotenuse of the prism such that it is not perpendicular to the base of the prism. Note the rotation and tilt of the letter "F".

With a data base of lens system and the "tools" described above, the engineer can develop his mechanical system to satisfy his requirements and still be confident that the optical functions are also properly met.

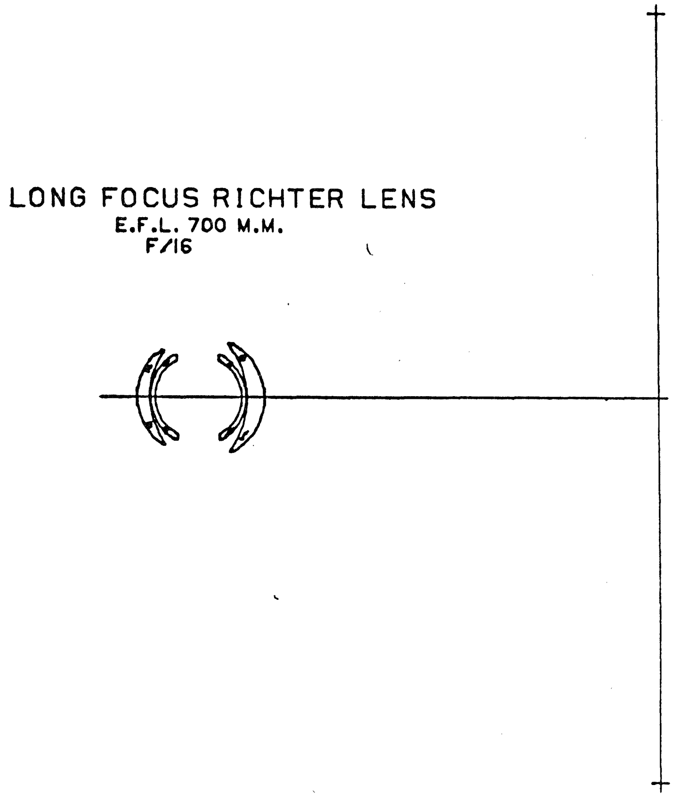
EXTENSIONS

In addition to what has been described above, the package can provide drawings for fabrication, assembly, illustration, or patent purposes. A standard form drawing for individual element specification has been provided for use by the optical shop or for outside quotation. Figure 11. The database can, of course, be extended to provide the information for typical computer aided manufacturing uses.

DESIGN GOALS

1. Active Interface
2. Data Base of Optical Designs
3. Optical Analysis
4. Integrated Process Through CAD
5. Learning Tool

Figure 1



SURF	RADIUS	SEP	C.A.	GLASS
1	84.432	18.94	139.8	
2	124.65	0.54	135.9	
3	72.83	6.39	123.8	
4	60.59	129.21	108.5	
5	-60.59	6.39	108.5	
6	-72.83	1.33	123.8	
7	-128.14	25.81	151.3	
8	-90.91	591.72	160.0	

DIMENSIONS IN M.M.

Figure 2

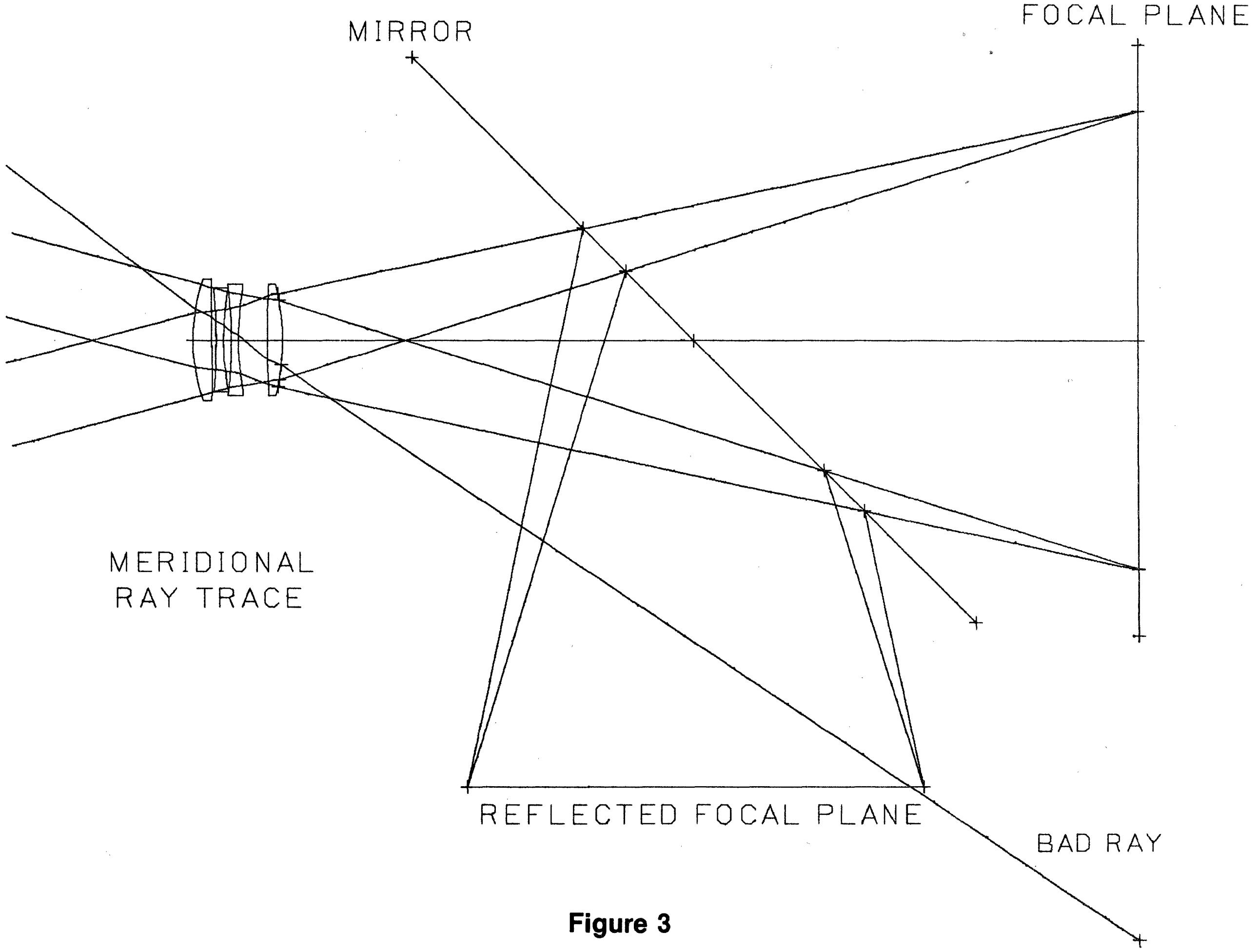


Figure 3

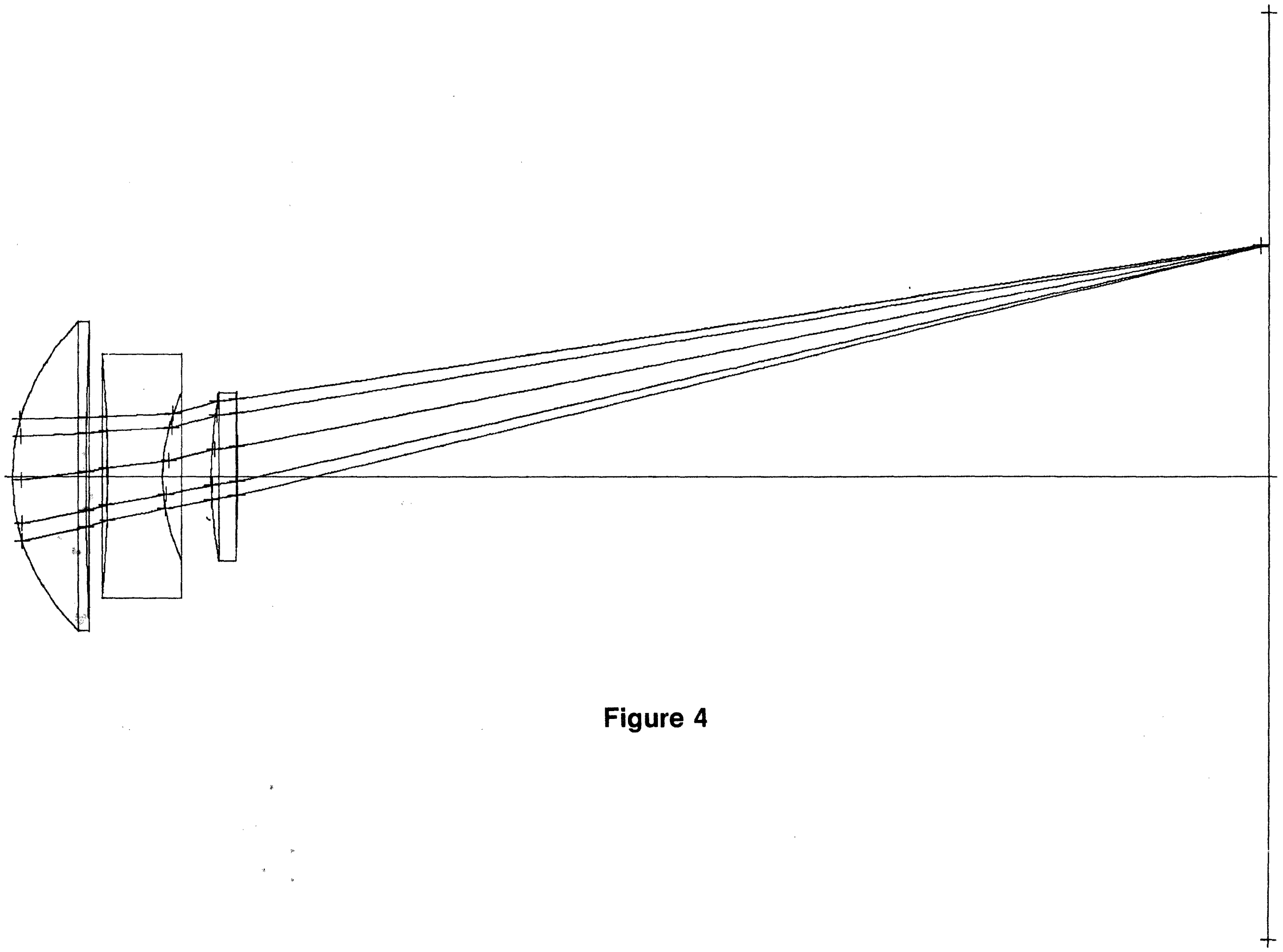


Figure 4

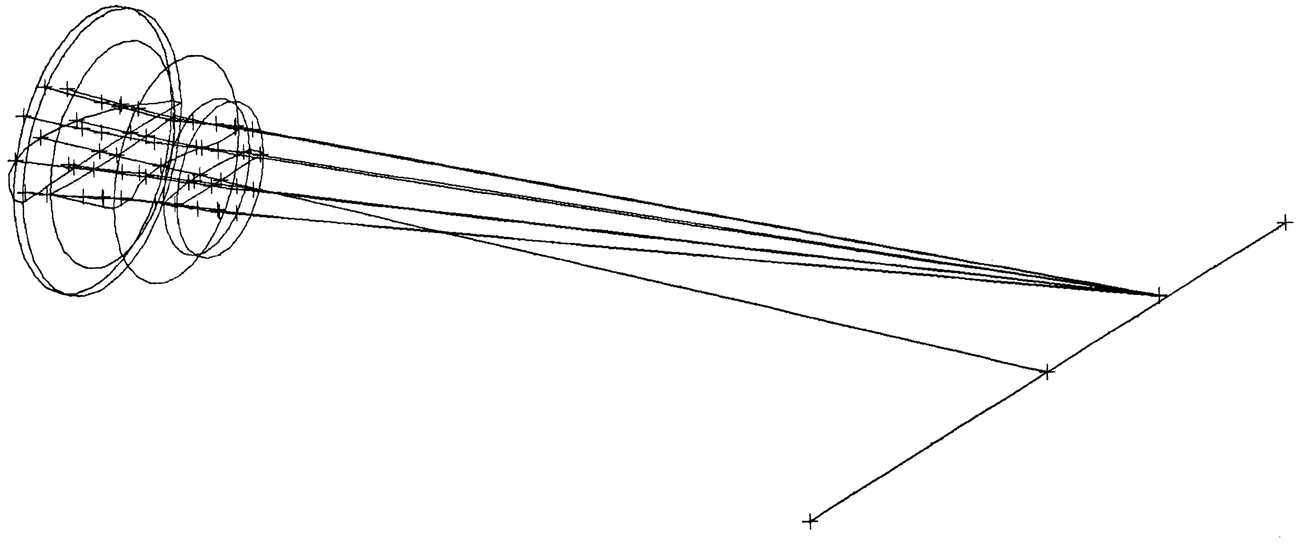


Figure 5

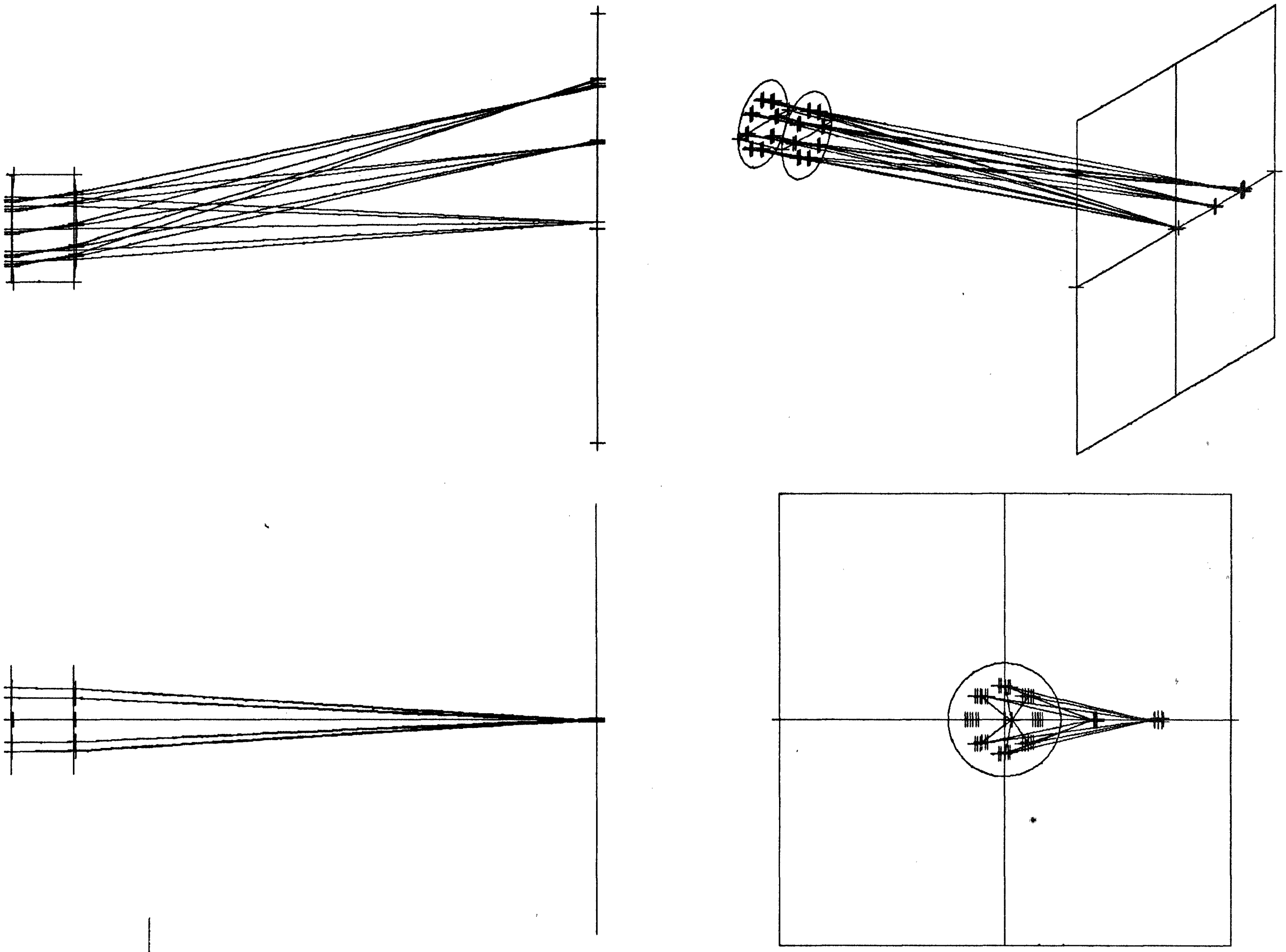


Figure 6

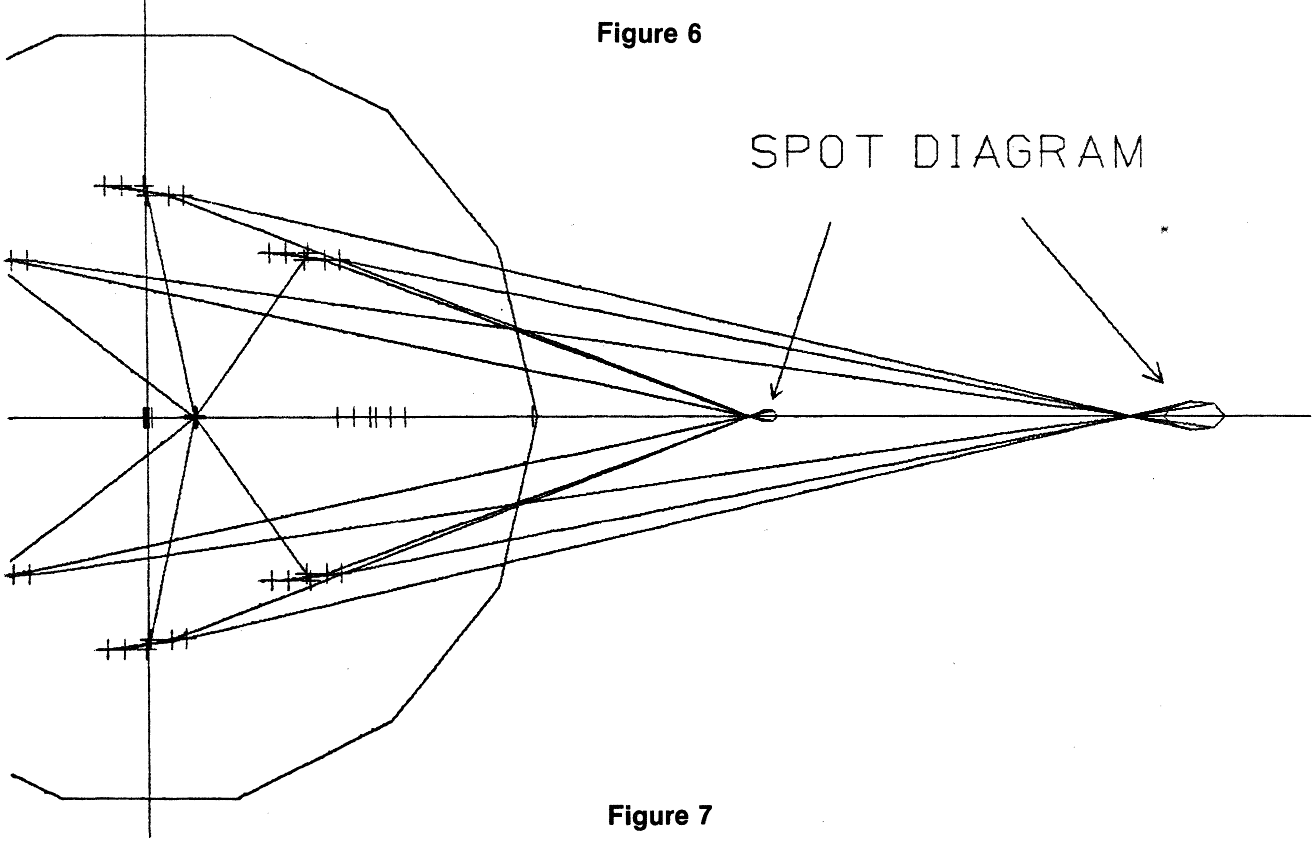


Figure 7

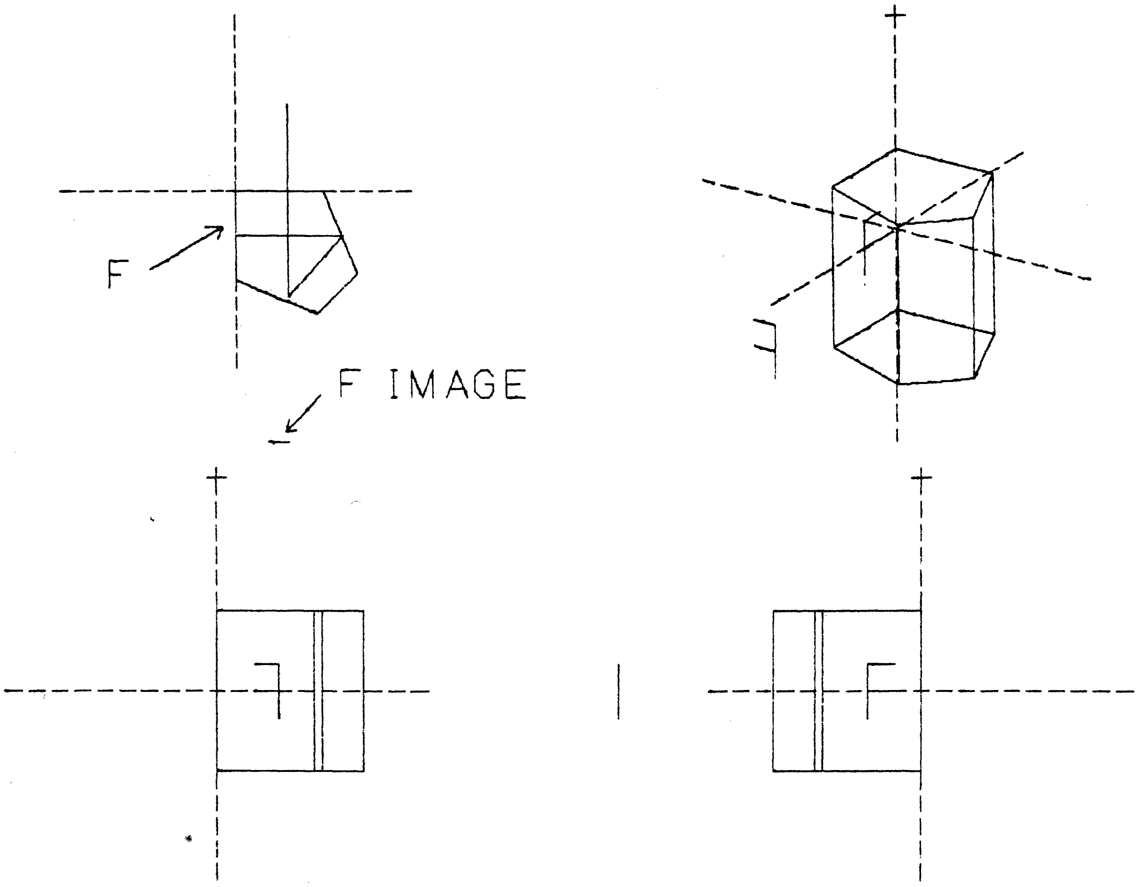


Figure 8

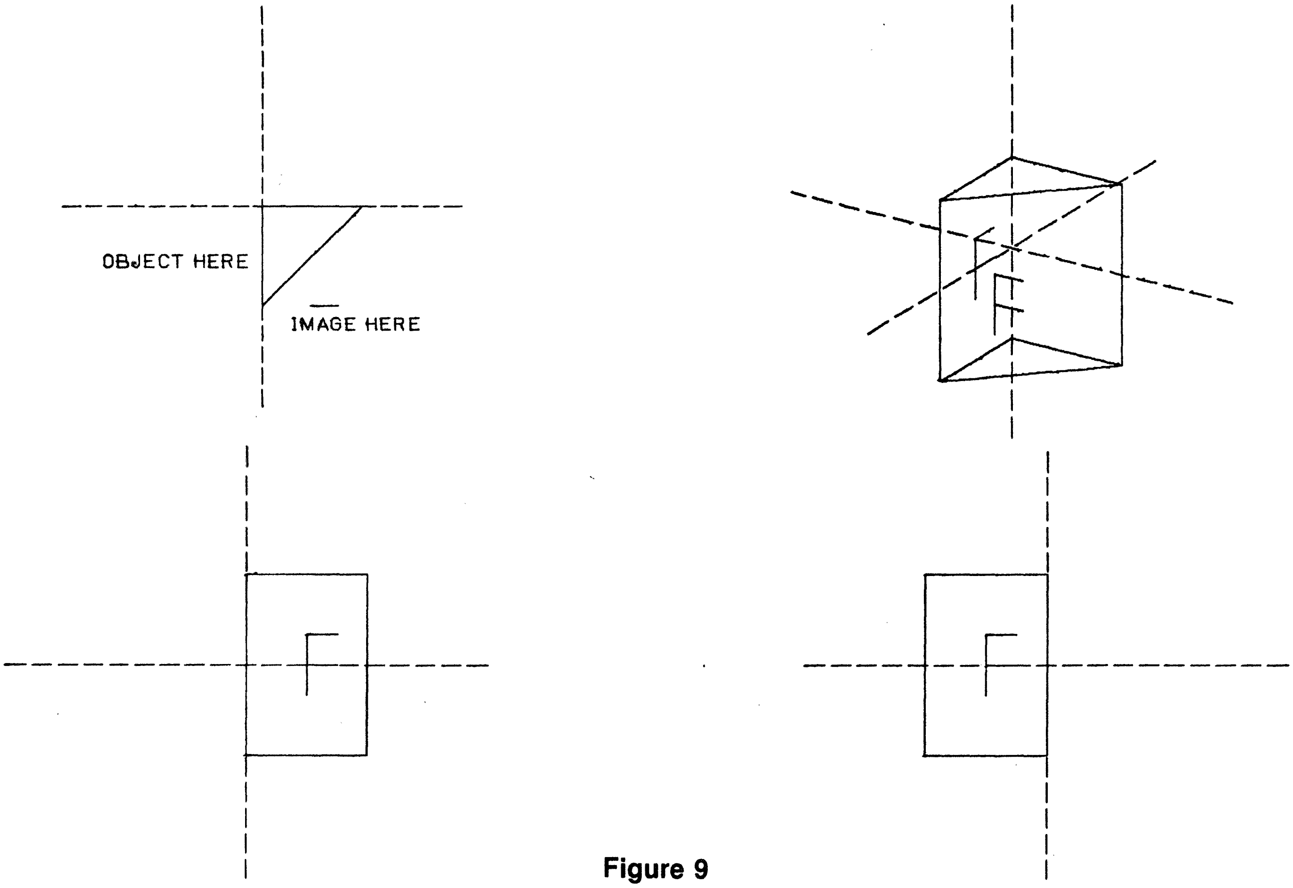


Figure 9

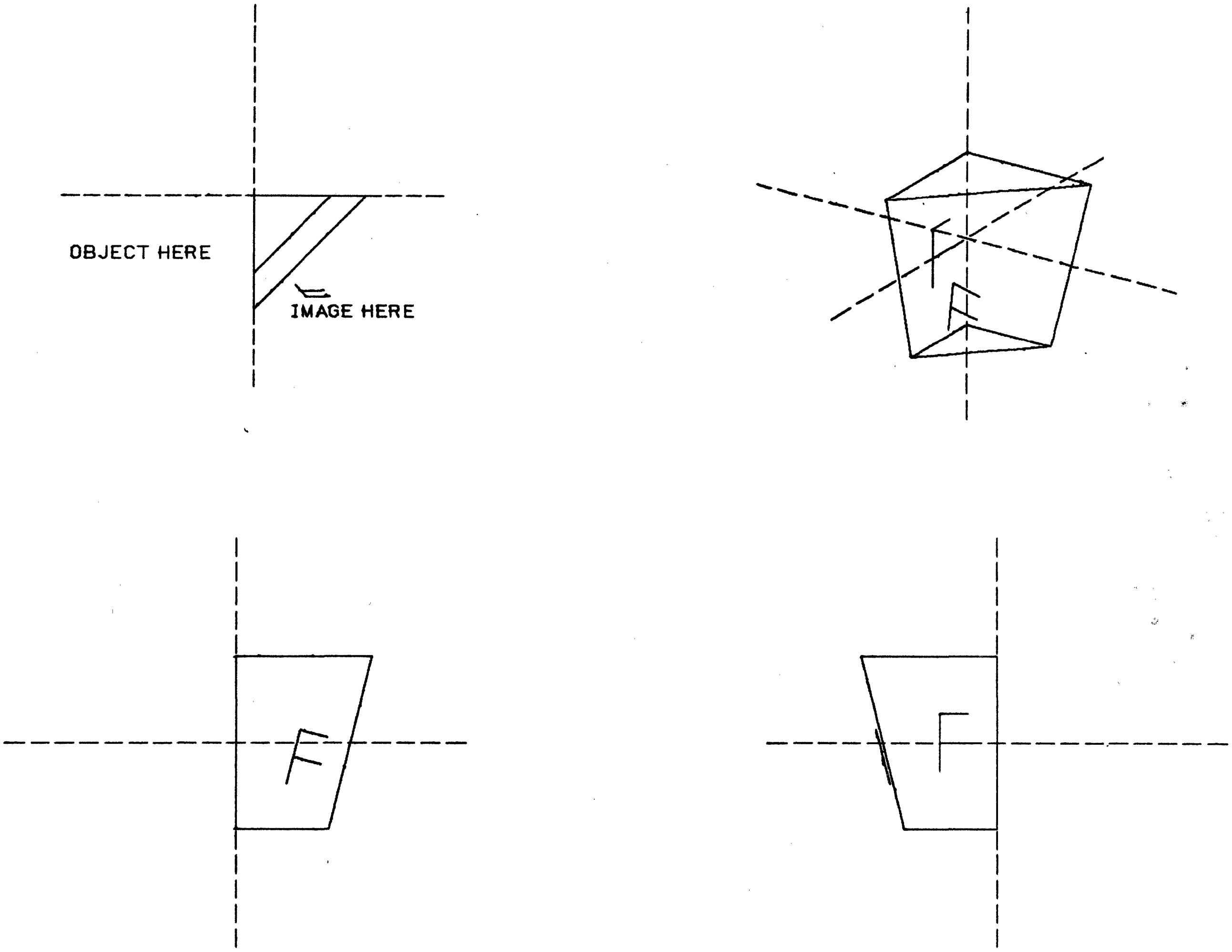
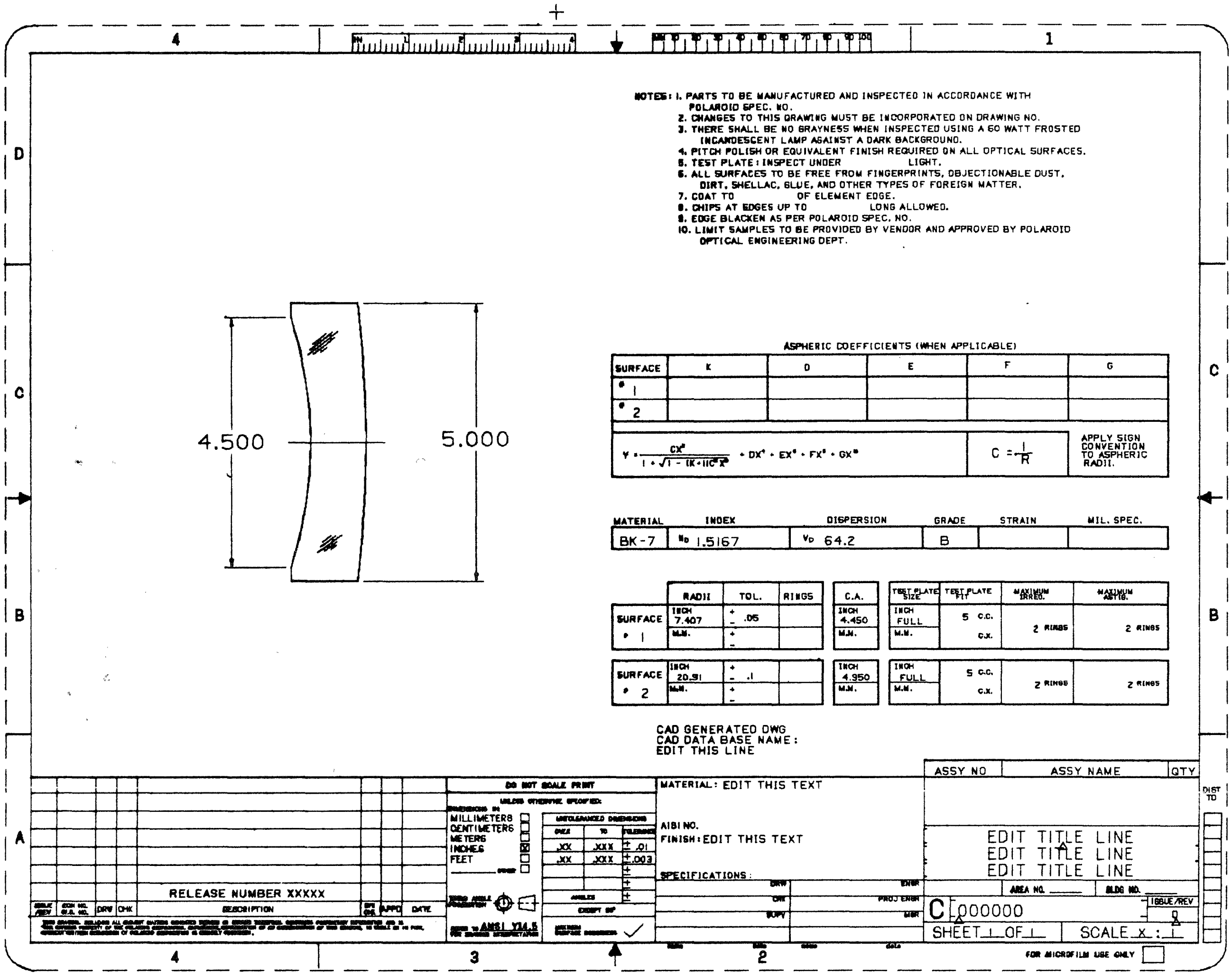


Figure 10



NOTES: 1. PARTS TO BE MANUFACTURED AND INSPECTED IN ACCORDANCE WITH POLAROID SPEC. NO.
2. CHANGES TO THIS DRAWING MUST BE INCORPORATED ON DRAWING NO.
3. THERE SHALL BE NO GRAYNESS WHEN INSPECTED USING A 60 WATT FROSTED INCANDESCENT LAMP AGAINST A DARK BACKGROUND.
4. PITCH POLISH OR EQUIVALENT FINISH REQUIRED ON ALL OPTICAL SURFACES.
5. TEST PLATE: INSPECT UNDER LIGHT.
6. ALL SURFACES TO BE FREE FROM FINGERPRINTS, OBJECTIONABLE DUST, DIRT, SHELLAC, GLUE, AND OTHER TYPES OF FOREIGN MATTER.
7. COAT TO OF ELEMENT EDGE.
8. CHIPS AT EDGES UP TO LONG ALLOWED.
9. EDGE BLACKEN AS PER POLAROID SPEC. NO.
10. LIMIT SAMPLES TO BE PROVIDED BY VENDOR AND APPROVED BY POLAROID OPTICAL ENGINEERING DEPT.

ASPHERIC COEFFICIENTS (WHEN APPLICABLE)

SURFACE	K	D	E	F	G
# 1					
# 2					

$$Y = \frac{CX^2}{1+\sqrt{1-(K+1)C^2X^2}} + DX^4 + EX^6 + FX^8 + GX^{10}$$

$$C = \frac{1}{R}$$

APPLY SIGN CONVENTION TO ASPHERIC RADII.

MATERIAL	INDEX	DISPERSION	GRADE	STRAIN	MIL. SPEC.
BK-7	N_D 1.5167	V_D 64.2	B		

	RADII	TOL.	RINGS	C.A.	TEST PLATE SIZE	TEST PLATE FIT	MAXIMUM IRREG.	MAXIMUM ASTIG.
SURFACE # 1	INCH 7.407	+ − .05		INCH 4.450	INCH FULL	5 C.C.	2 RINGS	2 RINGS
	M.M.	+ −		M.M.	M.M.	C.X.		
SURFACE # 2	INCH 20.91	+ − .1		INCH 4.950	INCH FULL	5 C.C.	2 RINGS	2 RINGS
	M.M.	+ −		M.M.	M.M.	C.X.		

CAD GENERATED DWG
CAD DATA BASE NAME:
EDIT THIS LINE

ASSY NO	ASSY NAME	QTY

RELEASE NUMBER XXXXX

ISSUE /REV	ECN NO. O.R. NO.	DRW	CHK	DESCRIPTION	DFT CHK	APPD	DATE

DO NOT SCALE PRINT

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS IN:
MILLIMETERS □
CENTIMETERS □
METERS □
INCHES ☒
FEET □
______ OTHER □

UNTOLERANCED DIMENSIONS

OVER	TO	TOLERANCE
.XX	.XXX	± .01
.XX	.XXX	± .003
		±
		±
ANGLES		±
EXCEPT AS		

THIRD ANGLE PROJECTION

REFER TO ANSI Y14.5 FOR DRAWING INTERPRETATION

MAXIMUM SURFACE ROUGHNESS ✓

MATERIAL: EDIT THIS TEXT

AISI NO.
FINISH: EDIT THIS TEXT

SPECIFICATIONS:

DRW
ENGR
CHK
PROJ ENGR
SUPV
MGR

EDIT TITLE LINE
EDIT TITLE LINE
EDIT TITLE LINE

AREA NO. ______ BLDG NO. ______

C 000000

ISSUE/REV 0

SHEET 1 OF 1 SCALE X : 1

FOR MICROFILM USE ONLY □

Figure 11

APPLICATIONS OF NODAL TABULATED DRAWINGS

B. Higgins
and
C. Zimmerer
General Electric
Room Air Conditioning
Appliance Park
AP6-247
Louisville, Kentucky 40225
502-452-3024

ABSTRACT

At Room Air Conditioning in Louisville, Kentucky, we have had a CV system since January 1980. We look forward to the day when we can use it as a design tool more than as a Drafting tool. After acquiring the system, we began to build a data base, that is, we began to put manual drawings into the system. This has been and still is a long, tedious process.

Tabulated drawings have been part of the Drafting world for years. They allow a savings because so many parts can be defined on one drawing without having a detailed drawing for each part. Even though the majority of our work is 3-D, we elected to put 2-D tabulated drawings on the system allowing a significant savings of disc space. For system management this was nice, but for design purposes it was not because we needed to have access to each part. The designer needed to be able to add these parts in 3-D to his assembly or layout drawings.

By calling a master execute file which controls the building of the part according to the specifications on the tabulated drawing, the designer now has the ability to add that specific part to his drawing. The master execute file controls execution of a series of data extract/data merge, edit, CV Teco, and Parex file routines which build the part per the specification. The designer only needs to know the tabulated drawing part number to add the part to his drawing.

At the same time, we also realized that the use of nodal information on the tabulated drawing would not only allow the use of data extract for drawing purposes, but also aid to further design development from an engineer's point of view. These 2-D tabulated drawings can now be used for other extraction, such as formulating a listing of certain types of parts along with any desired information from the tabulated drawing. This is accomplished with very little effort since only a short data extract needs to be written.

Use of these tabulated drawings allow us to carry one CAD file and 3–4 system files in exchange for a 3-D drawing of each part on the tabulated drawing. We plan to expand our use of data extract procedures for record keeping of drawing information, material extraction and etc.

INTRODUCTION

A few months after acquiring a system we decided to investigate what we could do with Parex Files. We wrote files that could draw a crude fan motor, a part that could be used in assembly drawings. The operator input certain variables, such as shaft lengths and diameters, and the system would build that fan motor. (The operator had to know the specifications of that particular fan motor.)

The ideal situation would allow the operator to input the part number and the system would draw a 3-D part according to the specifications of that particular part. A procedure allowing this is now available. This procedure uses a series of data extract, data merge, edit, CVTECO, Parex and PEP Files to build the part by the input of the part number. The operator does not need to know any of the dimensional specifications of the part, just the part number.

MAP OF PROCEDURE

This process uses an extract file, 3 Parex Files (2 of which are standard for all parts), 2 text files, and a 2-D CADDS File (which can be archived) to support the sequence of data extract/data merge, edit and building of the part by the Parex File. We can access an unlimited number of specific 3-D fan motors with 7 files and not have to change anything after we have called for the part.

A master file shows the parts available to the designer and calls the master Parex File. The master Parex file controls the command sequence including filing the CADDS File the designer is now working on, building and temporarily filing the part, and then returning the designer to his original CADDS File ready to add the part just created. Another Parex File contains the commands needed to build the part. And a 2-D tabulated drawing is created in CADDS from which a master text file is made containing the specific dimensions for each part.

NODAL TABULATED DRAWINGS

The text on the drawing format (See Figure #1, Items 1-8) is Universal Nodal Text. This applies to all of our drawings whether they are tabulated or not. Only the drawing number is necessary for our drawing build-up, but the test is used for other programs to keep track of drawing information.

On tabulated drawings, each line of tabulation is a nodal figure which has been inserted into the drawing. The text nodes have been properly spaced to allow the nodal text to appear under its appropriate column, and specifications are built into them to designate the text height and width and "force" the text to a certain layer. (In order to use data extract, nodal text must be assigned a specific layer or its text node a specific property.) Due to the large number of tabulations for each tabulated part, we decided to give each one a specified unique layer.

After the nodal figure for line of tabulation has been inserted, the nodal text can now be inserted. This process can be repeated for each line of tabulation or the first line can be copied into the remaining lines and the text edited.

The general fan motor data (see Figure #1 Items 9–15) is also needed for our 3-D construction. It cannot be entered into our program as a constant because other fan motors differ in these areas. To extract this information, properties were added to Computervision's Data. Prop File (see Figure #3). These properties were then assigned to the text nodes that contain the fan motor general data.

Now, the fan motor drawing is ready for data extract.

DATA EXTRACT

Here, we define what items are to be extracted from the drawing and specify the format of the text file to be created.

Our data extract file name is PX.FAN. Extract (See Figure #2). Line one begins Section #1 in which the extracted items are identified and given a name to be used in Section #8.

There are two types of information to be extracted (universal and component) so they must be specified. Then each extracted Nodal Text is specifically named either by the attached property or by layer.

Next is Section #2, the page format specifications. The first item listed is 10,000. This is the number of lines of information between each reprint of the header. The next item is the number of columns per page. Seventy-two is standard and this was just enough for our needs. Next is the number of lines within the header, which is seven in our case. The last item of Section Two is the number of items within Section #8.

The next section is seven, the header. This prints out in our text file above each page.

Finally, Section Eight is listed. Here the extracted information is placed in columns, left-justified and the priority order is specified by drawing number and part number indicated by the (1) and (2) across from part and drawing.

After Px.FAN.Extract is compiled, it can be used to extract information from only one drawing, several drawings, or a specified set of drawings. The resulting text file was then added to our Master Tabulated Drawing text file (Tx.Master.Tab, see Figure 4).

This file has all of the needed information to draw a fan motor in 3 dimensions. What we need now is to be able to access this information.

PX.TAB FILE

The PX.Tab File (see Figure 5) allows access to the Master Parex File. At the same time, instruction in using the file, as well as the current tabulated parts on file, are displayed to the designer. After these instructions, the file calls the Master Parex File.

THE MASTER PAREX FILE

The Master Parex File (see Figure 6) controls the command sequence, so that data may be gathered to build the part. This file first asks for input from the designer: his present part file name as well as the name of part to be built (such as fan, cap or topcover), the drawing number of the part to be built, and the user's initials.

After receiving this information, the current display is saved and the part is filed. System level is entered and the long edit command is used to edit a text file (see Figure 7) with the part number and then this is filed under an auxiliary file name (see Figure 8), so the original text file format is left intact to be used again at a later time.

The auxiliary text file now contains the part number and the variable names used in the part Parex File. This file is now merged with the text file created by the data extract on the tabulated drawing.

The MERGE FILE command is the main key to bring the drawing information into a temporary file to be used for construction.

Merge file transfers specified data from one text file to another text file based upon matching elements.

Here, we want to draw from our main source of information, Tx.Master.Tab and we want to transfer specified data to our temporary file Tx.Master.2.

After these two have been entered as our source file and destination file, the system responds, with *Merge Key Elements*: Following this, the column numbers of the information in the source file (Tx.Master.Tab) are entered, a colon, and the column number of the destination file are entered. This transfer is repeated until all of the desired information from the source file is transferred into the destination file. A double carriage return then returns you to CADDS level. The variables now contain the specific data for the unique part (see Figure 9), but this data is in the wrong syntax to be used by the Parex File.

CV TECO is now entered and the Auxiliary Text File is edited to the right syntax and the change filed (see Figure 10). The part Parex File is then edited and the former variables deleted and the new variables are extracted from the Auxiliary Text File and inserted into the proper sequence in the part Parex File. These changes are made and a temporary CADDS part is entered and the part Parex File is run. The part Parex File draws the part per the specifications and then files the part. The designer's original part file is entered; the display is restored and he is ready to add any modifiers to insert the part. The designer has typed 4 variables and has received a 3-D model of the part ready for insertion in his original work.

PAREX PART FILE

This file is names PX.FAN (or PX.CAP, etc.), see Figure 11. This file contains the commands necessary to draw the part with enough detail to be used in assembly and layout drawings. This Parex File also calls a PEP file (see Figure 12). The PEP File was made necessary for the use of conditional statements.

ADDITIONAL BENEFITS

An additional benefit from the nodal drawing was realized after the data was input in the drawing. All of the tabulated data is accessible by data extract. The electrical data on the drawing is constantly being used by our engineers, and to have this data prioritized by the computer would be of great benefit to them. A sample of a data extract and listing is shown in Figure #13 and #14.

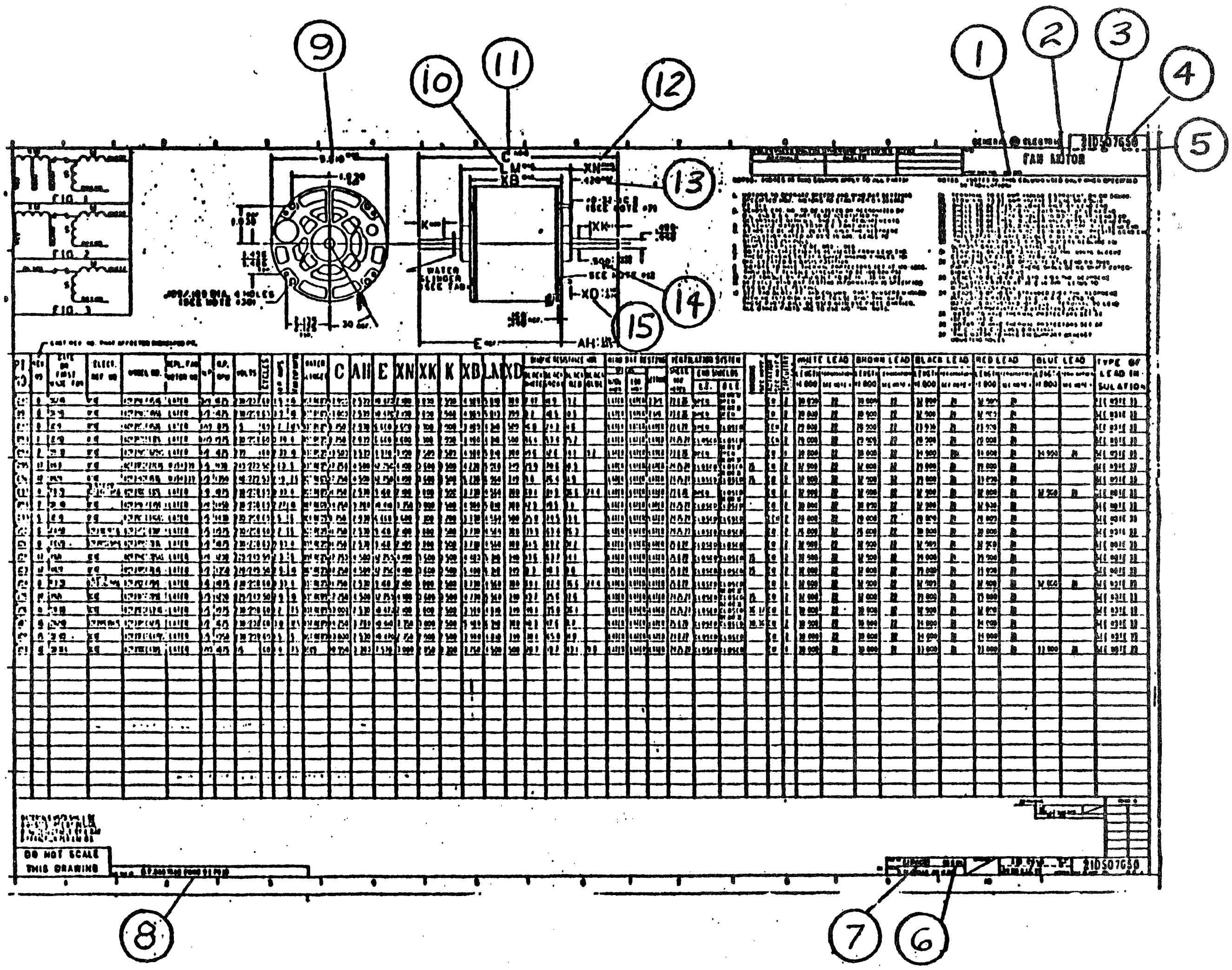
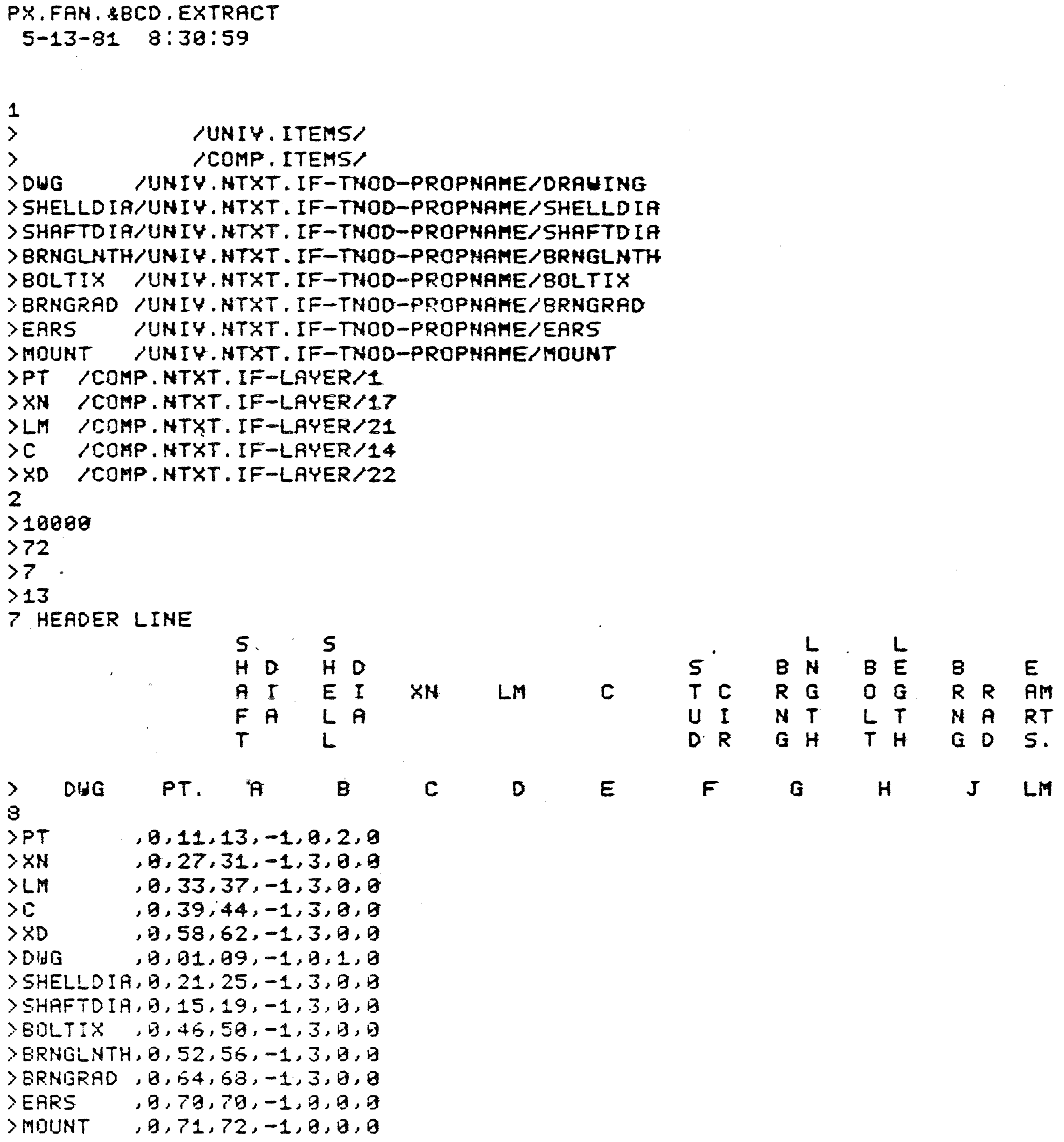
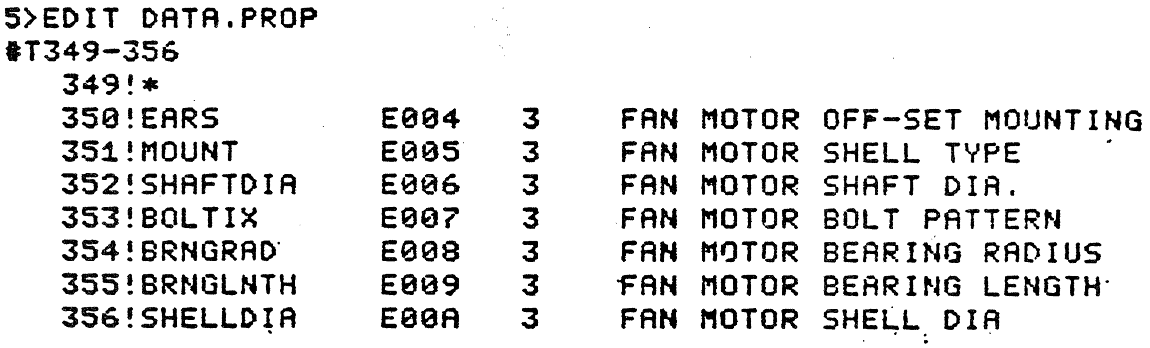


Figure 1



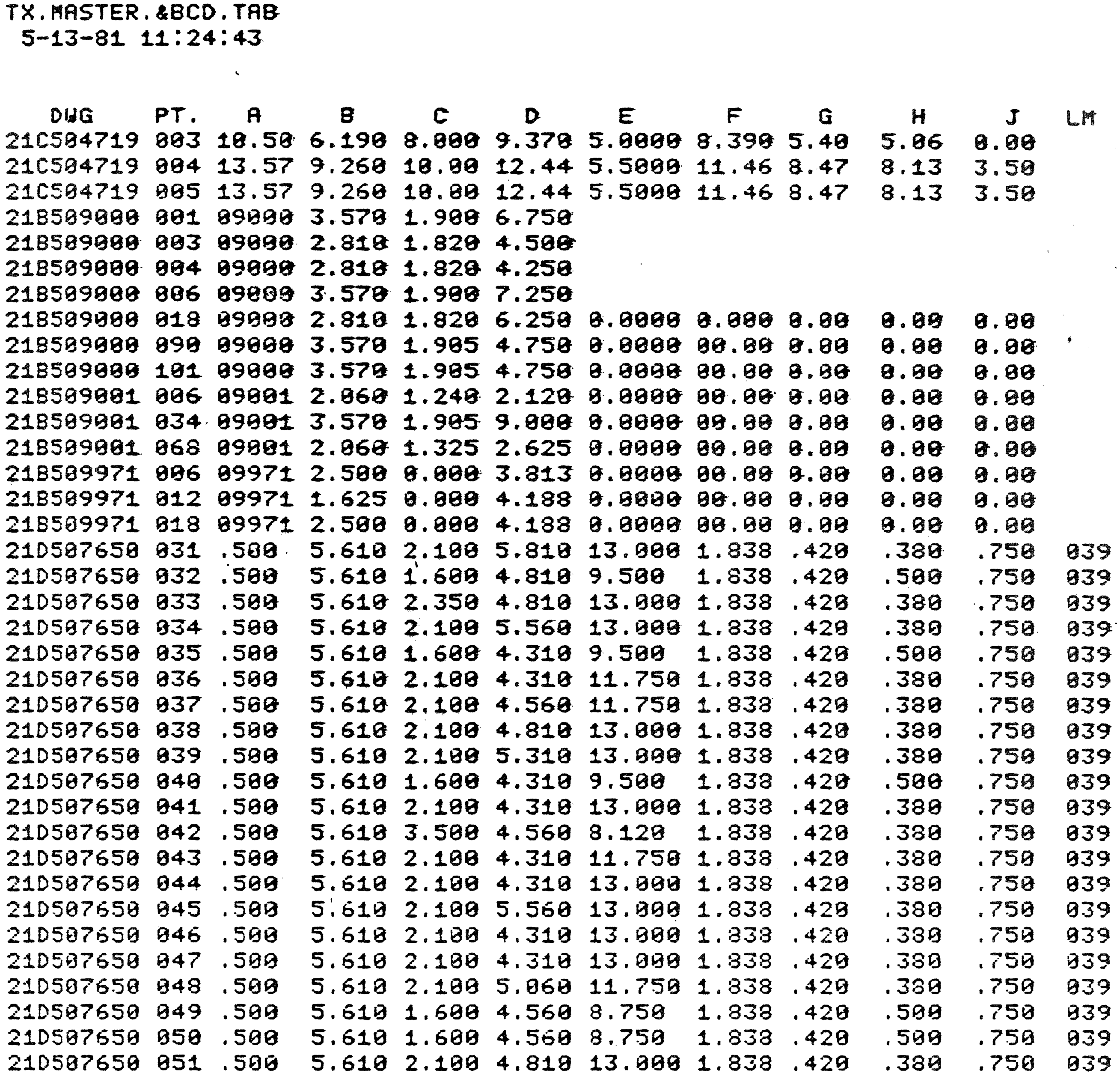
```
PX.FAN.&BCD.EXTRACT
 5-13-81  8:30:59


1
>            /UNIV.ITEMS/
>            /COMP.ITEMS/
>DWG     /UNIV.NTXT.IF-TNOD-PROPNAME/DRAWING
>SHELLDIA/UNIV.NTXT.IF-TNOD-PROPNAME/SHELLDIA
>SHAFTDIA/UNIV.NTXT.IF-TNOD-PROPNAME/SHAFTDIA
>BRNGLNTH/UNIV.NTXT.IF-TNOD-PROPNAME/BRNGLNTH
>BOLTIX  /UNIV.NTXT.IF-TNOD-PROPNAME/BOLTIX
>BRNGRAD /UNIV.NTXT.IF-TNOD-PROPNAME/BRNGRAD
>EARS    /UNIV.NTXT.IF-TNOD-PROPNAME/EARS
>MOUNT   /UNIV.NTXT.IF-TNOD-PROPNAME/MOUNT
>PT  /COMP.NTXT.IF-LAYER/1
>XN  /COMP.NTXT.IF-LAYER/17
>LM  /COMP.NTXT.IF-LAYER/21
>C   /COMP.NTXT.IF-LAYER/14
>XD  /COMP.NTXT.IF-LAYER/22
2
>10000
>72
>7
>13
7 HEADER LINE
                S     S                                     L     L
                H D   H D                          S     B N   B E   B   E
                A I   E I   XN    LM     C         T C   R G   O G   R R AM
                F A   L A                          U I   N T   L T   N A RT
                T     L                            D R   G H   T H   G D S.

>   DWG     PT.   A     B     C     D     E      F     G     H     J   LM
8
>PT      ,0,11,13,-1,0,2,0
>XN      ,0,27,31,-1,3,0,0
>LM      ,0,33,37,-1,3,0,0
>C       ,0,39,44,-1,3,0,0
>XD      ,0,58,62,-1,3,0,0
>DWG     ,0,01,09,-1,0,1,0
>SHELLDIA,0,21,25,-1,3,0,0
>SHAFTDIA,0,15,19,-1,3,0,0
>BOLTIX  ,0,46,50,-1,3,0,0
>BRNGLNTH,0,52,56,-1,3,0,0
>BRNGRAD ,0,64,68,-1,3,0,0
>EARS    ,0,70,70,-1,0,0,0
>MOUNT   ,0,71,72,-1,0,0,0
```

Figure 2



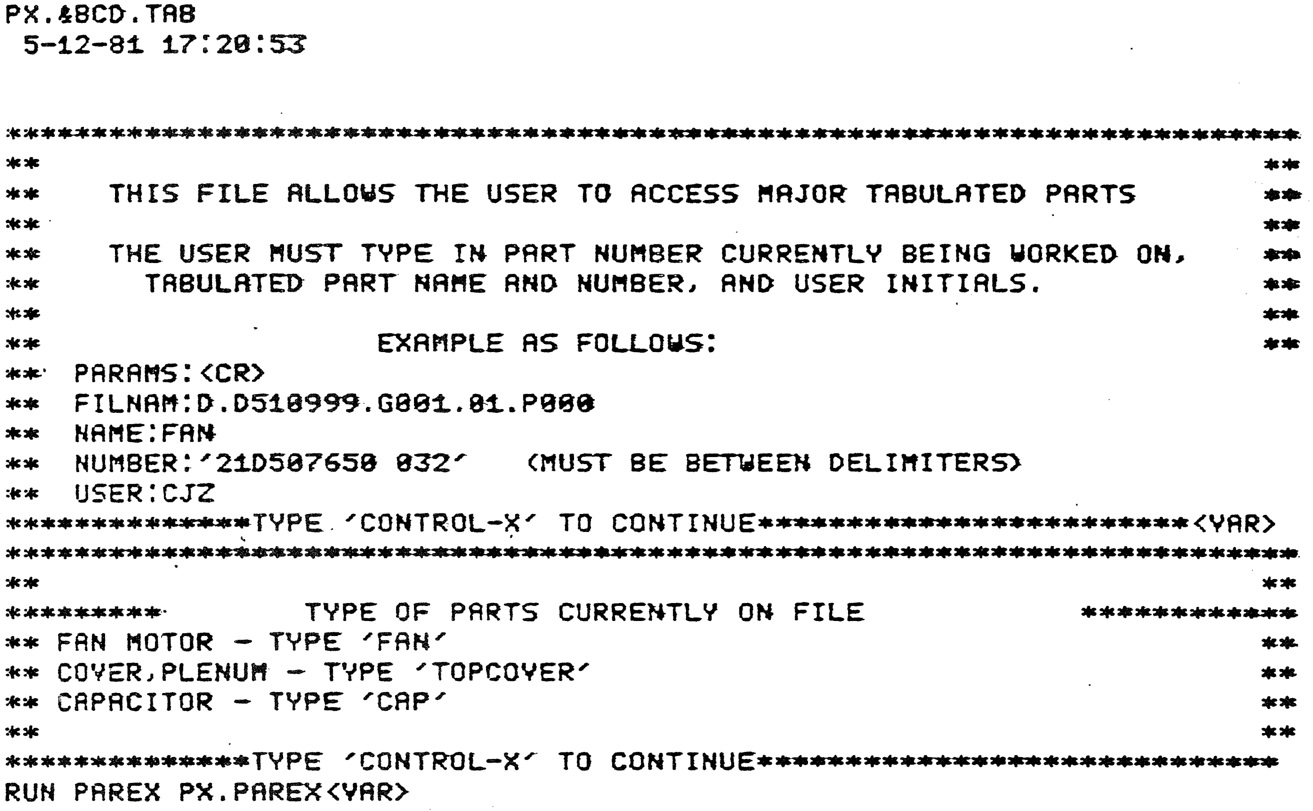
```
5>EDIT DATA.PROP
#T349-356
   349!*
   350!EARS          E004    3    FAN MOTOR OFF-SET MOUNTING
   351!MOUNT         E005    3    FAN MOTOR SHELL TYPE
   352!SHAFTDIA      E006    3    FAN MOTOR SHAFT DIA.
   353!BOLTIX        E007    3    FAN MOTOR BOLT PATTERN
   354!BRNGRAD       E008    3    FAN MOTOR BEARING RADIUS
   355!BRNGLNTH      E009    3    FAN MOTOR BEARING LENGTH
   356!SHELLDIA      E00A    3    FAN MOTOR SHELL DIA
```

Figure 3



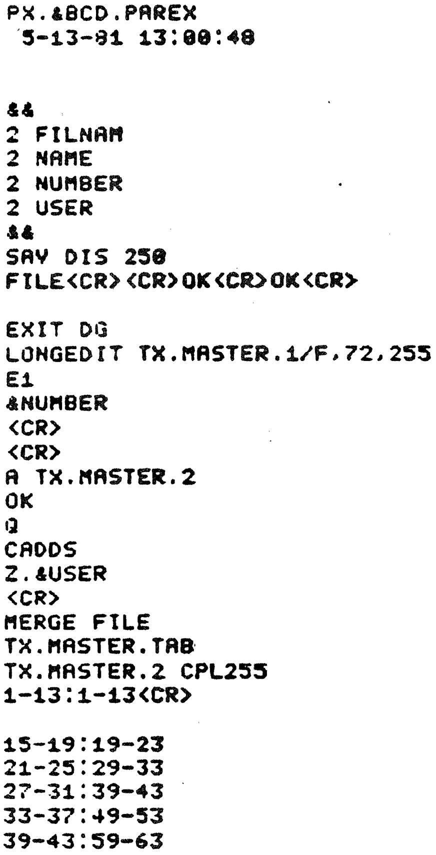
```
TX.MASTER.&BCD.TAB
 5-13-81 11:24:43


   DWG      PT.    A      B      C      D      E       F      G      H      J    LM
21C504719 003 10.50 6.190 8.000 9.370 5.0000 8.390 5.40  5.06  0.00
21C504719 004 13.57 9.260 10.00 12.44 5.5000 11.46 8.47  8.13  3.50
21C504719 005 13.57 9.260 10.00 12.44 5.5000 11.46 8.47  8.13  3.50
21B509000 001 09000 3.570 1.900 6.750
21B509000 003 09000 2.810 1.820 4.500
21B509000 004 09000 2.810 1.820 4.250
21B509000 006 09009 3.570 1.900 7.250
21B509000 018 09000 2.810 1.820 6.250 0.0000 0.000 0.00  0.00  0.00
21B509000 090 09000 3.570 1.905 4.750 0.0000 00.00 0.00  0.00  0.00
21B509000 101 09000 3.570 1.905 4.750 0.0000 00.00 0.00  0.00  0.00
21B509001 006 09001 2.060 1.240 2.120 0.0000 00.00 0.00  0.00  0.00
21B509001 034 09001 3.570 1.905 9.000 0.0000 09.00 0.00  0.00  0.00
21B509001 068 09001 2.060 1.325 2.625 0.0000 00.00 0.00  0.00  0.00
21B509971 006 09971 2.500 0.000 3.813 0.0000 00.00 0.00  0.00  0.00
21B509971 012 09971 1.625 0.000 4.188 0.0000 00.00 0.00  0.00  0.00
21B509971 018 09971 2.500 0.000 4.188 0.0000 00.00 0.00  0.00  0.00
21D507650 031 .500  5.610 2.100 5.810 13.000 1.838 .420  .380  .750  039
21D507650 032 .500  5.610 1.600 4.810 9.500  1.838 .420  .500  .750  039
21D507650 033 .500  5.610 2.350 4.810 13.000 1.838 .420  .380  .750  039
21D507650 034 .500  5.610 2.100 5.560 13.000 1.838 .420  .380  .750  039
21D507650 035 .500  5.610 1.600 4.310 9.500  1.838 .420  .500  .750  039
21D507650 036 .500  5.610 2.100 4.310 11.750 1.838 .420  .380  .750  039
21D507650 037 .500  5.610 2.100 4.560 11.750 1.838 .420  .380  .750  039
21D507650 038 .500  5.610 2.100 4.810 13.000 1.838 .420  .380  .750  039
21D507650 039 .500  5.610 2.100 5.310 13.000 1.838 .420  .380  .750  039
21D507650 040 .500  5.610 1.600 4.310 9.500  1.838 .420  .500  .750  039
21D507650 041 .500  5.610 2.100 4.310 13.000 1.838 .420  .380  .750  039
21D507650 042 .500  5.610 3.500 4.560 8.120  1.838 .420  .380  .750  039
21D507650 043 .500  5.610 2.100 4.310 11.750 1.838 .420  .380  .750  039
21D507650 044 .500  5.610 2.100 4.310 13.000 1.838 .420  .380  .750  039
21D507650 045 .500  5.610 2.100 5.560 13.000 1.838 .420  .380  .750  039
21D507650 046 .500  5.610 2.100 4.310 13.000 1.838 .420  .380  .750  039
21D507650 047 .500  5.610 2.100 4.310 13.000 1.838 .420  .380  .750  039
21D507650 048 .500  5.610 2.100 5.060 11.750 1.838 .420  .380  .750  039
21D507650 049 .500  5.610 1.600 4.560 8.750  1.838 .420  .500  .750  039
21D507650 050 .500  5.610 1.600 4.560 8.750  1.838 .420  .500  .750  039
21D507650 051 .500  5.610 2.100 4.810 13.000 1.838 .420  .380  .750  039
```

Figure 4



```
PX.&BCD.TAB
 5-12-81 17:20:53


**************************************************************************
**                                                                      **
**    THIS FILE ALLOWS THE USER TO ACCESS MAJOR TABULATED PARTS         **
**                                                                      **
**    THE USER MUST TYPE IN PART NUMBER CURRENTLY BEING WORKED ON,      **
**      TABULATED PART NAME AND NUMBER, AND USER INITIALS.              **
**                                                                      **
**                   EXAMPLE AS FOLLOWS:                                **
**  PARAMS:<CR>
**  FILNAM:D.D510999.G001.01.P000
**  NAME:FAN
**  NUMBER:'21D507650 032'    (MUST BE BETWEEN DELIMITERS)
**  USER:CJZ
**************TYPE 'CONTROL-X' TO CONTINUE**************************<VAR>
**************************************************************************
**                                                                      **
*********        TYPE OF PARTS CURRENTLY ON FILE             ************
** FAN MOTOR - TYPE 'FAN'                                               **
** COVER,PLENUM - TYPE 'TOPCOVER'                                       **
** CAPACITOR - TYPE 'CAP'                                               **
**                                                                      **
**************TYPE 'CONTROL-X' TO CONTINUE*******************************
RUN PAREX PX.PAREX<VAR>
```

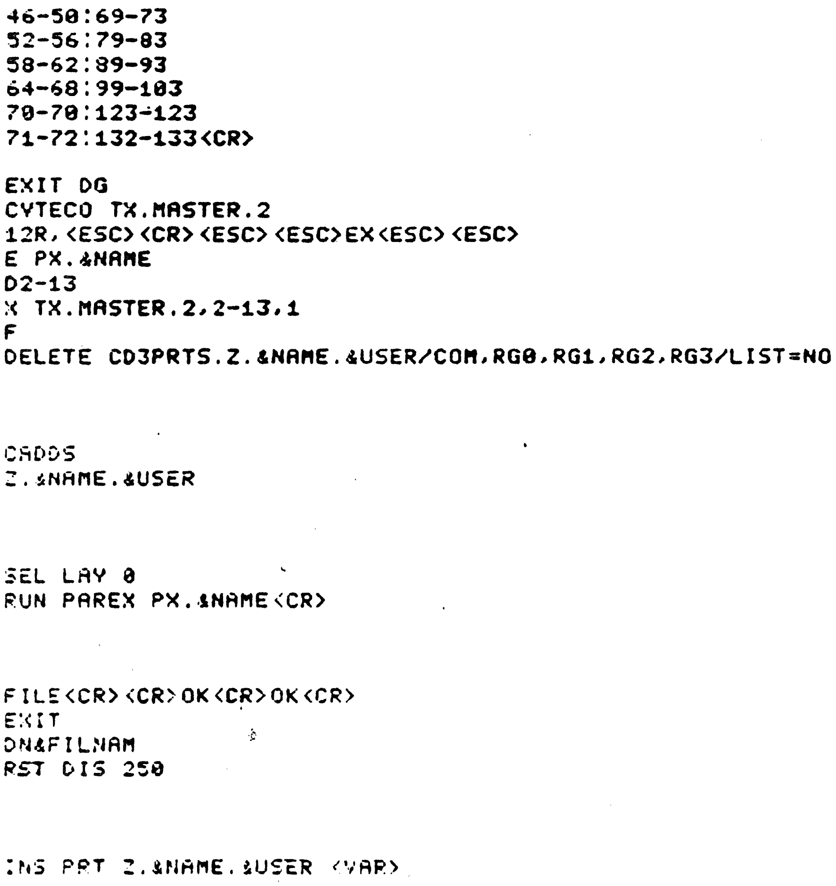
Figure 5



```
PX.&BCD.PAREX
 5-13-81 13:00:40


&&
2 FILNAM
2 NAME
2 NUMBER
2 USER
&&
SAV DIS 250
FILE<CR><CR>OK<CR>OK<CR>

EXIT DG
LONGEDIT TX.MASTER.1/F,72,255
E1
&NUMBER
<CR>
<CR>
A TX.MASTER.2
OK
Q
CADDS
Z.&USER
<CR>
MERGE FILE
TX.MASTER.TAB
TX.MASTER.2 CPL255
1-13:1-13<CR>

15-19:19-23
21-25:29-33
27-31:39-43
33-37:49-53
39-43:59-63
```

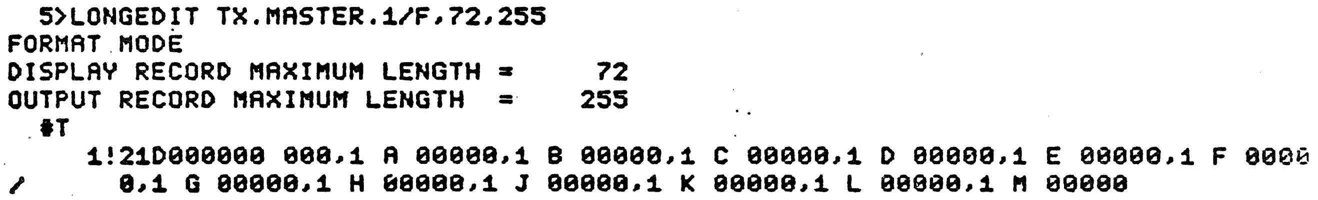
Figure 6

(continued)



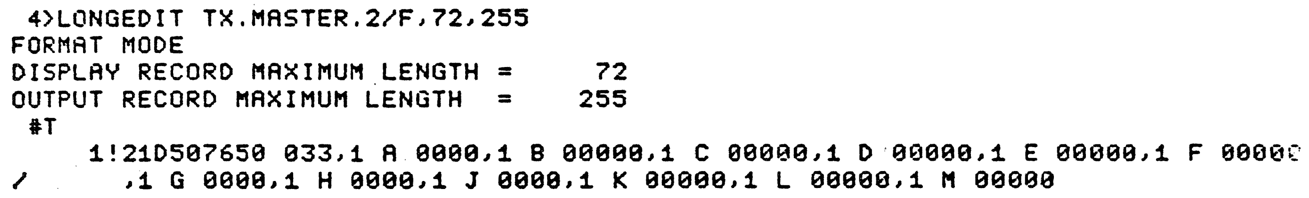
```
46-50:69-73
52-56:79-83
58-62:89-93
64-68:99-103
70-70:123-123
71-72:132-133<CR>

EXIT DG
CYTECO TX.MASTER.2
12R,<ESC><CR><ESC><ESC>EX<ESC><ESC>
E PX.&NAME
D2-13
X TX.MASTER.2,2-13,1
F
DELETE CD3PRTS.Z.&NAME.&USER/COM,RG0,RG1,RG2,RG3/LIST=NO


CADDS
Z.&NAME.&USER


SEL LAY 0
RUN PAREX PX.&NAME<CR>


FILE<CR><CR>OK<CR>OK<CR>
EXIT
DN&FILNAM
RST DIS 250


INS PRT Z.&NAME.&USER <VAR>
```

Figure 6 (continued)



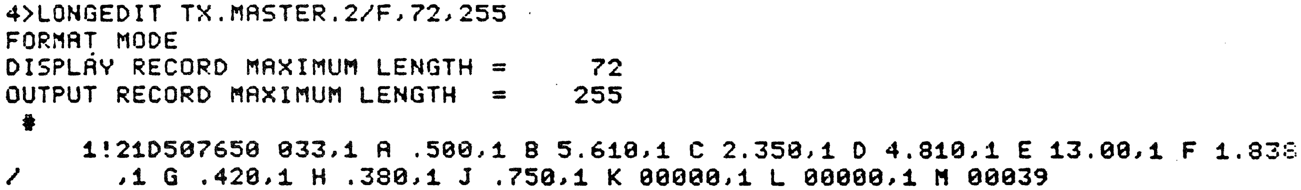
```
  5>LONGEDIT TX.MASTER.1/F,72,255
FORMAT MODE
DISPLAY RECORD MAXIMUM LENGTH =     72
OUTPUT RECORD MAXIMUM LENGTH  =    255
  #T
     1!21D000000 000,1 A 00000,1 B 00000,1 C 00000,1 D 00000,1 E 00000,1 F 0000
/       0,1 G 00000,1 H 00000,1 J 00000,1 K 00000,1 L 00000,1 M 00000
```

Figure 7



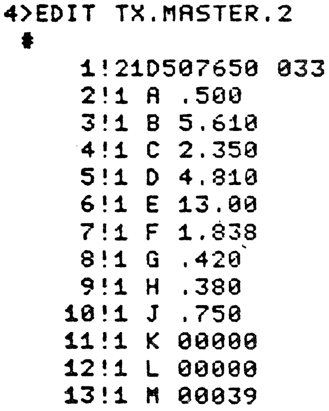
```
 4>LONGEDIT TX.MASTER.2/F,72,255
FORMAT MODE
DISPLAY RECORD MAXIMUM LENGTH =     72
OUTPUT RECORD MAXIMUM LENGTH  =    255
 #T
     1!21D507650 033,1 A 0000,1 B 00000,1 C 00000,1 D 00000,1 E 00000,1 F 00000
/       ,1 G 0000,1 H 0000,1 J 0000,1 K 00000,1 L 00000,1 M 00000
```

Figure 8



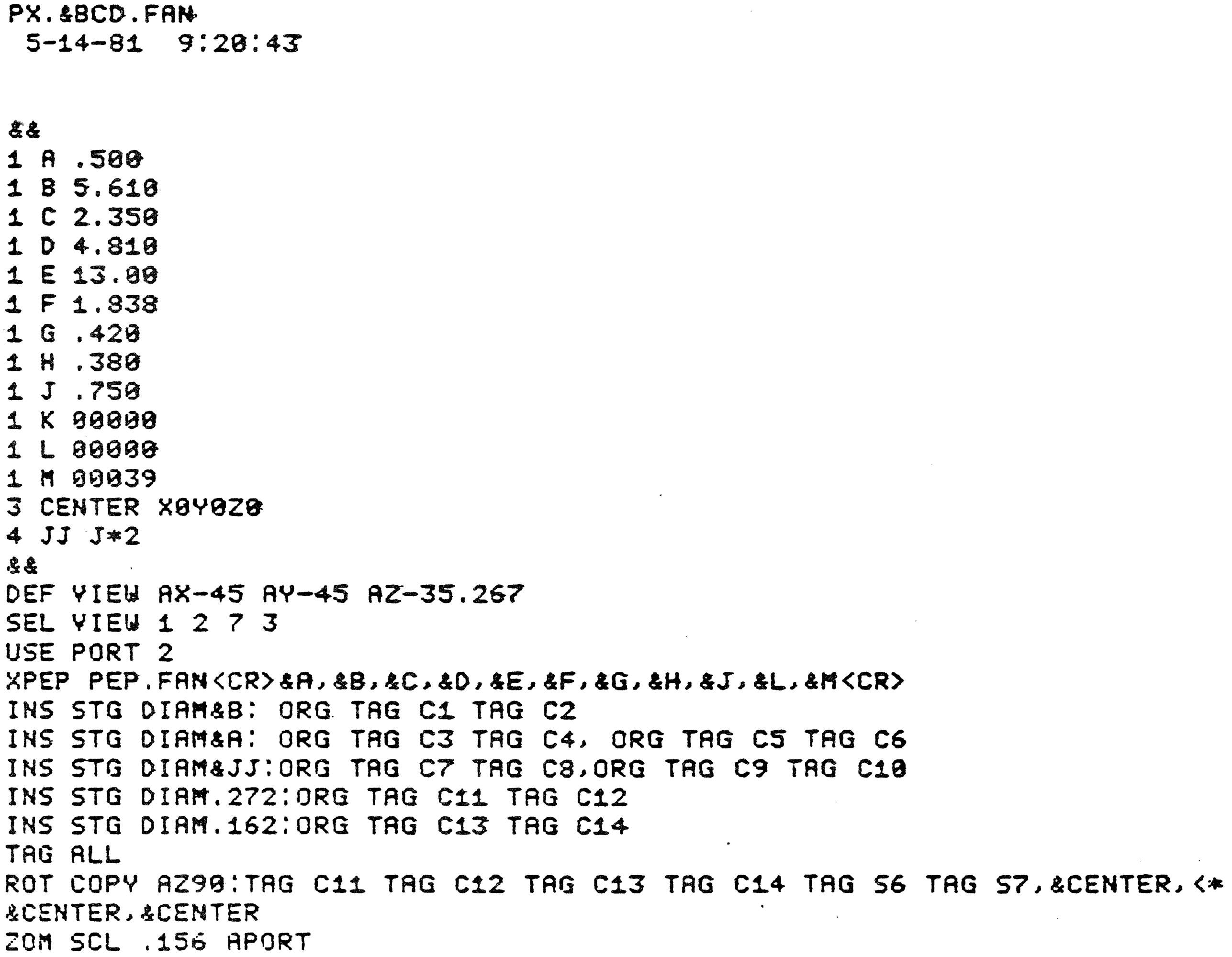
```
4>LONGEDIT TX.MASTER.2/F,72,255
FORMAT MODE
DISPLAY RECORD MAXIMUM LENGTH =      72
OUTPUT RECORD MAXIMUM LENGTH  =     255
 #
    1!21D507650 033,1 A .500,1 B 5.610,1 C 2.350,1 D 4.810,1 E 13.00,1 F 1.838
/      ,1 G .420,1 H .380,1 J .750,1 K 00000,1 L 00000,1 M 00039
```

Figure 9



```
4>EDIT TX.MASTER.2
 #
    1!21D507650 033
    2!1 A .500
    3!1 B 5.610
    4!1 C 2.350
    5!1 D 4.810
    6!1 E 13.00
    7!1 F 1.838
    8!1 G .420
    9!1 H .380
   10!1 J .750
   11!1 K 00000
   12!1 L 00000
   13!1 M 00039
```

Figure 10

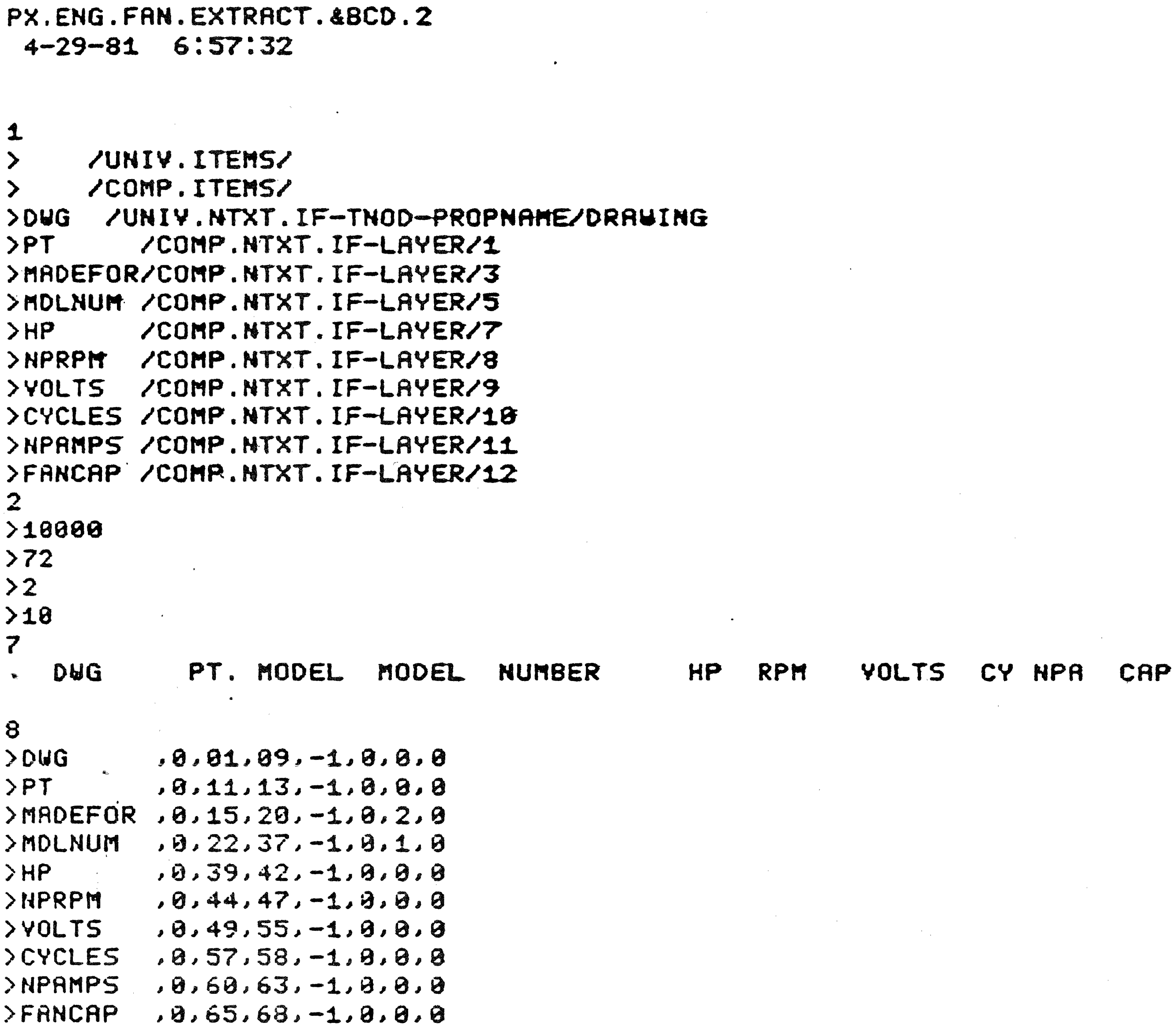


```
PX.&BCD.FAN
 5-14-81  9:20:43


&&
1 A .500
1 B 5.610
1 C 2.350
1 D 4.810
1 E 13.00
1 F 1.838
1 G .420
1 H .380
1 J .750
1 K 00000
1 L 00000
1 M 00039
3 CENTER X0Y0Z0
4 JJ J*2
&&
DEF VIEW AX-45 AY-45 AZ-35.267
SEL VIEW 1 2 7 3
USE PORT 2
XPEP PEP.FAN<CR>&A,&B,&C,&D,&E,&F,&G,&H,&J,&L,&M<CR>
INS STG DIAM&B: ORG TAG C1 TAG C2
INS STG DIAM&A: ORG TAG C3 TAG C4, ORG TAG C5 TAG C6
INS STG DIAM&JJ:ORG TAG C7 TAG C8,ORG TAG C9 TAG C10
INS STG DIAM.272:ORG TAG C11 TAG C12
INS STG DIAM.162:ORG TAG C13 TAG C14
TAG ALL
ROT COPY AZ90:TAG C11 TAG C12 TAG C13 TAG C14 TAG S6 TAG S7,&CENTER,<*
&CENTER,&CENTER
ZOM SCL .156 APORT
```

Figure 11

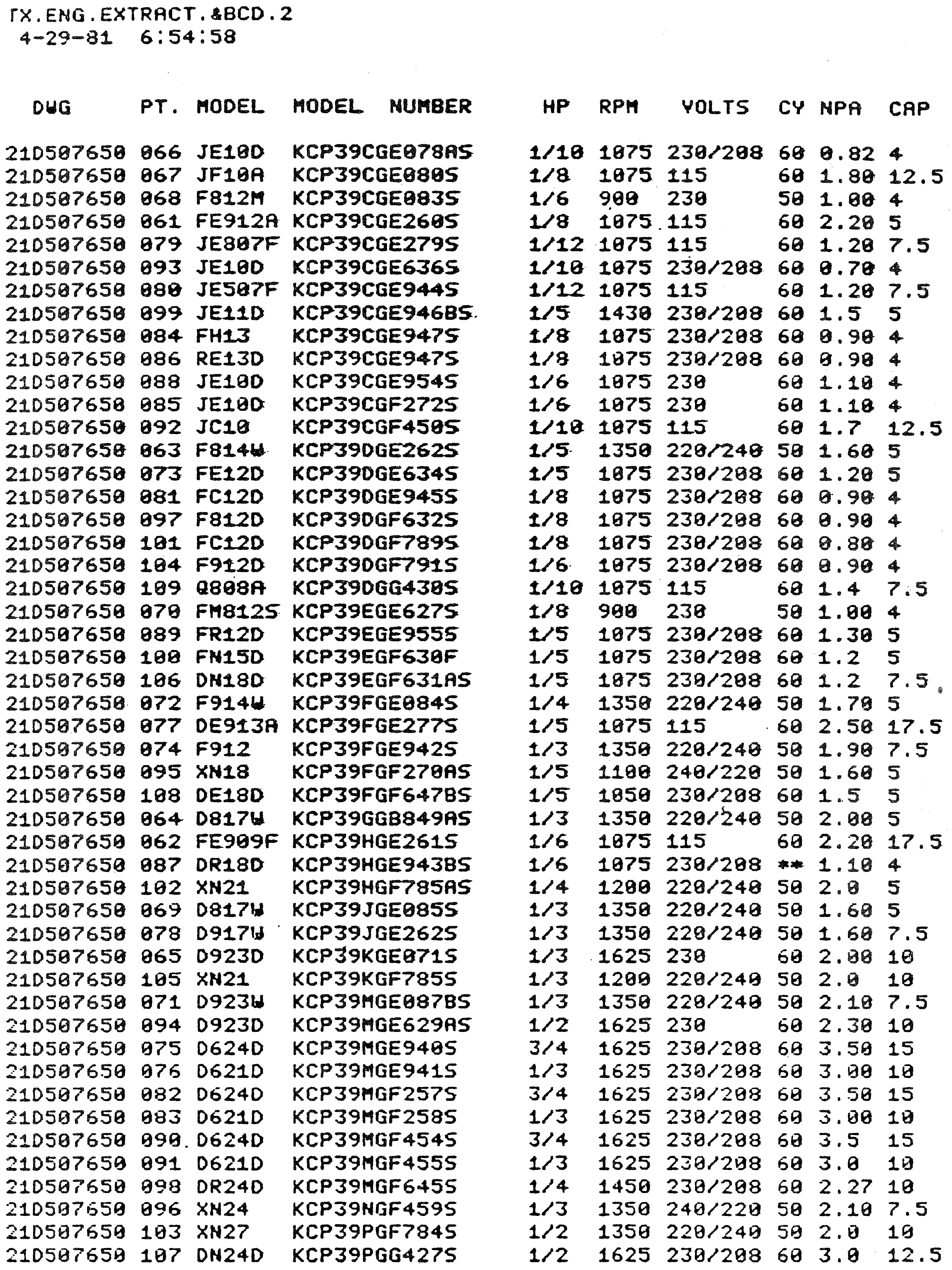
```
  B,C,D,E,F,G,H,J,L,M
  ,JUMPTO/LF



 ;9,JUMPTO/L10


 (2*G))

 5


 CENTER,0,0,-.1875,RADIUS,BB
 'CENTER,0,0,-FF,RADIUS,BB
 'CENTER,0,0,-GG,RADIUS,AA
 'CENTER,0,0,-KK,RADIUS,AA
 'CENTER,0,0,HH,RADIUS,AA
 /CENTER,0,0,LL,RADIUS,AA
 .0,JUMPTO/L30
 /CENTER,0,0,-FF,RADIUS,J
 /CENTER,0,0,-GG,RADIUS,J
 /CENTER,0,0,-.1875,RADIUS,J
 E/CENTER,0,0,HH,RADIUS,J
 .0,JUMPTO/LF
.E/CENTER,F,F,-.1875,RADIUS,.186
.E/CENTER,F,F,0,RADIUS,.186
.E/CENTER,F,F,0,RADIUS,.083
.E/CENTER,F,F,H,RADIUS,.083
:
```

Figure 12



```
PX.ENG.FAN.EXTRACT.&BCD.2
 4-29-81  6:57:32


1
>    /UNIV.ITEMS/
>    /COMP.ITEMS/
>DWG  /UNIV.NTXT.IF-TNOD-PROPNAME/DRAWING
>PT      /COMP.NTXT.IF-LAYER/1
>MADEFOR/COMP.NTXT.IF-LAYER/3
>MDLNUM /COMP.NTXT.IF-LAYER/5
>HP      /COMP.NTXT.IF-LAYER/7
>NPRPM  /COMP.NTXT.IF-LAYER/8
>VOLTS  /COMP.NTXT.IF-LAYER/9
>CYCLES /COMP.NTXT.IF-LAYER/10
>NPAMPS /COMP.NTXT.IF-LAYER/11
>FANCAP /COMP.NTXT.IF-LAYER/12
2
>10000
>72
>2
>10
7
.  DWG      PT. MODEL  MODEL  NUMBER     HP  RPM   VOLTS  CY NPA  CAP

8
>DWG      ,0,01,09,-1,0,0,0
>PT       ,0,11,13,-1,0,0,0
>MADEFOR ,0,15,20,-1,0,2,0
>MDLNUM  ,0,22,37,-1,0,1,0
>HP       ,0,39,42,-1,0,0,0
>NPRPM    ,0,44,47,-1,0,0,0
>VOLTS    ,0,49,55,-1,0,0,0
>CYCLES   ,0,57,58,-1,0,0,0
>NPAMPS   ,0,60,63,-1,0,0,0
>FANCAP   ,0,65,68,-1,0,0,0
```

Figure 13



```
FX.ENG.EXTRACT.&BCD.2
 4-29-81  6:54:58

  DWG       PT. MODEL  MODEL  NUMBER       HP   RPM   VOLTS   CY NPA  CAP

21D507650 066 JE10D  KCP39CGE078AS     1/10 1075 230/208 60 0.82 4
21D507650 067 JF10A  KCP39CGE080S      1/8  1075 115     60 1.80 12.5
21D507650 068 F812M  KCP39CGE083S      1/6  900  230     50 1.00 4
21D507650 061 FE912A KCP39CGE260S      1/8  1075 115     60 2.20 5
21D507650 079 JE807F KCP39CGE279S      1/12 1075 115     60 1.20 7.5
21D507650 093 JE10D  KCP39CGE636S      1/10 1075 230/208 60 0.70 4
21D507650 080 JE507F KCP39CGE944S      1/12 1075 115     60 1.20 7.5
21D507650 099 JE11D  KCP39CGE946BS     1/5  1430 230/208 60 1.5  5
21D507650 084 FH13   KCP39CGE947S      1/8  1075 230/208 60 0.90 4
21D507650 086 RE13D  KCP39CGE947S      1/8  1075 230/208 60 0.90 4
21D507650 088 JE10D  KCP39CGE954S      1/6  1075 230     60 1.10 4
21D507650 085 JE10D  KCP39CGF272S      1/6  1075 230     60 1.10 4
21D507650 092 JC10   KCP39CGF450S      1/10 1075 115     60 1.7  12.5
21D507650 063 F814W  KCP39DGE262S      1/5  1350 220/240 50 1.60 5
21D507650 073 FE12D  KCP39DGE634S      1/5  1075 230/208 60 1.20 5
21D507650 081 FC12D  KCP39DGE945S      1/8  1075 230/208 60 0.90 4
21D507650 097 F812D  KCP39DGF632S      1/8  1075 230/208 60 0.90 4
21D507650 101 FC12D  KCP39DGF789S      1/8  1075 230/208 60 0.80 4
21D507650 104 F912D  KCP39DGF791S      1/6  1075 230/208 60 0.90 4
21D507650 109 Q808A  KCP39DGG430S      1/10 1075 115     60 1.4  7.5
21D507650 070 FM812S KCP39EGE627S      1/8  900  230     50 1.00 4
21D507650 089 FR12D  KCP39EGE955S      1/5  1075 230/208 60 1.30 5
21D507650 100 FN15D  KCP39EGF630F      1/5  1075 230/208 60 1.2  5
21D507650 106 DN18D  KCP39EGF631AS     1/5  1075 230/208 60 1.2  7.5
21D507650 072 F914W  KCP39FGE084S      1/4  1350 220/240 50 1.70 5
21D507650 077 DE913A KCP39FGE277S      1/5  1075 115     60 2.50 17.5
21D507650 074 F912   KCP39FGE942S      1/3  1350 220/240 50 1.90 7.5
21D507650 095 XN18   KCP39FGF270AS     1/5  1100 240/220 50 1.60 5
21D507650 108 DE18D  KCP39FGF647BS     1/5  1050 230/208 60 1.5  5
21D507650 064 D817W  KCP39GGB849AS     1/3  1350 220/240 50 2.00 5
21D507650 062 FE909F KCP39HGE261S      1/6  1075 115     60 2.20 17.5
21D507650 087 DR18D  KCP39HGE943BS     1/6  1075 230/208 ** 1.10 4
21D507650 102 XN21   KCP39HGF785AS     1/4  1200 220/240 50 2.0  5
21D507650 069 D817W  KCP39JGE085S      1/3  1350 220/240 50 1.60 5
21D507650 078 D917W  KCP39JGE262S      1/3  1350 220/240 50 1.60 7.5
21D507650 065 D923D  KCP39KGE071S      1/3  1625 230     60 2.00 10
21D507650 105 XN21   KCP39KGF785S      1/3  1200 220/240 50 2.0  10
21D507650 071 D923W  KCP39MGE087BS     1/3  1350 220/240 50 2.10 7.5
21D507650 094 D923D  KCP39MGE629AS     1/2  1625 230     60 2.30 10
21D507650 075 D624D  KCP39MGE940S      3/4  1625 230/208 60 3.50 15
21D507650 076 D621D  KCP39MGE941S      1/3  1625 230/208 60 3.00 10
21D507650 082 D624D  KCP39MGF257S      3/4  1625 230/208 60 3.50 15
21D507650 083 D621D  KCP39MGF258S      1/3  1625 230/208 60 3.00 10
21D507650 090 D624D  KCP39MGF454S      3/4  1625 230/208 60 3.5  15
21D507650 091 D621D  KCP39MGF455S      1/3  1625 230/208 60 3.0  10
21D507650 098 DR24D  KCP39MGF645S      1/4  1450 230/208 60 2.27 10
21D507650 096 XN24   KCP39NGF459S      1/3  1350 240/220 50 2.10 7.5
21D507650 103 XN27   KCP39PGF784S      1/2  1350 220/240 50 2.0  10
21D507650 107 DN24D  KCP39PGG427S      1/2  1625 230/208 60 3.0  12.5
```

Figure 14

THE SHOCK ABSORBER — AN EXAMPLE OF VARIANT DESIGN WITH VARPRO AT VOLKSWAGEN

Egbert Elsholz
Volkswagonwerk, AG
Postfach
3180 Wolfsburg 1
Germany

INTRODUCTION

The most important benefit of CAD systems is the gain in productivity. The ideal system would have the following CV command:

```
CON RAB DOORS 4 HP 75 CYLS 4
```

(which stands for "construct Rabbit"). It would design a complete car called the "Rabbit" with all its components. The user would have to define the number of doors and the engine would be a four cylinder 75 hp engine. Unfortunately this is impossible today and will be so in future.

Although it is not possible for a complete car it need not be for some of its components. One of these components might be the shock absorber consisting of a number of different parts, most of which are simple contours and can be described as rotational shapes. The typical shape of each of these parts will not be changed from one model to another, however the dimensioning of some of the geometrical data will be different, depending on the different forces, weight and other mechanical surroundings defined by the particular car model.

Other examples for this application in the automotive industry could be brake design or parts of the engine like the crankshaft, camshaft, or gear.

ANALYSIS AND AIM OF THE PROJECT

We decided to program the shock absorber, because the pilot user in the research and development at Volkswagen was the axle design department who thus could make use of this technique first. The aim of the project was to generate technical drawings (2D-information only) for further use in the prototype shop.

Before starting this project we had to examine the different programming facilities within CV systems. We discussed the usage of PAREX or EXECUTE files, as well as PEP programs; also programming in FORTRAN seemed to be possible. Finally we came to the conclusion that VARPRO would be the tool to do this job.

At that point we had our task defined and we had the tool to do it with. Now the discussions with the designers had to go into details. The problem is: the designers in general have not been educated in programming; likewise the programmers are not educated and experienced in designing. Also both groups talk in their own languages from which quite a few misunderstandings develop.

On the evolutionary basis of trial and error we finally came to an agreement. The basic programming started in CADDS-3 Rev., 10.03 and used the old VARPRO revision. At the present we have a general structure running, as well as about half of the mechanical parts programmed. A lot of work is still to be done. However, the new VARPRO with Rev. 11.0E will be used and should simplify a lot of problems.

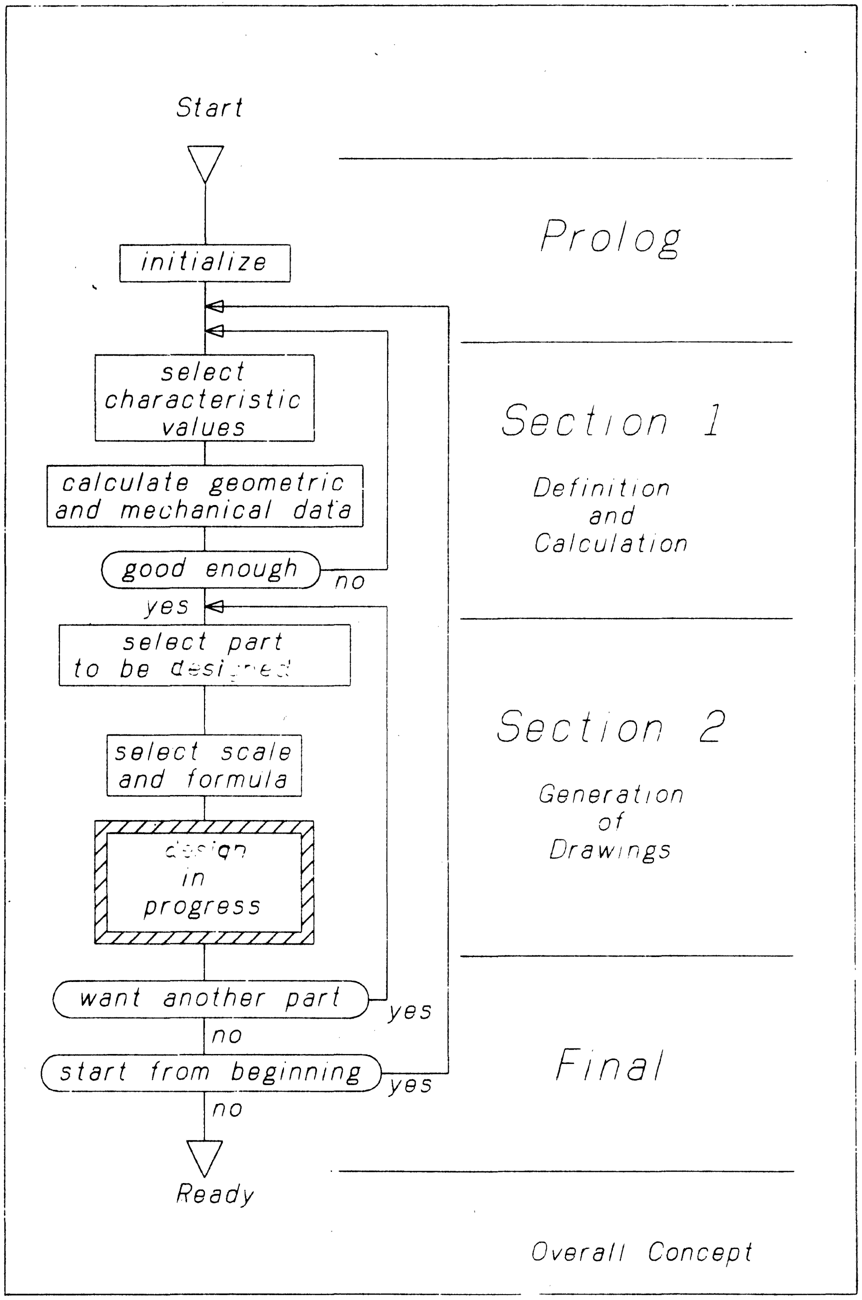


Figure 1

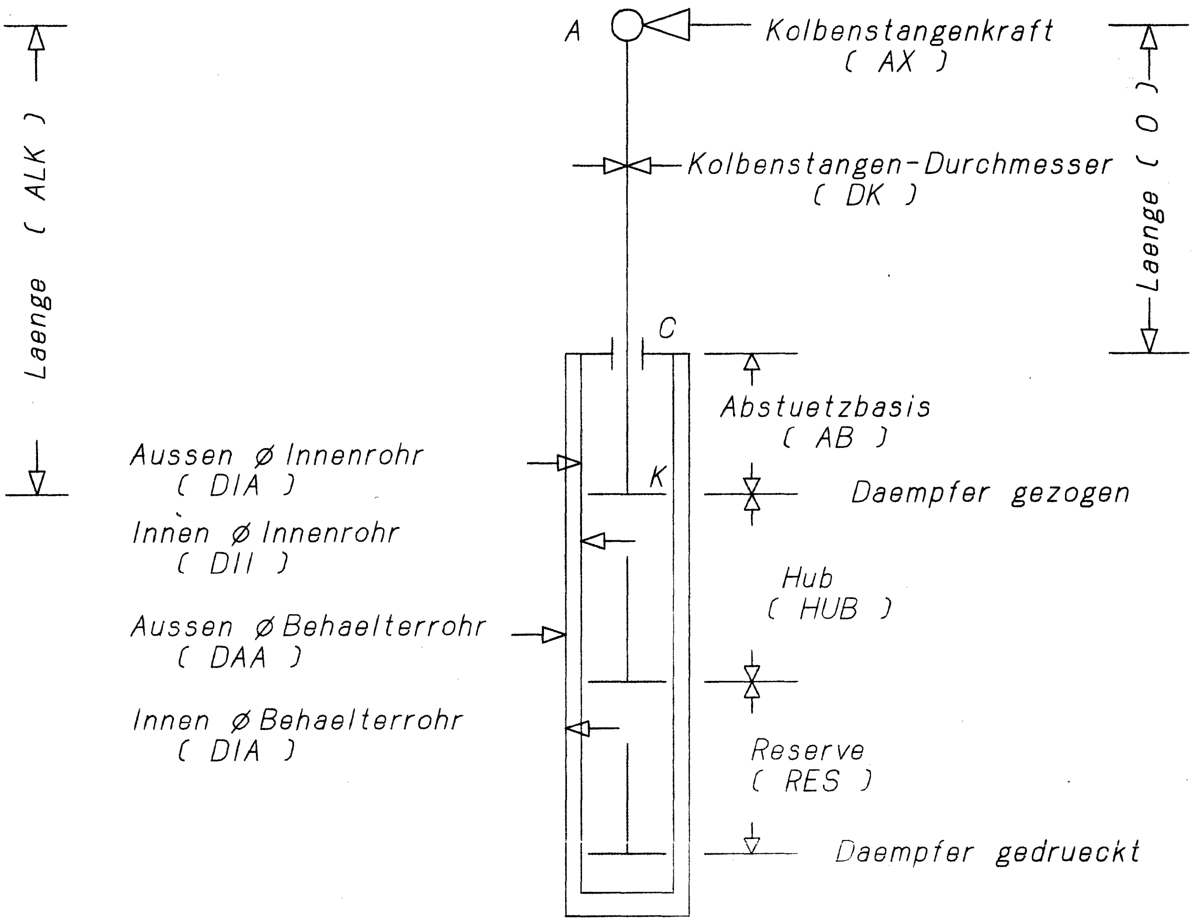
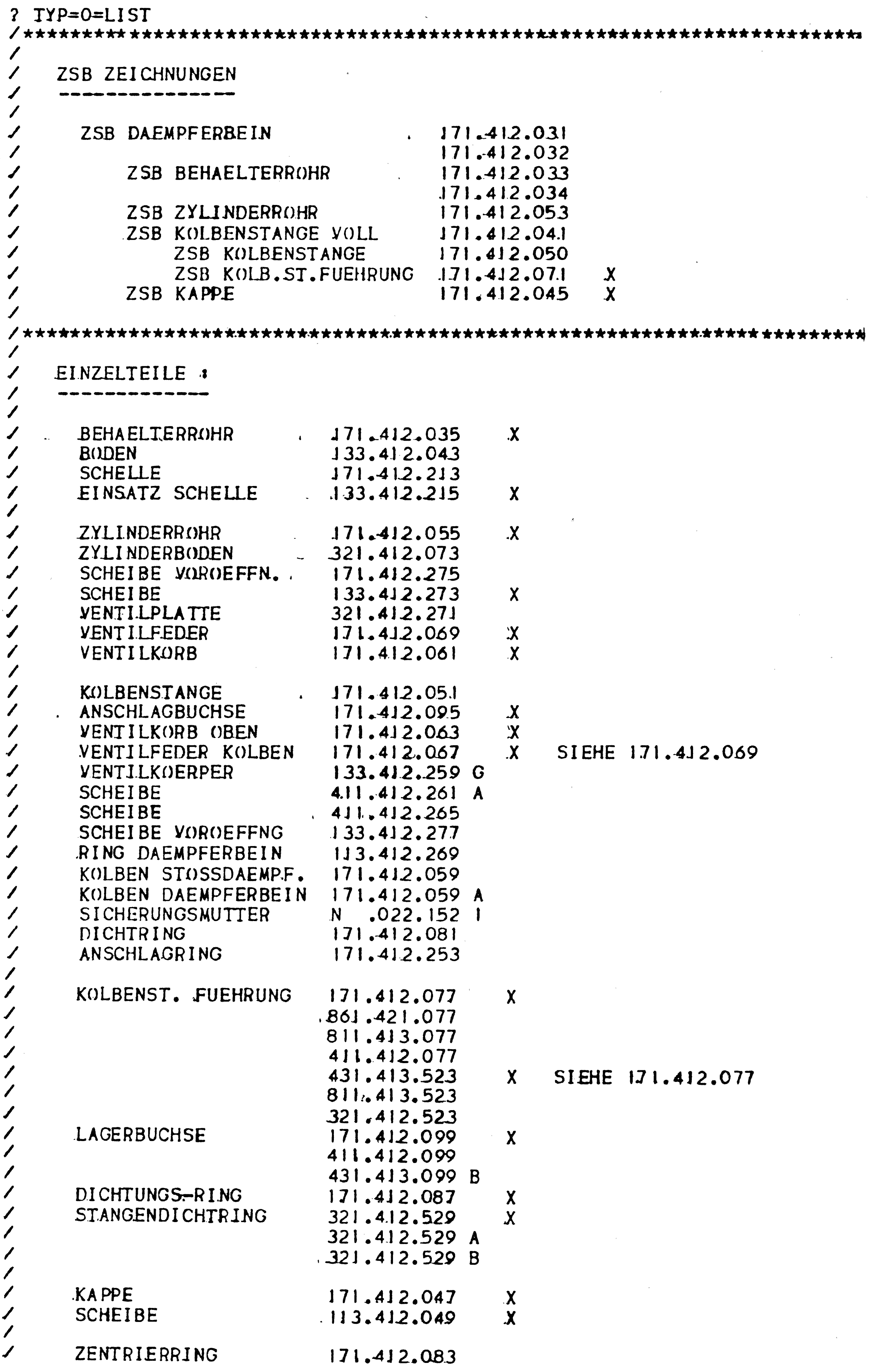
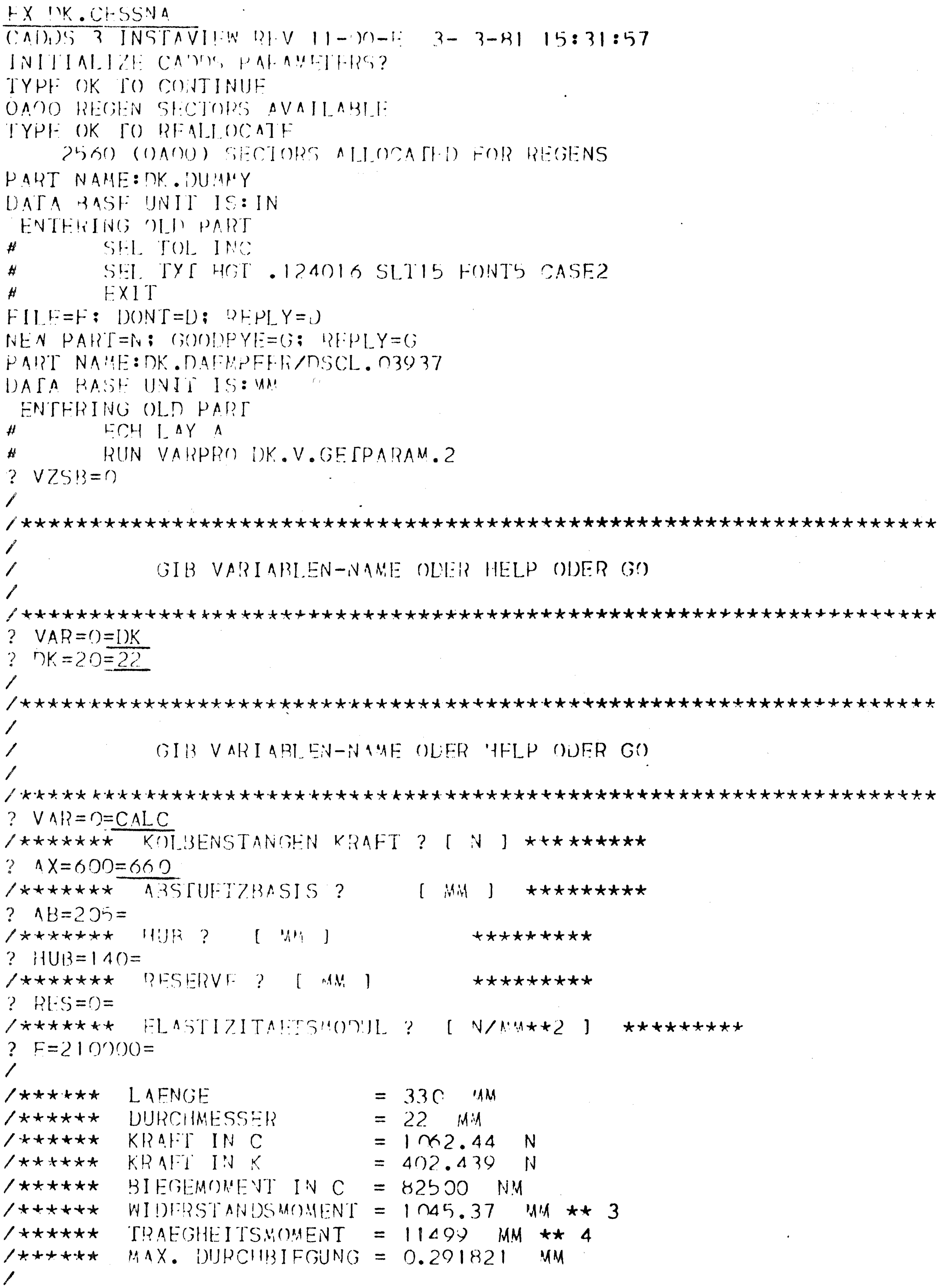

A
Kolbenstangenkraft (AX)
Kolbenstangen-Durchmesser (DK)
Laenge (O)
Laenge (ALK)
C
Abstuetzbasis (AB)
K
Daempfer gezogen
Aussen Ø Innenrohr (DIA)
Innen Ø Innenrohr (DII)
Hub (HUB)
Aussen Ø Behaelterrohr (DAA)
Innen Ø Behaelterrohr (DIA)
Reserve (RES)
Daempfer gedrueckt

Figure 2



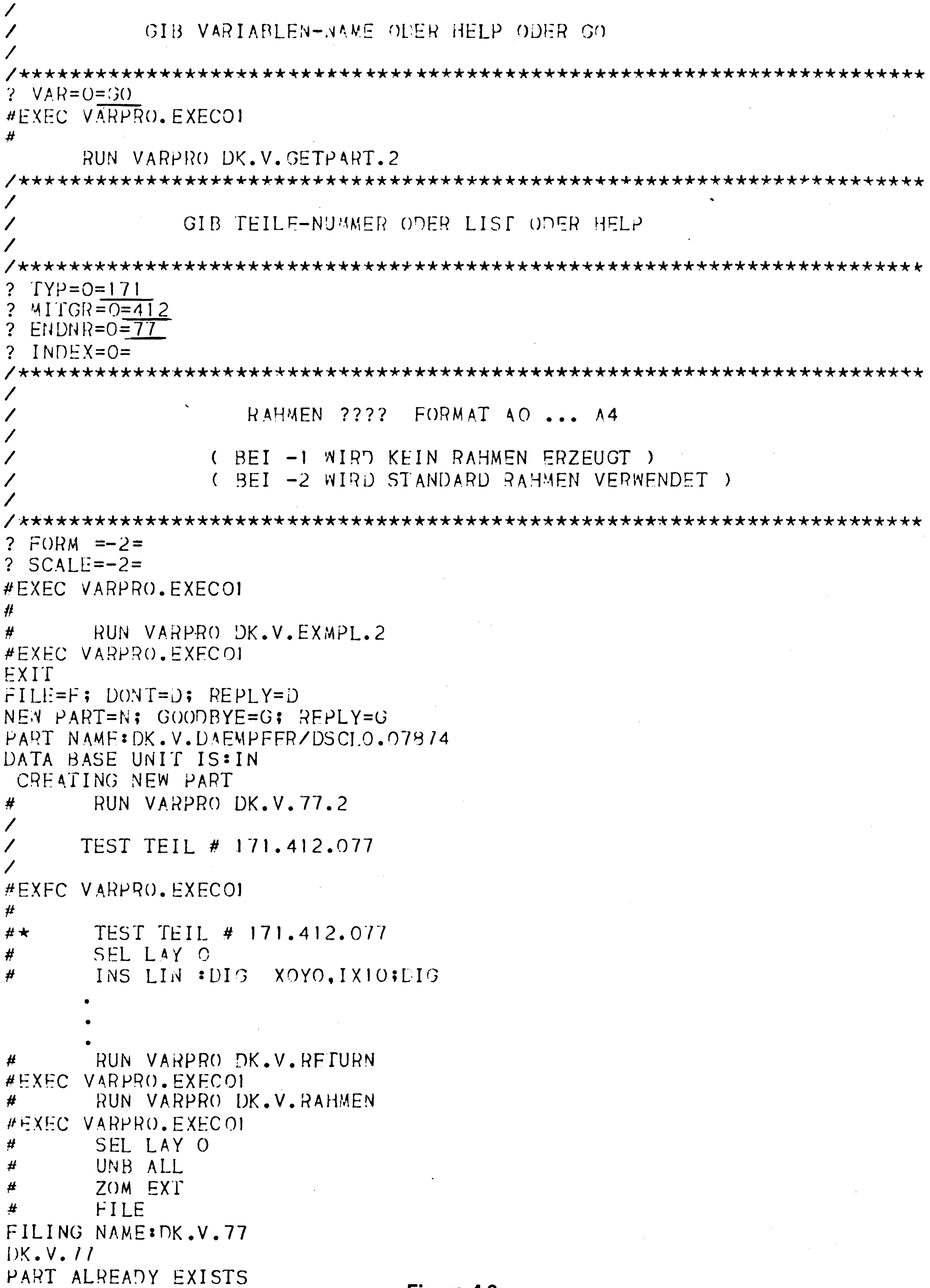
```
? TYP=O=LIST
/*************************************************************************
/
/   ZSB ZEICHNUNGEN
/   ---------------
/
/     ZSB DAEMPFERBEIN               171.412.031
/                                    171.412.032
/         ZSB BEHAELTERROHR          171.412.033
/                                    171.412.034
/         ZSB ZYLINDERROHR           171.412.053
/         ZSB KOLBENSTANGE VOLL      171.412.041
/             ZSB KOLBENSTANGE       171.412.050
/             ZSB KOLB.ST.FUEHRUNG   171.412.071   X
/         ZSB KAPPE                  171.412.045   X
/
/*************************************************************************
/
/   EINZELTEILE :
/   -------------
/
/     BEHAELTERROHR        171.412.035    X
/     BODEN                133.412.043
/     SCHELLE              171.412.213
/     EINSATZ SCHELLE      133.412.215    X
/
/     ZYLINDERROHR         171.412.055    X
/     ZYLINDERBODEN        321.412.073
/     SCHEIBE VOROEFFN.    171.412.275
/     SCHEIBE              133.412.273    X
/     VENTILPLATTE         321.412.271
/     VENTILFEDER          171.412.069    X
/     VENTILKORB           171.412.061    X
/
/     KOLBENSTANGE         171.412.051
/     ANSCHLAGBUCHSE       171.412.095    X
/     VENTILKORB OBEN      171.412.063    X
/     VENTILFEDER KOLBEN   171.412.067    X   SIEHE 171.412.069
/     VENTILKOERPER        133.412.259 G
/     SCHEIBE              411.412.261 A
/     SCHEIBE              411.412.265
/     SCHEIBE VOROEFFNG    133.412.277
/     RING DAEMPFERBEIN    113.412.269
/     KOLBEN STOSSDAEMPF.  171.412.059
/     KOLBEN DAEMPFERBEIN  171.412.059 A
/     SICHERUNGSMUTTER     N   .022.152 I
/     DICHTRING            171.412.081
/     ANSCHLAGRING         171.412.253
/
/     KOLBENST. FUEHRUNG   171.412.077    X
/                          861.421.077
/                          811.413.077
/                          411.412.077
/                          431.413.523    X   SIEHE 171.412.077
/                          811.413.523
/                          321.412.523
/     LAGERBUCHSE          171.412.099    X
/                          411.412.099
/                          431.413.099 B
/     DICHTUNGS-RING       171.412.087    X
/     STANGENDICHTRING     321.412.529    X
/                          321.412.529 A
/                          321.412.529 B
/
/     KAPPE                171.412.047    X
/     SCHEIBE              113.412.049    X
/
/     ZENTRIERRING         171.412.083
```

Figure 3



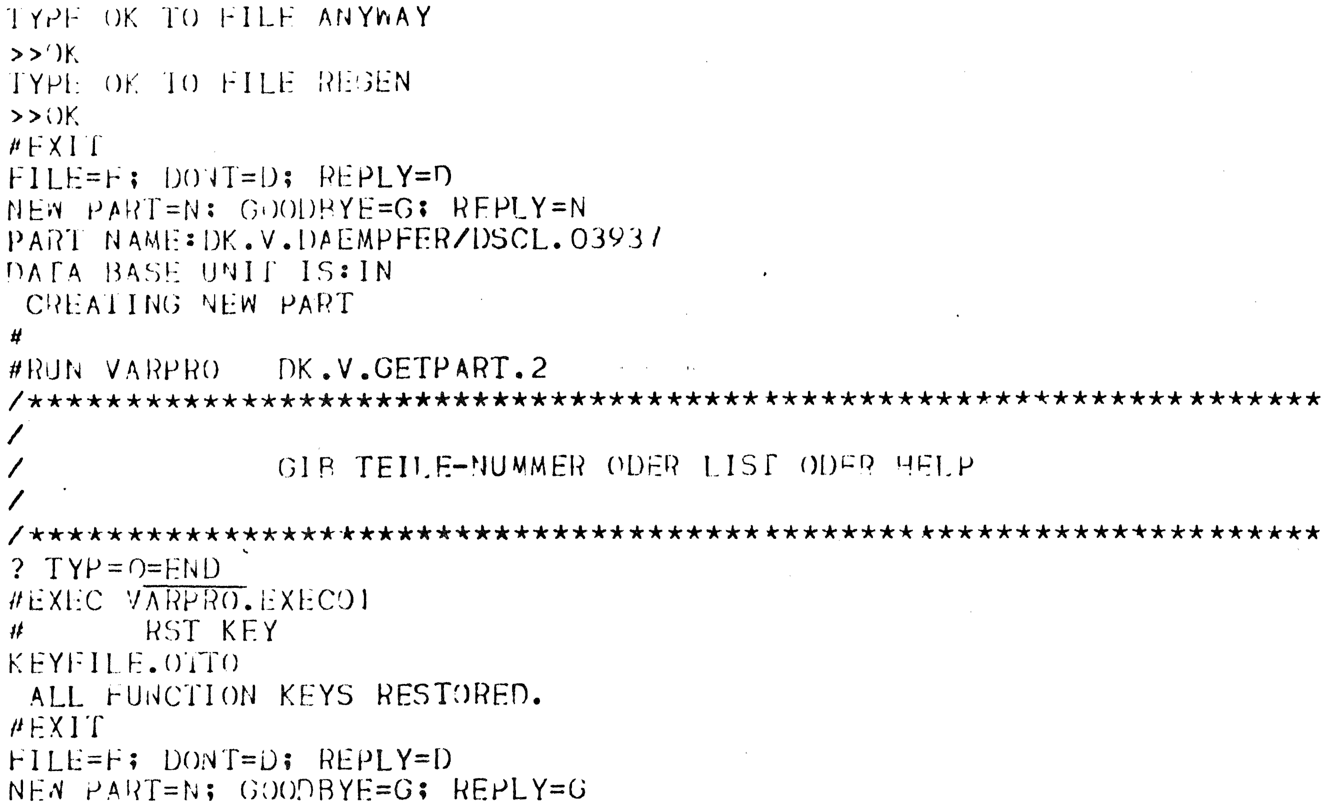
```
EX DK.CESSNA
CADDS 3 INSTAVIEW REV 11-00-E  3- 3-81 15:31:57
INITIALIZE CADDS PARAMETERS?
TYPE OK TO CONTINUE
OA00 REGEN SECTORS AVAILABLE
TYPE OK TO REALLOCATE
    2560 (OA00) SECTORS ALLOCATED FOR REGENS
PART NAME:DK.DUMMY
DATA BASE UNIT IS:IN
 ENTERING OLD PART
#       SEL TOL INC
#       SEL TYT HGT .124016 SLT15 FONT5 CASE2
#       EXIT
FILE=F; DONT=D; REPLY=D
NEW PART=N; GOODPYE=G; REPLY=G
PART NAME:DK.DAEMPFER/DSCL.03937
DATA BASE UNIT IS:MM
 ENTERING OLD PART
#       ECH LAY A
#       RUN VARPRO DK.V.GETPARAM.2
? VZSB=0
/
/*****************************************************************
/
/           GIB VARIABLEN-NAME ODER HELP ODER GO
/
/*****************************************************************
? VAR=0=DK
? DK=20=22
/
/*****************************************************************
/
/           GIB VARIABLEN-NAME ODER HELP ODER GO
/
/*****************************************************************
? VAR=0=CALC
/*******  KOLBENSTANGEN KRAFT ? [ N ] *********
? AX=600=660
/*******  ABSTUETZBASIS ?      [ MM ]  *********
? AB=205=
/*******  HUB ?   [ MM ]           *********
? HUB=140=
/*******  RESERVE ?  [ MM ]         *********
? RES=0=
/*******  ELASTIZITAETSMODUL ?  [ N/MM**2 ]  *********
? E=210000=
/
/******  LAENGE            = 330  MM
/******  DURCHMESSER       = 22  MM
/******  KRAFT IN C        = 1062.44  N
/******  KRAFT IN K        = 402.439  N
/******  BIEGEMOMENT IN C  = 82500  NM
/******  WIDERSTANDSMOMENT = 1045.37  MM ** 3
/******  TRAEGHEITSMOMENT  = 11499  MM ** 4
/******  MAX. DURCHBIEGUNG = 0.291821  MM
/
/
```

Figure 4-1



```
/
/           GIB VARIABLEN-NAME ODER HELP ODER GO
/
/**********************************************************************
? VAR=0=GO
#EXEC VARPRO.EXEC01
#
      RUN VARPRO DK.V.GETPART.2
/**********************************************************************
/
/               GIB TEILE-NUMMER ODER LIST ODER HELP
/
/**********************************************************************
? TYP=0=171
? MITGR=0=412
? ENDNR=0=77
? INDEX=0=
/**********************************************************************
/
/                     RAHMEN ????  FORMAT A0 ... A4
/
/                 ( BEI -1 WIRD KEIN RAHMEN ERZEUGT )
/                 ( BEI -2 WIRD STANDARD RAHMEN VERWENDET )
/
/**********************************************************************
? FORM =-2=
? SCALE=-2=
#EXEC VARPRO.EXEC01
#
#       RUN VARPRO DK.V.EXMPL.2
#EXEC VARPRO.EXEC01
EXIT
FILE=F; DONT=D; REPLY=D
NEW PART=N; GOODBYE=G; REPLY=G
PART NAME:DK.V.DAEMPFER/DSCL0.07874
DATA BASE UNIT IS:IN
 CREATING NEW PART
#       RUN VARPRO DK.V.77.2
/
/      TEST TEIL # 171.412.077
/
#EXEC VARPRO.EXEC01
#
#*      TEST TEIL # 171.412.077
#       SEL LAY 0
#       INS LIN :DIG  X0Y0,IX10;DIG
        .
        .
        .
#       RUN VARPRO DK.V.RETURN
#EXEC VARPRO.EXEC01
#       RUN VARPRO DK.V.RAHMEN
#EXEC VARPRO.EXEC01
#       SEL LAY 0
#       UNB ALL
#       ZOM EXT
#       FILE
FILING NAME:DK.V.77
DK.V.77
PART ALREADY EXISTS
```

Figure 4-2



```
TYPE OK TO FILE ANYWAY
>>OK
TYPE OK TO FILE REGEN
>>OK
#EXIT
FILE=F; DONT=D; REPLY=D
NEW PART=N; GOODBYE=G; REPLY=N
PART NAME:DK.V.DAEMPFER/DSCL.0393/
DATA BASE UNIT IS:IN
 CREATING NEW PART
#
#RUN VARPRO   DK.V.GETPART.2
/**********************************************************************
/
/                GIB TEILE-NUMMER ODER LIST ODER HELP
/
/**********************************************************************
? TYP=0=END
#EXEC VARPRO.EXEC01
#       RST KEY
KEYFILE.OTTO
 ALL FUNCTION KEYS RESTORED.
#EXIT
FILE=F; DONT=D; REPLY=D
NEW PART=N; GOODBYE=G; REPLY=G
```

Figure 4-3

THE CONCEPT OF THE PROGRAM

Section 1: Definition and Calculation

The overall concept of the program is shown in fig. 1. After initialization of the default parameters the user is questioned to select the desired variable data such as the diameter of the piston rod, maximum length of the shock absorber, maximum diameter of the outer tube, etc. This is established by first entering the name of the variable as shown in a schematic at the graphical screen (see fig. 2). The program outputs the default value of the desired variable and accepts any changes at that point. This can be done for as many values as necessary and as often as desired.

The next step is the stress calculation of the most critical parts. As this problem is solved by relatively simple mechanical formulas, this process is integrated in the same program — finite element analyses are not done at this stage.

To calculate these data some more information is entered into the system, i.e. acting forces on the parts, material constants, position of the moving parts (drawn position, standard position etc.). The result is output on the alfa screen (see table 1). Besides the geometric information, mechanical data is shown, like forces at different locations, bending moment, section modulus and maximum bending of the piston rod.

Table 1

```
*****   LAENGE                = 192 MM   ( length )
*****   DURCHMESSER           = 20 MM    ( diameter )
*****   KRAFT IN C            = 337.92 N ( force at location C )
*****   KRAFT IN K            = 249.92 N ( force at location K )
*****   BIEGEMOMENT IN C      = 12496 NM ( bending moment at C )
*****   WIDERSTANDSMOMENT     = 785.397 MM**3 ( section modulus )
*****   TRAEGHEITSMOMENT      = 7853.97 MM**4 (moment of inertia)
*****   MAX. DURCHBIEGUNG     = 0.179308 MM    (max. bending )
```

At that stage of the program the designer might be content with what he created. If not, he could start all over again thus optimizing his part in a very short period of time.

Section 2: Generation of Drawings

The next section of the program will be entered on command of the user and will request the designer to select one of the parts to be drawn with the characteristical data he had defined in the previous section. The part is selected by entering the part number (nine digits and an index — if present). A list of these parts is shown in fig. 3 and can be reproduced on the screen. A number of assembly drawings are defined that are created by calling each of the detail programs in sequence.

Another option for the user is to enter "ALL" which results in the program to generate a complete set of all parts at the time.

In the next step the user is requested to help the program with organizational information for the output. He is asked to select a standard design formula of a desired size. (DIN A0 through DIN A4). He also can tell the program about the scale of the drawing. If he does not enter a value for those requests default values are used that are defined in the routines. The default values will also be used if the complete set of drawings was selected. After that entry the user might send the program into background thus clearing the terminal for other work.

The result of the program is one or more CADDS-3 parts in the CV data base. These drawings contain most of the necessary data. However, the user is required to have a critical eye on the results and should — if necessary — complete the drawings with tolerances, general text, machining information etc. On the other hand the program will generate nongraphical data like the weight of the part automatically and note this value at the specified location in the drawing. Also other predefined nongraphical data like material information are entered into the drawing. After manual completion of the drawing the result can be drawn on a plotter or sent to other users.

PROGRAMMING AND USER AIDS

Similar to the online documentation facilities of CV this program offers the possibility of a HELP command at both major sections. This produces a list of possible commands at the alfa screen.

For manual usage after the design process a layer description is included in each drawing on layer 255, so that anyone who can handle CV systems also can handle those drawings very easily.

A selection of examples is shown in the appendix.

A protocol of a dialog session is shown in fig. 4. The underlined information has to be typed in by the user. The result of this example is included in the appendix with the part number 171.412.077.

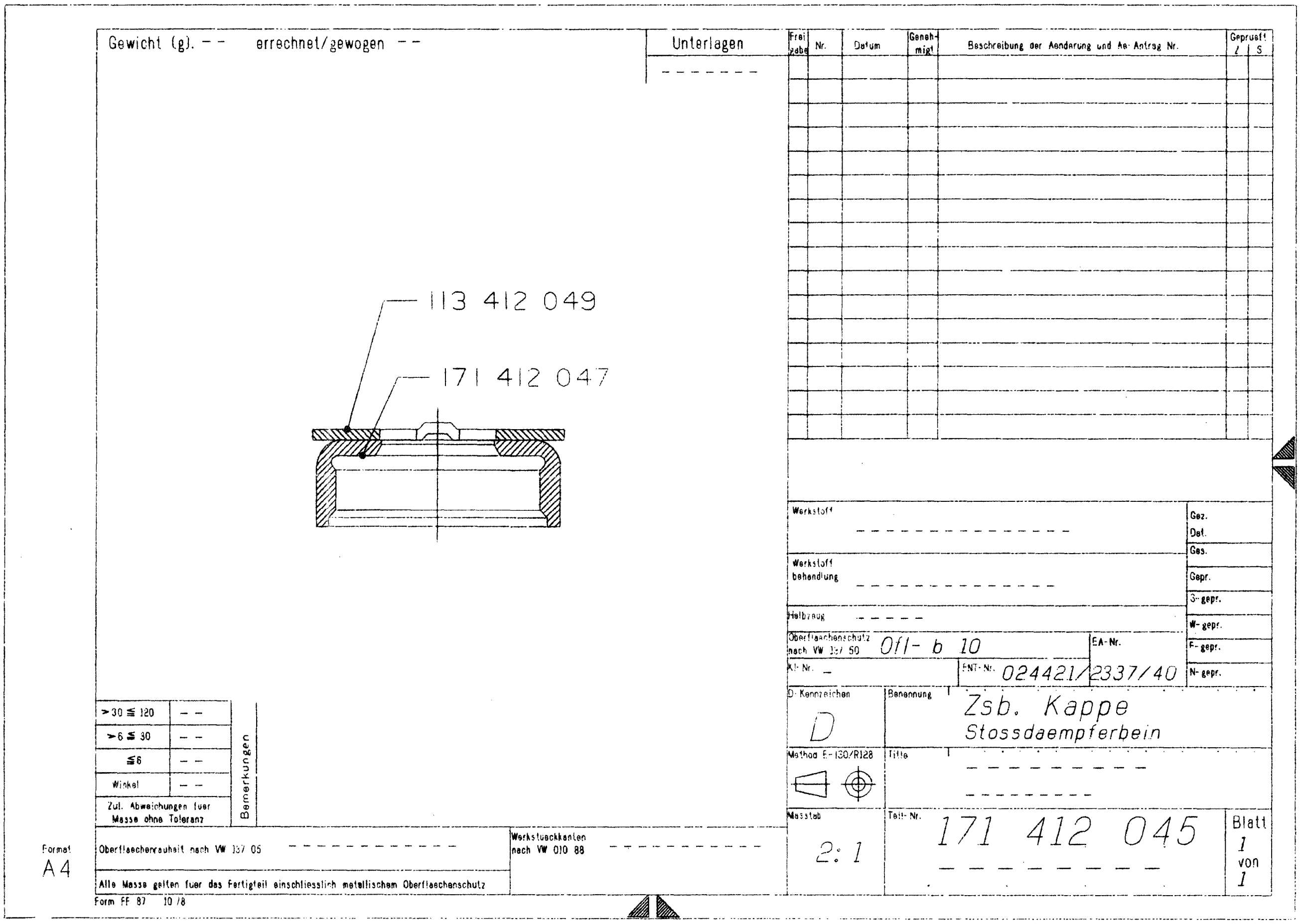
Fig. 5 reflects a programming skeleton, which acts as a very effective programming aid. Because most of the parts of the shock absorber — as well as many other parts in civil engineering — are defined by rotational shapes, the general layout of the program describing these parts will be the same. Thus the skeleton requests the programmer to fill in the part specific data only. Parameter handling and layering will be organized identically, independent of the programmer and the time the program was created.

FURTHER DEVELOPMENT

Further development will be done on the project to connect this task to existing programs on other computers. On our Cyber mainframe in the computing centre several programs have been developed for kinematics calculation for the front axle, center of mass calculation for the complete car and others. The use of the results of these programs as input for the described program can be realized by replacing the user requests in step 2 with a data exchange with the mainframe via a HASP data link.

Expanding this concept to other components of the car as discussed in the beginning will also be a way of proving the effectiveness of this technique.

Appendix 1



Gewicht (g). -- errechnet/gewogen --

Unterlagen

Freigabe	Nr.	Datum	Genehmigt	Beschreibung der Aenderung und Ae-Antrag Nr.	Geprueft	
					l	S

>30 ≦ 120	--
>6 ≦ 30	--
≦6	--
Winkel	--
Zul. Abweichungen fuer Masse ohne Toleranz	

Bemerkungen

Format A4

Oberflaechenrauheit nach VW 137 05 ------------

Werkstueckkanten nach VW 010 88 ------------

Alle Masse gelten fuer das Fertigteil einschliesslich metallischem Oberflaechenschutz

Form FF 87 10 78

Werkstoff ---------------

Werkstoff behandlung ---------------

Halbzeug -----

Oberflaechenschutz nach VW 137 50 Ofl- b 10

EA-Nr.

Kl-Nr. –

ENT-Nr. 024421/2337/40

Gez.
Dat.
Ges.
Gepr.
S-gepr.
W-gepr.
F-gepr.
N-gepr.

D-Kennzeichen D

Benennung Zsb. Kappe Stossdaempferbein

Method E-130/R128

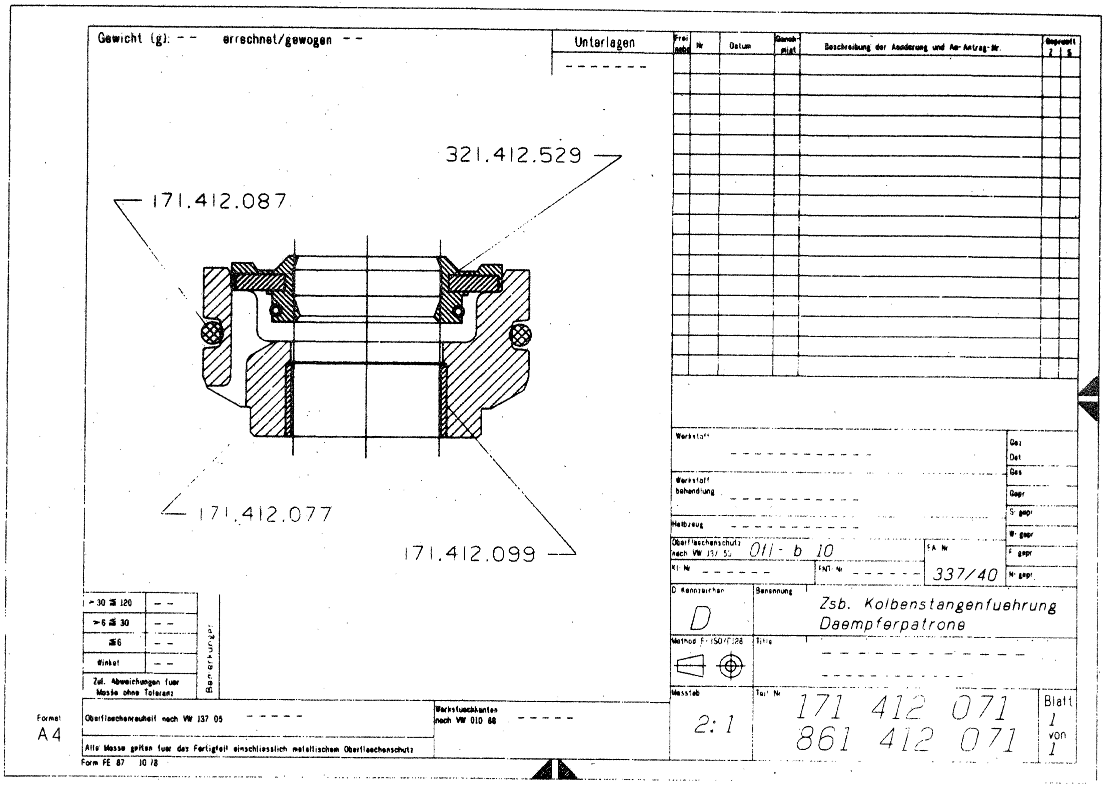
Title ---------

Masstab 2: 1

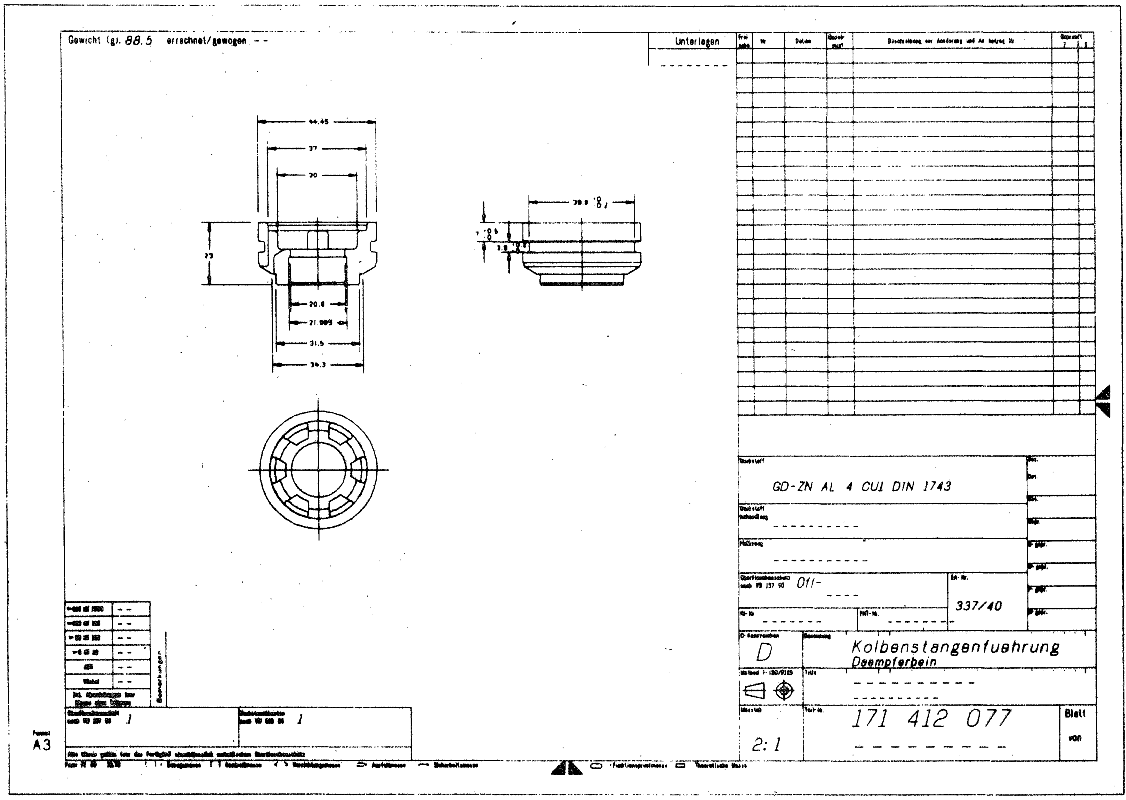
Teil-Nr. 171 412 045

Blatt 1 von 1

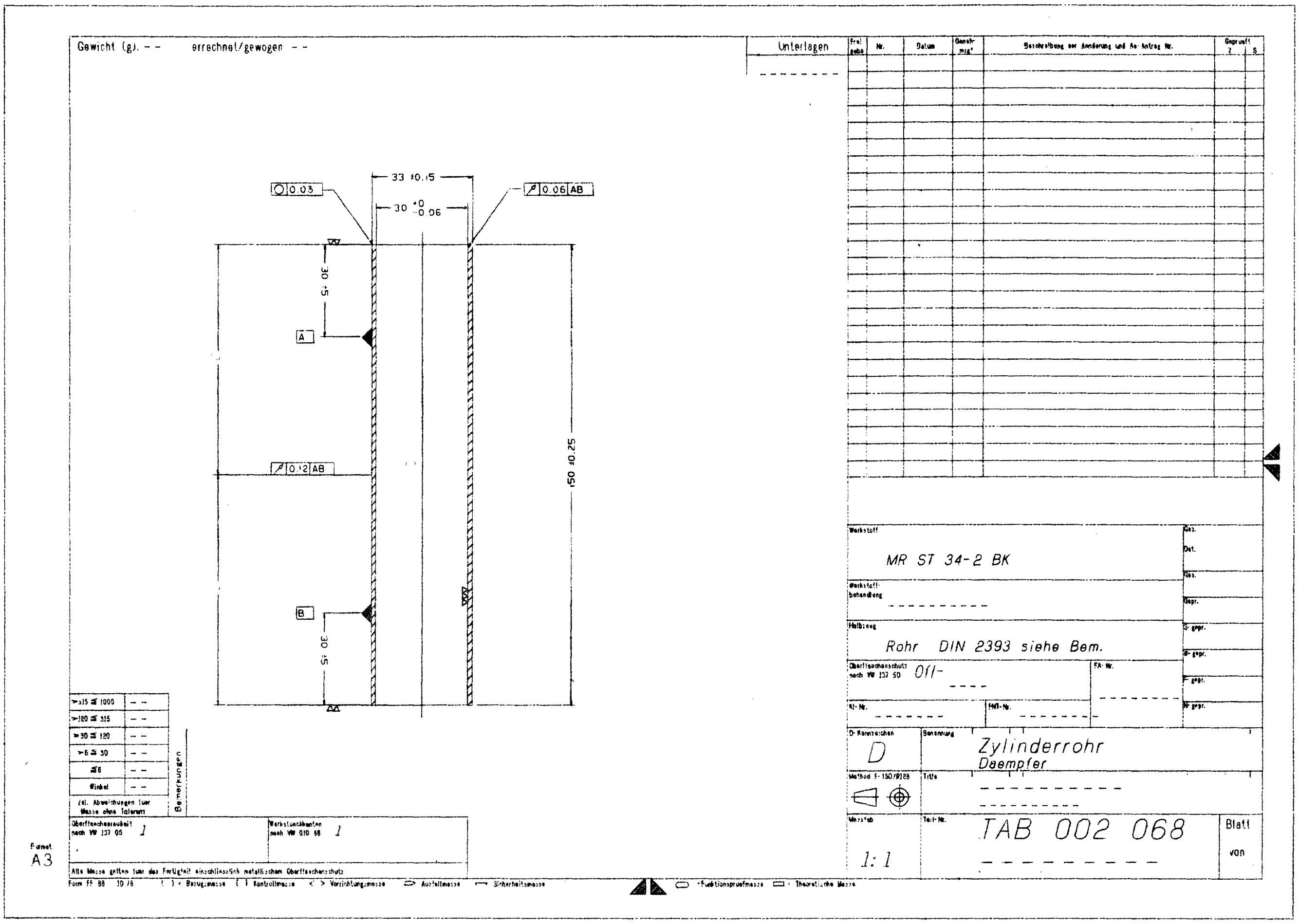
Appendix 2



Appendix 3



Appendix 4



Gewicht (g). - - errechnet/gewogen - -

Unterlagen

- - - - - - - -

Frei gabe	Nr.	Datum	Genehmigt	Beschreibung der Aenderung und Ae-Antrag Nr.	Geprueft Z	S

Werkstoff: MR ST 34-2 BK

Werkstoffbehandlung: - - - - - - - - - -

Halbzeug: Rohr DIN 2393 siehe Bem.

Oberflaechenschutz nach VW 137 50: Ofl- - - - -

FA-Nr. - - - - - - - -

Kl-Nr. - - - - - - - HMT-Nr. - - - - - - - -

Gez. Dat. Ges. Gepr. G-gepr. Q-gepr. F-gepr. N-gepr.

D-Kennzeichen: D

Benennung: Zylinderrohr Daempfer

Method E-ISO/R128

Titel: -

Massstab: 1:1

Teil-Nr.: TAB 002 068 - - - - - - - - -

Blatt von

>315 ≦ 1000	- -
>120 ≦ 315	- -
>30 ≦ 120	- -
>6 ≦ 30	- -
≦6	- -
Winkel	- -
Zul. Abweichungen fuer Masse ohne Toleranz	

Bemerkungen

Oberflaechenrauheit nach VW 137 05: 1

Werkstueckkanten nach VW 010 88: 1

Format A3

Alle Masse gelten fuer das Fertigteil einschliesslich metallischem Oberflaechenschutz

Form FF 88 10 78 ( ) = Bezugsmasse [ ] Kontrollmasse < > Vorrichtungsmasse ⊂⊃ Ausfallmasse ─ Sicherheitsmasse ▭ = Funktionspruefmasse ▭ = Theoretische Masse

IMPLEMENTATION OF FORGING LOAD AND STRESS ANALYSIS ON A COMPUTERVISION CADDS 3 SYSTEM

Ally Badawy,
Carl Billhardt,
Taylan Altan
Battellis Columbus Laboratories
505 King Avenue
Columbus, OH 43201
614-424-7864, 7860, 7859

ABSTRACT

Recently, approximate methods of analysis have been developed for predicting stresses and loads in impression die forging. These methods analyze a forging by considering it in terms of plane strain and axisymmetric cross sections. It is desirable to augment the drafting and NC capabilities of a stand-alone CAD/CAM system with the mathematical design analysis techniques used in forging die design. With this goal in mind, an existing method for calculating forging stresses and load has been implemented on a Computervision CADDS 3 system. This paper describes the procedure used in this implementation and gives examples of the practical application of this new capability.

BACKGROUND

Forging is the process of bulk deforming a material (usually a metal) in order that a product of some desired geometry is generated. The forming takes place by placing the material between dies and then forcing the dies together. The dies can either be flat pieces (open dies) or sculptured in three dimensions (closed or impression dies). When impression dies are used, generally the dies do not close completely tight but rather leave a certain clearance all the way around between the dies. This clearance is called flash and is provided to give excess material a place to escape. The material which escapes to the flash is scrap and is trimmed from the part after forging. Generally open die forging is used when:

- Parts are very large (i.e. giant crankshafts and rotors).
- Part geometry is simple.
- Quantity of parts to be made is low.

Impression die forging is used under the opposite conditions:

- Parts are small to moderately large
- Part geometry is complex
- Quantity of parts is large.

A special case of impression die forging is precision, trapped die or flashless forging. This process produces a highly defined part but requires special care in preparing the forging material and machining the dies.

Based on tonnage and quantity of forgings, the impression die (with flash) process is the most used both in the U.S. and overseas.

Among the factors considered in designing a set of dies is the maximum stress the dies will be subjected to, and the total force required to forge the part. The maximum stress must be considered to avoid overloading the die and cracking it. The level of stress will also affect the rate at which the die wears and therefore its expected life. The total force required to form the part determines the size of the equipment which must be used. If too small a press is used the part may not fully fill the die, or the press may be overloaded to the point of catastrophic failure.

Until recently, estimating the load and stress on a die set was based on the judgment of an experienced designer. His estimate was based on comparing a propsective forging to similar ones he had worked on previously. A good designer was one who had more good estimates than bad estimates. However, he could always err on the conservative side and use dies or a press that were much larger than necessary. Both would avoid failure but at an unknown cost for oversizing.

Using a designer's experience to estimate forging loads and stresses is becoming less reliable for two reasons. First, experienced designers are retiring with no trained people to replace them. Their experience is being lost. Secondly, new materials and forging processes are being developed for which there is no prior experience for guidance. This is especially true in forgings for the aerospace industries. For certain applications, materials are being developed for their high performance characteristics in critical applications (engines and air frame components). Once proved by the material scientist and the design engineer, it becomes the forging engineer's problem to determine how to work the material into the desired form. The use of difficult to form materials had led to the development of new forging techniques, such as the isothermal and hot die forging processes where die materials are very expensive and die design for complex shapes becomes very critical.

STRESS AND LOAD ANALYSIS

The need for a practical method for predicting forging load and stresses led to the application of the "slab method" technique in a numerical and computerized form. (1,2)* The slab method can be used to estimate loads and stresses for plane strain (forging an infinitely long bar with material flow in two dimensions) or axisymmetric (part of rotation) flow. For this purpose the planes of metal flow are considered in a given forging, Figure 1 (1). Although techniques for analyzing three dimensional metal flow are being developed, they are not yet available for general use. They also require considerable computing time and power, and may only be suitable for execution on large, scientific processors.

To function the slab analysis needs the following information:

- The geometry of the upper and lower die surface at the cross section being considered. This geometry includes the flash width and thickness. From this geometry, a model of how the material will flow can be developed. Depending on the geometry, material will either flow by sliding against the die (sliding flow) or by internally shearing against itself (shear flow).
- The flow stress of the material (its inherent resistance to deformation). This is not a single unique value but depends on the temperature and rate of deformation (strain rate). Generally, there is a direct relationship between strain rate and flow stress. Increasing the deformation rate increases the flow stress.

  The deformation rate also has an indirect effect on flow stress. When forging at low speeds, the material is in contact with the dies for longer time than when forging at high speed, thus lowering the material temperature and increasing the flow stress. In addition to heat loss from the die affecting flow stress, there is also heat gain as a result of deformation and friction. The net effect of all of these variables is impossible to predict by judgment.

*Numbers in parenthesis refer to References given at the end of the paper.

The slab method functions first by determining an approximate "flow surface" in a given cross section. The flow surface represents the approximate mode of metal flow in the section. The flow surface is divided into a series of trapezoidal elements (2). Knowing the geometry of each element, and the beginning flow stress and the friction factor or coefficient (sliding friction or internal shear), the load and pressure acting on the element can be found, as seen in Figure 2 (3). In addition, the flow stress at the end of the element can also be determined. By working from both outside edges toward the center simultaneously, the total load, maximum stress, and average pressure can be determined for a section in plane strain. For an axisymmetric section, these values are found by treating the trapezoidal elements in order, from the flash to the centerline.

IMPLEMENTATION

The slab method for forging load and stress analysis has previously been implemented on different computers, including CDC-6500 and DEC-PDP 11/40 processors (4,5). The limitation of these implementations was that generating the definition of the section to be considered was a cumbersome process. The section data was prepared off-line as a series of coordinates defining the polygon formed by the tangents at each corner or fillet. Each corner or fillet coordinate had a radius value associated with it. The designer would generate the data for sections of interest, and then process the data in an interactive or batch mode. Although he could interactively modify the flash dimensions, he would have to take his results back to his drafting table to complete his drawings.

Recently, Battelle re-hosted this forging analysis on a Computervision CADDS 3 system using CGOS 200. This was implemented as the user-written function FORGE. The graphic information indicated to FORGE by the user are two "strings" representing the upper and lower die surfaces, Figure 2. Modifiers pertaining to material and process conditions may be entered by the user following CV's conventional practices. The two input strings are generated using the GEN STG command. The strings may be generated left to right or vice versa, and either the top of bottom string may be indicated first. The only limitations placed on the strings is that they must be monotonic in X and when indicated, they must lie in a plane parallel to the current view.

FORGE is written entirely in Fortran, and currently uses three coreloads. The major functions of these coreloads is:

- Coreload 1. Modifier and Get Data processors, followed by formulation of material flow model.
- Coreload 2. Calculation of material flow stress in the flash and cavity regions based on default or modifier values for such parameters as material code, die and stock temperatures, and ram speed.
- Coreload 3. Calculation of load, maximum stress, and center of load.

Sections may be generated either manually in 2-D or by using a cutting plane to section a 3-D model. The results of FORGE are used in two ways. As each section is processed, the resulting stress values are used to determine if the flash dimensions need to be adjusted. A high stress may exceed the load carrying capacity of the die material. In this case, the flash would be opened to reduce its tendency to restrict the flow. If the stress if low, the material may not actually fill the extreme sections of the die cavity. In either case, the user would return to CADDS, modify the geometry as appropriate, and run FORGE again.

The calculated results for stress, load, and center of load are output to the telewriter as printed numeric values. In addition, the calculated flow surfaces and the stress distrubition are plotted on the graphics display.

The second way in which the output of FORGE is used is to obtain the load. This is done by keeping a manual tally for each section as it is processed. The sum of the individual loads can then be used to determine the press size needed to forge the part.

A forging such as a connecting rod, Figure 3, can be correctly processed by FORGE even though it has some sections that are in plane strain flow and some that are in axisymmetric flow. The strain or flow condition to be used for a particular section is indicated as a modifier.

GENERATION OF THE FLOW SURFACE

Towards the end of the forging stroke, material does not flow along the die surface at all points as the dies come together, Figure 2. In areas such as under tall ribs material in the rib stays stationary once the rib is filled. Below the rib, material continues to flow but by shearing within itself rather than by sliding across the die surface. The slab method uses the geometry of the flow (sliding friction or shear) surfaces rather than the actual die surfaces to predict stress and load.

A concept for formulating the flow surface based on the geometry of the die surfaces was developed and implemented in FORGE. This is a new capability, not available elsewhere in such a general form. The flow surface formulation assumes that shear occurs whenever the angle of the die surface exceeds 45 degrees. The friction factor or co-efficient for the trapezoidal deformation elements are set as appropriate, based on this geometric consideration. Beneath a rib, flow often goes from sliding friction to internal shear and then back to friction. The presence of ribs is automatically determined and the flow surface is modified accordingly.

The procedure used to formulate the flow surface is known to fail with certain very complex die surfaces. However, the slab method of analysis is also not entirely valid for these geometries, and the user will be aware of a problem when he examines the display generated for the flow surface.

VALUES OF FLOW STRESS AND FRICTION

To estimate the stresses and the load in forging, it is necessary to know, in addition to the part geometry and the type of metal flow, (a) the flow stress of the deforming material under the forging conditions and (b) the friction at the die material interface.

Estimation of Friction

The friction shear stress, τ, between two surfaces, sliding relative to each other, is given by $\tau = m\sigma/\sqrt{3} = f\sigma$, where σ is the flow stress of the deforming material and m is the friction shear factor ($0 < m < 1$). When the material shears within itself, then friction is maximum, i.e. $m = 1$, and τ is equal to the shear stress of the material. When material flows by sliding friction over the die surface, then m is determined from a friction test. The test most commonly used for forging is the so called ring test (6). This test gives values of $0.05 < m < 0.15$ for cold forging and $0.2 < m < 0.4$ for hot forging.

Estimation of Flow Stress

For a given material, the flow stress (or the instantaneous yield stress), σ, is influenced by (a) the amount of deformation or strain, (b) the rate of deformation or strain-rate, and (c) the deformation temperature. The amount of deformation is obtained from the initial and final heights of the forging. The deformation rate is determined by the speed of the ram and by the height of the forging. The temperature during forging is influenced by (a) the initial stock and die temperatures, (b) the heat transfer from the hot material to the colder dies and (c) the heat generation due to friction and deformation energies dissipated during forging.

All three variables — deformation, deformation rate, and temperature — vary throughout the forging depending upon the geometry. Thus, for practical purposes, it is a reasonable assumption to consider these variables and consequently the flow stress, σ, to have different average values (a) in the cavity or impression area and (b) in the flash region of the cross section.

The present forging analysis software can either accept estimated values of the flow stress in the flash and cavity regions or it can estimate these values from the known process variables, i.e. die and stock temperatures, die closure speed at the time of die-material contact, heat transfer coefficient, forging material and friction factor.

EXAMPLES

To evaluate the correctness and accuracy of the stress and load calculations, three examples were considered. Predictions were made with the design analysis implemented on Computervision and compared with experimental data or predictions made with other well proven techniques.

Estimation of Forging Load for a Connecting Rod

The blocker (preform) geometry of the connecting rod, Figure 3, was analyzed as an example, since detailed information and experimental load data were available from an earlier Battelle study (3). Figure 3 illustrates the geometry and the cross sections of this connecting rod as well as the direction of metal flow. For the purposes of the example calculation, metal flow in sections A-A and C-C is considered axisymmetric with a containing angle of 360 degrees. In section B-B, metal flow is plane strain. The dimensions of the analyzed cross sections are given in Figure 4. The input values for the flow stress and friction factor of each cross section are given in Table 1. The output values for forging load and maximum stress, for each cross section, are given in Table 2.

For comparison, the results of the simple analysis given in Reference (3), the experimental results given in the the same reference, and the FORGE CAD system are summarized in Table 3. As seen in Table 3, the results of the FORGE system are within practical engineering accuracy. The stress distribution and flow model for one of the cross sections (A-A) as displayed on the Computervision's Designer II graphics terminal, are shown in Figure 5.

Cross Section of an Isothermally Forged Titanium Part

In an earlier study conducted at Battelle, a design analysis was made of the dies for isothermal forging of a titanium alloy connecting link, shown in Figure 6. For cross section J-J of Figure 6, the stresses and unit loads were calculated using the same values of flow stress (1000 psi) and friction (0.25) used in the earlier analysis. The stress distribution on the cross section (plane strain) is shown in Figure 7. The forging load estimated using CV-FORGE for the same cross section is 45 tons. The forging load estimated earlier in Reference (7) was 48.8 tons. The small difference between the two predictions is essentially due to the slightly different flow models used in both cases.

TABLE 1:
INPUT VALUES FOR FLOW STRESS AND FRICTION FACTOR USED IN DIFFERENT CROSS SECTIONS OF CONNECTING ROD[3]

	Section A-A	Section B-B	Section C-C
Flow Stress in Flash	20,800 Psi	20,800 Psi	20,800 Psi
Flow Stress in Cavity	18,000 Psi	17,000 Psi	16,000 Psi
Friction Shear Factor	0.3	0.3	0.3

TABLE 2:
ESTIMATED LOAD AND MAXIMUM STRESS FOR THREE CROSS SECTIONS

	Section A-A	Section B-B	Section C-C
Forging Load	164 tons	109.0 tons	57 tons
Maximum Stress	136,790 Psi	107,513 Psi	104,566 Psi

TABLE 3:
SUMMARY AND COMPARISON OF FORGING LOADS

	Simple Analysis from Reference(7)	FORGE (CV Software)	Experimental
Total Forging Load (tons)	311.7	330	321.67

The Main Bearing Cross Section of a Crank Shaft

In a previous Battelle study, the cross section of the main journal of a crankshaft forging seen in Figure 8, was analyzed. The same cross section, together with the same flow stress values in flash (20,000 psi) and cavity (12,000 psi) and friction shear factor (0.43), used is considered here. The stress distribution on the given cross section is shown in Figure 8. The estimated forging load using CV-FORGE for the cross section is 257 tons. The forging load, estimated earlier with another Forging-CAD system, was 259 tons.

LIMITATIONS AND AVAILABILITY

The present version of CV-FORGE assumes the input sections are dimensioned in inches. The absolute size of the die sections are only used to determine the density of the trapezoids used for the slab analysis. Modifier inputs and calculated results are also given in English units. It would be possible, without great difficulty, to allow metric dimensions to be used.

The FORGE software is available from Computervision. At the time of this writing (May 1981), the exact method of obtaining the software and documentation was unknown. Inquiries would best be directed to Computervision, Bedford, Massachusetts.

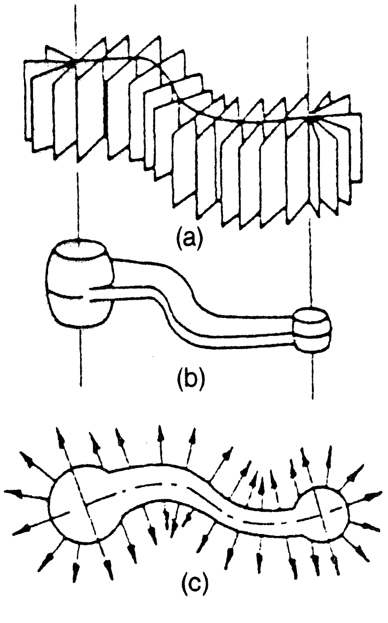


Figure 1: Planes and Directions of Metal Flow During Forging of a Relatively Complex Shape.
(a) Planes of flow
(b) Finish forged shape
(c) Directions of flow

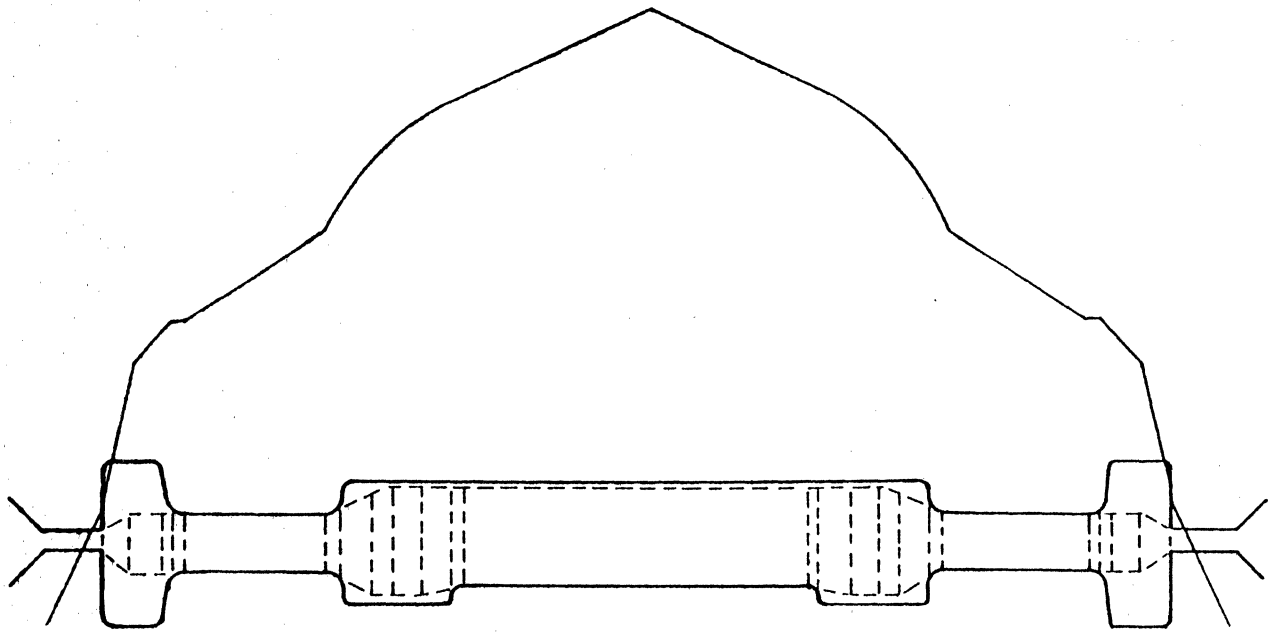
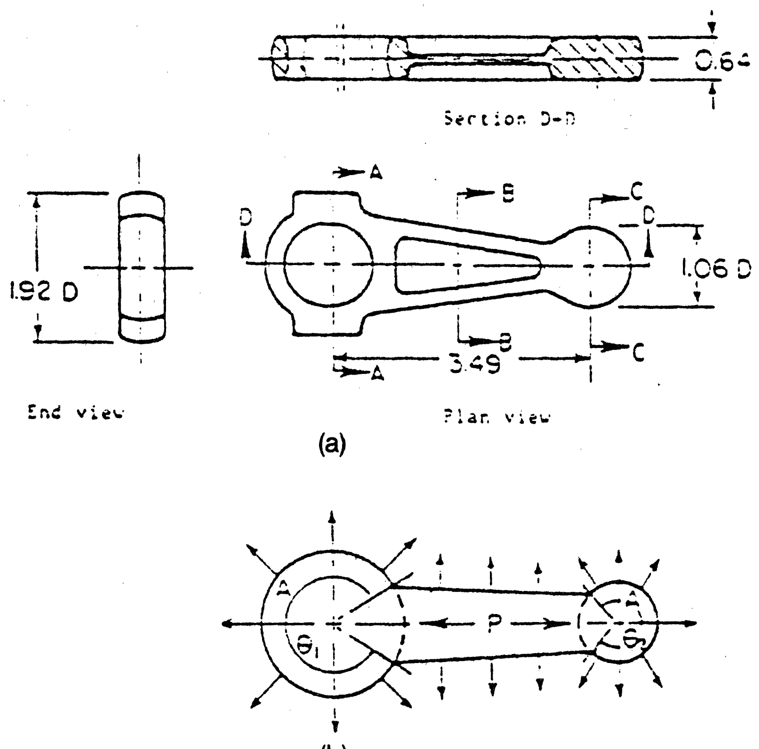


Figure 2: Example Display of the Stress Distribution and the Flow Model Obtained by Computer Aided Analysis of a Cross Section.
Flash Thickness = 0.100, Flash Width = 0.300, Finish Area = 2.85 in², Forging Pressure = 163,167.8 PSI, Load = 1,006,770 lb acting at $_x$ = 0.020, Maximum Stress = 235,503.566



(a)

(b)

Figure 3: Geometry, Directions of Metal Flow and Representative Cross Sections of a Connecting Rod.
(a) Cross Sectional Views of the Connecting Rod
(b) Directions of Metal Flow — A = Axisymmetric P = Plane Strain

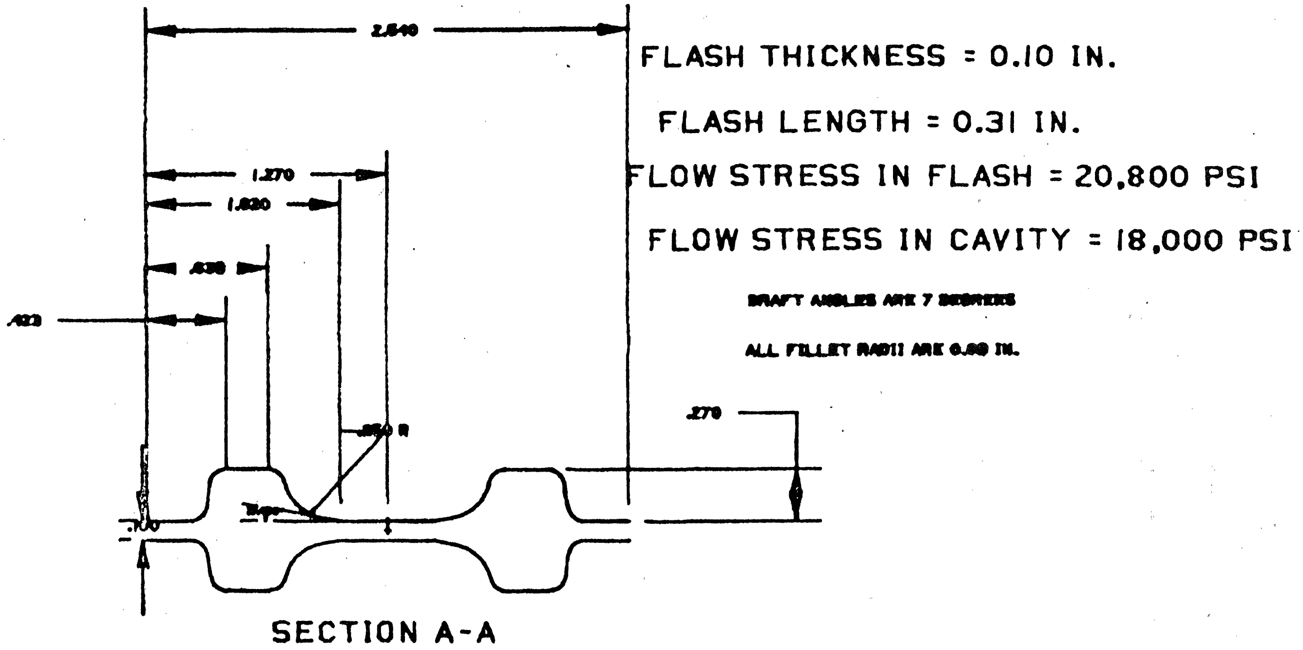

2.540
FLASH THICKNESS = 0.10 IN.
FLASH LENGTH = 0.31 IN.
FLOW STRESS IN FLASH = 20,800 PSI
FLOW STRESS IN CAVITY = 18,000 PSI
1.270
1.020
.630
DRAFT ANGLES ARE 7 DEGREES
.423
ALL FILLET RADII ARE 0.00 IN.
.270
.250 R
.100
SECTION A-A

Figure 4: Dimensions of the Blocker Cross Sections of the Connecting Rod, seen in Figure 3.
(a) Section A-A

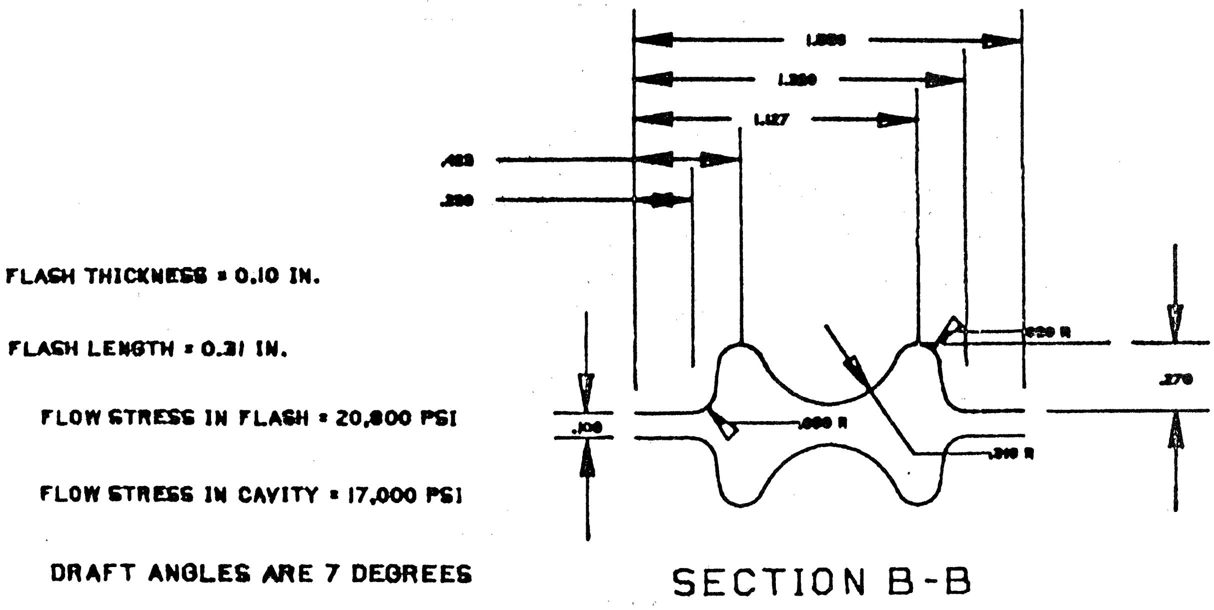

1.880
1.380
1.127
.483
.330
FLASH THICKNESS = 0.10 IN.
FLASH LENGTH = 0.31 IN.
.020 R
.270
FLOW STRESS IN FLASH = 20,800 PSI
.100
.080 R
.310 R
FLOW STRESS IN CAVITY = 17,000 PSI
DRAFT ANGLES ARE 7 DEGREES
SECTION B-B

Figure 4: (continued) Dimensions of the Blocker Cross Sections of the Connecting Rod, seen in Figure 3.
(b) Section B-B

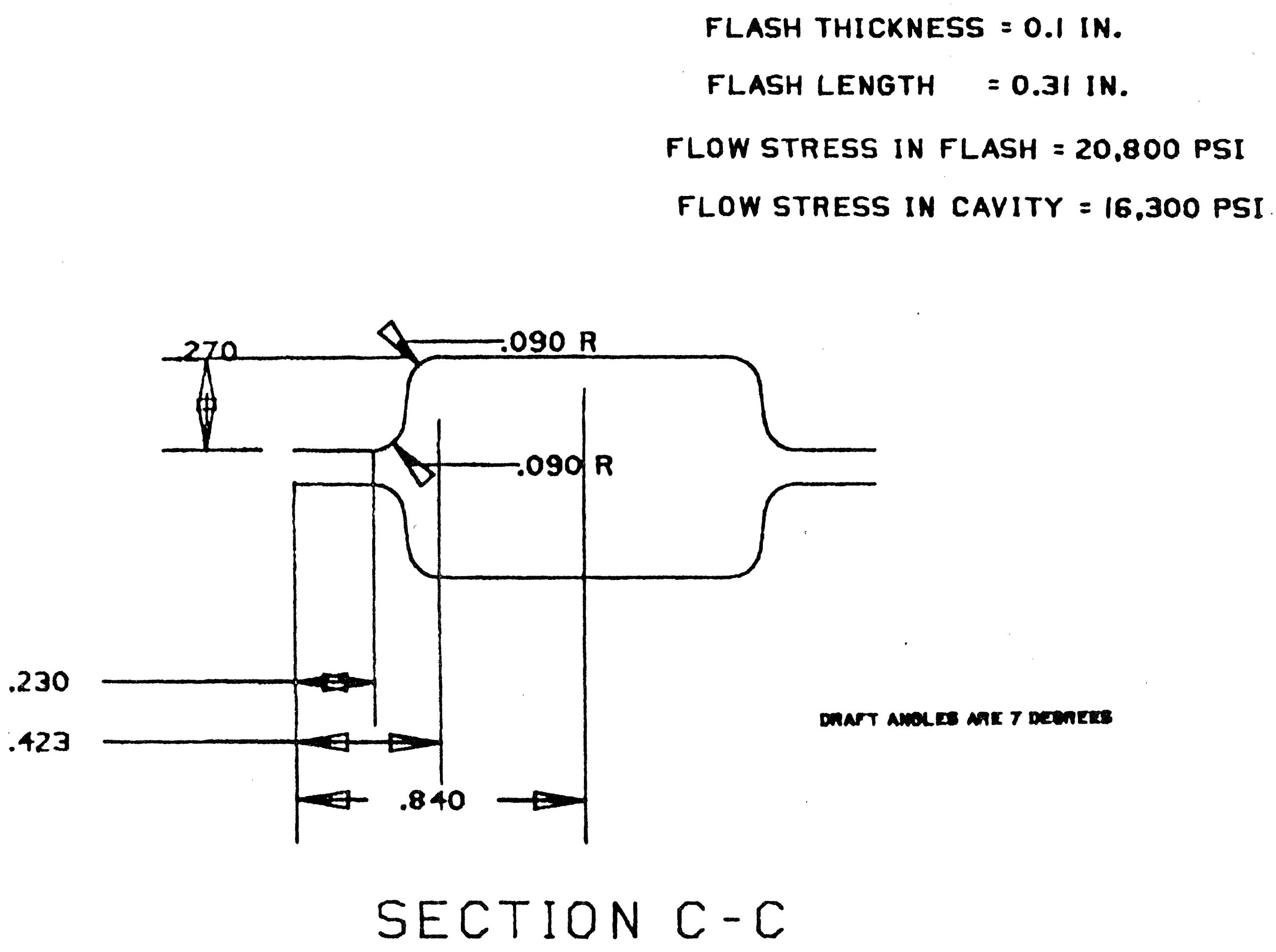

FLASH THICKNESS = 0.1 IN.
FLASH LENGTH = 0.31 IN.
FLOW STRESS IN FLASH = 20,800 PSI
FLOW STRESS IN CAVITY = 16,300 PSI
.270
.090 R
.090 R
.230
.423
.840
DRAFT ANGLES ARE 7 DEGREES
SECTION C-C

Figure 4: (continued) Dimensions of the Blocker Cross Sections of the Connecting Rod, seen in Figure 3.
(c) Section C-C

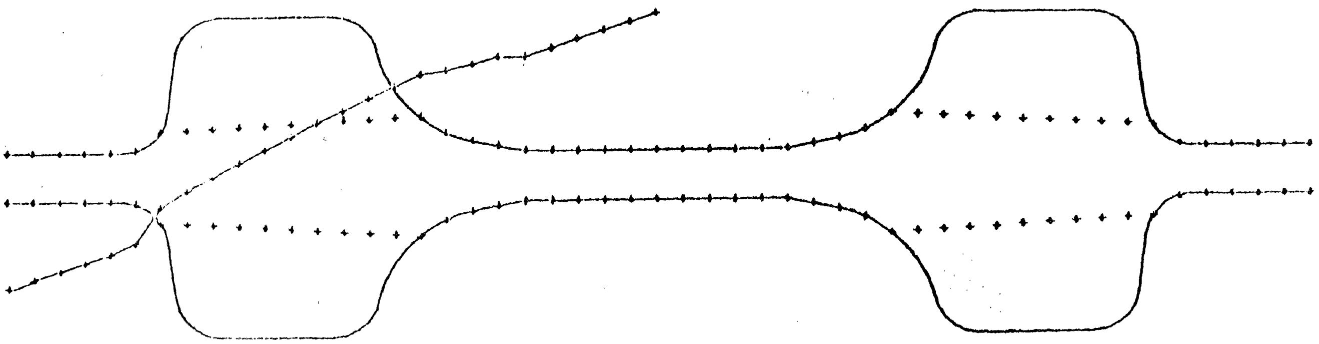


Figure 5: An Example Cross Section, the Flow Model and the Stress Distribution, as Calculated and Displayed by CV-Forge System.

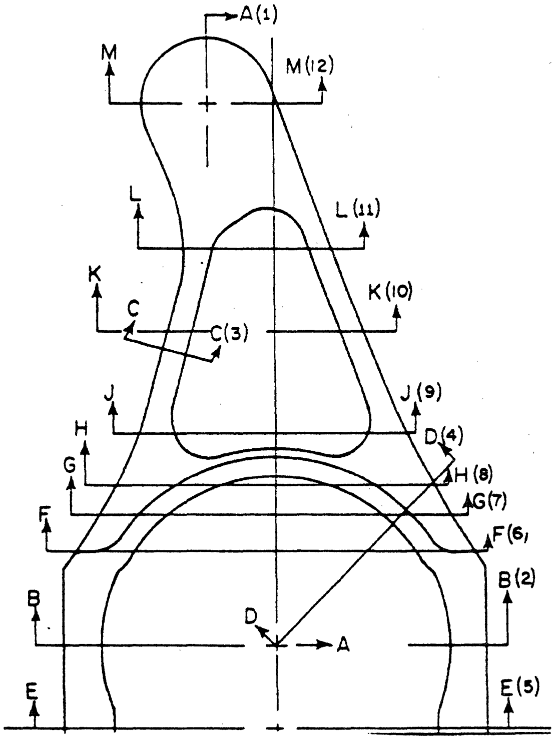


Figure 6: Plan View of a Titanium Alloy Connecting Link and the Sections Analyzed in Previous Study[7]

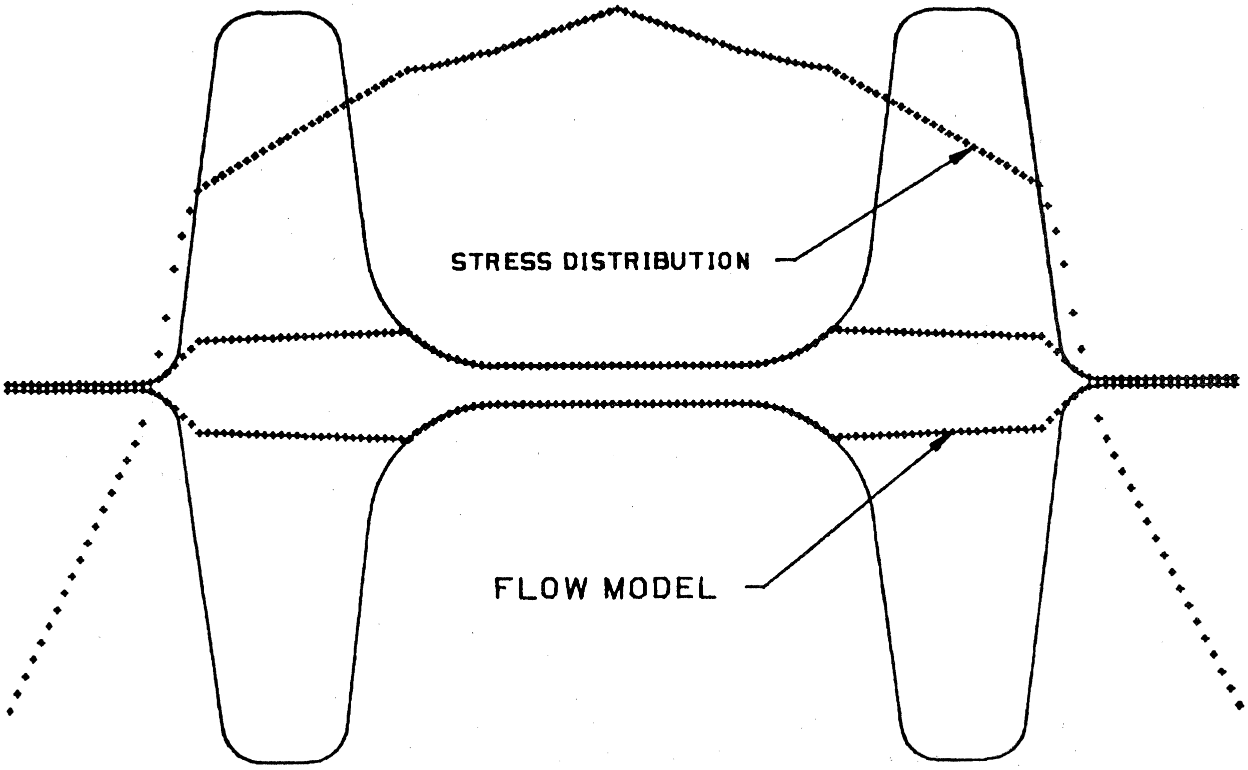


Figure 7: Stress Distribution on Cross Section J-J of the Connecting Link for an Assumed Flow Stress of 1000 PSI[7]
(Section Width Including Flash = 7.6 inch, Section Height = 4.7 inch)

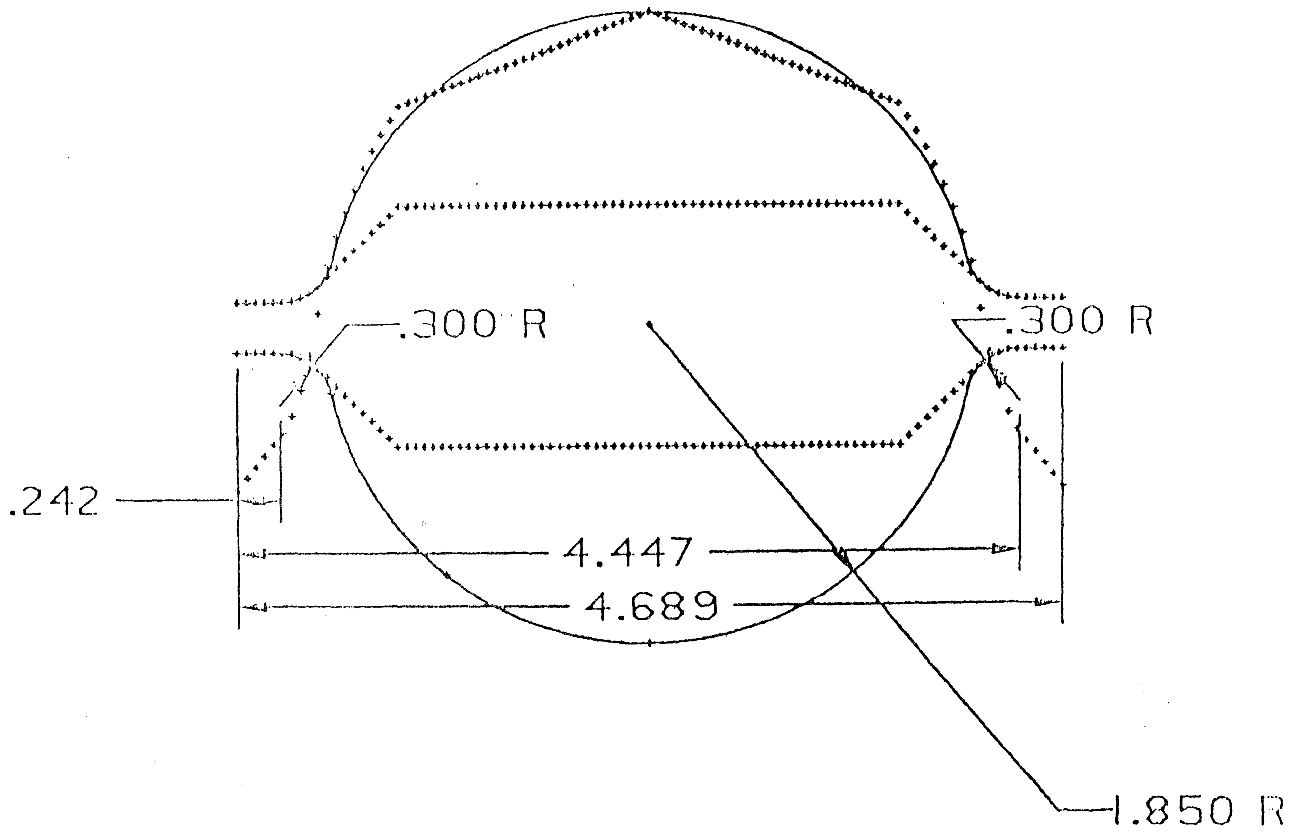


Figure 8: Main Bearing Cross Section of a Forged Crankshaft with Flow Model and Stress Distribution.

REFERENCES

(1) Altan, T., et. al., "Forging Equipment, Materials, and Practices", Handbook 03, 1973, Metals & Ceramics Information Center, Battelle, Columbus, Ohio.

(2) Akgerman, N., and Altan, T., "Modular Analysis of Geometry and Stresses in Closed-Die Forging: Application to a Structural Part", ASME Transactions, J. Engr. for Industry, Vol. 94, Series B, November 1972, p 1025.

(3) Subramanian, T. L., and Altan, T., "Practical Method for Estimating Forging Loads with the Use of a Programmable Calculator", ASM's Journal of Applied Metalworking, V1., No. 2, January 1980, p 60.

(4) Akgerman, N., Subramanian, T. L., and Altan, T., "CAD/CAM as Applied to Closed-Die Forging", SME Paper MS-75-516, presented at the CAD/CAM III Conference in Chicago, February 13, 1975,

(5) Billhardt, C. F. Akgerman, N., and Altan, T., "CAD/CAM for Closed-Die Forging of Track Shoes and Links", SME Paper MS76-739, October 1976.

(6) Lahoti, G. D., Nagpal, V., and Altan, T., "Selection of Lubricants in Hot Forging and Extrusion Processes", paper presented at the "Conference on Lubrication Challenges in Metalworking and Processing", IITRI, June 7-9, 1978, Chicago, Illinois, published in Proceedings.

(7) Subramanian, T. L., and Altan, T., "Application of Computer-Aided Techniques to Precision Close-Die Forging", Annals of CIRP, Vol. 27/1/1978, p 123.

THREE-DIMENSIONAL DATA ANALYSIS USING COMPUTERVISION CADDS4

David I. Terrill
Dresser Magcobar
10205 Westheimer
Houston, TX 77042
(713) 972-5972

Any engineer required to analyze or report test results can appreciate the difficulty of examining the reams of paper often generated by analysis programs or measurement tests. This paper presents a reasonably simple approach to display, verification, analysis, and documentation of test data using the three-dimensional graphics and user-oriented programming capabilities of the Computervision CADDS4 interactive graphics system.

The first requirement at Dresser Magcobar for a graphical display of test data was to display the results of an analysis program which had been generated on an engineering computer system. The program had output the results of seven different dependent variables, called ''sensitivities'' or uncertainties, by calculating their values through the ranges of the two independent variables, called ''TFO'' and ''Ixy''. The results were seven sets of data, each containing 703 data points. Because of the large quantity of data points, the data was very difficult to analyze, and manually preparing the data into a reportable form would have been prohibitively time consuming.

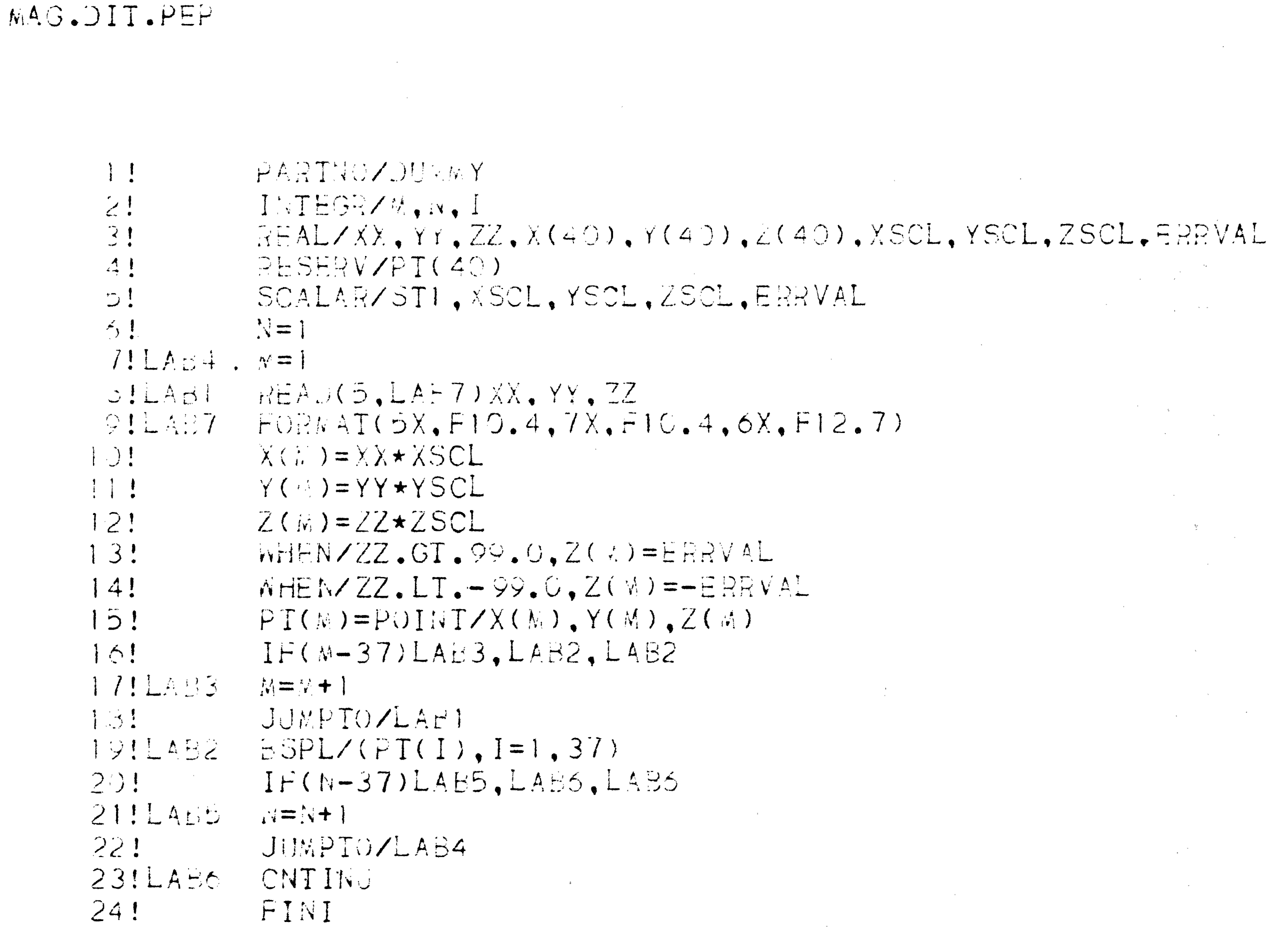
The approach taken was to write all seven sets of data onto standard ASCII character code magnetic tape files, then to read these files into text files on the Computervision system. Then a short, simple PEP program was written to read all the data points and generate three dimensional points and B-spline curves from the data. Then, B-spline patch surfaces were created from the curves, the results were rotated into an orientation for optimum display, and the axes were labelled using three-dimensional model text entities.

Figure 1 shows the PEP source file, Figure 2 shows a sample input data file, and Figures 3 thru 5 represent the resultant graphics files for three of the dependent variables in this first application. FORTRAN formatted READ statements were used to read the input data. Since the values of the dependent variables were very small numbers (see Figure 2), and option for scaling the values of X, Y, or Z data was added at line 5 of the PEP source file, then the scale factors were added to the beginning of the data files. For this reason, the ''sensitivity'' values represented in Figures 3 thru 5 appear considerably larger than the annotation on the ''sensitivity'' axis indicates. Also, certain occurrences of the data had infinite or indeterminate values, representing an error condition. These values were arbitrarily set to a large value and an option added to the PEP source file to revise the magnitude of the error condition for display purposes, as indicated by the ''INF'' annotation on the ''sensitivity'' axis.

Having thus created a graphical representation of his data, the engineer is now able to interactively examine, verify, or plot his data. In addition, in this particular application, it is important to the engineer to examine which area of "TFO" and "Ixy" fall within an acceptably low value of "sensitivity". Assuming that ±0.5 is his acceptable limit (reference Figure 5), he can define a view and use Z — clipping to display only the portions of the B-spline surface that lie between the planes $Z = \pm 0.5$. This capability is illustrated in Figure 6.

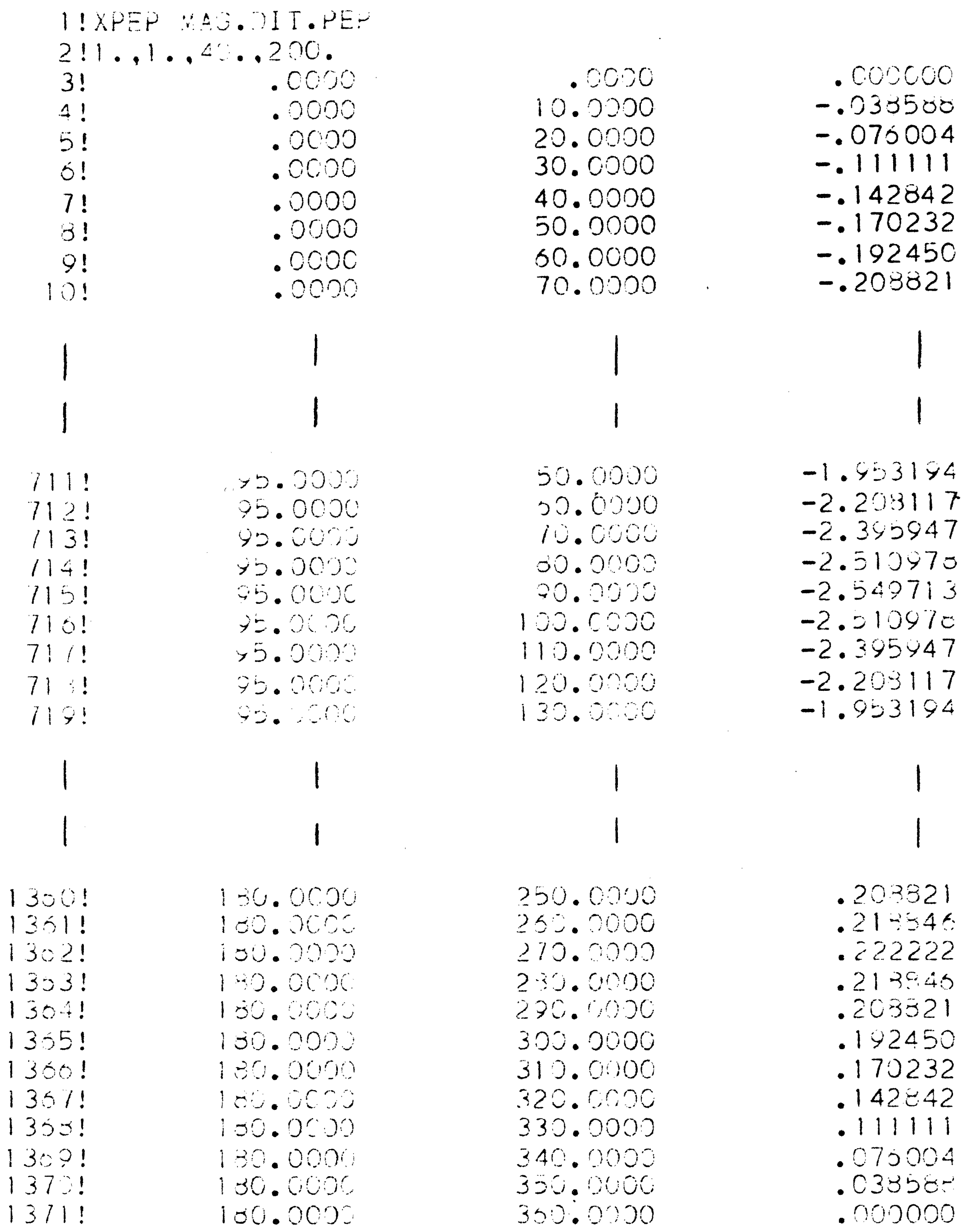
The same PEP source program, with minor modifications, can be used for a wide variety of different plot requirements. Another application is illustrated in Figures 7 thru 10, representing the results of test measurements in a survey area. Figures 7 and 8 show a rather "noisy" set of data points, and Figures 9 and 10 show a relatively stable set of points. Although the data in this case was generated manually and had to be manually input into the Computervision system, the entire project illustrated in Figures 7 thru 10 was accomplished with less than eight man-hours of effort.

In addition to the many features provided by Computervision CADDS4 for Mechanical Design and Drafting, from conceptual design through analysis to final documentation, the system also provides Dresser Magcobar a significant capability for interactive analysis and reporting of test data. Even in cases with large numbers of data points, completely annotated graphics files can now be generated with about two man-hours of effort. In highly repetitive projects where several graphics files are created using similar data, part figures can be created for axis annotation and execute files created for drawing generation, reducing the effort to produce graphics files to little more than system execution time.



```
MAG.DIT.PEP

        PARTNO/DUMMY
        INTEGR/M,N,I
        REAL/XX,YY,ZZ,X(40),Y(40),Z(40),XSCL,YSCL,ZSCL,ERRVAL
        RESERV/PT(40)
        SCALAR/STI,XSCL,YSCL,ZSCL,ERRVAL
        N=1
LAB4 .  M=1
LAB1    READ(5,LAB7)XX,YY,ZZ
LAB7    FORMAT(5X,F10.4,7X,F10.4,6X,F12.7)
        X(M)=XX*XSCL
        Y(M)=YY*YSCL
        Z(M)=ZZ*ZSCL
        WHEN/ZZ.GT.99.0,Z(M)=ERRVAL
        WHEN/ZZ.LT.-99.0,Z(M)=-ERRVAL
        PT(M)=POINT/X(M),Y(M),Z(M)
        IF(M-37)LAB3,LAB2,LAB2
LAB3    M=M+1
        JUMPTO/LAB1
LAB2    BSPL/(PT(I),I=1,37)
        IF(N-37)LAB5,LAB6,LAB6
LAB5    N=N+1
        JUMPTO/LAB4
LAB6    CNTINU
        FINI
```

Figure 1: PEP source file.



```
   1!XPEP MAG.DIT.PEP
   2!1.,1.,40.,200.
   3!          .0000               .0000          .000000
   4!          .0000             10.0000         -.038588
   5!          .0000             20.0000         -.076004
   6!          .0000             30.0000         -.111111
   7!          .0000             40.0000         -.142842
   8!          .0000             50.0000         -.170232
   9!          .0000             60.0000         -.192450
  10!          .0000             70.0000         -.208821

   |             |                  |                |

   |             |                  |                |

 711!        95.0000             50.0000        -1.953194
 712!        95.0000             60.0000        -2.208117
 713!        95.0000             70.0000        -2.395947
 714!        95.0000             80.0000        -2.510976
 715!        95.0000             90.0000        -2.549713
 716!        95.0000            100.0000        -2.510976
 717!        95.0000            110.0000        -2.395947
 718!        95.0000            120.0000        -2.208117
 719!        95.0000            130.0000        -1.953194

   |             |                  |                |

   |             |                  |                |

1360!       180.0000            250.0000          .208821
1361!       180.0000            260.0000          .218546
1362!       180.0000            270.0000          .222222
1363!       180.0000            280.0000          .218546
1364!       180.0000            290.0000          .208821
1365!       180.0000            300.0000          .192450
1366!       180.0000            310.0000          .170232
1367!       180.0000            320.0000          .142842
1368!       180.0000            330.0000          .111111
1369!       180.0000            340.0000          .076004
1370!       180.0000            350.0000          .038588
1371!       180.0000            360.0000          .000000
```

Figure 2: Input Data file.

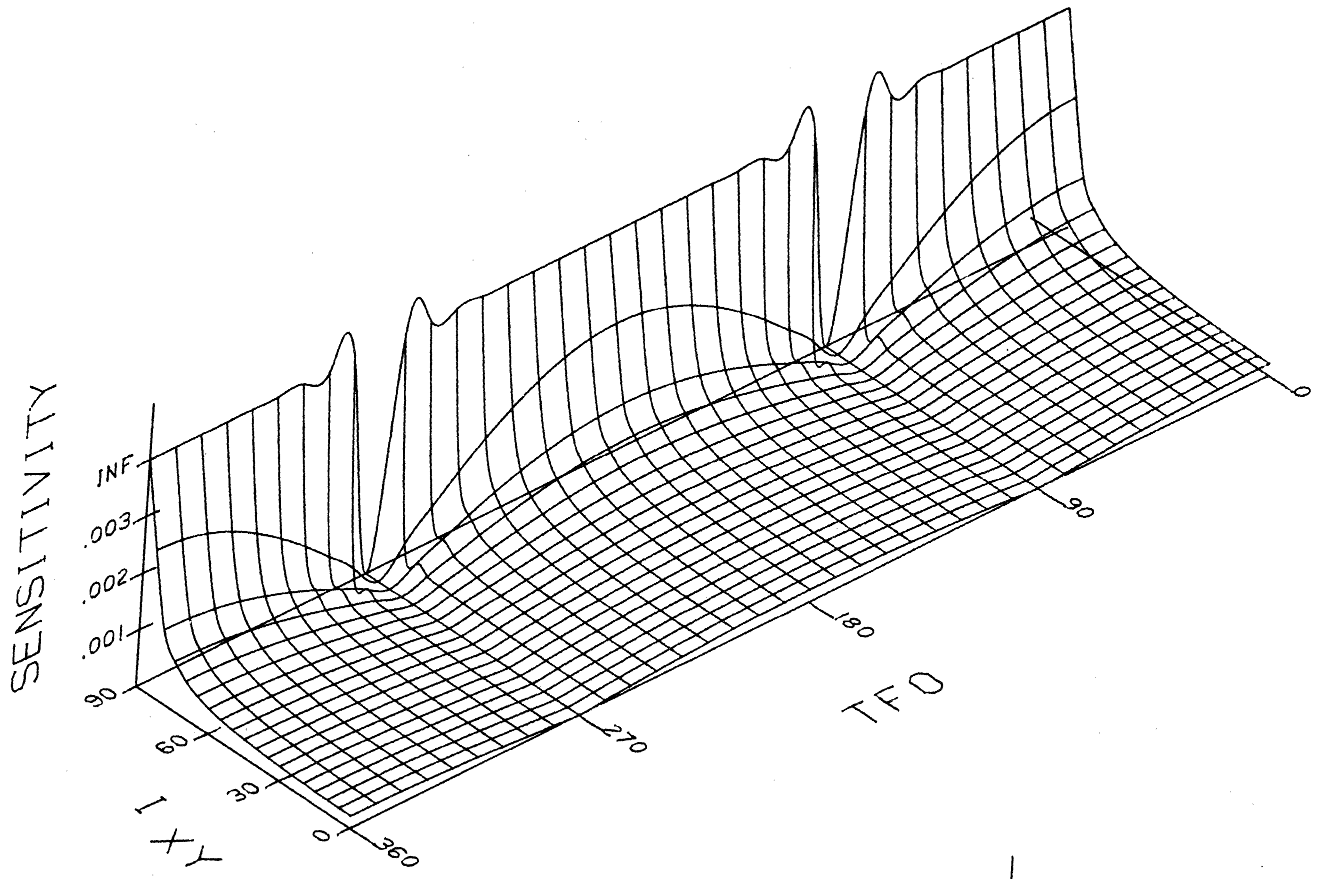


Figure 3: Graphics file containing 703 data points.

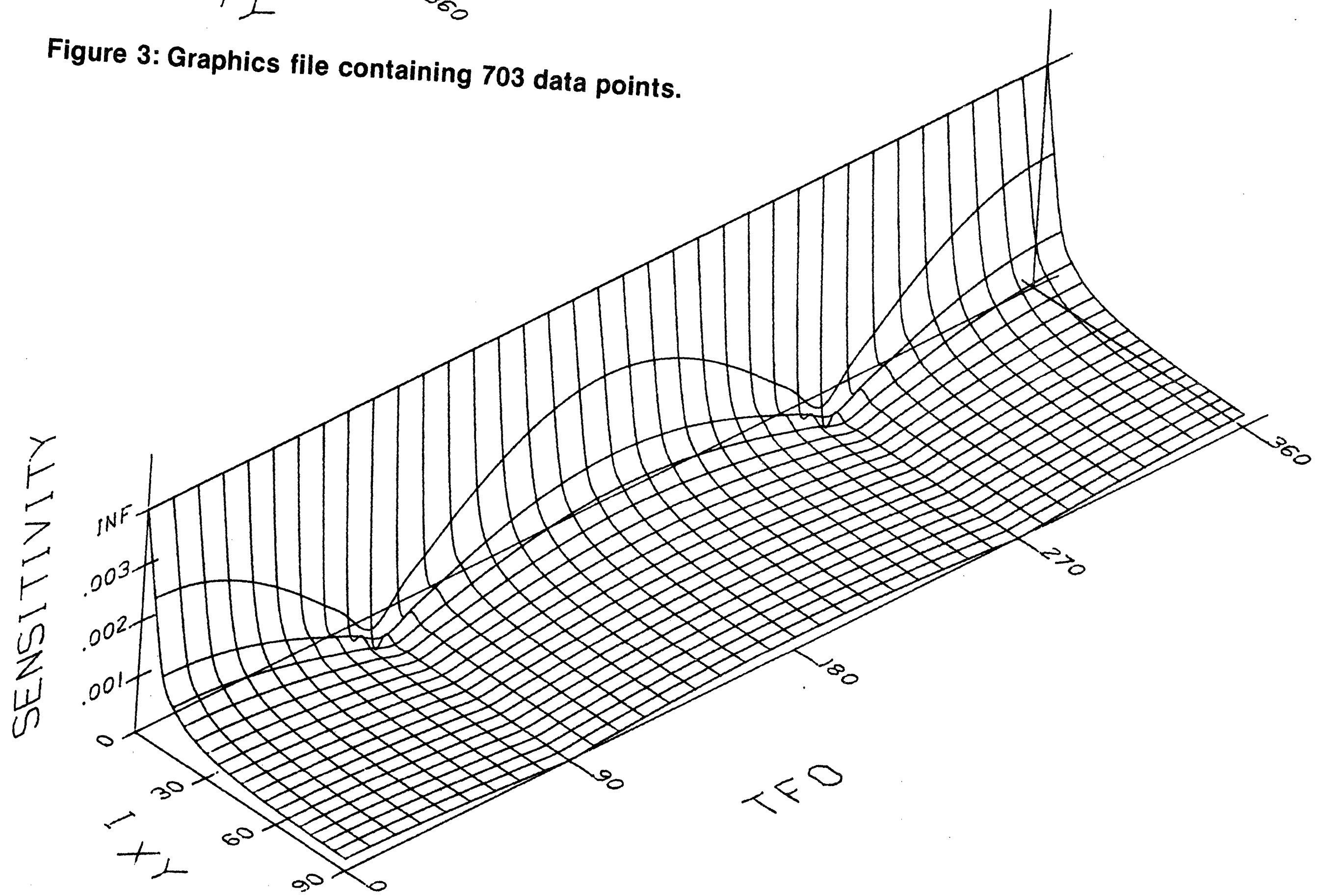


Figure 4: Graphics file containing 703 data points.

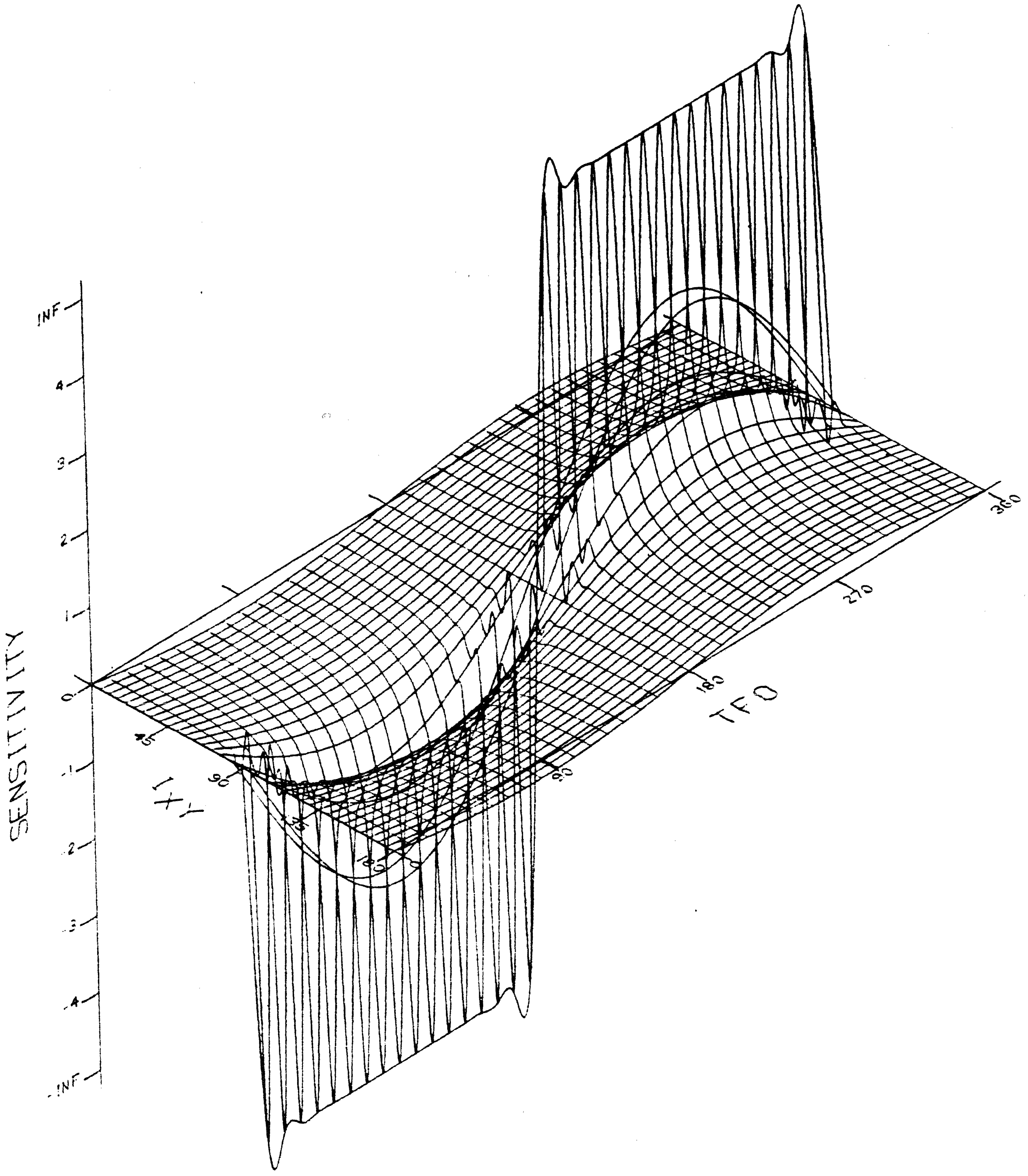


Figure 5: Graphics file containing 1369 data points.

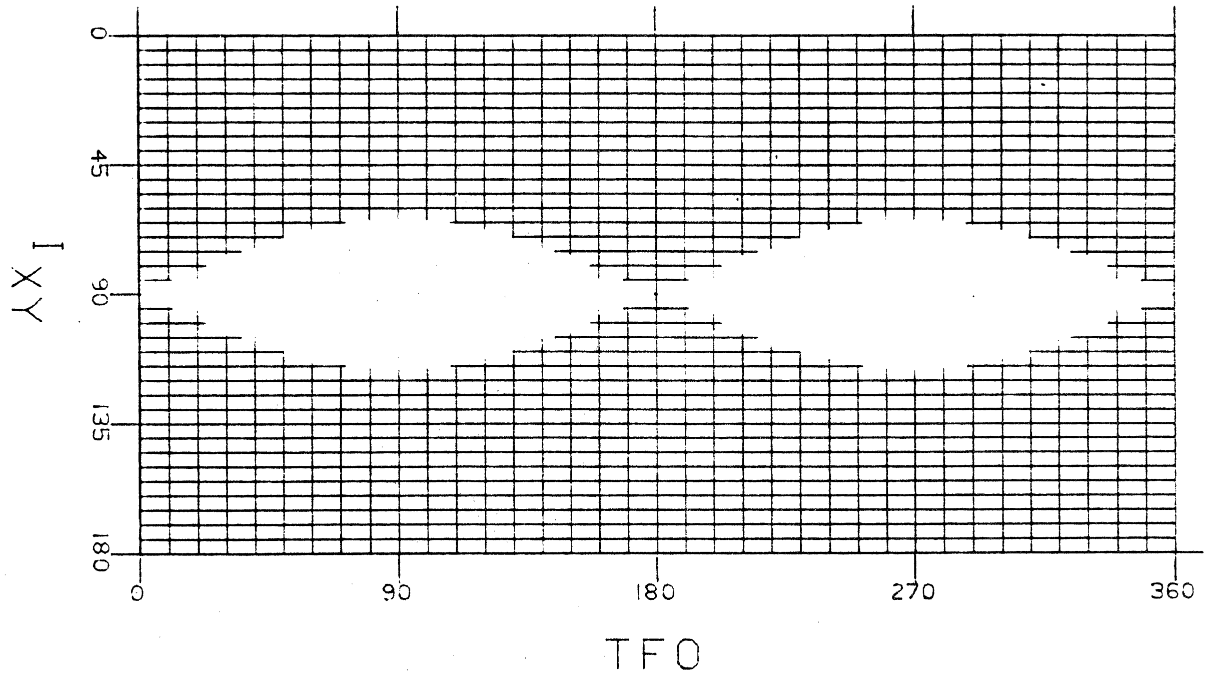


Figure 6: Using Z — clipping to display acceptable sensitivity values.

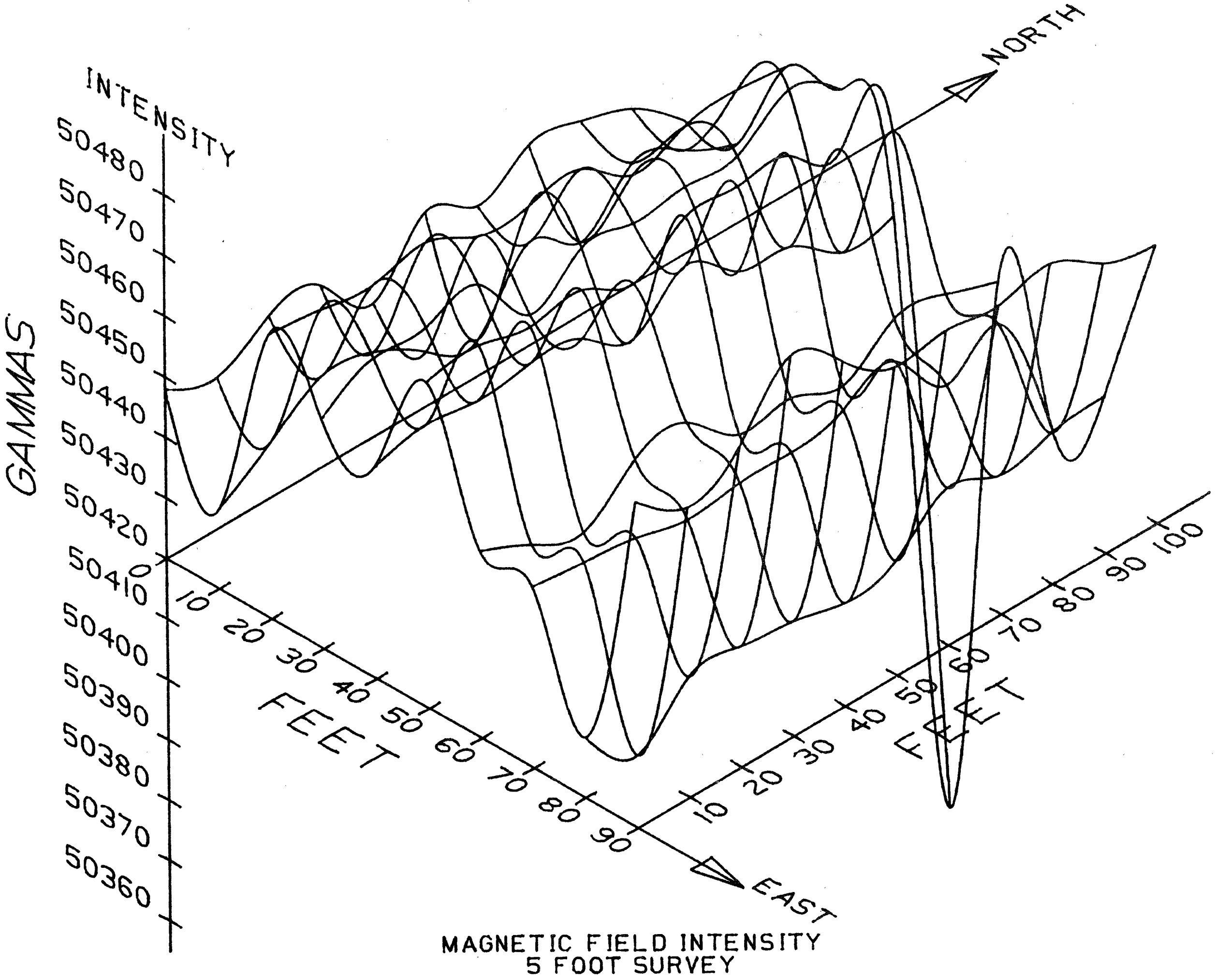


Figure 7: Graphics file for test measurement application.

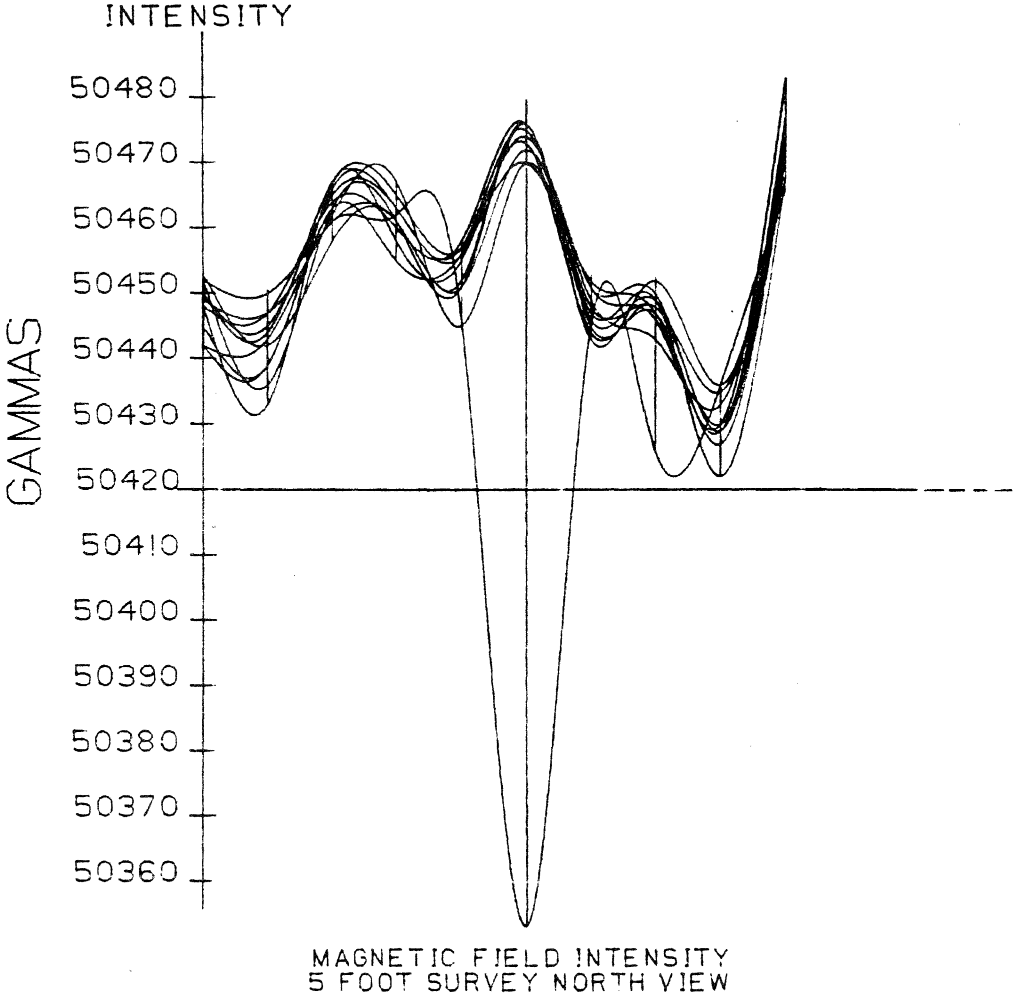


Figure 8: Orthographic view of data shown in Figure 7.

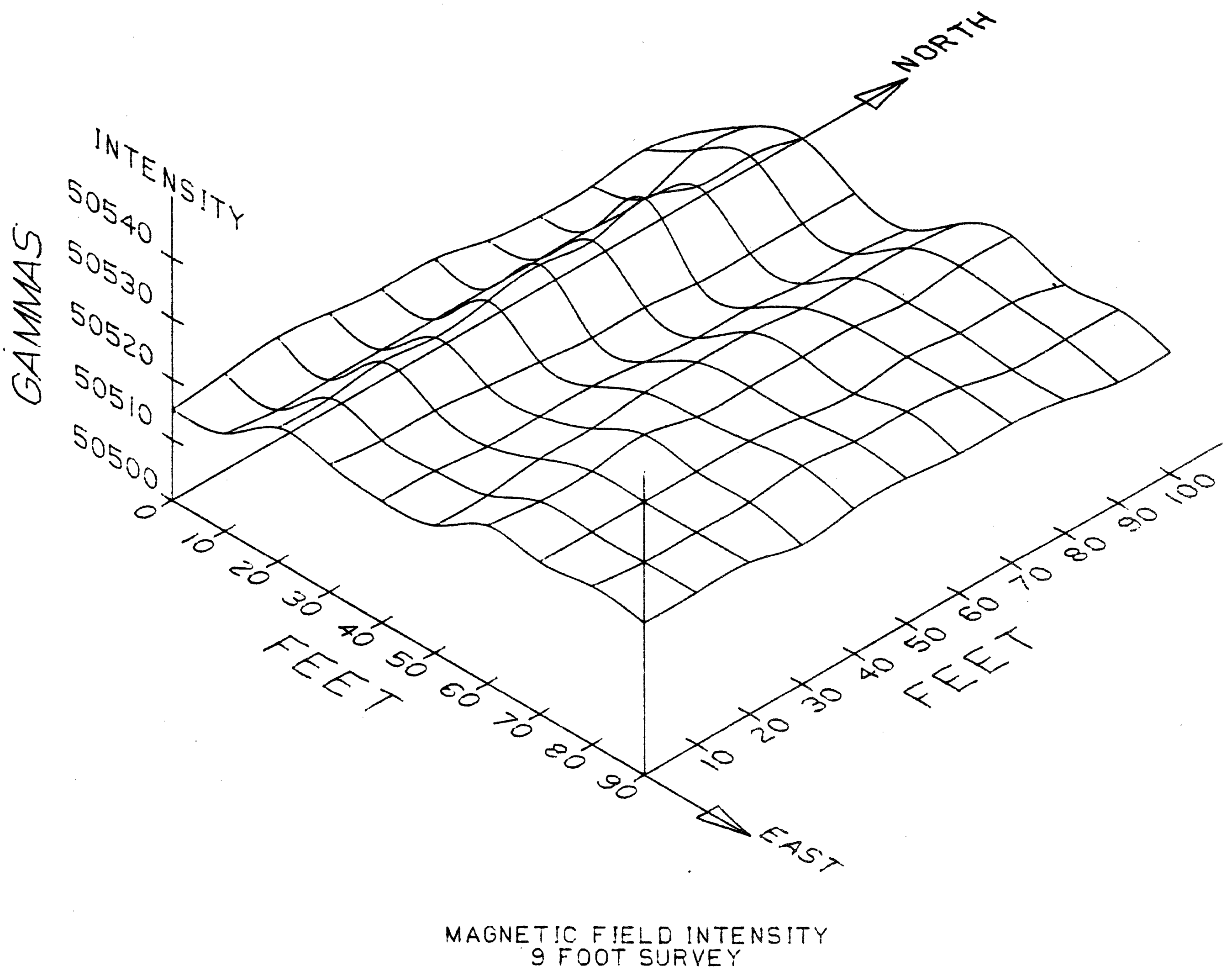


Figure 9: Graphics file for test measurement application.

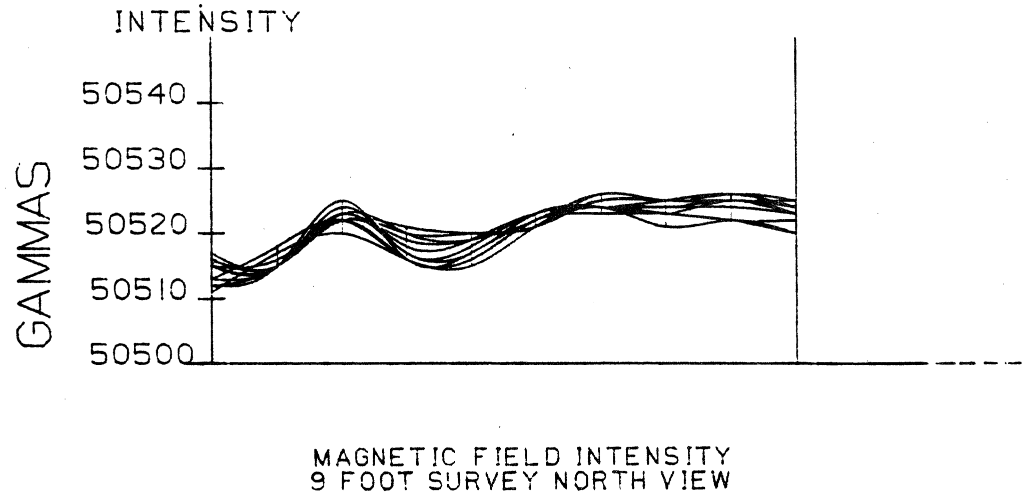


Figure 10: Orthographic view of data shown in Figure 9.

A CUSTOMISED PROCEDURE FOR ASSEMBLY DRAWINGS AND LIST OF PARTS

John Painter
Computervision Centre
1040 Uxbridge Rd.
Hayes, Middlesex
England
1-561-2626

APPENDICES
1. Part Layering scheme
2. The drawing sheet on balloons
3. Figures — a typical application
4. Programs and execute files.

ABSTRACT

Companies have many different reasons for acquiring CAD/CAM systems, but foremost in the requirements of most companies is the need to produce complex engineering drawings as quickly as possible. This paper outlines techniques for creating arrangement drawings quickly and easily using the facilities of the Computervision system.

The techniques make use of a number of features of the system including 'VARPRO' programs to handle standard procedures, 'libraries' of standard and parameterised parts and 'Data Extract' to generate information for a bill of materials report.

Many experienced users are developing, or have developed, techniques such as these described here to satisfy various requirements, which include the need to produce assembly and arrangement drawings, layouts and tender documents for a variety of applications in mechanical engineering design and draughting.

This paper is primarily directed to the newer users who may be looking for methods such as these to solve problems in such applications, but I hope a few of the more experienced users will also pick us a few 'TIPS'.

THE METHOD

The method uses a procedure for creating an engineering drawing which begins with the insertion of a standard drawing form. The actual drawing is created mainly by inserting standard components, from a library of such items, with additional information inserted using the standard CADD's commands, as necessary. The information for the bill of materials is contained within 'balloons' which can be inserted with the library parts or separately. Data extract procedures are used to create a bill of materials report.

The techniques involve the use of EXECUTE files and VARPRO programs to standardise and automate much of the work, some examples of these programs are included at the end of this paper.

The details of the method, as described in the following paragraphs, are based on a particular application at a particular company but the actual examples and the execute files have been modified to suit the needs of this presentation. Each application is individual and the techniques will be developed to suit the requirements of the particular application.

THE DRAWING SHEETS

The production of an engineering drawing begins with the insertion of a standard drawing format.

Drawing sheets are held on the system in the standard sizes used by the company — A0 to A3. Each sheet comprises two parts — an outline frame and certain other data to be inserted into the engineering drawing as individual entities (i.e. as a 'part'), and fixed text and titling boxes to be inserted as a single 'figure'.

The PART includes the outside edge and inside frame of the drawing, which may then be used for reference if necessary. It also includes 'text nodes' which describe the text to be added to the titling box, defining its position, font, height and other parameters, as required. The TNODES also contain properties for use by the data extract procedure.

The FIGURE contains the remainder of the drawing sheet, the internal boxing, the company name and other annotation, symbols and other information contained in the company's standard drawing form.

The parts that make up a typical drawing sheet are shown in Appendix 2 (figs. 1-3). Many variations on the theme may exist, for example the same fixed standard titling box may be inserted in various sized frames, or there may be a completely different layout for each size drawing.

Layering conventions must be used, the particular requirements determine the complexity of the scheme. The layering scheme applicable to this example is given in the appendices.

INITIALISING A DRAWING

An engineering drawing is initialised by beginning a new part into which the drawing frame is inserted. To standardise and simplify the procedure a VARPRO program is used to begin the drawing, insert a drawing sheet of the required size and set the layering and other parameters necessary.

The VARPRO program X.DRG.RUN-BEG-1D, included in the appendices, satisfies the basic requirements for simple 2D drawings. Many applications require a more powerful program for complex 2D and 3D drawings which would pre-define a number of 'parts' automatically.

Also included in an execute file X.DRG.INS-TITLES which requests data and inserts it on the correct text nodes in the titling box.

THE BALLOONS

In an engineering drawing balloons are used to identify the individual items of an assembly and reference them to a parts listing.

Arrangement and assembly drawings are built up by adding parts which represent the items which make up the assembly. In many applications involving a schematic representation of the finished component, e.g. piping and printed circuit boards, the individual items are added as figures to provide a pictorial presentation of the assembly, however in many mechanical engineering applications it is more convenient to add the items and sub assemblies as parts so that the individual entities can be modified to suit a particular need. To provide a textural description of the component, enabling a parts list to automatically be generated, a balloon is added as a figure.

The balloons are parts comprising the necessary graphics, a circle, a connect node to which the string (a nodal line) is connected, and text nodes with properties to position the reference number and the remaining text. In this application it was more convenient to include all the descriptive information as text (on layers to be echoed off), in other cases it may be more satisfactory to add it as a property.

The balloons are added to the drawing as a nodal figure and then connected to the item referenced by the nodal line which terminates at a small circle (with a connect node at its centre) on the item being referenced. Once added the balloons may be moved around very easily. Other methods of implementation include terminating the string with an arrow, and omitting the circle around the reference number (as in this particular application).

Since there are many instances in which a balloon must reference an item which already has a description (in another view for instance) a 'null balloon', i.e. one without the property information, is also provided. The balloons are shown pictorially in Appendix 2 (figs. 4–6).

USING THE BALLOONS

The balloons can be added to the assembly drawing after the individual items have been inserted, however they are usually a part of the library part. They may also be added automatically after the item has been drawn (by a PEP or VARPRO program).

To simplify the insertion of balloons a number of VARPRO Programs have been developed, X.B.INS-BALLOON and others are included in the appendices. A program for inserting balloons into 3D parts is also included.

The method offers extreme flexibility since the individual parts of an assembly can be created in the most convenient manner only the balloon need conform to the standards required for the data extract procedures.

THE LIBRARY PARTS

An assembly drawing is generated by adding together items and sub assemblies which are held in a 'library of parts'.

The library comprises the individual items and frequently used sub assemblies. Individual parts normally contain just the geometric description but sub assemblies may include cross-hatching, centre lines and hidden detail in the way in which they would normally appear in the finished drawing, some examples are given in Appendix 3 (figs. 1–2). The sub assemblies, and where appropriate the individual items, also include the balloons suitably annotated and placed in a position convenient for the finished drawing. Other annotation may be included to assist the user. Further examples are given in figures 3–7.

The library also includes certain standard sections and views which are required for the finished drawing (figs. 8–9). Similar libraries in other applications may also include standard blocks of notes, data tables, and other similar items.

The majority of the library items in this application have been constructed and stored as 'parts' on the system, but many applications will make use of 'parameterised' parts generated using PEP and/or VARPRO programs which may automatically add the balloon after creating the part.

The library is built up on an 'as needed' basis, i.e. the first time an item is required it is created and used, then it is filed as a library part for immediate use on the next occasion. Clearly it takes a little time to build up a useful library but a careful study of future needs and time spent on building an effective library will pay considerable dividends.

CONSTRUCTING THE DRAWING

The construction of the drawing is a straight-forward process. The various stages are shown in Appendix 3. After the drawing sheet has been inserted using the initialisation procedure (fig. 10), the items and sub assemblies are added. Various stages in the insertion of the first few items are shown (figs. 11–16), together with the completed main view (fig. 17). Alternative methods of construction include building the views separately and inserting them into the drawing sheet on completion, a particular convenient method if the size of completed component cannot easily be ascertained at the start.

The completed drawing is shown, with the titling block annotated (figs. 18–19) and finally the drawing ready for plotting (fig. 20).

EXTRACTING THE BOM

The list of parts (bill of materials) is obtained using 'data extract'. This package provides the facilities for extracting non-graphic information from a drawing to build a file of textural information, and may be used to create a variety of reports from a part or set of parts.

The list of parts is created from the information contained in the balloons and certain header information is extracted from the titling box. The data to be extracted and the manner of presentation of the report is described by an 'extract definition file' (EDF). The EDF and the VARPRO program which runs the complete process are also included in the appendices.

In this particular application there is a further requirement to post process the basic bill of materials report. The output from data extract is restructured to provide trailer lines for change information, double line spacing of items, a tally of the number off of each item from counts on the balloon, and other minor modifications. The report is then output to a line printer in a way such that it can be cut into A4 sheets that closely resemble the existing (non-computer generated) forms. Other applications are not, usually, as demanding!

POST SCRIPT

There are many different ways in which the methods outlined above can be implemented, each application must be examined individually. The basic procedure will be the same, but the drawing formats and balloons can be constructed to suit the standards of the particular company.

Many experienced users have already developed their own procedures, I hope that this paper will prompt other users to look at similar applications and help them to build their own methods.

APPENDIX 1
PART LAYERING SCHEME FOR MDD APPLICATIONS

The following pages define a general layering scheme for mechanical design and draughting and associated applications.

LAYERS — SUMMARY OF LAYER ALLOCATION

0 temporary work, construction etc. — to be erased on completion — not to be drawn

1–99 for design and draughting office use — inc. geometric description, drawing annotation, etc.

100–199 for associated applications or further design use depending on part requirements.

200–255 for management information use and other non-geometric/ draughting information.

NOTE Many other aspects of system management and administration depend on the layering strategy. It is important that it is well-defined, comprehensive yet not complex, and the number layers allocated should be kept to a minimum with room for future development, and tailoring.

LAYERS — DRAWING MANAGEMENT INFORMATION AREA

201–209 balloon graphics and text and/or similar data in drawing views for part list information.

210–229 reserved for future development for drawing management systems.

230–239 engineering information — mass properties and flow/stress analysis calcs and results.

240 drawing frame — outer edge of sheet and inner frame, 'keep in' area.

241 drawing blank outlines and title block lines, company name and similar data.

242–243 text and figures in drawing blank for pen thicknesses — fine (242) and standard (243).

244–249 other text on standard sheet for specific applications, used as necessary.

250 text nodes and text for title box data, (spread to following layers if necessary).

255 textural description of information contained in this part (kept to a minimum).

NOTE The actual number of layers allocated is small — there is considerable room for future development and tailoring to satisfy particular requirements.

APPENDIX 2
FIGURES THE DRAWING SHEET AND BALLOONS

Fig. 1 the drawing frames and text nodes.
Fig. 2 the text nodes in the titling box.
Fig. 3 the titling box figure.
Fig. 4 the balloons showing annotation.
Fig. 5 the balloons echoed for drawing.
Fig. 6 the balloons showing the properties.

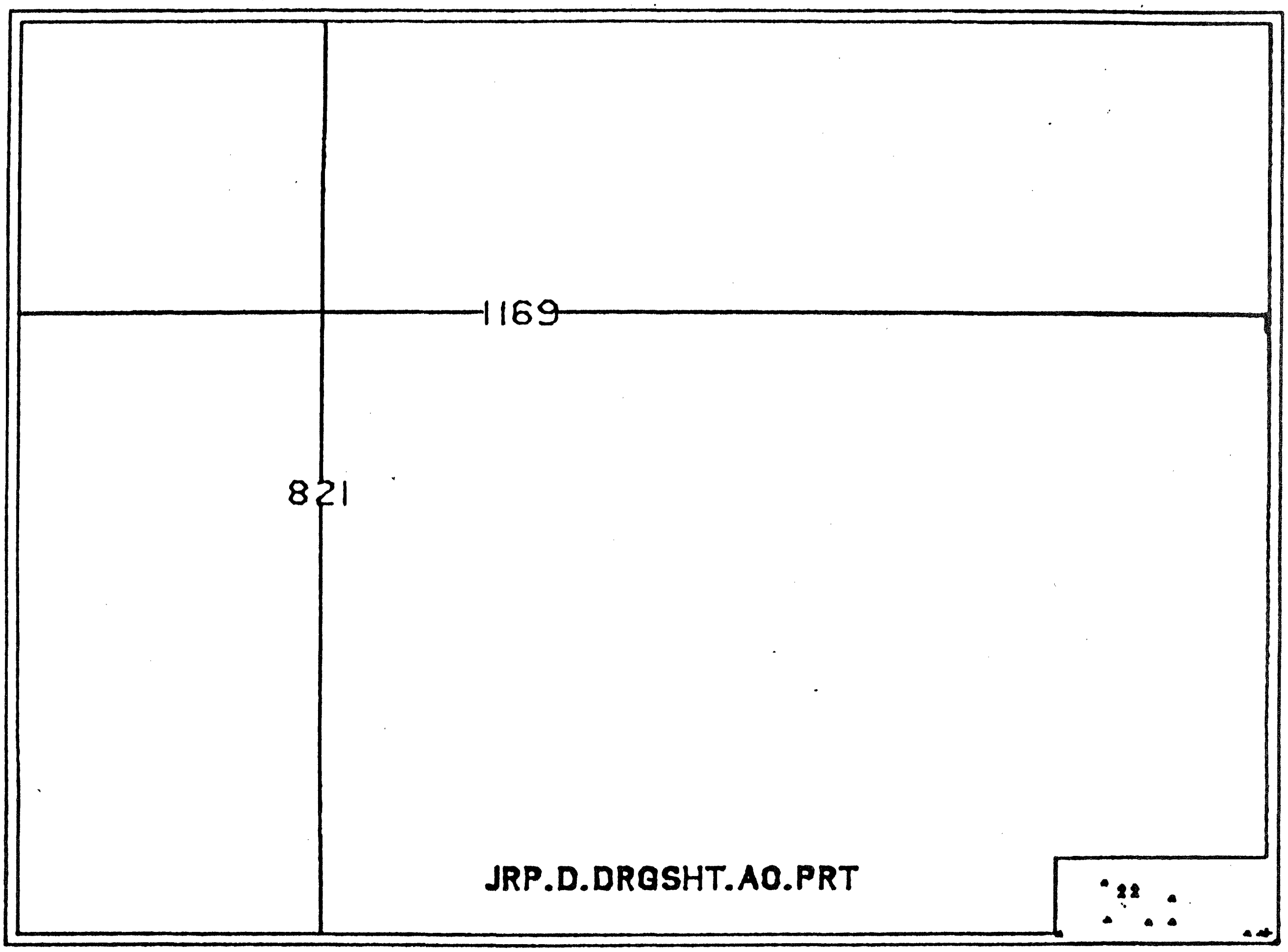


Figure 1

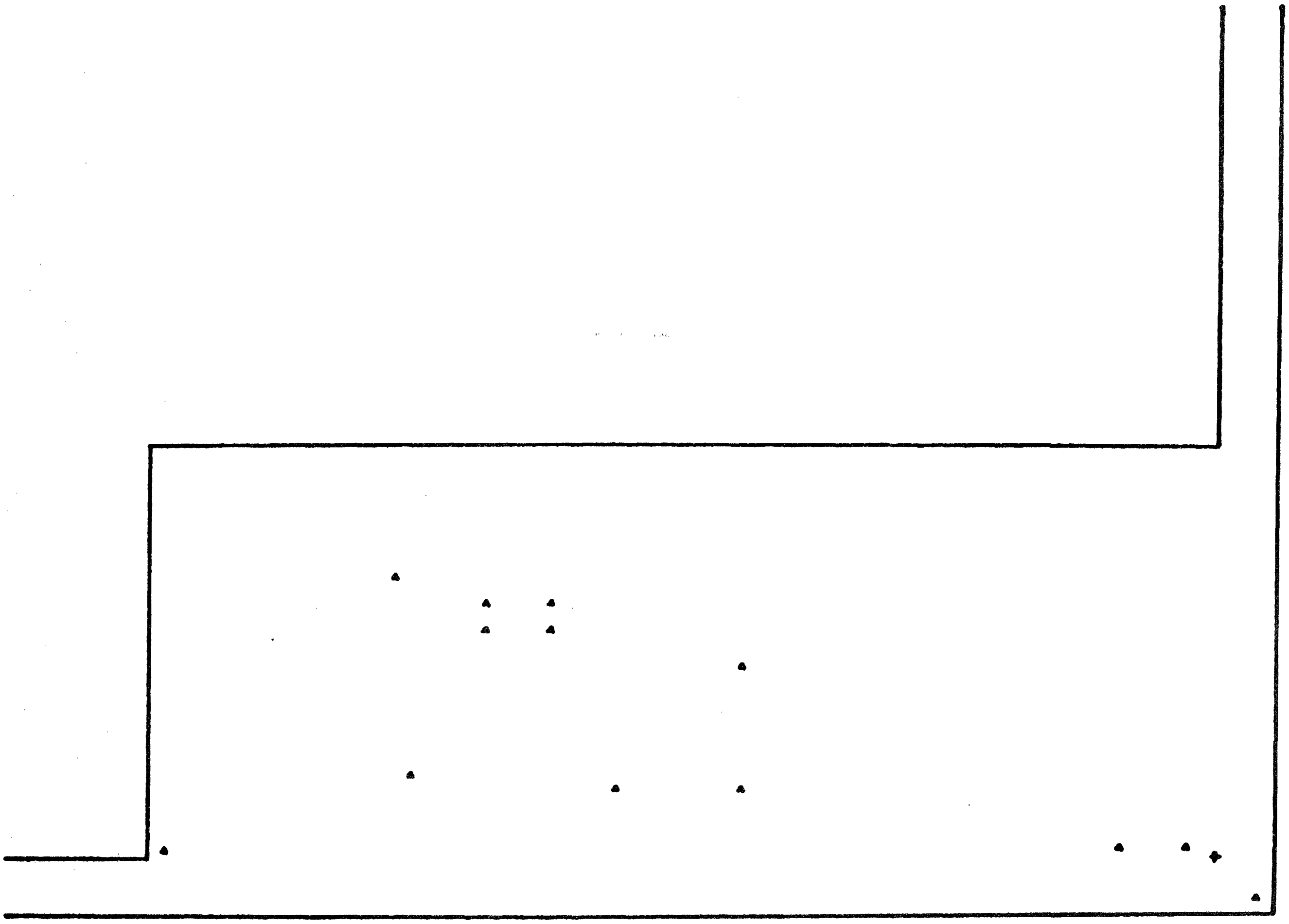


Figure 2

JRP.D.DRGSHT.AO.FIG

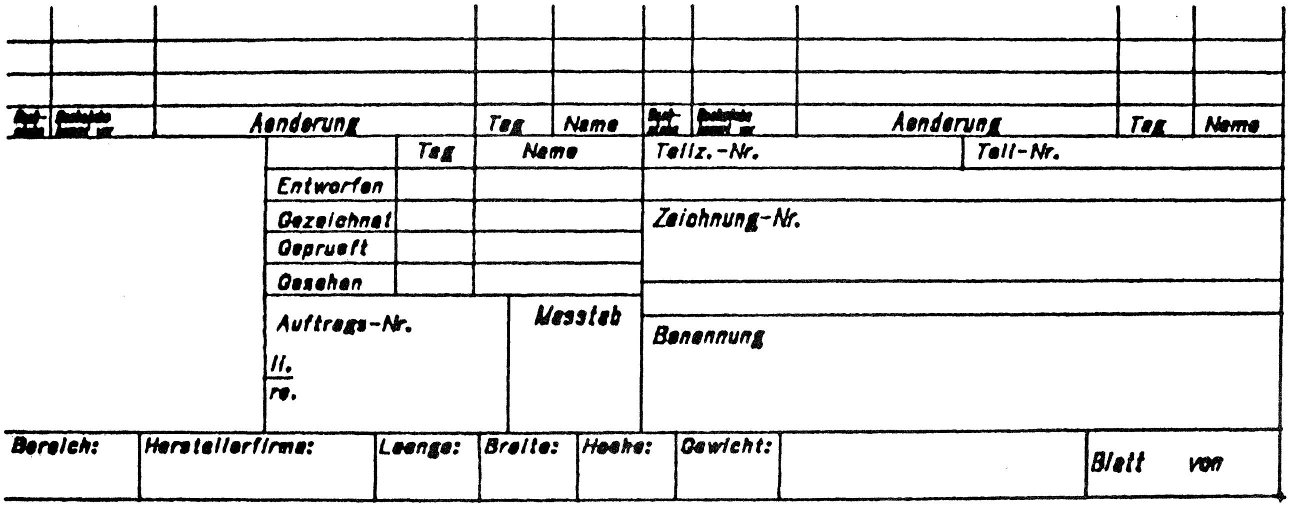
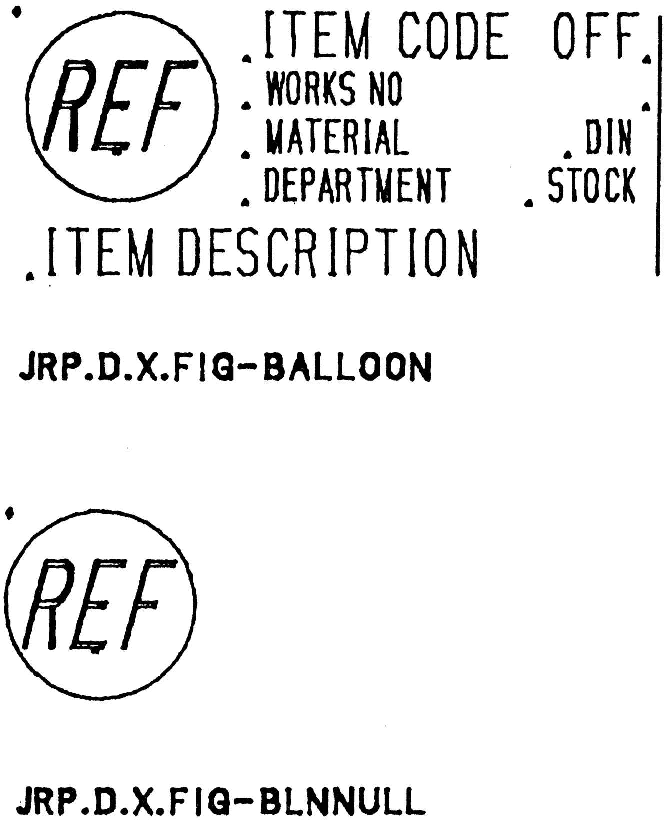


Figure 3



JRP.D.X.FIG-BALLOON

JRP.D.X.FIG-BLNNULL

Figure 4

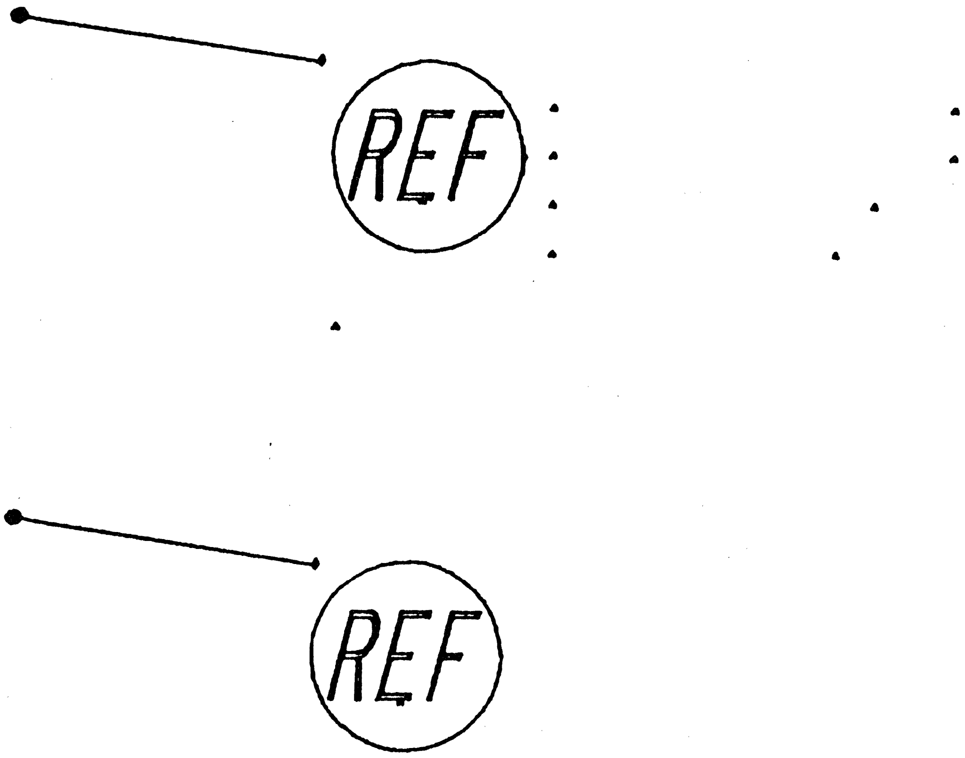


Figure 5

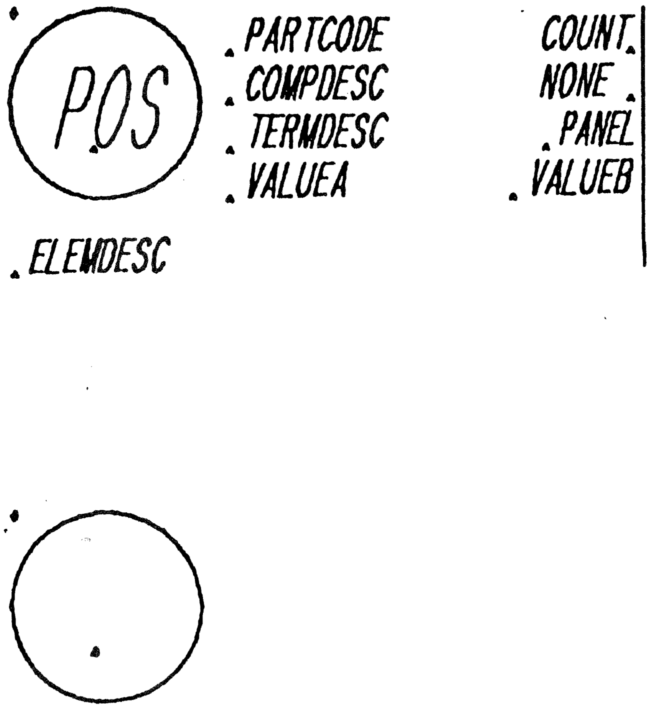

POS
PARTCODE
COMPDESC
TERMDESC
VALUEA
COUNT
NONE
PANEL
VALUEB
ELEMDESC

Figure 6

APPENDIX 3
FIGURES A TYPICAL APPLICATION

Fig. 1-2 library part sub assemblies
Fig. 3-7 library parts showing annotation.
Fig. 8-9 library part section and view.
Fig. 10 the drawing sheet inserted.
Fig. 11 the first part inserted.
Fig. 12 unnecessary annotation removed.
Fig. 13 the second part inserted,
Fig. 14 and suitably re-arranged.
Fig. 15 the third part inserted.
Fig. 16 the fourth part inserted.
Fig. 17 the completed view.
Fig. 18 the completed drawing.
Fig. 19 the titling box.
Fig. 20 the drawing ready for plotting.
The list of parts report.

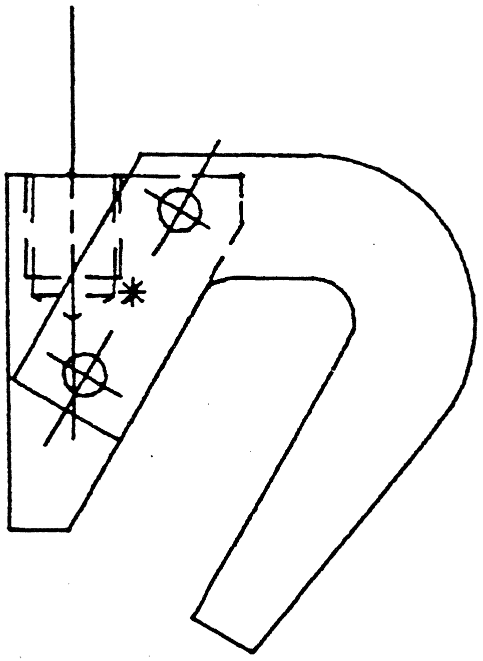


Figure 1

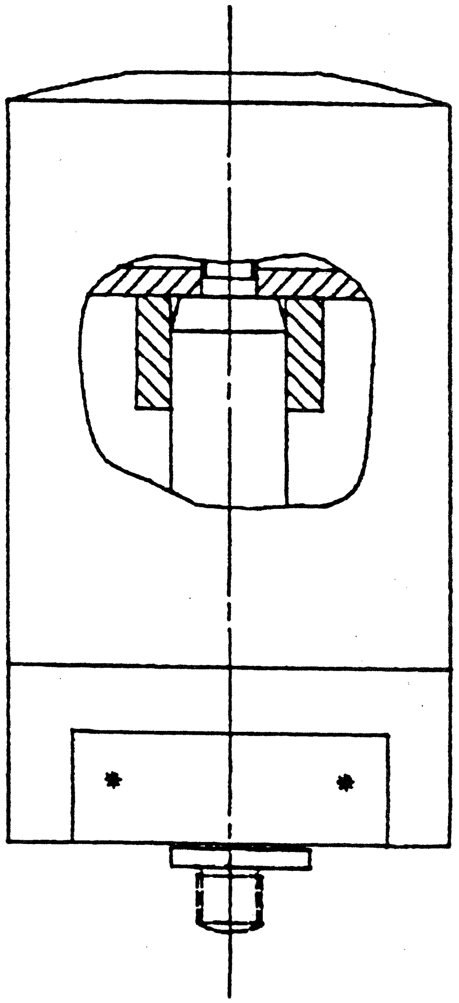
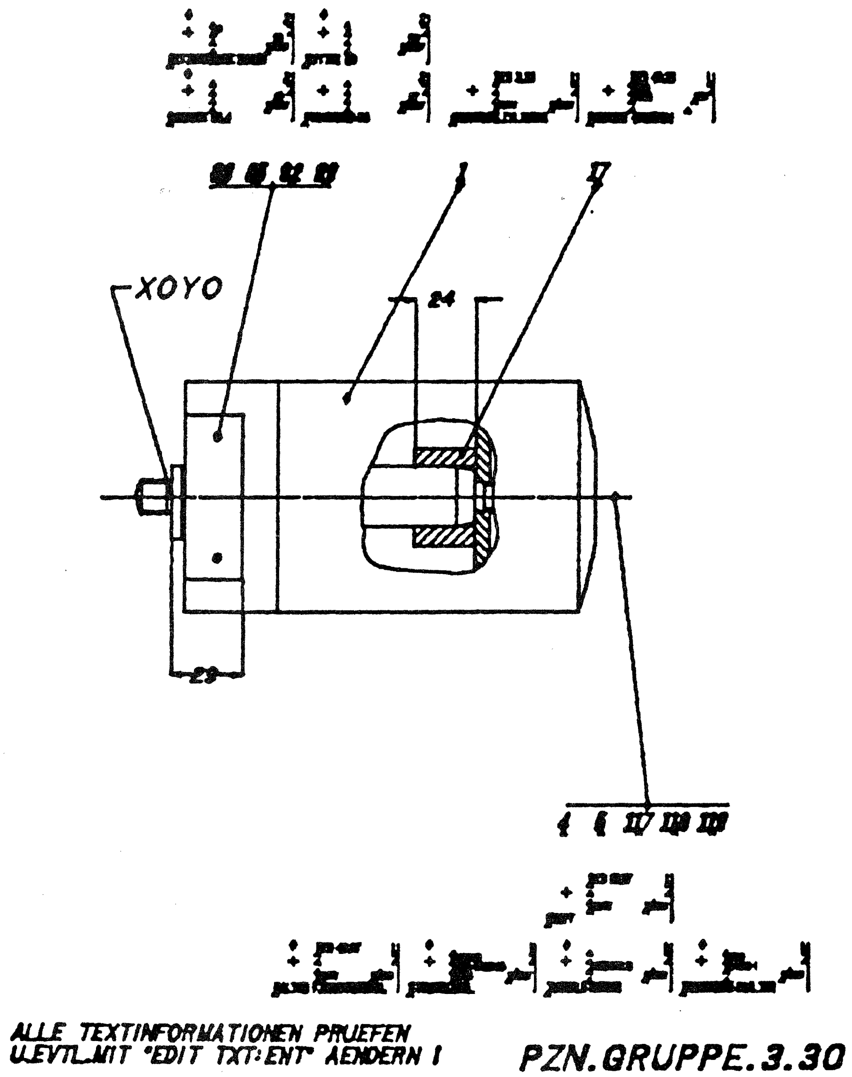


Figure 2



1
17
XOYO
24
29
4 6 117 118 119
ALLE TEXTINFORMATIONEN PRUEFEN
U.EVTL.MIT 'EDIT TXT:ENT' AENDERN !
PZN.GRUPPE.3.30

Figure 3

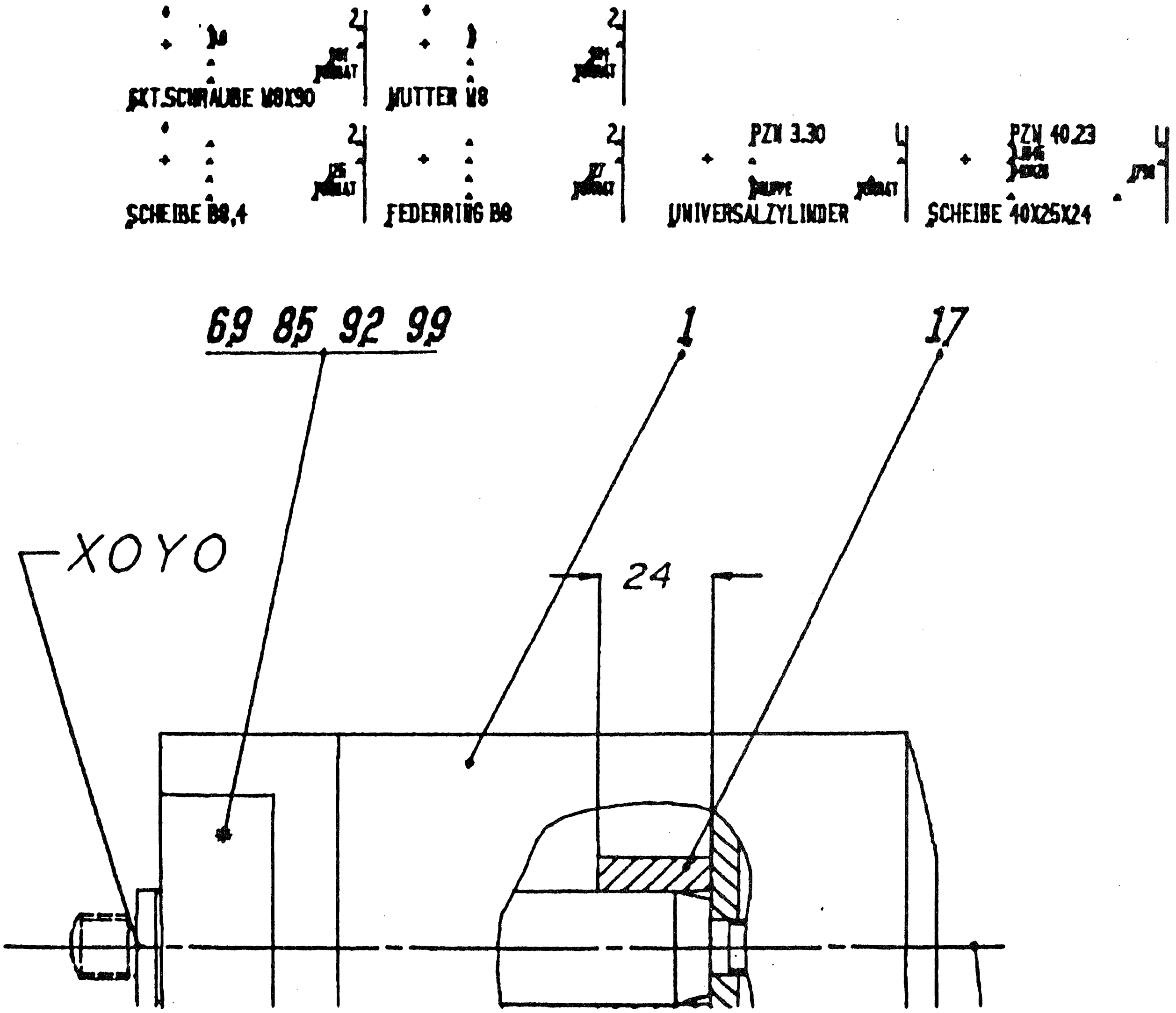
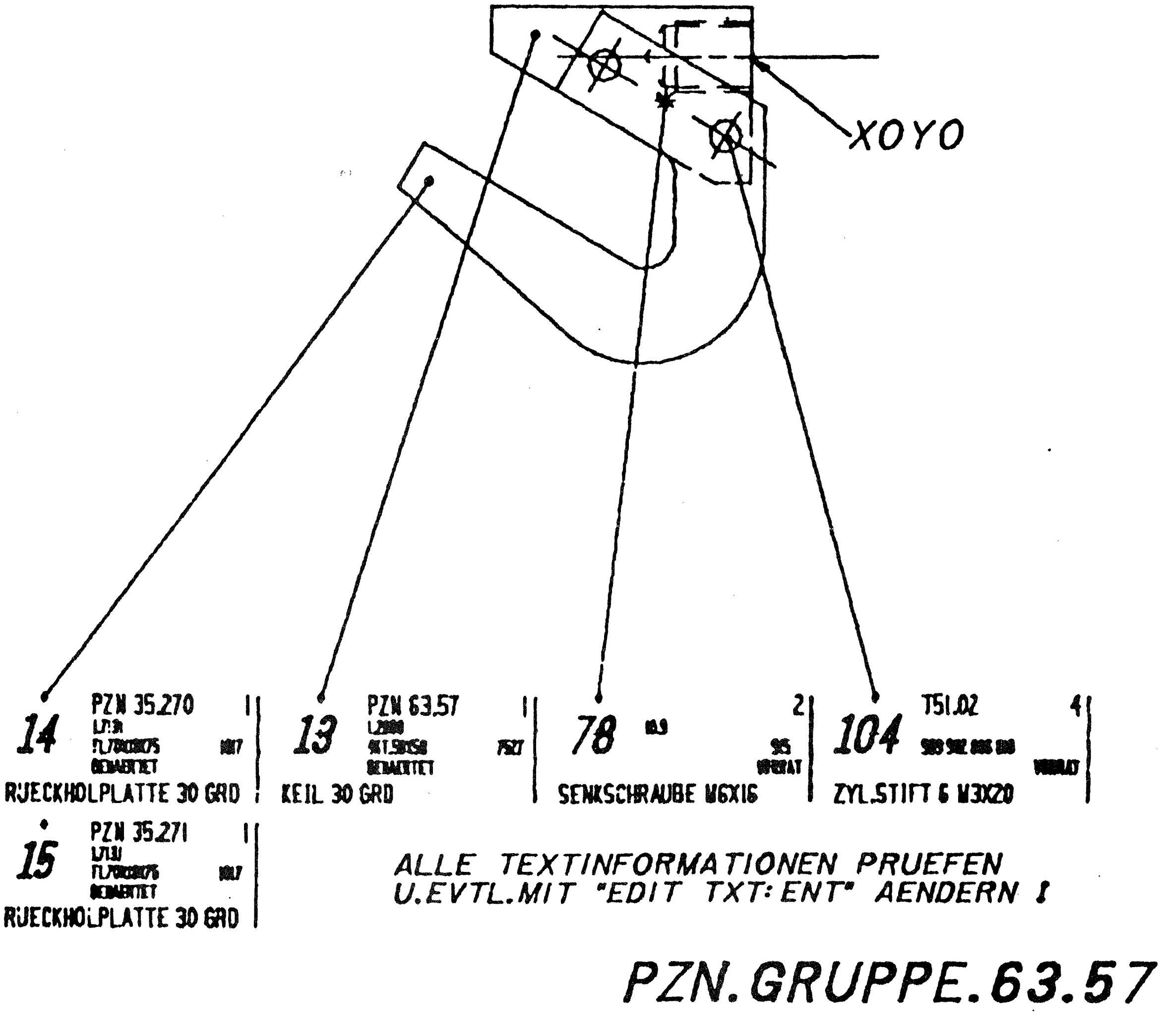

SKT.SCHRAUBE M8X90
MUTTER M8
SCHEIBE B8,4
FEDERRING B8
PZN 3.30
UNIVERSALZYLINDER
PZN 40.23
SCHEIBE 40X25X24
69 85 92 99
1
17
XOYO
24

Figure 4



XOYO
14
PZN 35.270
1
RUECKHOLPLATTE 30 GRD
13
PZN 63.57
1
KEIL 30 GRD
78
2
SENKSCHRAUBE M6X16
104
T51.02
4
ZYL.STIFT 6 M3X20
15
PZN 35.271
1
RUECKHOLPLATTE 30 GRD
ALLE TEXTINFORMATIONEN PRUEFEN
U.EVTL.MIT "EDIT TXT:ENT" AENDERN !
PZN.GRUPPE.63.57

Figure 5

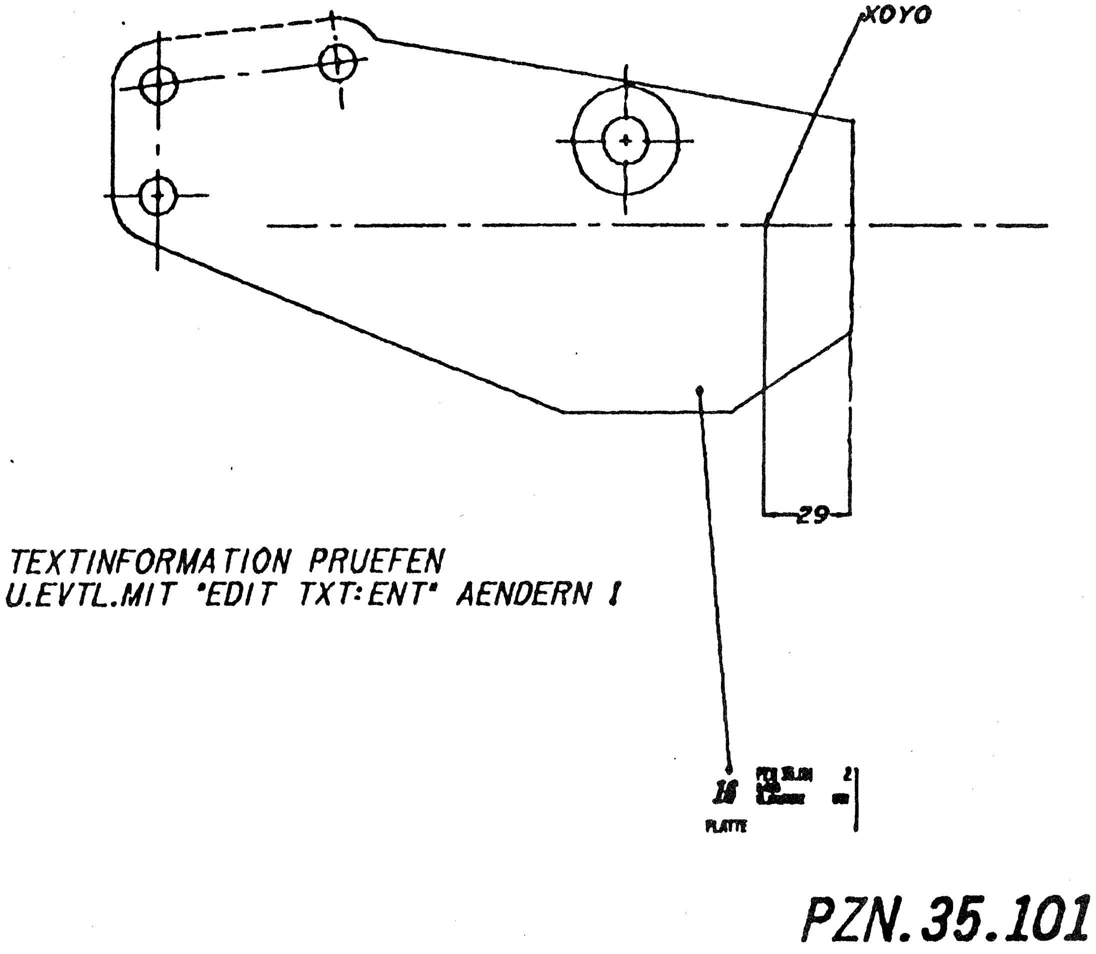

XOYO
29
TEXTINFORMATION PRUEFEN
U.EVTL.MIT 'EDIT TXT:ENT' AENDERN !
16
2
PLATTE
PZN.35.101

Figure 6

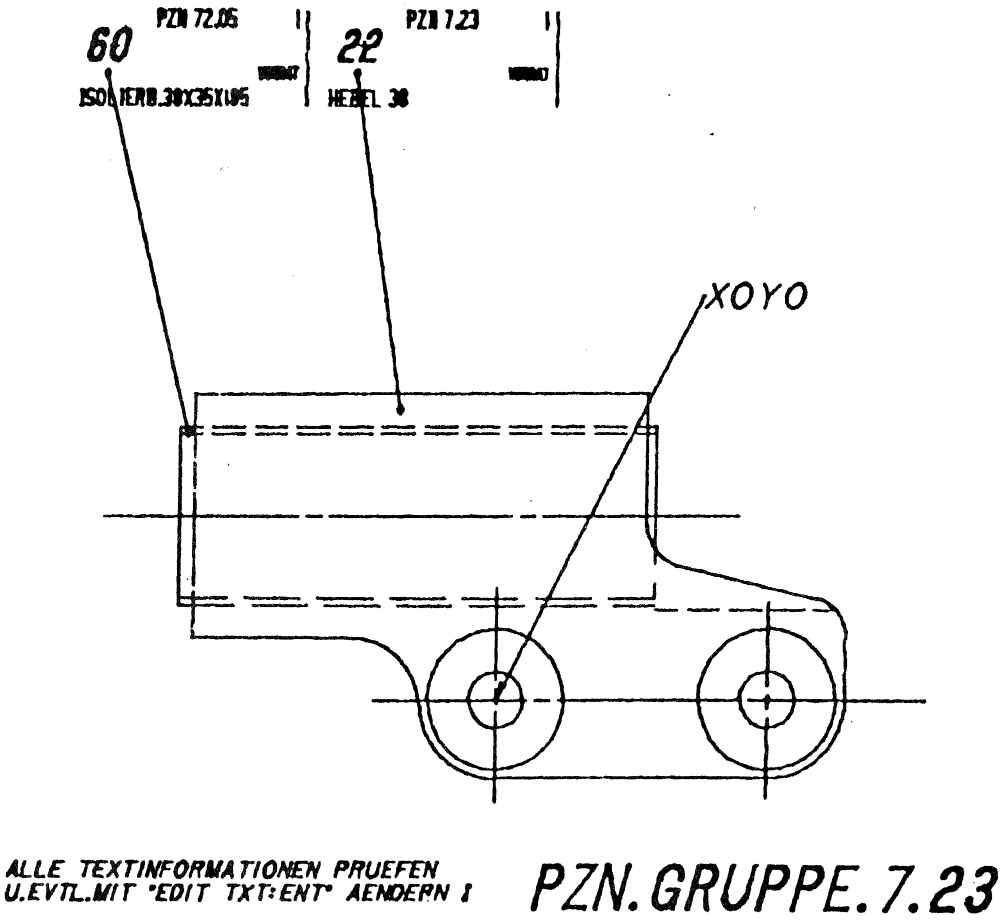

PZN 72.05
60
ISOLIERB.30X35X105
PZN 7.23
22
HEBEL 38
XOYO
ALLE TEXTINFORMATIONEN PRUEFEN
U.EVTL.MIT 'EDIT TXT:ENT' AENDERN !
PZN.GRUPPE.7.23

Figure 7

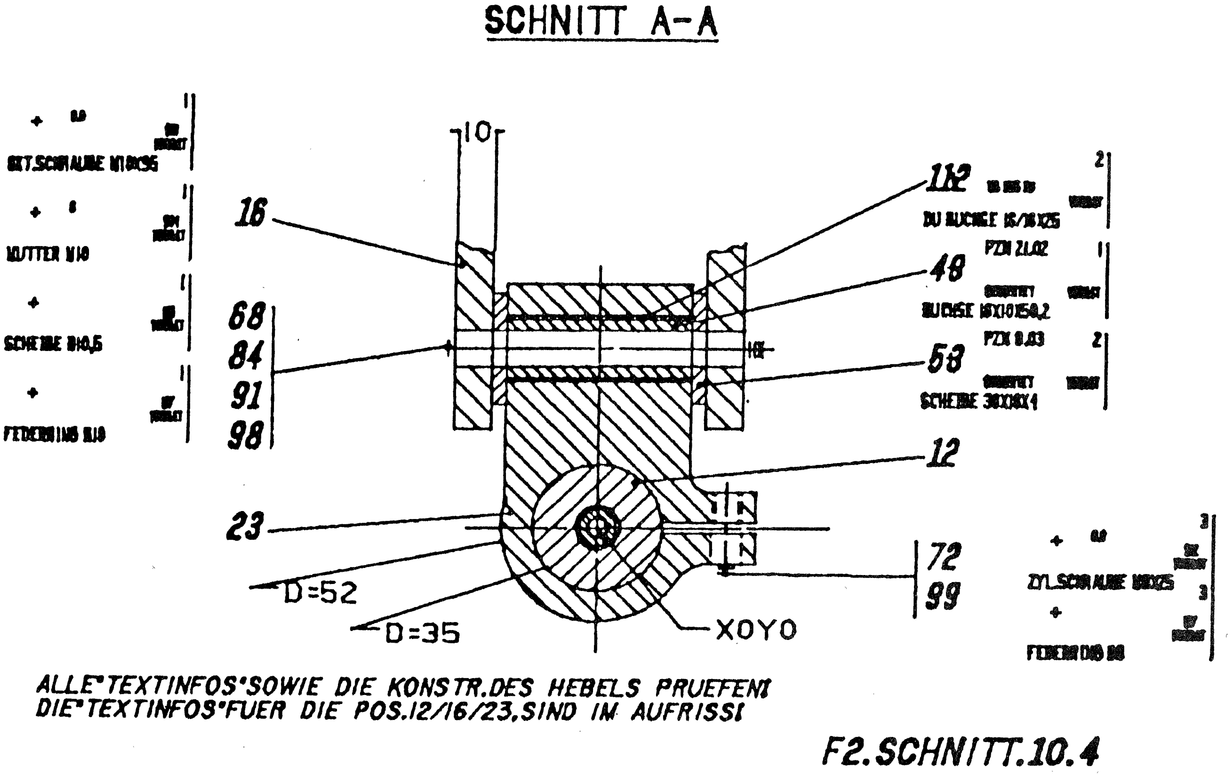


Figure 8

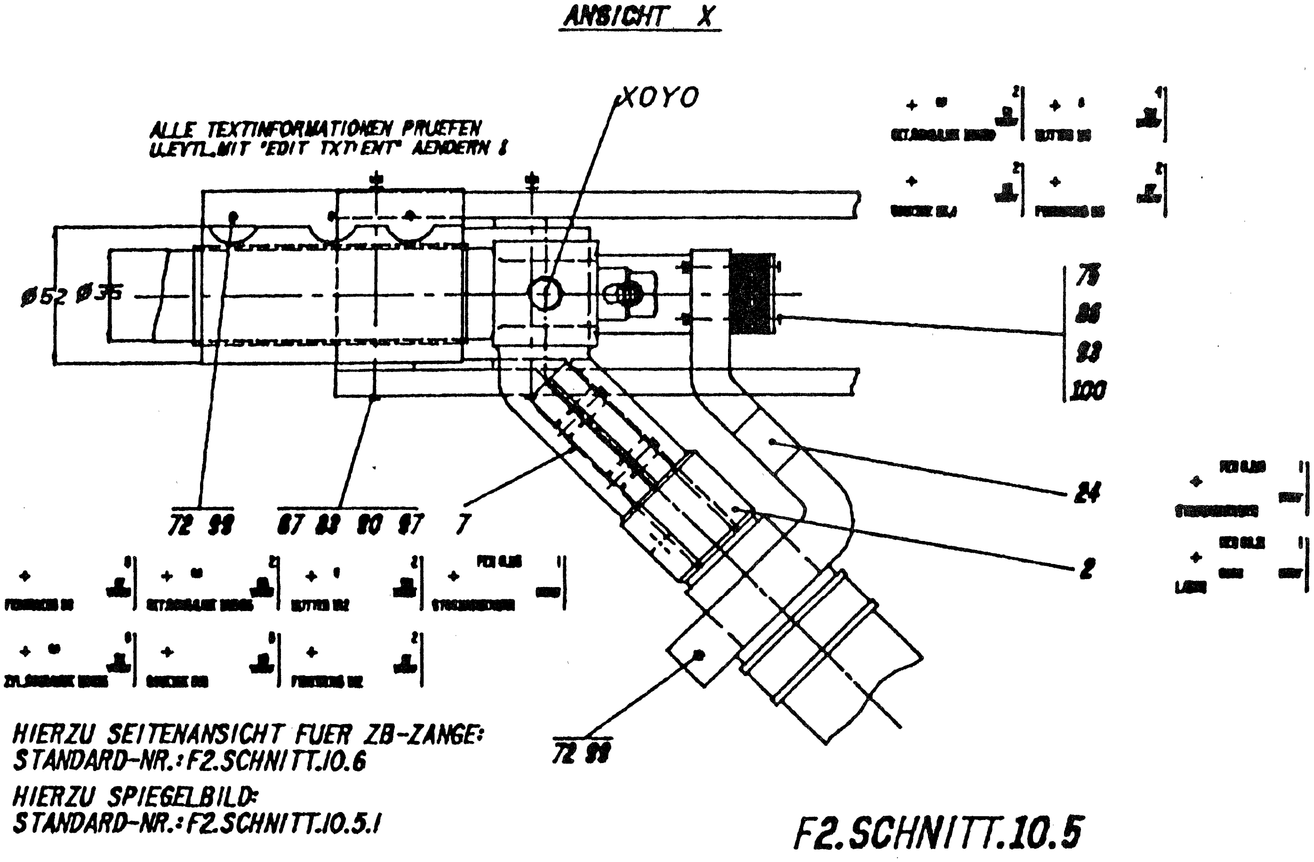

ANSICHT X
XOYO
ALLE TEXTINFORMATIONEN PRUEFEN
U.EVTL.MIT 'EDIT TXT ENT' AENDERN !
ø52
ø35
75
86
93
100
24
2
72 99
87 88 90 97
7
72 99
MUTTER
FEDERRING
HIERZU SEITENANSICHT FUER ZB-ZANGE:
STANDARD-NR.:F2.SCHNITT.10.6
HIERZU SPIEGELBILD:
STANDARD-NR.:F2.SCHNITT.10.5.1
F2.SCHNITT.10.5

Figure 9

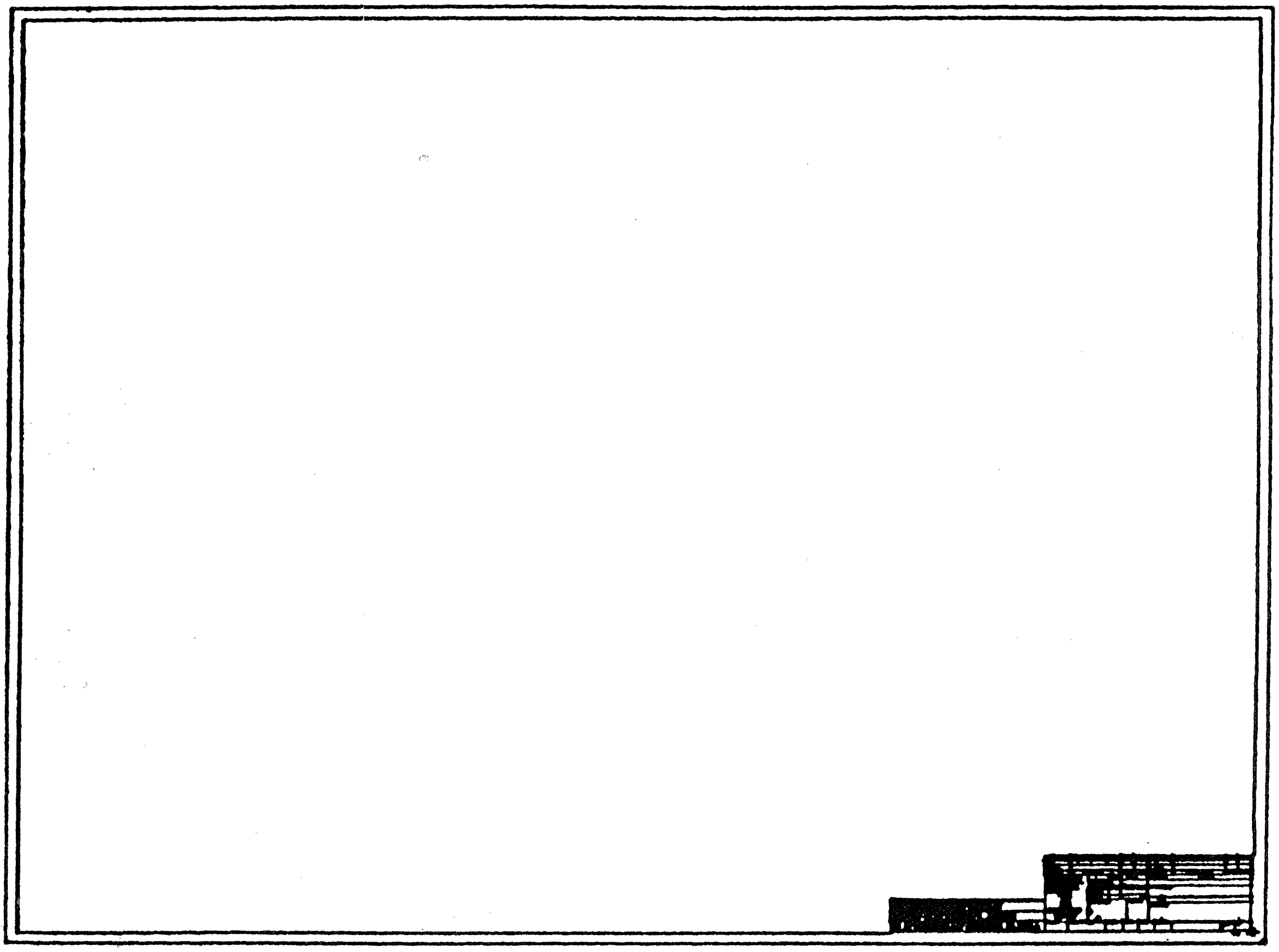


Figure 10

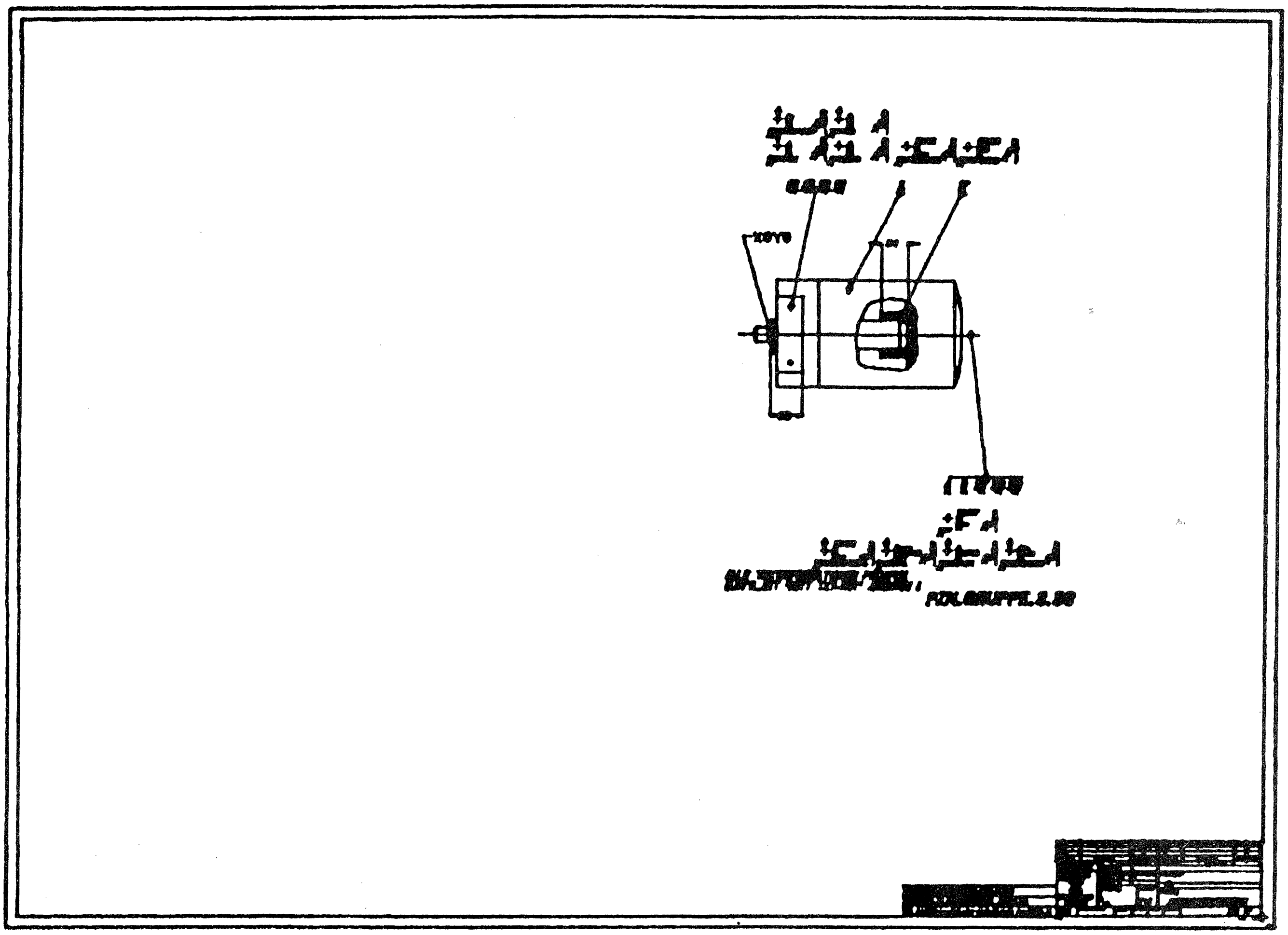


Figure 11

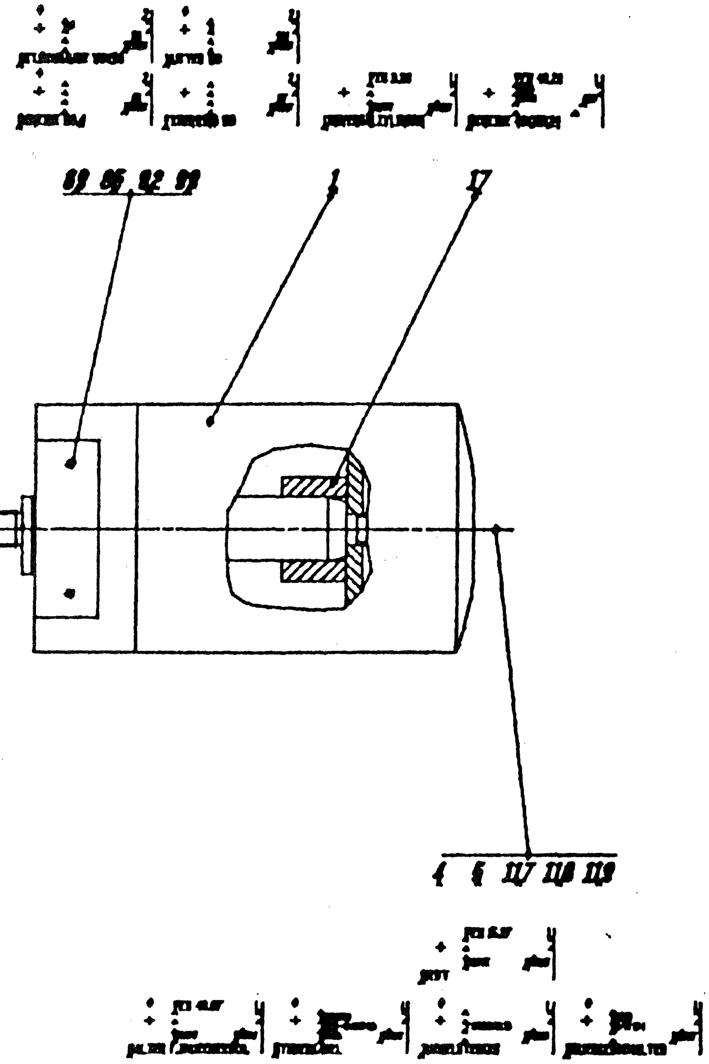


Figure 12

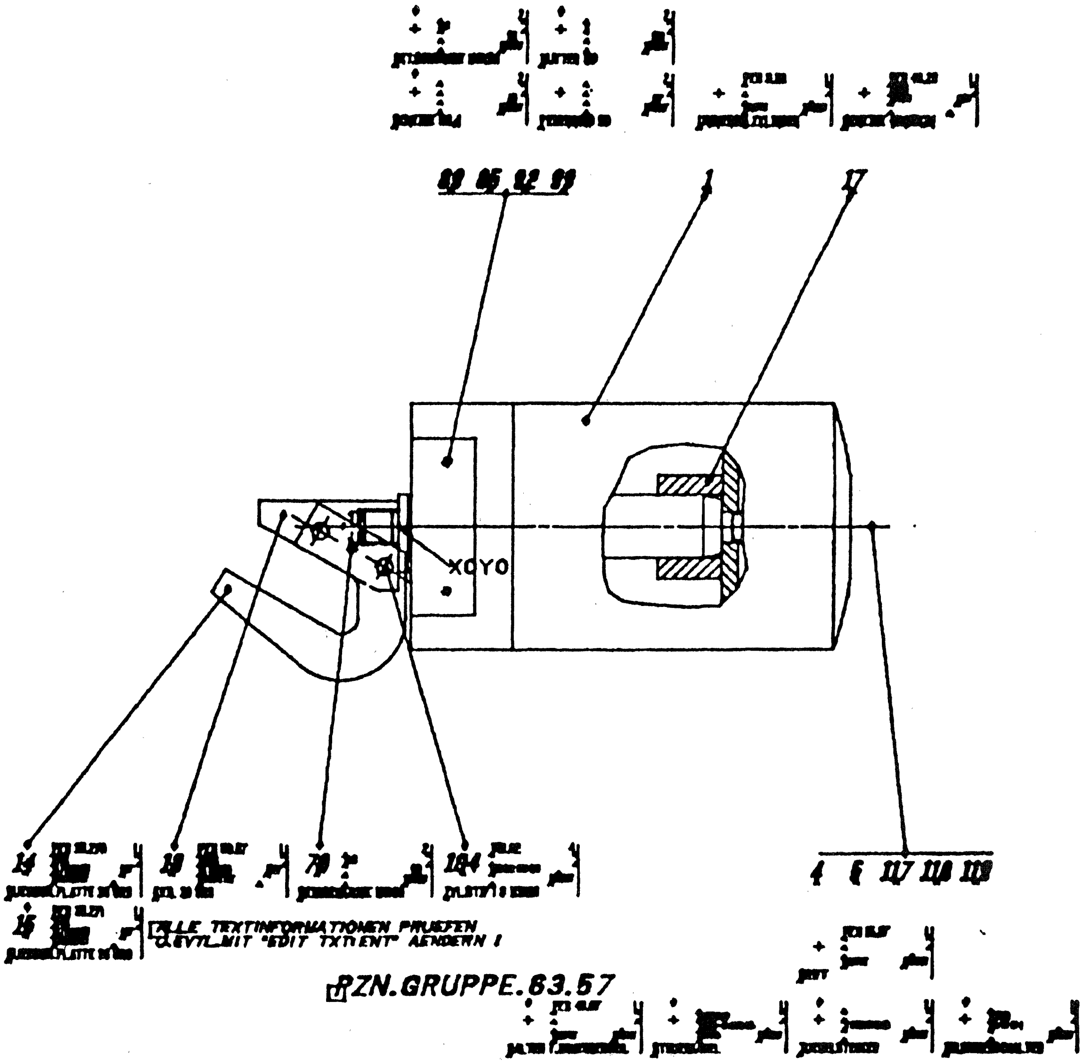

1
17
XOYO
14
18
79
164
15
4 6 117 118 119
ALLE TEXTINFORMATIONEN PRUEFEN
EVTL MIT "EDIT TXTIENT" AENDERN !
RZN.GRUPPE.63.57

Figure 13

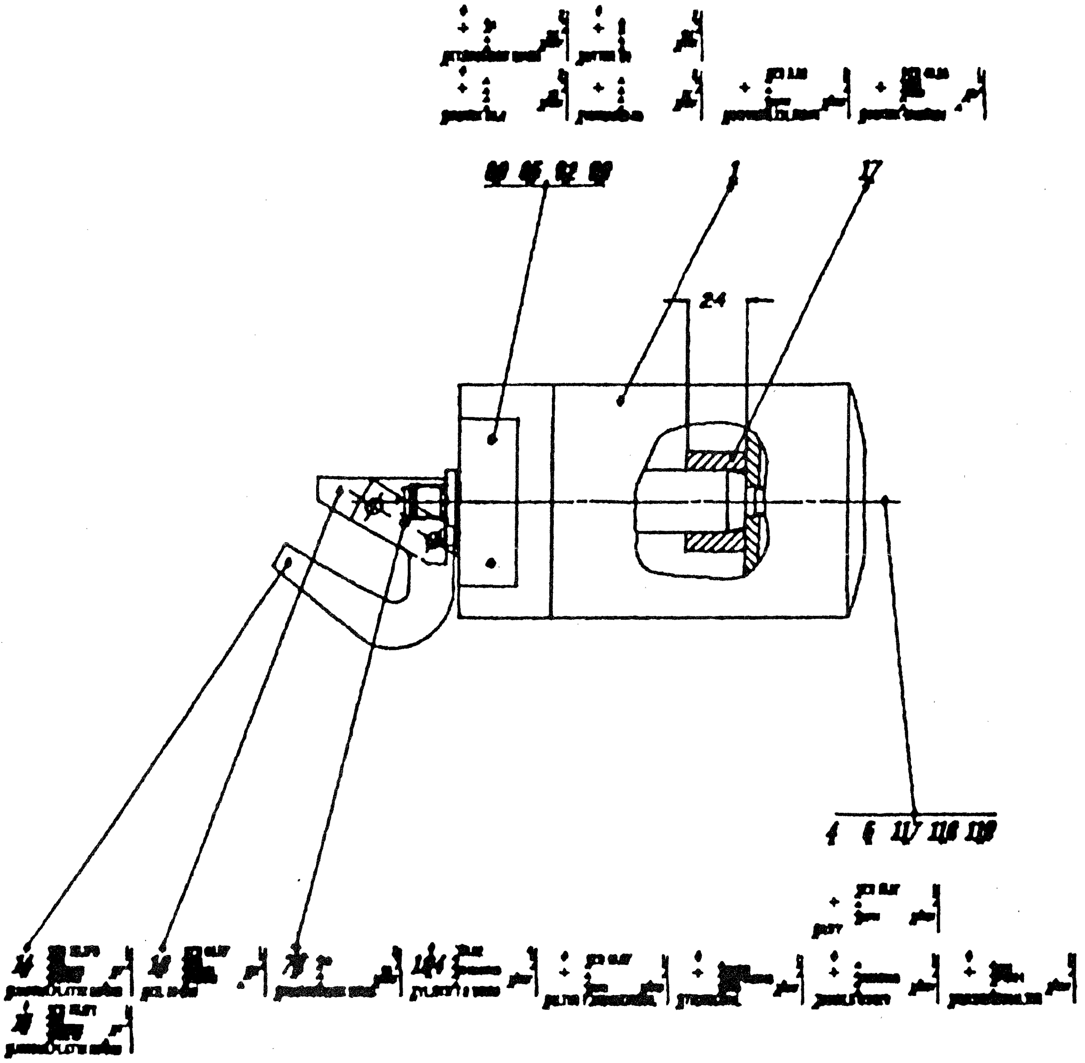


Figure 14

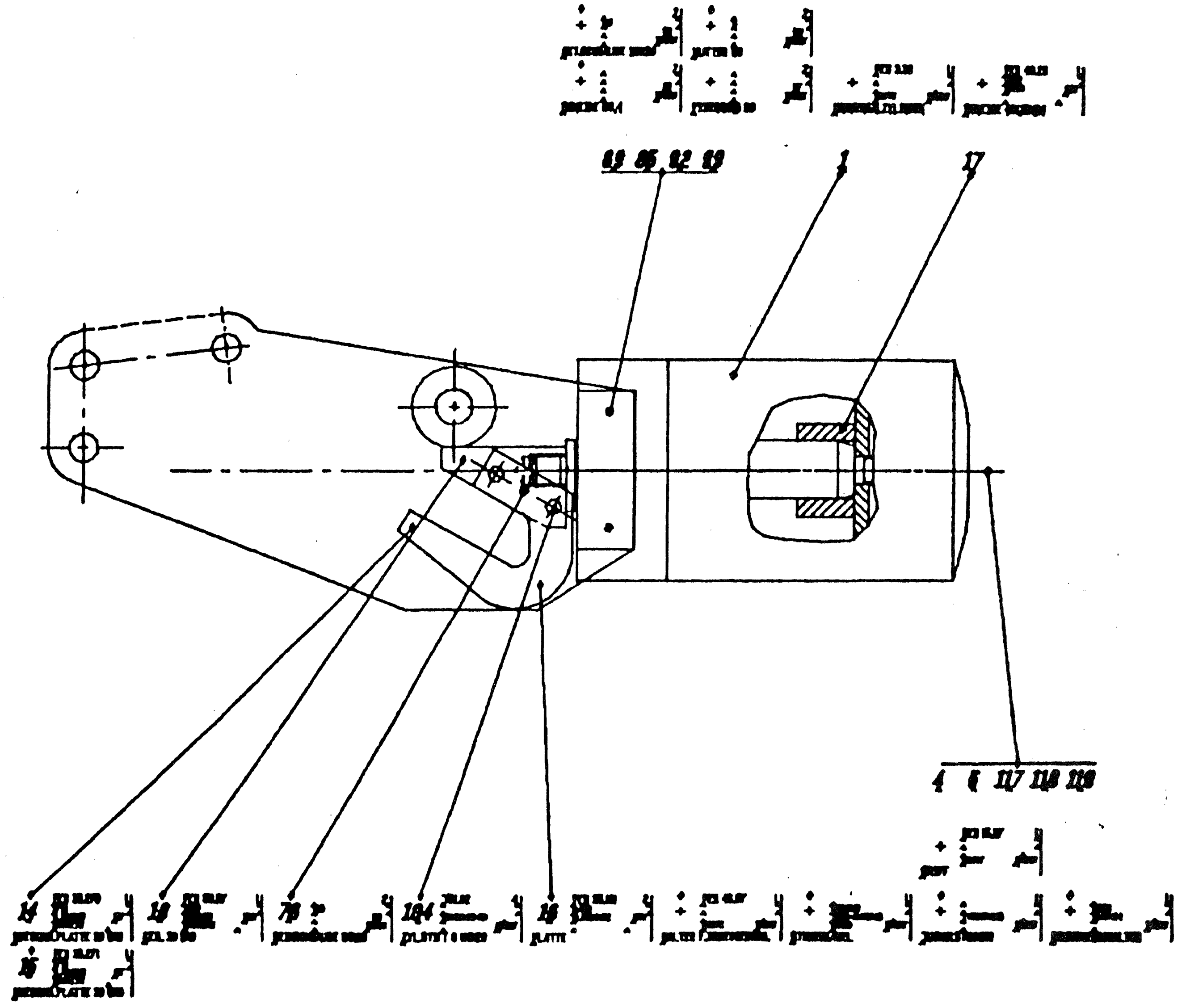


Figure 15

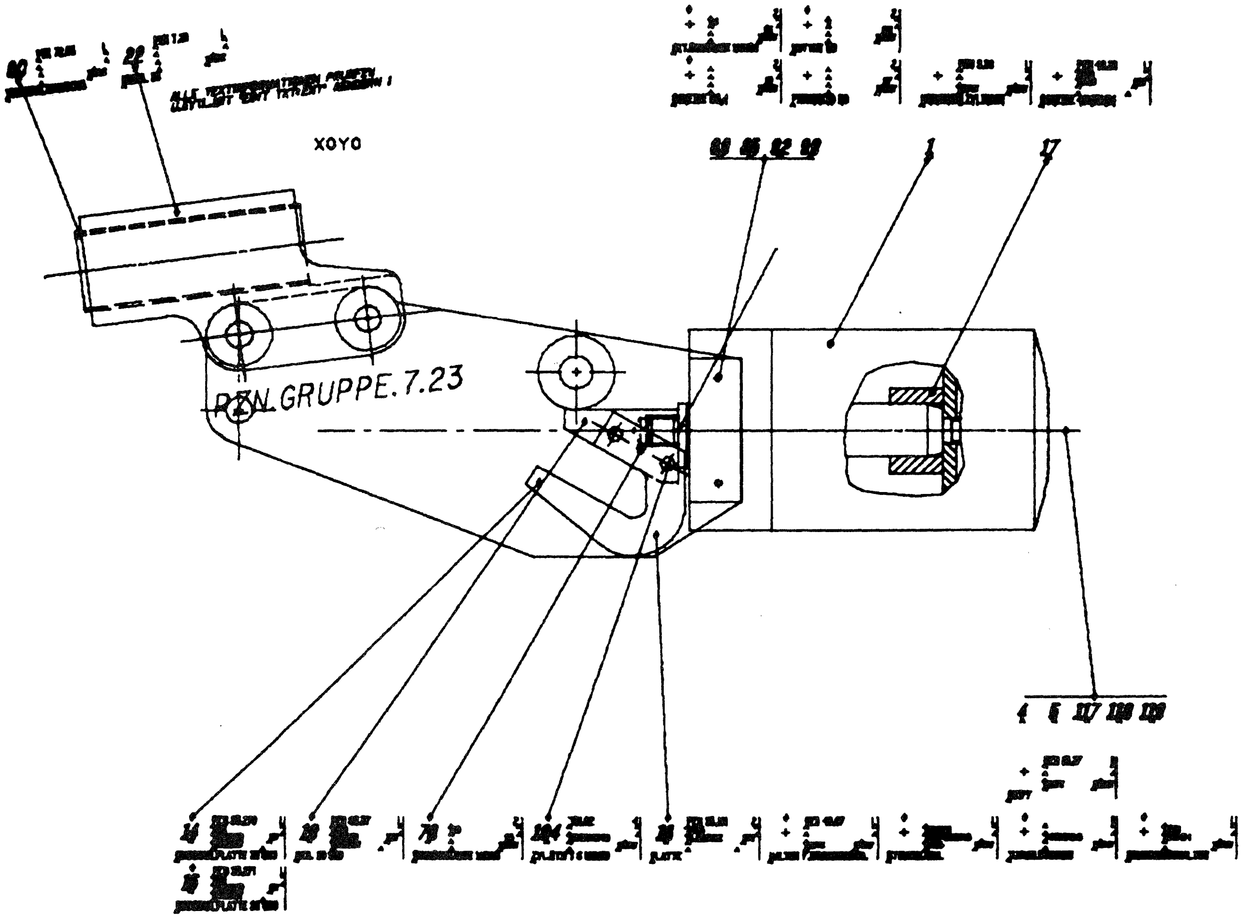


Figure 16

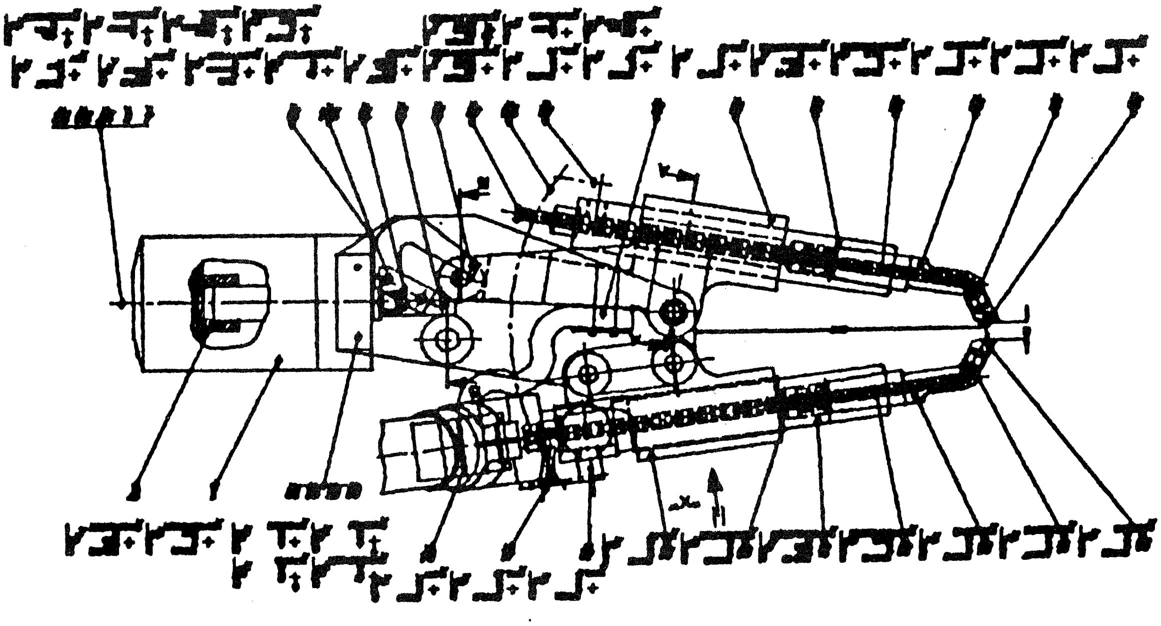


Figure 17

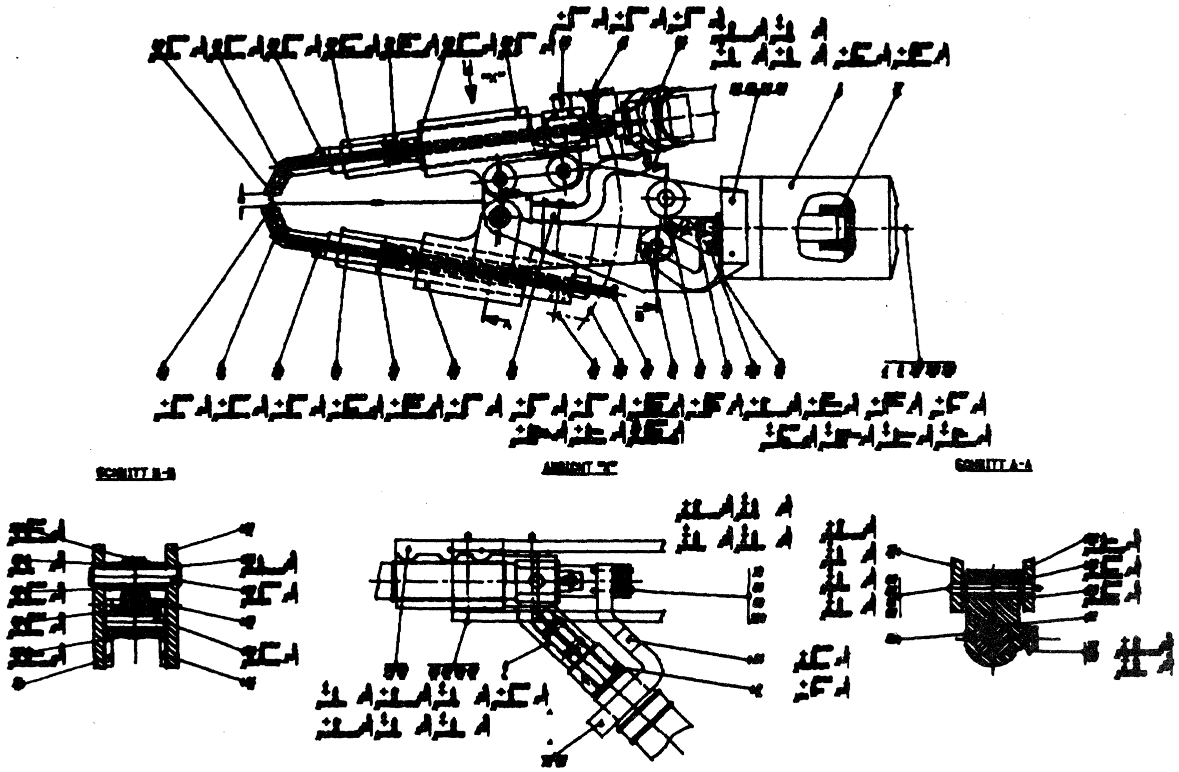
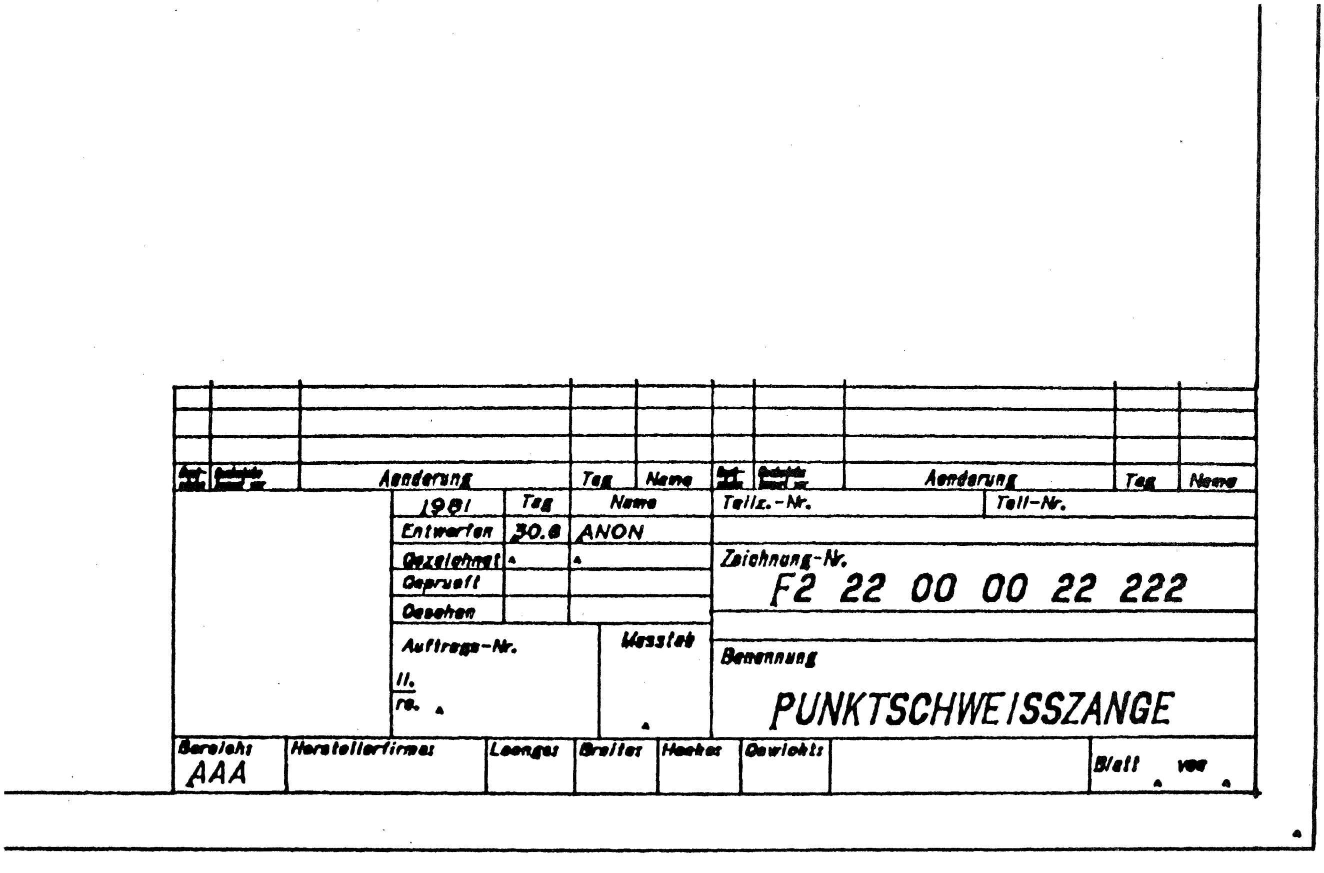


Figure 18



Aenderung	Tag	Name	Aenderung	Tag	Name

1981	Tag	Name
Entwerfen	30.8	ANON
Gezeichnet		
Geprueft		
Gesehen		

Teilz.-Nr.

Teil-Nr.

Zeichnung-Nr.
F2 22 00 00 22 222

Auftrags-Nr.
II.
re.

Masstab

Benennung
PUNKTSCHWEISSZANGE

Bereich:	Herstellerfirma:	Laenge:	Breite:	Hoehe:	Gewicht:		Blatt von
AAA							

Figure 19

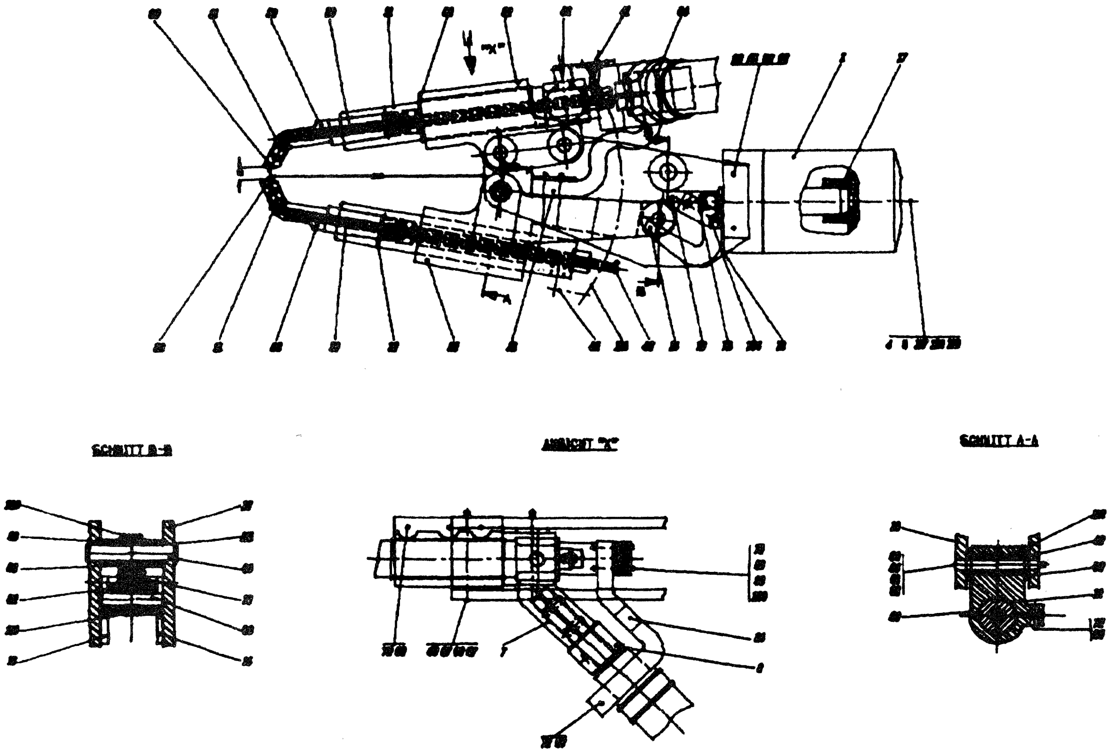


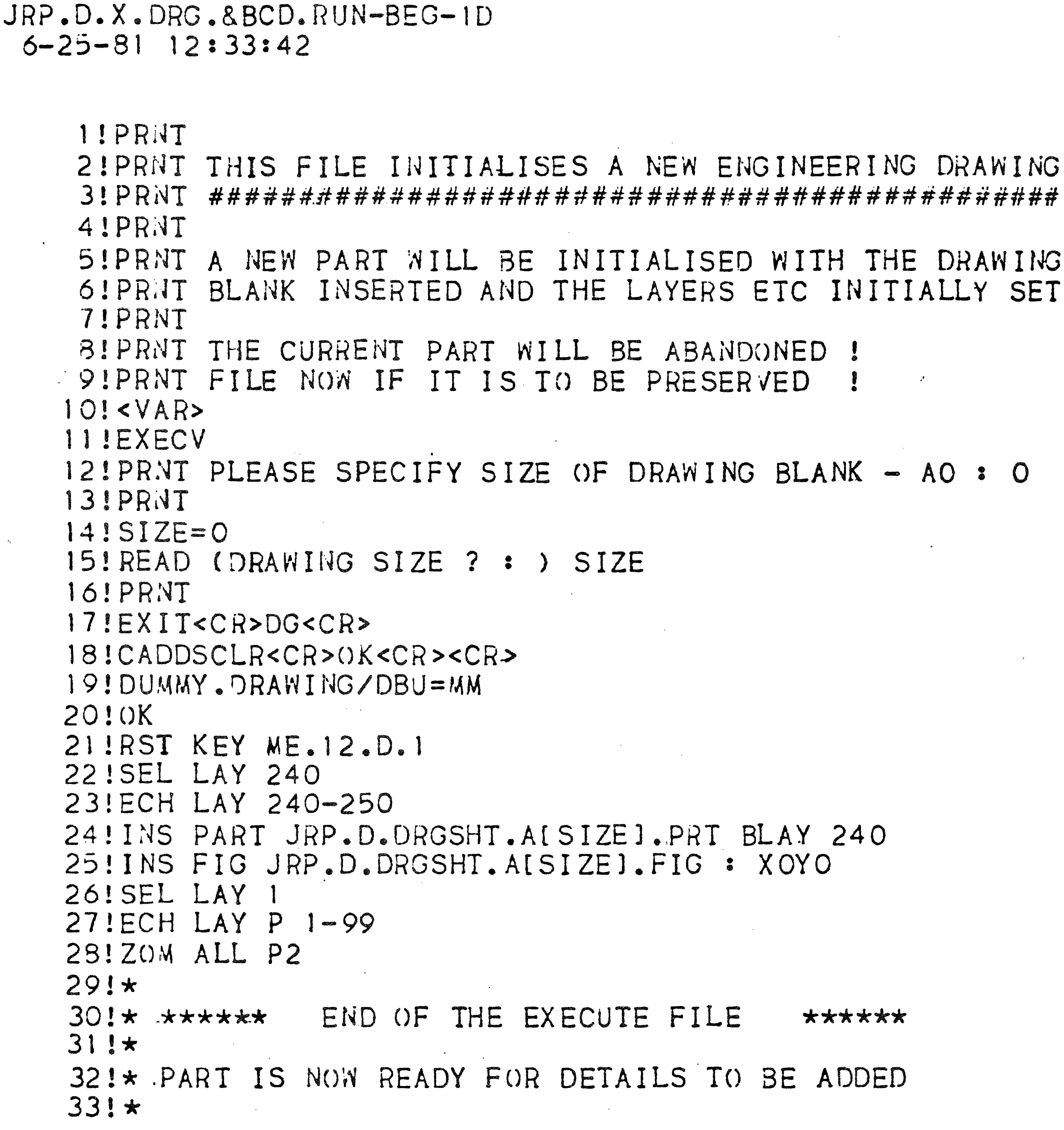
Figure 20

```
JOP.D.TEST.22222.&BCD.BOM
7- 1-81 19:31:50

!
!
!---------------------------------------------------------------------------------------------------------------!
!                                                                                                               !
!                                                                                                               !
!                                                                                                               !
! !------------------------------------------------------------------------------------------------------------!
! !                       !     STUECKLISTE INDIREKTER BEREICH      ! KA !         ! WA-NR              ! BLATT  1  VON  4 !
! !                       !     ==============================      !    !         !                    !                  !
! !------------------------------------------------------------------------------------------------------------!
! ! ZEICHNUNGS-NR / BAUMUSTER ! BG  UG ! SACHBENENNUNG                  ! ABT       NAME              ! DATUM     !
! ! F2 22 00 00 22 222        !        ! PUNKTSCHWEISSZANGE             ! AAA       ANON              ! 30-JUN-81 !
! !------------------------------------------------------------------------------------------------------------!
! ! KA NR      ST   BENENNUNG              NORM-NUMMER   WERKST   ROHMASS / TYPE     DIN  STCKZ / BEMERKUNG / LAGER !
! ! -- --      --   ---------              -----------   ------   -------   ----     ---  -----   ---------   ----- !
! !     1       1  UNIVERSALZYLINDER       PZN 3.30                                       GRUPPE               VORRAT
! !
! !     2       1  LAGER                   PZN 60.21                                      GRUPPE               VORRAT
! !
! !     4       1  GRIFF                   PZN 15.27                                      GRUPPE               VORRAT
! !
! !     5       1  HALTER F.DRUECKERSCH.   PZN 46.07                                      GRUPPE               VORRAT
! !
! !     7       1  STROMABNEHMER           PZN 6.115                                                           VORRAT
! !
! !    11       1  ELEKTRODENH. D35X230    PZN 5.422     2.1293   D35X233           1756
! !
! !    12       1  ELEKTRODENHALTER 222    PZN 5.401     2.1293   R36X12X225        1754
! !
! !    13       1  KEIL 30 GRD             PZN 63.57     1.2080   4KT.50X50         7527  GEHAERTET
! !
! !    14       1  RUECKHOLPLATTE 30 GRD   PZN 35.270    1.7131   FL70X10X75        1017  GEHAERTET
! !
! !    15       1  RUECKHOLPLATTE 30 GRD,  PZN 35.271    1.7131   FL70X10X75        1017  GEHAERTET
! !
! !    16       2  PLATTE                  PZN 35.101    3.4335   BL.10X115X262     1793
! !
! !    17       1  SCHEIBE 40X25X24        PZN 40.23     3.1645   D40X26            1798
! !
! !    21       1  KUPFERBAND                            2.0060   FL30X0,15X215     1791  100STCK CA0,9KG      VORRAT
! !
! !    22       1  HEBEL 38                PZN 7.23                                                            VORRAT
! !
! !    23       1  HEBEL 52                PZN 7.39                                                            VORRAT
! !
! !    24       1  STROMABNEHMER           PZN 6.218                                                           VORRAT
! !
! !    29       2  ELEKTRODE 125           PZN 2.140                                                           VORRAT
! !------------------------------------------------------------------------------------------------------------!
! ! GEPRUEFT     ! AEND.MITTLG.  !    !    !    !    !    !    !    !    !    !    !    !    !    !    !
! !              ! NUMMER/JAHR.  !    !    !    !    !    !    !    !    !    !    !    !    !    !    !
! !------------------------------------------------------------------------------------------------------------!
!---------------------------------------------------------------------------------------------------------------!
!
!
!
!
!
!
!
!
!---------------------------------------------------------------------------------------------------------------!
!                                                                                                               !
!                                                                                                               !
!                                                                                                               !
! !------------------------------------------------------------------------------------------------------------!
! !                       !     STUECKLISTE INDIREKTER BEREICH      ! KA !         ! WA-NR              ! BLATT  2  VON  4 !
! !                       !     ==============================      !    !         !                    !                  !
! !------------------------------------------------------------------------------------------------------------!
! ! ZEICHNUNGS-NR / BAUMUSTER ! BG  UG ! SACHBENENNUNG                  ! ABT       NAME              ! DATUM     !
! ! F2 22 00 00 22 222        !        ! PUNKTSCHWEISSZANGE             ! AAA       ANON              ! 30-JUN-81 !
! !------------------------------------------------------------------------------------------------------------!
! ! KA NR      ST   BENENNUNG              NORM-NUMMER   WERKST   ROHMASS / TYPE     DIN  STCKZ / BEMERKUNG / LAGER !
! ! -- --      --   ---------              -----------   ------   -------   ----     ---  -----   ---------   ----- !
! !    30       2  ELEKTRODE 25            PZN-2.650                                                           VORRAT
! !
! !    31       2  ELEKTR.SCHAFT 13        PZN 2.487                                                           VORRAT
! !
! !    36       2  WASSERZUFUEHRUNG 105    PZN 26.03                                      D1=6:D2=5            VORRAT
! !
! !    40       1  NIPPEL 170              PZN 20.13                                                           VORRAT
! !
! !    41       1  NIPPEL 190              PZN 20.46                                                           VORRAT
! !
! !    42       2  NIPPEL B                PZN 20.01                                      D=6                  VORRAT
! !
! !    49       1  BUCHSE 16X10X50,2       PZN 21.02                                      GEHAERTET            VORRAT
! !
! !    50       2  BUCHSE 25X15,1X17       PZN 21.13                                                           VORRAT
! !
! !    52       2  SCHEIBE 30X12X2         PZN 8.02                                                            VORRAT
! !
! !    53       2  SCHEIBE 30X10X4         PZN 8.03                                       GEHAERTET            VORRAT
! !
! !    54       2  PLATTE                  PZN 11.10                                                           VORRAT
! !
! !    58       1  ZYL.STIFT 12M6X57       PZN 51.42                                                           VORRAT
! !
! !    60       1  ISOLIERB.38X35X105      PZN 72.05                                                           VORRAT
! !
!    63      1  BOLZEN 15X85           VN 589/10                                                            VORRAT
!
!    67      2  6KT.SCHRAUBE M12X95                   8.8                           931                     VORRAT
!
!    58      1  6KT.SCHRAUBE M10X95                   8.8                           931                     VORRAT
!
!    69      2  6KT.SCHRAUBE M8X90                    8.8                           931                     VORRAT
!-------------------------------------------------------------------------------------------------------------------!
! GEPRUEFT    ! AEND.MITTLG.   !     !     !     !     !     !     !     !     !     !     !     !     !     !      !
!             ! NUMMER/JAHR.   !     !     !     !     !     !     !     !     !     !     !     !     !     !      !
!-------------------------------------------------------------------------------------------------------------------!
!--------------------------------------------------------------------------------------------------------------------!
!
!
!
!
!
!
!
!
!--------------------------------------------------------------------------------------------------------------------!
!                                                                                                                    !
!                                                                                                                    !
!                                                                                                                    !
!-------------------------------------------------------------------------------------------------------------------!
!                 !      STUECKLISTE INDIREKTER BEREICH      ! KA !        ! WA-NR                 ! BLATT  3  VON  4 !
!                 !      ==============================      !    !        !                       !                  !
!-------------------------------------------------------------------------------------------------------------------!
! ZEICHNUNGS-NR / BAUMUSTER ! BG  UG ! SACHBENENNUNG                     ! ART        NAME             ! DATUM      !
! F2 22 00 00 22 222        !        ! PUNKTSCHWEISSZANGE                ! AAA        ANON             ! 30-JUN-81  !
!-------------------------------------------------------------------------------------------------------------------!
! KA NR      ST     BENENNUNG             NORM-NUMMER    WERKST   ROHMASS / TYPE       DIN  STCKZ / BEMERKUNG / LAGER !
! -- --      --     ---------             -----------    ------   -------   ----       ---  -----   ---------   ----- !
!    72      9  ZYL.SCHRAUBE M8X25                    8.8                           912                     VORRAT
!
!    75      2  6KT.SCHRAUBE M6X50                    8.8                           931                     VORRAT
!
!    78      2  SENKSCHRAUBE M6X16                   10.9                          915                     VORRAT
!
!    33      2  MUTTER M12                            6                             934                     VORRAT
!
!    34      1  MUTTER M10                            6                             934                     VORRAT
!
!    85      2  MUTTER M8                             6                             934                     VORRAT
!
!    86      4  MUTTER M6                             6                             934                     VORRAT
!
!    89      2  SCHEIBE B15                                                         125                     VORRAT
!
!    90      6  SCHEIBE B13                                                         125                     VORRAT
!
!    91      1  SCHEIBE B10,5                                                       125                     VORRAT
!
!    92      2  SCHEIBE B8,4                                                        125                     VORRAT
!
!    93      2  SCHEIBE B6,4                                                        125                     VORRAT
!
!    97      2  FEDERRING B12                                                       127                     VORRAT
!
!    98      1  FEDERRING B10                                                       127                     VORRAT
!
!    99     11  FEDERRING B8                                                        127                     VORRAT
!
!   100      2  FEDERRING 36                                                        127                     VORRAT
!
!   104      4  ZYL.STIFT 6 M3X20      T51.02                  999 902 006 010                              VORRAT
!-------------------------------------------------------------------------------------------------------------------!
! GEPRUEFT    ! AEND.MITTLG.   !     !     !     !     !     !     !     !     !     !     !     !     !     !      !
!             ! NUMMER/JAHR.   !     !     !     !     !     !     !     !     !     !     !     !     !     !      !
!-------------------------------------------------------------------------------------------------------------------!
!--------------------------------------------------------------------------------------------------------------------!
!
!
!
!
!
!
!
!
!--------------------------------------------------------------------------------------------------------------------!
!                                                                                                                    !
!                                                                                                                    !
!                                                                                                                    !
!-------------------------------------------------------------------------------------------------------------------!
!                 !      STUECKLISTE INDIREKTER BEREICH      ! KA !        ! WA-NR                 ! BLATT  4  VON  4 !
!                 !      ==============================      !    !        !                       !                  !
!-------------------------------------------------------------------------------------------------------------------!
! ZEICHNUNGS-NR / BAUMUSTER ! BG  UG ! SACHBENENNUNG                     ! ART        NAME             ! DATUM      !
! F2 22 00 00 22 222        !        ! PUNKTSCHWEISSZANGE                ! AAA        ANON             ! 30-JUN-81  !
!-------------------------------------------------------------------------------------------------------------------!
! KA NR      ST     BENENNUNG             NORM-NUMMER    WERKST   ROHMASS / TYPE       DIN  STCKZ / BEMERKUNG / LAGER !
! -- --      --     ---------             -----------    ------   -------   ----       ---  -----   ---------   ----- !
!   109      1  STUETZROLLE 35X15X19   03.04.120      INA      NATR 15 PPX                                  VORRAT
!
!   110      1  STUETZROLLE 32X12X15                  INA      NATR 12 PPX                                  VORRAT
!
!   111      2  SICHERUNGSRING 15X1                                                 471                     VORRAT
!
!   112      2  DU BUCHSE 16/18X25                             MB 1625 DU                                   VORRAT
!
!   116      1  WASSERSCHLAUCH                                 D7X3,5X400                                   VORRAT
!
```

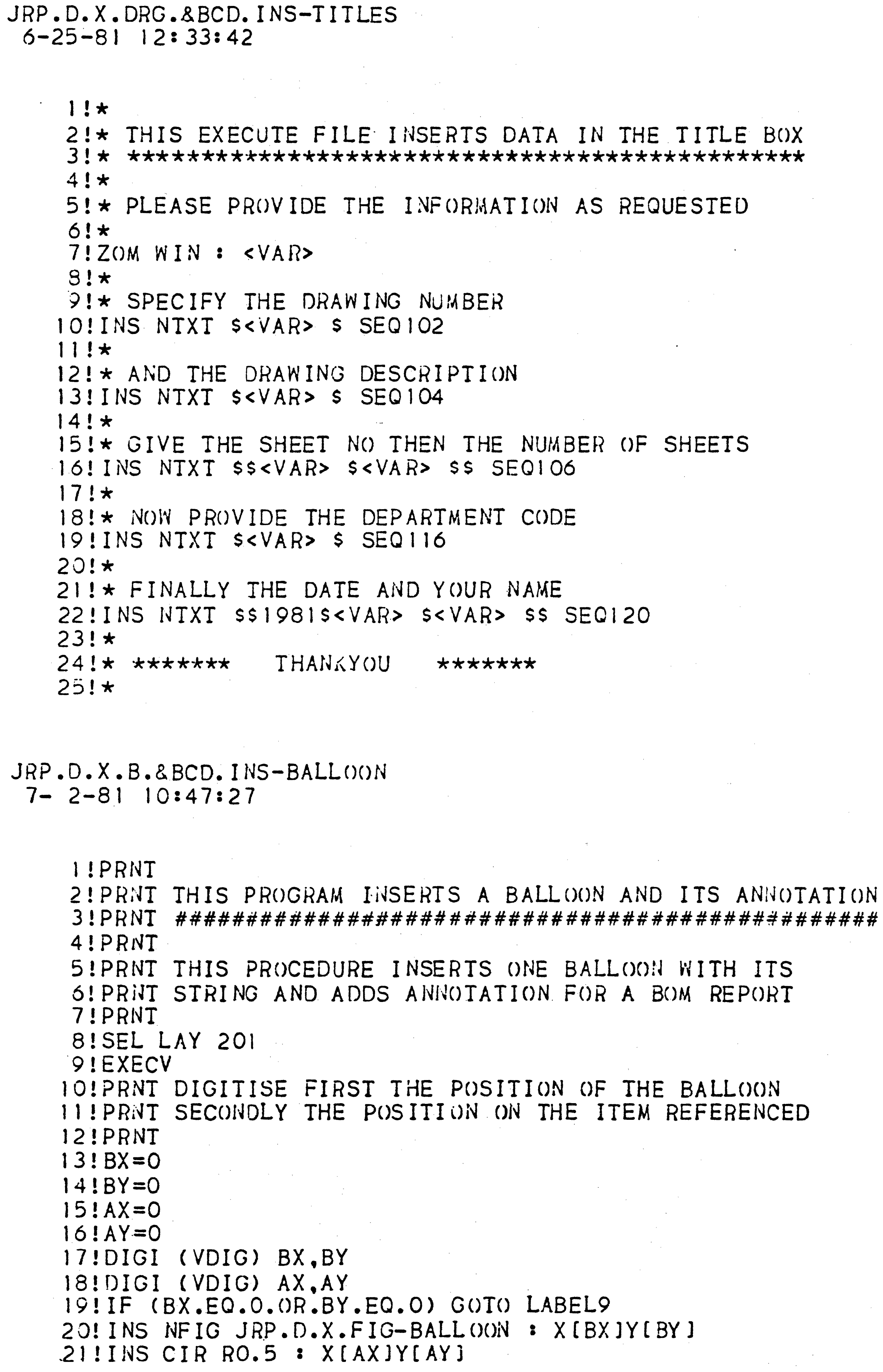
```
203! !  117     1  STEUERKABEL                          GUMMIKAB 1,5MM**2:4000 LG       2ADRIG               VORRAT
204! !
205! !  118     1  TUCHELSTECKER                                 T 3352:5POLIG                                 VORRAT
206! !
207! !  119     1  DRUECKERSCHALTER                     BOSCH     SH/TD 10-1                                    VORRAT
208! !
209! !
210! !
211! !
212! !
213! !
214! !
215! !
216! !
217! !
218! !
219! !
220! !
221! !
222! !
223! !
224! !
225! !
226! !---------------------------------------------------------------------------------------------------------------!
227! ! GEPRUEFT     ! AEND.MITTLG.   !     !     !     !     !     !     !     !     !     !     !     !     !     !     !
228! !              ! NUMMER/JAHR.   !     !     !     !     !     !     !     !     !     !     !     !     !     !     !
229! !---------------------------------------------------------------------------------------------------------------!
230!!-----------------------------------------------------------------------------------------------------------------!
231!!
232!!
```

APPENDIX 4
PROCEDURES PROGRAMS AND EXECUTE FILES

1. initialise a new engineering drawing.
2. insert data in the title box.
3. insert a balloon and its annotation.
4. insert a dummy balloon and a number.
5. insert balloons and text in 3D parts.
6. insert dummy balloons in 3D parts.
7. insert balloons and their strings.
8. provide annotation for the balloons.
9. run the BOM report procedures.
10. an EDF for MD applications.



```
JRP.D.X.DRG.&BCD.RUN-BEG-1D
 6-25-81 12:33:42


     1!PRNT
     2!PRNT THIS FILE INITIALISES A NEW ENGINEERING DRAWING
     3!PRNT ###############################################
     4!PRNT
     5!PRNT A NEW PART WILL BE INITIALISED WITH THE DRAWING
     6!PRNT BLANK INSERTED AND THE LAYERS ETC INITIALLY SET
     7!PRNT
     8!PRNT THE CURRENT PART WILL BE ABANDONED !
     9!PRNT FILE NOW IF IT IS TO BE PRESERVED  !
    10!<VAR>
    11!EXECV
    12!PRNT PLEASE SPECIFY SIZE OF DRAWING BLANK - AO : O
    13!PRNT
    14!SIZE=O
    15!READ (DRAWING SIZE ? : ) SIZE
    16!PRNT
    17!EXIT<CR>DG<CR>
    18!CADDSCLR<CR>OK<CR><CR>
    19!DUMMY.DRAWING/DBU=MM
    20!OK
    21!RST KEY ME.12.D.1
    22!SEL LAY 240
    23!ECH LAY 240-250
    24!INS PART JRP.D.DRGSHT.A[SIZE].PRT BLAY 240
    25!INS FIG JRP.D.DRGSHT.A[SIZE].FIG : XOYO
    26!SEL LAY 1
    27!ECH LAY P 1-99
    28!ZOM ALL P2
    29!*
    30!* ******   END OF THE EXECUTE FILE   ******
    31!*
    32!* PART IS NOW READY FOR DETAILS TO BE ADDED
    33!*
```



```
JRP.D.X.DRG.&BCD.INS-TITLES
 6-25-81 12:33:42


*
* THIS EXECUTE FILE INSERTS DATA IN THE TITLE BOX
* *********************************************
*
* PLEASE PROVIDE THE INFORMATION AS REQUESTED
*
ZOM WIN : <VAR>
*
* SPECIFY THE DRAWING NUMBER
INS NTXT $<VAR> $ SEQ102
*
* AND THE DRAWING DESCRIPTION
INS NTXT $<VAR> $ SEQ104
*
* GIVE THE SHEET NO THEN THE NUMBER OF SHEETS
INS NTXT $$<VAR> $<VAR> $$ SEQ106
*
* NOW PROVIDE THE DEPARTMENT CODE
INS NTXT $<VAR> $ SEQ116
*
* FINALLY THE DATE AND YOUR NAME
INS NTXT $$1981$<VAR> $<VAR> $$ SEQ120
*
* *******   THANKYOU   *******
*
```

```
JRP.D.X.B.&BCD.INS-BALLOON
 7- 2-81 10:47:27


PRNT
PRNT THIS PROGRAM INSERTS A BALLOON AND ITS ANNOTATION
PRNT ################################################
PRNT
PRNT THIS PROCEDURE INSERTS ONE BALLOON WITH ITS
PRNT STRING AND ADDS ANNOTATION FOR A BOM REPORT
PRNT
SEL LAY 201
EXECV
PRNT DIGITISE FIRST THE POSITION OF THE BALLOON
PRNT SECONDLY THE POSITION ON THE ITEM REFERENCED
PRNT
BX=0
BY=0
AX=0
AY=0
DIGI (VDIG) BX,BY
DIGI (VDIG) AX,AY
IF (BX.EQ.0.OR.BY.EQ.0) GOTO LABEL9
INS NFIG JRP.D.X.FIG-BALLOON : X[BX]Y[BY]
INS CIR R0.5 : X[AX]Y[AY]
```

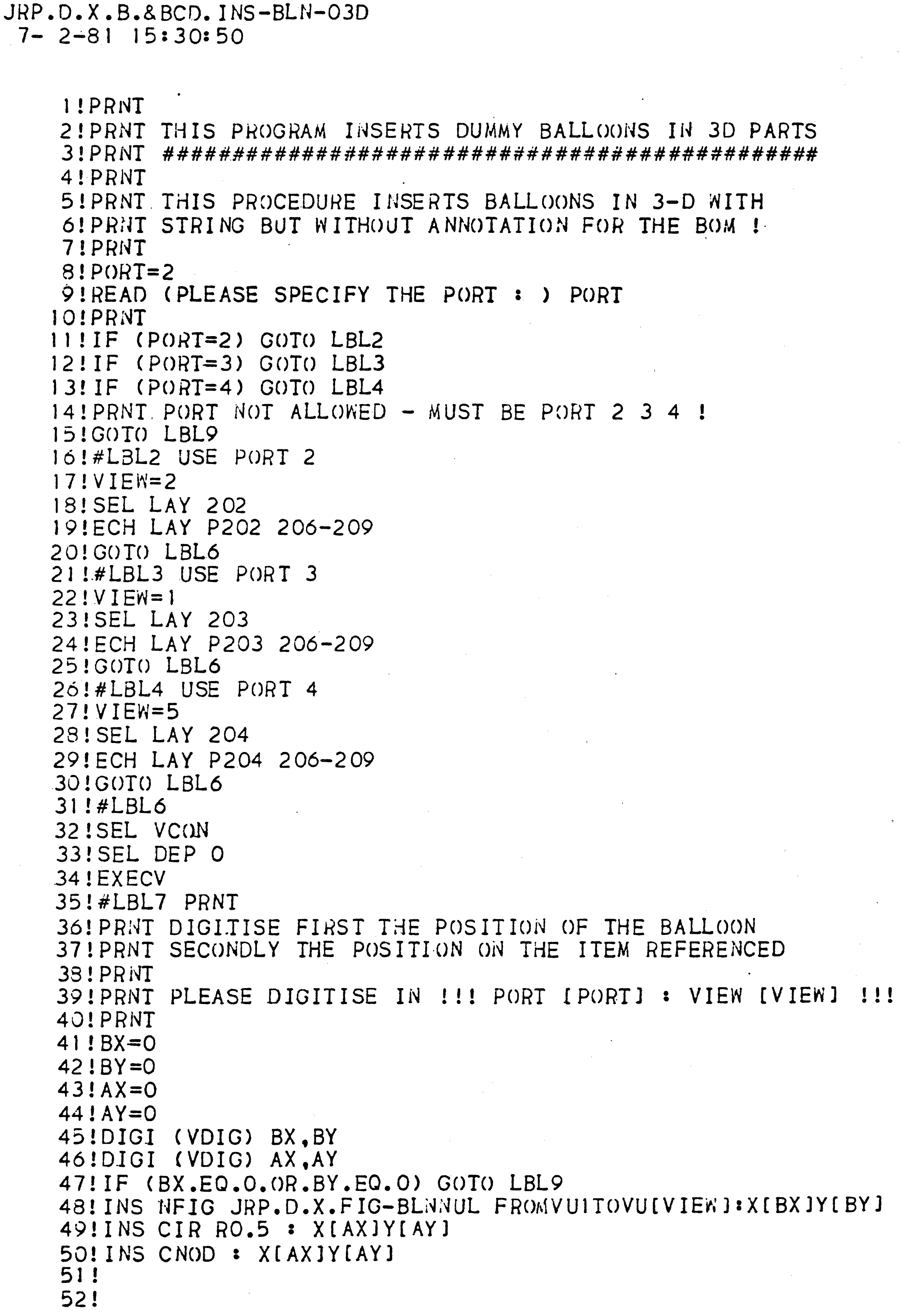
```
22!INS CNOD : X[AX]Y[AY]
23!INS NLIN : X[BX]Y[BY],X[AX]Y[AY]
24!<CR>
25!EXECV
26!PRNT THIS SECTION PROVIDES ANNOTATION FOR THE BALLOONS
27!PRNT
28!PRNT NOTE :- PLEASE FOLLOW THE INSTRUCTIONS CAREFULLY !
29!PRNT TEXT MUST NOT EXCEED STATED NUMBER OF CHARACTERS !
30!PRNT
31!PRNT CHARACTER 1-20 : 12345678901234567890
32!READ (ENTER POS  - 3 : ) &POS
33!READ (BENENNUNG - 21 : ) &BEN
34!READ (NORM NR   - 14 : ) &NRM
35!READ (STUECK AUS - 3 : ) &STK
36!READ (WERKSTOFF  - 8 : ) &WRK
37!READ (ROHMASS   - 18 : ) &RHM
38!READ (DIN NUMMER - 5 : ) &DIN
39!READ (BEMERKUNG - 16 : ) &BMK
40!READ (VORRAT ?   - 6 : ) &LGR
41!INS NTXT <*
42!$$ [&POS] $[&BEN] $[&NRM] $[&STK] $<*
43![&WRK] $[&RHM] $[&DIN] $[&BMK] $[&LGR] $$ <*
44!NFIG : X[BX]Y[BY]
45!#LABEL9 *
46!* ****** END OF VARPRO PROGRAM ******
47!*
```

```
JRP.D.X.B.&BCD.INS-BLN-NUL
 7- 2-81 10:48:11
```

```
 1!PRNT
 2!PRNT THIS PROGRAM INSERTS A DUMMY BALLOON AND A NUMBER
 3!PRNT ##############################################
 4!PRNT
 5!PRNT THIS PROCEDURE INSERTS ONE BALLOON WITH ITS
 6!PRNT STRING BUT WITHOUT ANNOTATION FOR THE BOM !
 7!PRNT
 8!SEL LAY 201
 9!EXECV
10!PRNT DIGITISE FIRST THE POSITION OF THE BALLOON
11!PRNT SECONDLY THE POSITION ON THE ITEM REFERENCED
12!PRNT
13!BX=0
14!BY=0
15!AX=0
16!AY=0
17!DIGI (VDIG) BX,BY
18!DIGI (VDIG) AX,AY
19!IF (BX.EQ.0.OR.BY.EQ.0) GOTO LABEL9
20!INS NFIG JRP.D.X.FIG-BLNNULL : X[BX]Y[BY]
21!INS CIR R0.5 : X[AX]Y[AY]
22!INS CNOD : X[AX]Y[AY]
23!INS NLIN : X[BX]Y[BY],X[AX]Y[AY]
24!<CR>
```

```
   25!EXECV
   26!PRNT THIS SECTION PROVIDES ANNOTATION FOR THE BALLOONS
   27!PRNT
   28!PRNT NOTE :- PLEASE FOLLOW THE INSTRUCTIONS CAREFULLY !
   29!PRNT TEXT MUST NOT EXCEED STATED NUMBER OF CHARACTERS !
   30!PRNT
   31!READ (ENTER POS  - 3 : ) &POS
   32!INS NFIG $$ [&POS] $$ NFIG : X[BX]Y[BY]
   33!#LABEL9 *
   34!* ****** END OF VARPRO PROGRAM ******
   35!*


JRP.D.X.B.&BCD.INS-BLN-3-D
 7- 2-81 15:29:53


    1!PRNT
    2!PRNT THIS PROGRAM INSERTS BALLOONS & TEXT IN 3D PARTS
    3!PRNT ###############################################
    4!PRNT
    5!PRNT THIS PROCEDURE INSERTS BALLOONS IN 3-D WITH
    6!PRNT STRING AND ADDS ANNOTATION FOR A BOM REPORT
    7!PRNT
    8!PORT=2
    9!READ (PLEASE SPECIFY THE PORT : ) PORT
   10!PRNT
   11!IF (PORT=2) GOTO LBL2
   12!IF (PORT=3) GOTO LBL3
   13!IF (PORT=4) GOTO LBL4
   14!PRNT PORT NOT ALLOWED - MUST BE PORT 2 3 4 !
   15!GOTO LBL9
   16!#LBL2 USE PORT 2
   17!VIEW=2
   18!SEL LAY 202
   19!ECH LAY P202 206-209
   20!GOTO LBL6
   21!#LBL3 USE PORT 3
   22!VIEW=1
   23!SEL LAY 203
   24!ECH LAY P203 206-209
   25!GOTO LBL6
   26!#LBL4 USE PORT 4
   27!VIEW=5
   28!SEL LAY 204
   29!ECH LAY P204 206-209
   30!GOTO LBL6
   31!#LBL6
   32!SEL VCON
   33!SEL DEP 0
   34!EXECV
   35!#LBL7 PRNT
   36!PRNT DIGITISE FIRST THE POSITION OF THE BALLOON
   37!PRNT SECONDLY THE POSITION ON THE ITEM REFERENCED
   38!PRNT
   39!PRNT PLEASE DIGITISE IN !!! PORT [PORT] : VIEW [VIEW] !!!
   40!PRNT
```

```
41!BX=0
42!BY=0
43!AX=0
44!AY=0
45!DIGI (VDIG) BX,BY
46!DIGI (VDIG) AX,AY
47!IF (BX.EQ.0.OR.BY.EQ.0) GOTO LBL9
48!INS NFIG JRP.D.X.FIG-BALLON FROMVU1TOVU[VIEW]:X[BX]Y[BY]
49!INS CIR R0.5 : X[AX]Y[AY]
50!INS CNOD : X[AX]Y[AY]
51!
52!
53!
54!
55!
56!
57!
58!
59!
60!
61!INS NLIN : X[BX-7]Y[BY-7],X[AX]Y[AY]
62!<CR>
63!EXECV
64!PRNT THIS SECTION PROVIDES ANNOTATION FOR THE BALLOONS
65!PRNT
66!PRNT NOTE :- PLEASE FOLLOW THE INSTRUCTIONS CAREFULLY !
67!PRNT TEXT MUST NOT EXCEED STATED NUMBER OF CHARACTERS !
68!PRNT
69!PRNT CHARACTER 1-20 : 12345678901234567890
70!READ (ENTER POS   - 3 : ) &POS
71!READ (BENENNUNG - 21 : ) &BEN
72!READ (NORM NR    - 14 : ) &NRM
73!READ (STUECK AUS - 3 : ) &STK
74!READ (WERKSTOFF  - 8 : ) &WRK
75!READ (ROHMASS    - 18 : ) &RHM
76!READ (DIN NUMMER - 5 : ) &DIN
77!READ (BEMERKUNG - 16 : ) &BMK
78!READ (VORRAT ?   - 6 : ) &LGR
79!INS NTXT <*
80!$$ [&POS] $[&BEN] $[&NRM] $[&STK] $<*
81![&WRK] $[&RHM] $[&DIN] $[&BMK] $[&LGR] $$ <*
82!NFIG : X[BX]Y[BY]
83!EXECV
84!PRNT
85!&ANS="NO"
86!READ (ANOTHER BALLOON -   Y/N ? : ) &ANS
87!IF (&ANS.EQ."Y") GOTO LBL7
88!#LBL9 *
89!* ****** END OF VARPRO PROGRAM ******
90!*
```



```
JRP.D.X.B.&BCD.INS-BLN-03D
 7- 2-81 15:30:50


     1!PRNT
     2!PRNT THIS PROGRAM INSERTS DUMMY BALLOONS IN 3D PARTS
     3!PRNT ##############################################
     4!PRNT
     5!PRNT THIS PROCEDURE INSERTS BALLOONS IN 3-D WITH
     6!PRNT STRING BUT WITHOUT ANNOTATION FOR THE BOM !
     7!PRNT
     8!PORT=2
     9!READ (PLEASE SPECIFY THE PORT : ) PORT
    10!PRNT
    11!IF (PORT=2) GOTO LBL2
    12!IF (PORT=3) GOTO LBL3
    13!IF (PORT=4) GOTO LBL4
    14!PRNT PORT NOT ALLOWED - MUST BE PORT 2 3 4 !
    15!GOTO LBL9
    16!#LBL2 USE PORT 2
    17!VIEW=2
    18!SEL LAY 202
    19!ECH LAY P202 206-209
    20!GOTO LBL6
    21!#LBL3 USE PORT 3
    22!VIEW=1
    23!SEL LAY 203
    24!ECH LAY P203 206-209
    25!GOTO LBL6
    26!#LBL4 USE PORT 4
    27!VIEW=5
    28!SEL LAY 204
    29!ECH LAY P204 206-209
    30!GOTO LBL6
    31!#LBL6
    32!SEL VCON
    33!SEL DEP 0
    34!EXECV
    35!#LBL7 PRNT
    36!PRNT DIGITISE FIRST THE POSITION OF THE BALLOON
    37!PRNT SECONDLY THE POSITION ON THE ITEM REFERENCED
    38!PRNT
    39!PRNT PLEASE DIGITISE IN !!! PORT [PORT] : VIEW [VIEW] !!!
    40!PRNT
    41!BX=0
    42!BY=0
    43!AX=0
    44!AY=0
    45!DIGI (VDIG) BX,BY
    46!DIGI (VDIG) AX,AY
    47!IF (BX.EQ.0.OR.BY.EQ.0) GOTO LBL9
    48!INS NFIG JRP.D.X.FIG-BLNNUL FROMVU1TOVU[VIEW]:X[BX]Y[BY]
    49!INS CIR R0.5 : X[AX]Y[AY]
    50!INS CNOD : X[AX]Y[AY]
    51!
    52!
```

```








INS NLIN : X[BX-7]Y[BY-7],X[AX]Y[AY]
<CR>
EXECV
PRNT THIS SECTION PROVIDES ANNOTATION FOR THE BALLOONS
PRNT
PRNT NOTE :- PLEASE FOLLOW THE INSTRUCTIONS CAREFULLY !
PRNT TEXT MUST NOT EXCEED STATED NUMBER OF CHARACTERS !
PRNT
READ (ENTER POS  - 3 : ) &POS
INS NTXT $$ [POS] $$ NFIG : X[BX]Y[BY]
EXECV
PRNT
&ANS="NO"
READ (ANOTHER BALLOON -   Y/N ? : ) &ANS
IF (&ANS.EQ."Y") GOTO LBL7
#LBL9 *
* ****** END OF VARPRO PROGRAM ******
*
```

JRP.D.X.B.&BCD.INS-BLN-FIG
6-25-81 19:47:27

```
*
* THIS EXECUTE FILE INSERTS BALLOONS AND THEIR STRINGS
* *************************************************
*
* THIS PROCEDURE ENABLES A NUMBER OF BALLOONS
* WITH THEIR STRINGS TO BE INSERTED MORE EASILY
*
* DIGITISE FIRST THE POSITIONS OF THE BALLOON
* SECONDLY THE POSITIONS ON THE ITEM REFERENCED
* THIRDLY AGAIN THE SAME POSITIONS ON THE ITEM
* FINALLY CONNECT THE TWO WITH TWO DIGITISES EACH
*
SEL LAY 201
INS NFIG JRP.D.X.FIG-BALLOON : <VAR>
INS CIR RO.5 : <VAR>
INS CNOD : ORG CIR <VAR>
INS NLIN : <VAR>
*
* ******   END OF EXECUTE FILE   ******
*
```

```
JRP.D.X.B.&BCD.INS-BLN-TXT
 6-25-81 17:57:26


    1!PRNT
    2!PRNT THIS PROGRAM PROVIDES ANNOTATION FOR THE BALLOONS
    3!PRNT ##################################################
    4!PRNT
    5!PRNT NOTE :- PLEASE FOLLOW THE INSTRUCTIONS CAREFULLY
    6!PRNT TEXT MUST NOT EXCEED STATED NUMBER OF CHARACTERS
    7!PRNT
    8!PRNT CHARACTER 1-20 : 12345678901234567890
    9!READ (ENTER POS   - 3 : ) &POS
   10!READ (BENENNUNG - 21 : ) &BEN
   11!READ (NORM NR    - 14 : ) &NRM
   12!READ (STUECK AUS - 3 : ) &STK
   13!READ (WERKSTOFF  - 8 : ) &WRK
   14!READ (ROHMASS    - 18 : ) &RHM
   15!READ (DIN NUMMER - 5 : ) &DIN
   16!READ (BEMERKUNG - 16 : ) &BMK
   17!READ (VORRAT ?   - 6 : ) &LGR
   18!INS NTXT <*
   19!$$ [&POS] $[&BEN] $[&NRM] $[&STK] $<*
   20![&WRK] $[&RHM] $[&DIN] $[&BMK] $[&LGR] $$ <*
   21!NFIG : <VAR>
   22!*
   23!* ****** END OF VARPRO PROGRAM ******
   24!*
```

```
JRP.D.X.DRG.&BCD.RUN-BOM-OP
 6-29-81 14:17:53


    1!PRNT
    2!PRNT THIS PROGRAM RUNS THE BOM REPORT PROCEDURES
    3!PRNT ##########################################
    4!PRNT
    5!PRNT THE CURRENT PART WILL BE ABANDONED !
    6!PRNT FILE NOW IF IT IS TO BE PRESERVED  !
    7!<VAR>
    8!EXECV
    9!READ (PLEASE SPECIFY PART NAME : )  &PART
   10!READ (NEXT THE BOM REPORT NAME : )  &NAME
   11!PRNT
   12!EXTR DATA JP.BOM.EDF.PS PART [&PART] NAME ZEIT.JPBOM.PS.EXTR
   13!<CR>
   14!RUN BOMSTK
   15!ZEIT.JPBOM.PS.EXTR
   16!ZEIT.JPBOM.PS.TEIL
   17!ZEIT.JPBOM.PS.SEIT
   18!<CR>
   19!RUN BOMFIN
   20!ZEIT.JPBOM.PS.SEIT
   21!ZEIT.JPBOM.PS.TEIL
   22![&NAME]
   23!<CR>
```

```
24!EXIT
25!DG
26!DELETE ZEIT.JPBOM.PS//NLEV
27!*
28!* ****** END OF BOM REPORT PROCEDURE ******
29!*
30!* ****** ON PARTNAME : [&PART]
31!*
32!* ****** REPORT FILE : [&NAME] !!!
33!*
```

```
JRP.D.DRG.X-EDF.&BCD.BOM
 7- 1-81 18:50:47
```

```
 1!*      28-NOV-80 : JRP (CVE) : VERSION 1
 2!*
 3!*      REFERENCE - J.R.PAINTER - CVE HQ
 4!*
 5!************************************************************
 6!*
 7!*      EXTRACT DEFINITION FILE DESCRIBING REQUIREMENTS FOR
 8!*
 9!*      BILLS OF MATERIALS FOR MECHANICAL TYPE APPLICATIONS
10!*
11!*      USING THE INFORMATION FROM BALLOONS ON THE DRAWINGS
12!*
13!************************************************************
14!*
15!*
16!*
17!************************************************************
18!*
19!1  SECTION #1 - DATA EXTRACT ITEMS
20!*
21!************************************************************
22!*
23!>              /UNIV.ITEMS   /
24!>              /COMP.ITEMS   /
25!*
26!>  ZEICHNNB  / UNIV.NTXT.IF-TNOD-PROPNAME / PARTNO
27!>  BENENNUN  / UNIV.NTXT.IF-TNOD-PROPNAME / TITLE
28!>  ABT       / UNIV.NTXT.IF-TNOD-PROPNAME / ENCLNAME
29!>  NAME      / UNIV.NTXT.IF-TNOD-PROPNAME / JACK
30!>  DATE      / UNIV.NTXT.IF-TNOD-PROPNAME / REGENDATE
31!>  WANB      / UNIV.NTXT.IF-TNOD-PROPNAME / DRAWING
32!>  BGR       / UNIV.NTXT.IF-TNOD-PROPNAME / XLOC
33!*
34!>  TEILNR    /COMP.NTXT.IF-TNOD-PROPNAME  / POS
35!>  STUECK    /COMP.NTXT.IF-TNOD-PROPNAME  / COUNT
36!>  BENENN    /COMP.NTXT.IF-TNOD-PROPNAME  / ELEMDESC
37!>  NORMNR    /COMP.NTXT.IF-TNOD-PROPNAME  / PARTCODE
38!>  WERKST    /COMP.NTXT.IF-TNOD-PROPNAME  / COMPDESC
39!>  ROHMAS    /COMP.NTXT.IF-TNOD-PROPNAME  / TERMDESC
40!>  DIN       /COMP.NTXT.IF-TNOD-PROPNAME  / PANEL
```

```
>  BMRKNG     /COMP.NTXT.IF-TNOD-PROPNAME  / VALUEA
>  LAGER      /COMP.NTXT.IF-TNOD-PROPNAME  / VALUEB
*
************************************************************
*
2  SECTION #2 - PAGE FORMAT SPECIFICATIONS:
*
************************************************************
*
*  NUMBER OF PRINT LINES PER PAGE :
*
>  43
*
*  NUMBER OF COLUMN POSITIONS PER PRINT LINE :
*
>  120
*
*  NUMBER OF PAGE HEADER LINES (SECTION 7) :
*
>  26
*
*  NUMBER OF ITEM FIELD DESCRIPTOR RECORDS (SECTION 8) :
*
>  16
*
************************************************************
*
3  SECTION #3 - PAGINATION SPECIFICATIONS (OPTIONAL) :
*
************************************************************
*
*  HEADER LINE NO,START COLUMN,END COLUMN,JUSTIFICATION :
*
*  NOTE: HEADER LINE NO IS ZERO IF NO PAGINATION DESIRED
*        JUSTIFICATION CODE : -1=LEFT,0=CENTER,+1=RIGHT.
*
>  7,105,107, +1
*
************************************************************
*
3  SECTION #4 - REPORT DATE SPECIFICATIONS (OPTIONAL) :
*
************************************************************
*
*  HEADER LINE NO,START COLUMN,END COLUMN,JUSTIFICATION :
*
*  NOTE: HEADER LINE NO IS ZERO IF NO DATE IS DESIRED
*        A MINIMUM OF 8 COLUMN POSITIONS MUST BE GIVEN.
*
>  0,105,107, +1
*
************************************************************
*
3  SECTION #5 - REPORT TIME SPECIFICATIONS (OPTIONAL) :
*
************************************************************
*
```

```
*  HEADER LINE NO,START COLUMN,END COLUMN,JUSTIFICATION :
*
*  NOTE: HEADER LINE NO IS ZERO IF NO TIME IS DESIRED
*        NEED A MINIMUM OF 5 (OR 9 IF SECS ALSO) COLUMNS
*
>  0,105,107, +1
*
*
*******************************************************
*
*  SECTION #6 - BILL-OF-MATERIALS TALLY (OPTIONAL) :
*
*******************************************************
*
*  BOM INDICATOR,START COLUMN,END COLUMN,JUSTIFICATION :
*
*  NOTE: BOM INDICATOR = 0 IF NOT APPLICABLE ELSE = 1
*
*  0,13,15,+1
```

```
117!*
118!****************************************** ****************
119!*
120!7  SECTION #7 - HEADER LINES
121!*
122!**********************************************************
123!*
124!*000000000111111111122222222223333333333444444444*555555555566666666667777777777888888888899999999990000000000111111111
125!*123456789012345678901234567890123456789012345678*012345678901234567890123456789012345678901234567890123456789012345678
126!>!
127!>!--------------------------------------------------------------------------------------------------------------------!
128!>!                                                                                                                    !
129!>!                                                                                                                    !
130!>!                                                                                                                    !
131!> !------------------------------------------------------------------------------------------------------------------!
132!> ! COMPUTERVISION. !      STUECKLISTE INDIREKTER BEREICH      ! KA !        ! WA-NR              ! BLATT OO  VON OO !
133!> ! ( EUROPE ) INC. !      ==============================      !    !        !                    !                  !
134!> !------------------------------------------------------------------------------------------------------------------!
135!> ! ZEICHNUNGS-NR / BAUMUSTER ! BG  UG ! SACHBENENNUNG                       ! ABT         NAME          ! DATUM     !
136!> !                           !        !                                     !                           !           !
137!> !------------------------------------------------------------------------------------------------------------------!
138!> ! KA NR     ST    BENENNUNG            NORM-NUMMER    WERKST   ROHMASS / TYPE       DIN  STCKZ / BEMERKUNG / LAGER !
139!> ! -- --     --    ---------            -----------    ------   -------   ----       ---  -----   ---------   ----- !
140!>
141!>
142!> ! THESE HEADER LINES ARE TO GIVE THE IMPRESSION OF THE FINISHED REPORT
143!> !  THE ARRANGEMENT OF THE HEADER LINES IS HANDLED BY A POST-PROCESSER.
144!>
145!>
146!> !------------------------------------------------------------------------------------------------------------------!
147!> ! GEPRUEFT    ! AEND.MITTLG.   !      !      !      !      !      !      !      !      !      !      !      !      !
148!> !             ! NUMMER/JAHR.   !      !      !      !      !      !      !      !      !      !      !      !      !
149!> !------------------------------------------------------------------------------------------------------------------!
150!>!--------------------------------------------------------------------------------------------------------------------!
151!>!
152!*
153!**********************************************************
154!8  SECTION #8 - ITEM FIELD DESCRIPTORS :
155!*
156!**********************************************************
157!*
158!> ZEICHNNB    ,11,  4, 29, -1, 0, 0, 0
159!> BGR         ,11, 32, 33, +1, 0, 0, 0
160!> BENENNUN    ,11, 41, 76, -1, 0, 0, 0
161!> WANB        , 8, 79, 97, -1, 0, 0, 0
162!> ABT         ,11, 79, 86, -1, 0, 0, 0
163!> NAME        ,11, 89,104, -1, 0, 0, 0
164!> DATE        ,11,107,115, +1, 0, 0, 0
165!*
166!> TEILNR      , 0,  4,  8, +1, 0,+1, 1
167!> STUECK      , 0, 13, 15, +1, 0, 0, 0
168!> BENENN      , 0, 18, 38, -1, 0, 0, 0
169!> NORMNR      , 0, 41, 54, -1, 0, 0, 0
170!> WERKST      , 0, 56, 63, -1, 0, 0, 0
171!> ROHMAS      , 0, 65, 82, -1, 0, 0, 0
172!> DIN         , 0, 84, 88, +1, 0, 0, 0
173!> BMRKNG      , 0, 91,106, -1, 0, 0, 0
174!> LAGER       , 0,109,115, +1, 0, 0, 0
175!*
176!**********************************************  END  ****
```

CREATING 2D DETAILS FROM 3D DESIGN LAYOUTS

Broncille Stephens
Polaroid
565 Tech Square 2G
Cambridge, MA 02139
617-577-2964

ABSTRACT

CREATING 2-D DETAILS FROM 3-D LAYOUTS

This paper demonstrates how to construct formatted 2D details from 3D design layouts. It contains a step by step example of how this is accomplished. The use of PEP and PAREX files as an aid will be discussed.

INTRODUCTION

The construction of formatted 2D details from 3D layouts can be easily accomplished. The use of Exec-Parex and the standardization of procedures can minimize the time, effort, and degree of expertise required of the computer operator.

3D LAYOUT — STANDARD STRUCTURING

STANDARDIZATION OF PROCEDURES whenever possible is essential.

1. One part constructed on one layer.
2. An information layer designated and which contains all the information necessary for persons other than the current designer/layout to work on the design. Below are some examples of the information required:

   — LAYOUT NAME
   — OPERATOR NAME
   — INDIVIDUAL PART NAMES
   — LAYERING STRUCTURE OF LAYOUT
   — QUANTITY

3. The savings of *basic* displays can be partially standardized. (FIGURES 1A, 1B, 1C)

   — 0 = ALL VIEWS, ALL PORTS
   — 1 = VIEW 1 or PORT 1
   — 2 = VIEW 2 or PORT 2
   — 3 = VIEW 7 or PORT 3

When the basic displays are consistent on all layouts and all details, it expedites the time required for familiarization of the design. This is a definite time saver when working on many different drawings created by different operators.

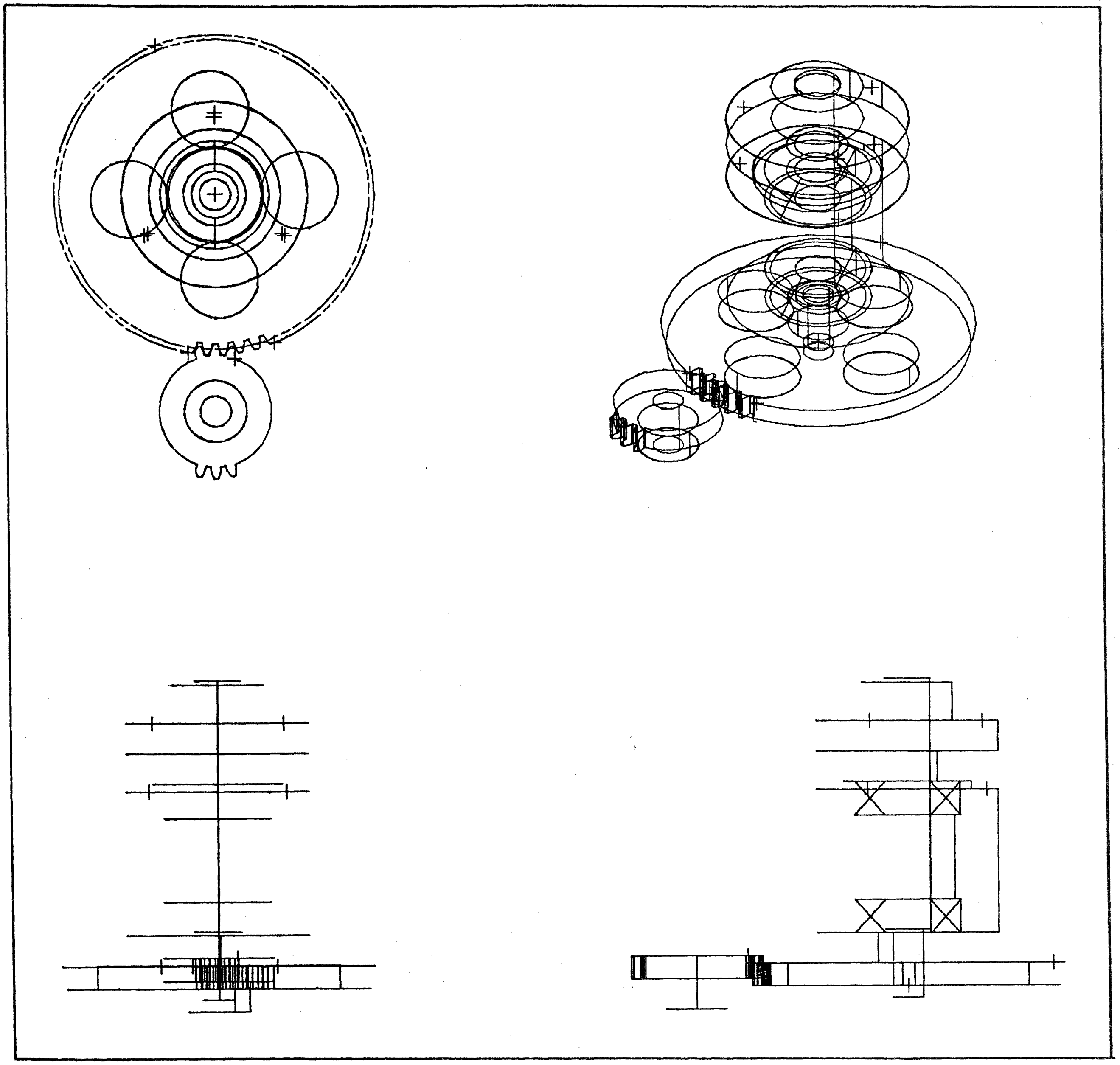


Figure 1A

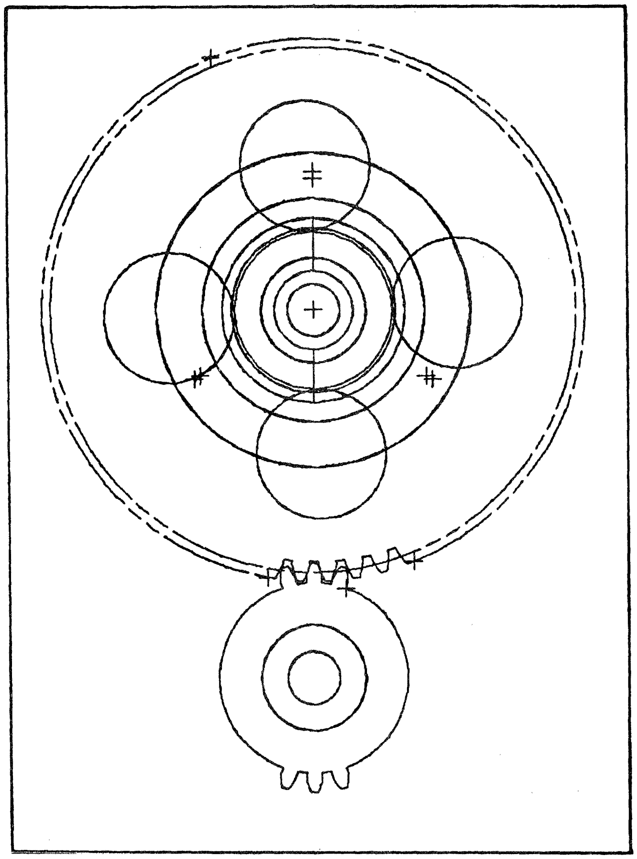


Figure 1B

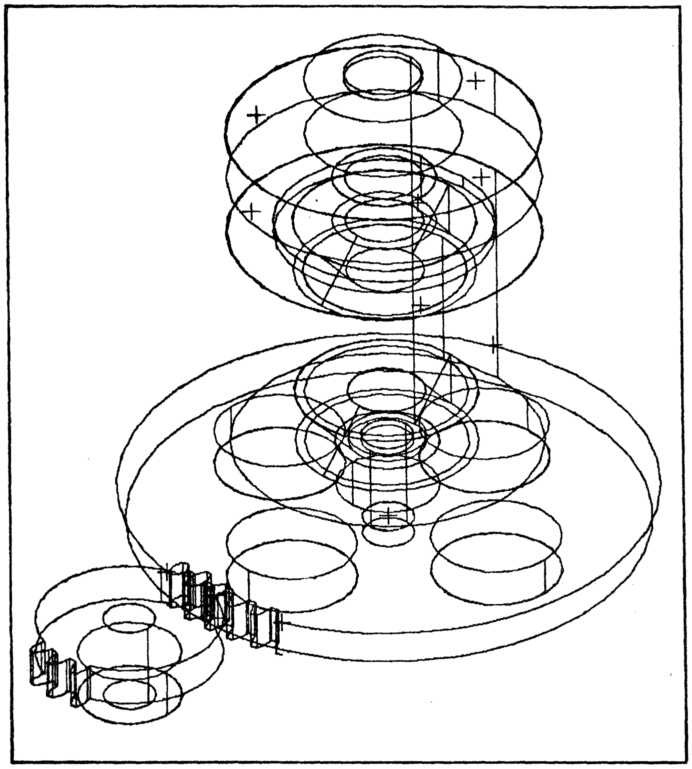


Figure 1C

2D DETAIL — STANDARD STRUCTURING

There are many areas that lend themselves to standardization when detailing.

1. Standard size formats should be created, overwrite protected and always on the disk. (FIGURE 2)
2. A standard layering structure should be established. (A layer or group of layers should be designated for specific information only.) Some examples are:

— Format
— Format Information
  — TITLE
  — ISSUE NUMBER
  — RELEASE NUMBER
  — NOTES
  — DRAWING NUMBER
  — OPERATOR'S NAME
  — GRAPHICS — PART
  — DIMENSIONS
  — STRINGS AT A WIDTH
  (Strings are used to show different line widths.)
  — XH (CROSS-HATCHING)

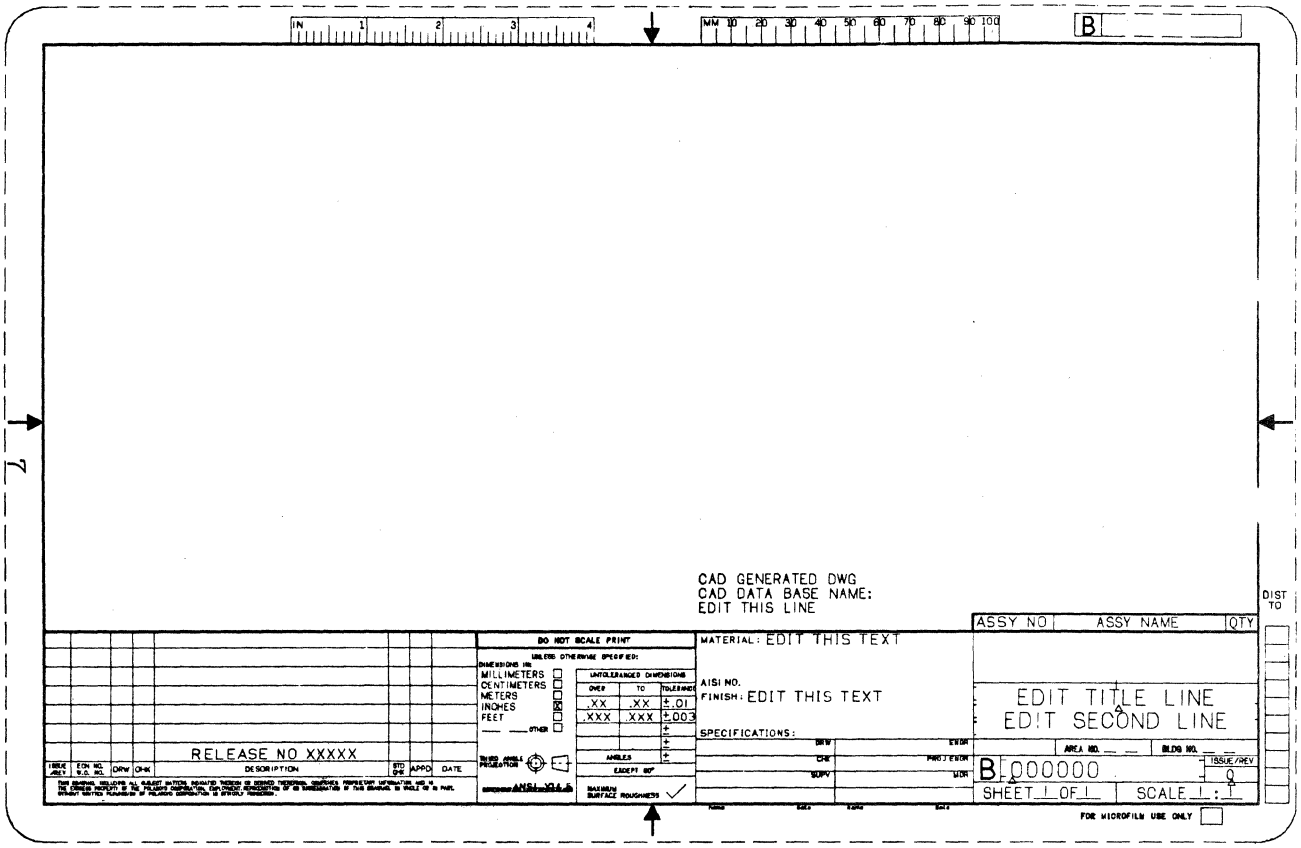


Figure 2

3. Basic displays can be saved here as well as in the layout. (Figures 3A, 3B, 3C)

   — TOTAL PICTURE
   — TITLE BOX
   — GRAPHIC VIEWS
   — NOTE COLUMN

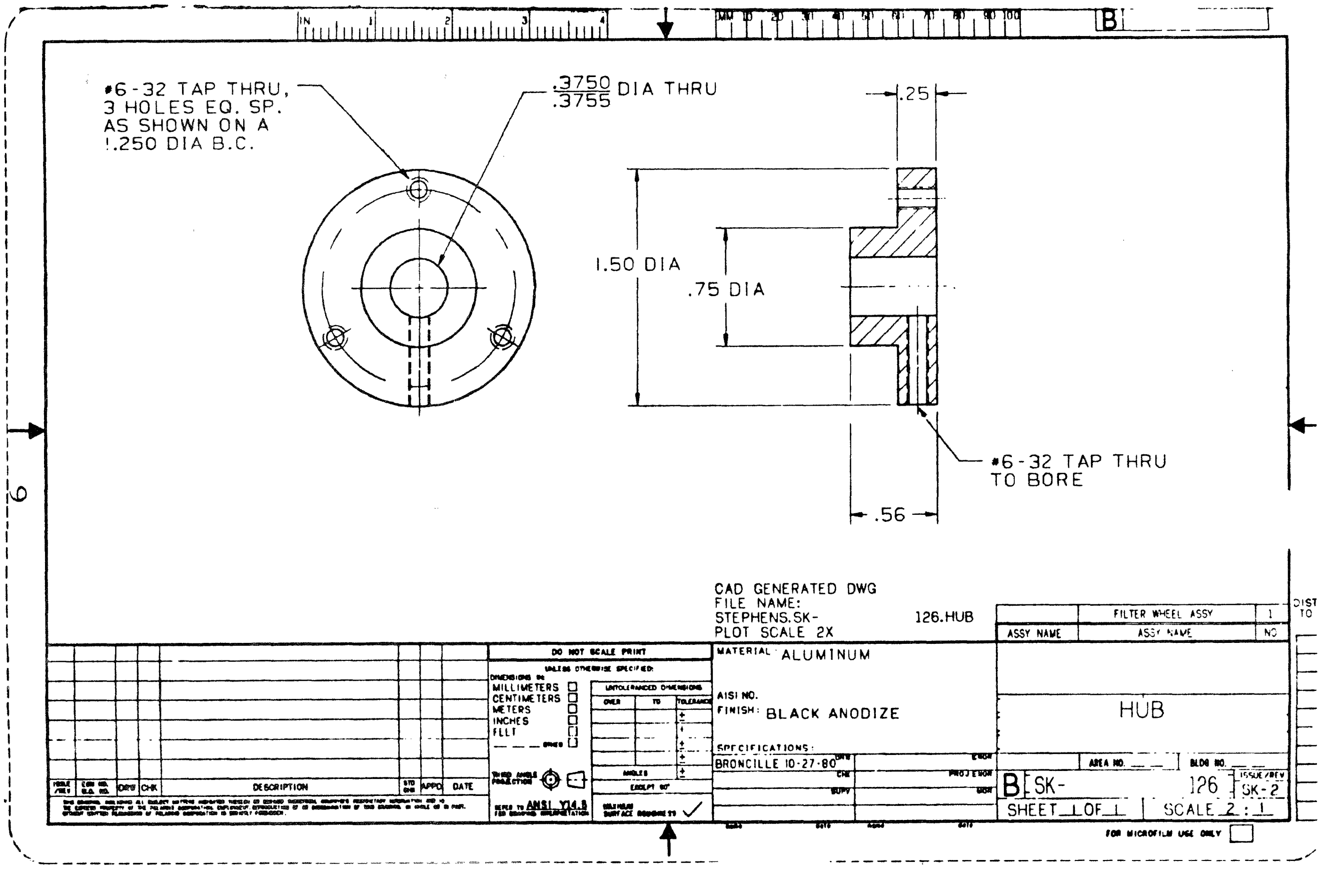


Figure 3A

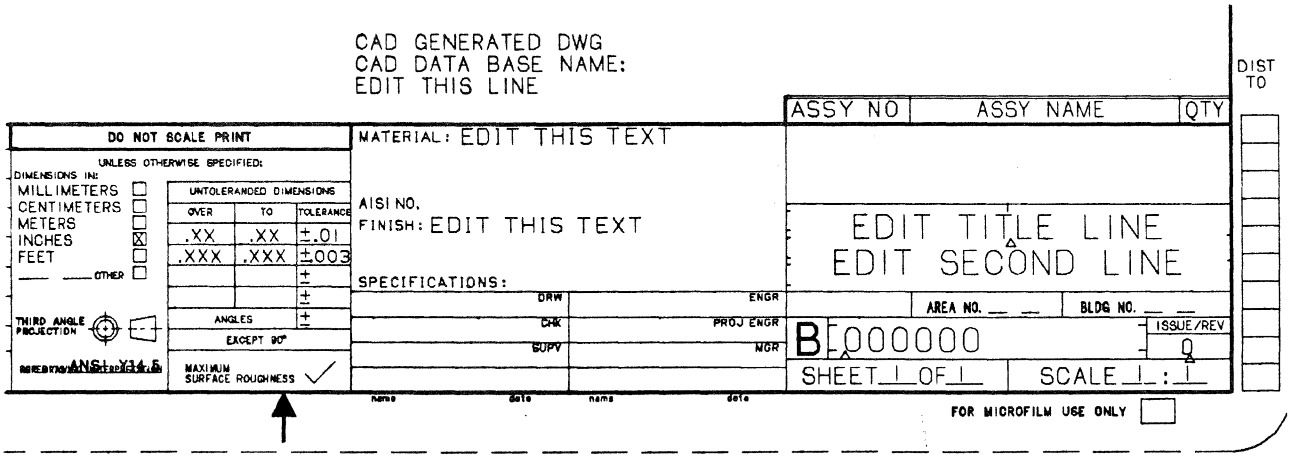


Figure 3B

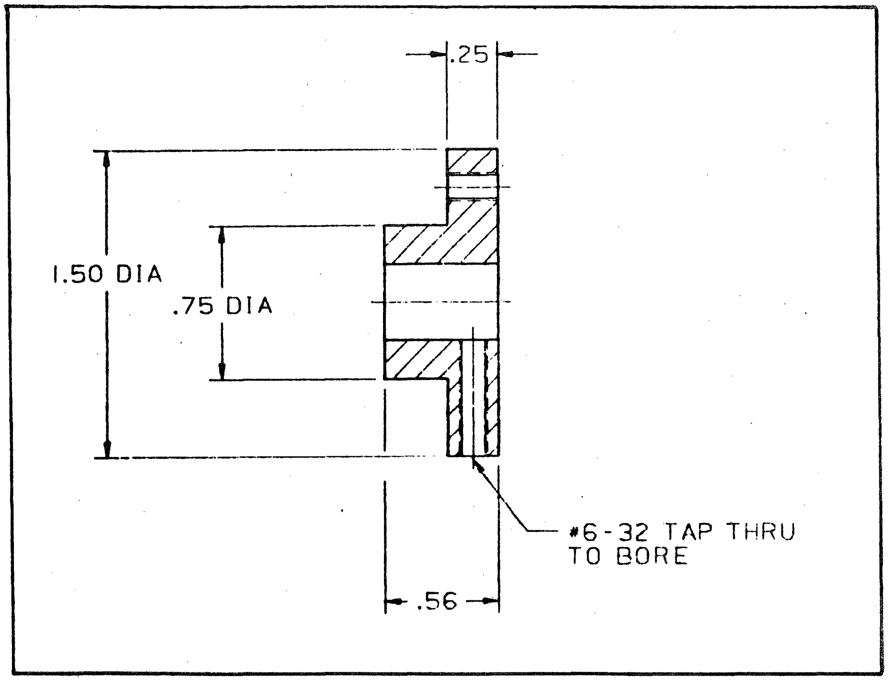


Figure 3C

PRELIMINARY INFORMATION

The information required from the designer/layout prior to detailing is:

1. LAYOUT NAME
2. LAYERS DESIRED DETAILED

DETAILING PROCEDURES
LAYOUT — 3D

— PART NAME: STEPHENS.PAPER.LAYOUT ↵ (Figure 4)
— #USE PORT 1 ↵
— #SEL LAY N APORT ↵
— #ECH LAYN APORT ↵
— #RST DIS O ↵ (Basic display with one layer in all ports visible.) (Figure 5)
— #CONST PRT STEPHENS.LAY N : ENT WIN DD (Temporary part) ↵

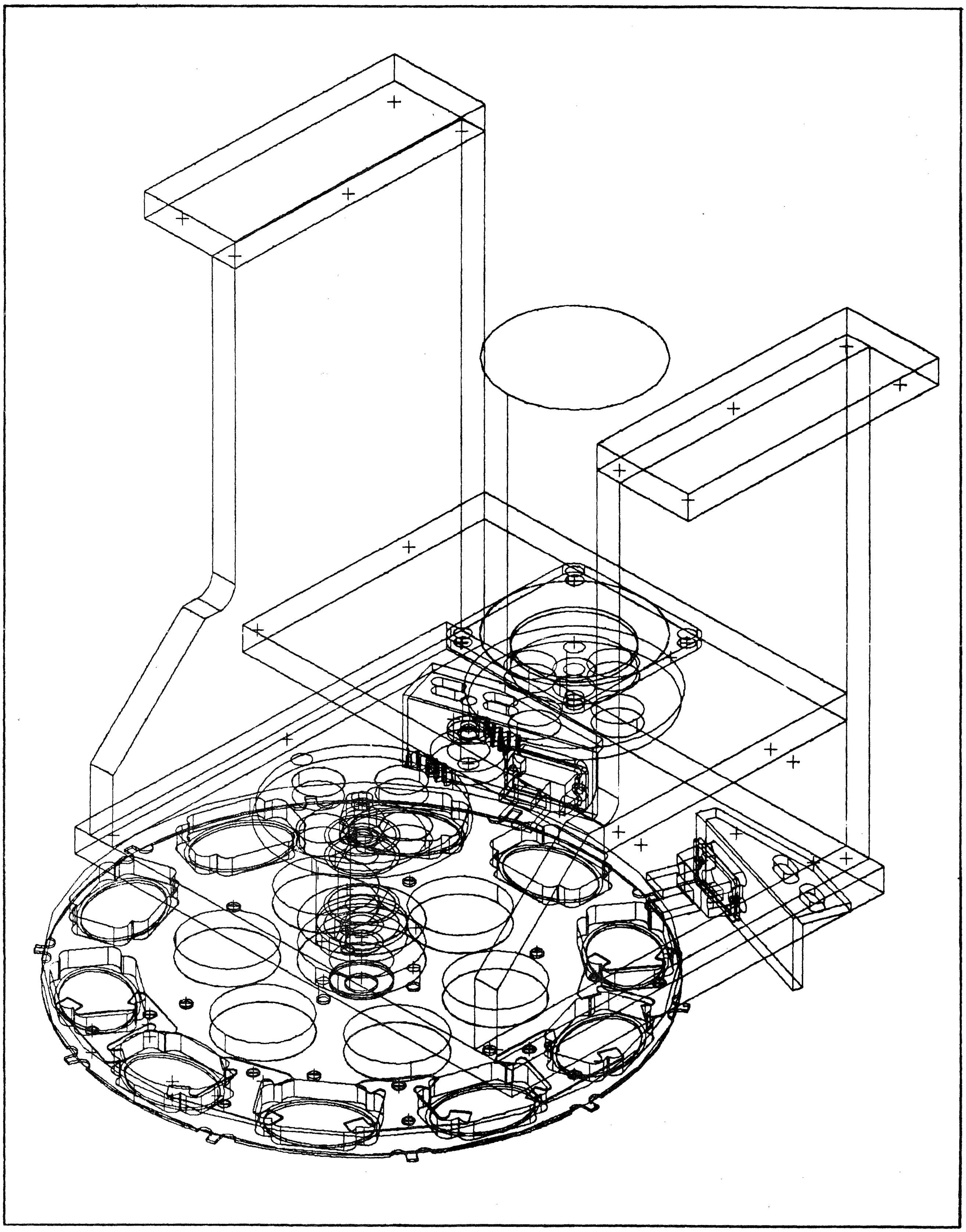


Figure 4

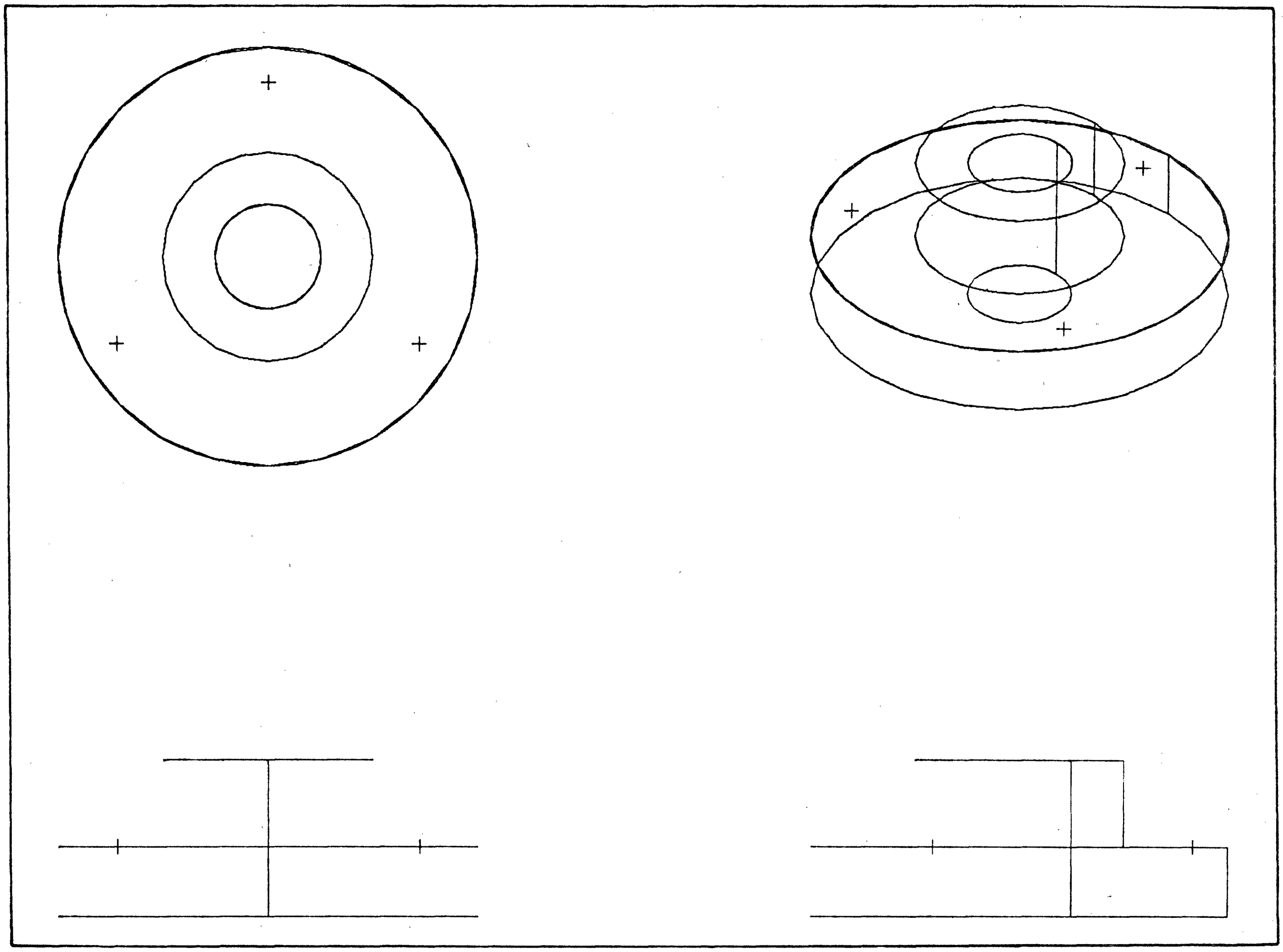


Figure 5

If desired, another lay could be selected and another part constructed before exiting from the layout.

— #EXIT
— #D N (Do not file) (New part)

TEMPORARILY CONSTRUCTED 3D PART

— PART NAME: STEPHENS LAYN
— #REG REG
— #ZOM ALL P10 APORT (Figure 6)
— #PLT DOT SCL1 SLEW20 (Plotting done on a versatic)

At this point, the detailer has a hard copy of the part to study with all the views at a scale of 1 to 1. He/she can now decide which views are necessary, what draw scale is required, and what size format is appropriate. This is a temporary part which will be deleted at another time.

— #EXIT
— #D N (Do not file) (New part)

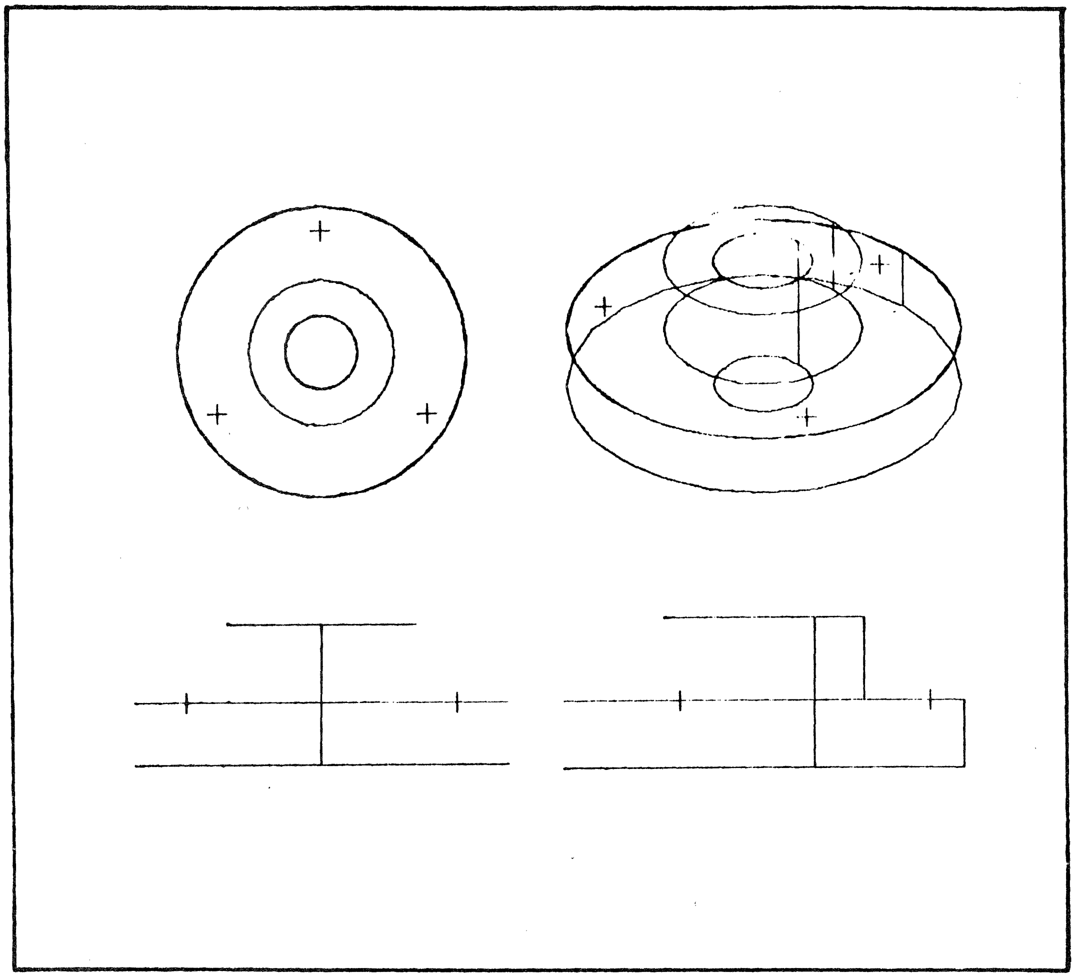


Figure 6

FORMAL DETAIL DRAWING — 2D

```
— PART NAME: B000000 /DSCL2* ↵
— #SEL LAYN ↵
— #ECH LAYA ↵
— #INS NFIG STANDARD.B-SHEET SCL.5 ** ↵
— #ZOM ALL P10 ↵
— #SAV DIS 0 ↵ (BASIC DISPLAY OF TOTAL PICTURE) (Figure 2)
```

***DSCL**
Enables the operator to create a drawing at a scale other than 1 to 1, but dimensions, leaders, text, arrowheads, etc remain at 1 to 1 scale.

**When using a DSCL, the format must be inserted at the scale factor (SCALE = 1 ÷ DSCL).

```
— #INS PRT STEPHENS.LAYN VIEWN ↵ ***
— #ZOM EXT ↵
— #CHG LAY N : ENT WIN DD ↵
— #MOV: ENT WIN DD, dig D dig D ↵ (Figure 7A)
— #INS LIN VER (HOR): dig END (ORG) D DIG D ↵
— #INS PRT STEPHENS.LAYN VIEWN ↲
— #ZOM EXT ↲
— #CHG LAY N : ENT WIN DD ↲
— #MOV: ENT WIN DD, END (ORG) D END (ORG) D ↲ (Figure 7B)
— #ERS LIN : ENT D ↲
— #ECH LAYA ↲
— #RST DIS 0 ↲
— #FILE ↲
— #SPACE BAR
— #OK
```

Default scale of insertion is 1.

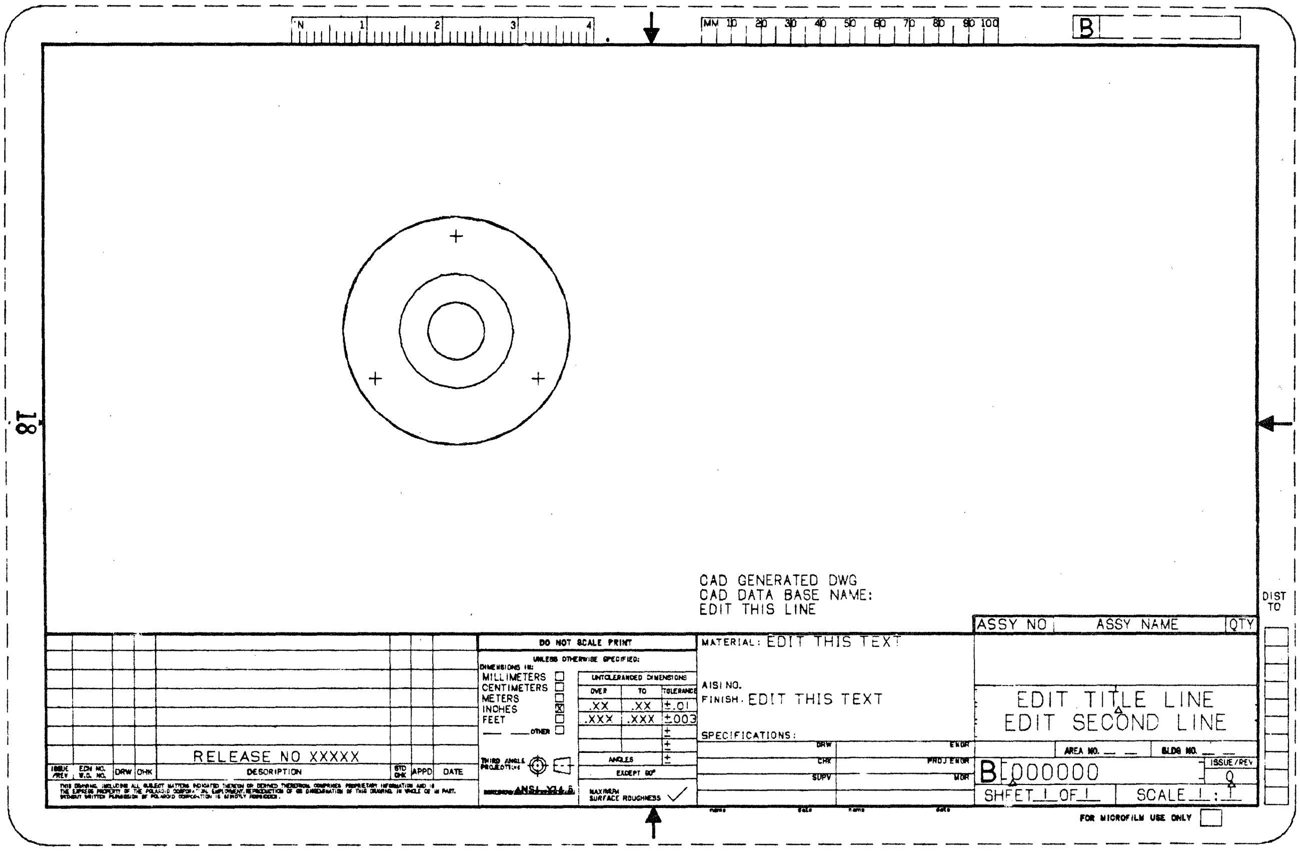


Figure 7A

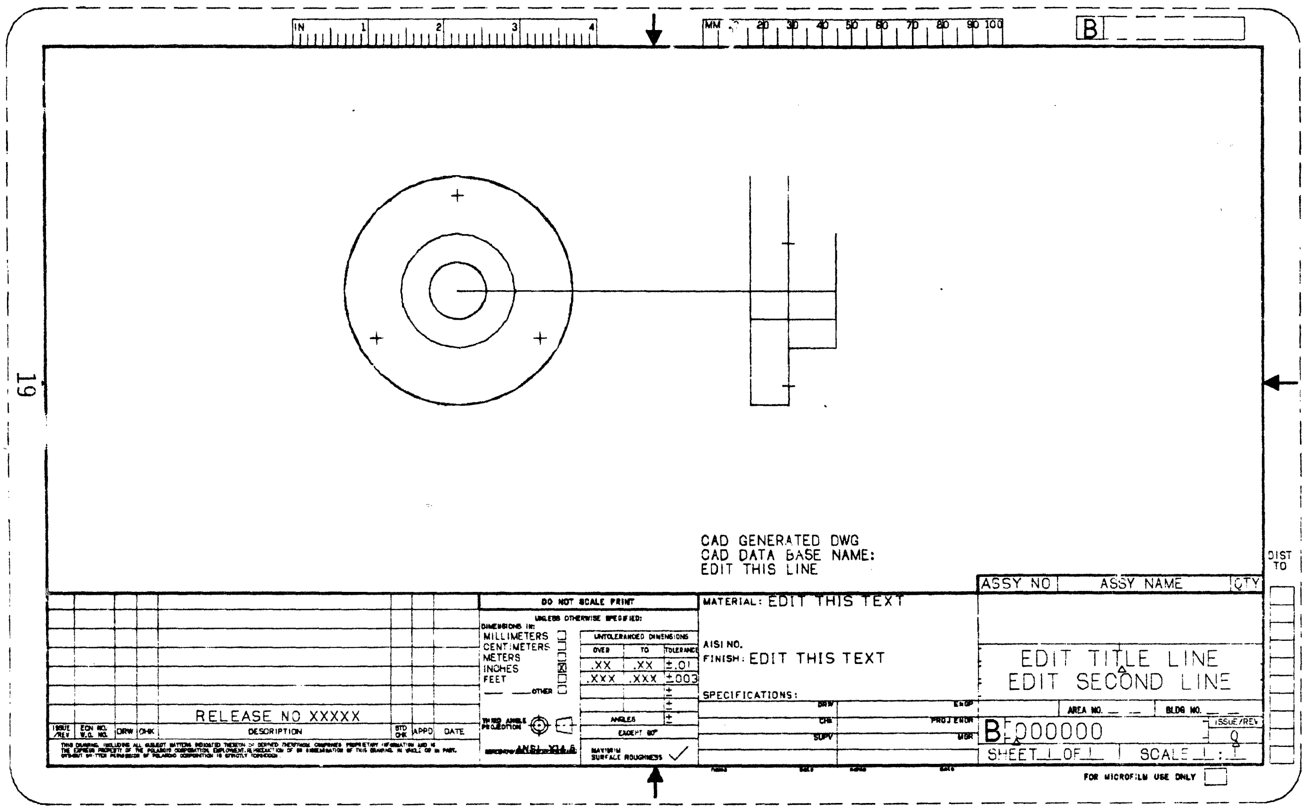


Figure 7B

This drawing is now ready to detail. This method creates a 2D detail of desired views from a 3D database (Figure 3A). Every view inserted in this manner is non-associative with each other and, therefore, lines can be inserted, erased, dashed or cross-hatched. A clean concise drawing can easily be obtained.

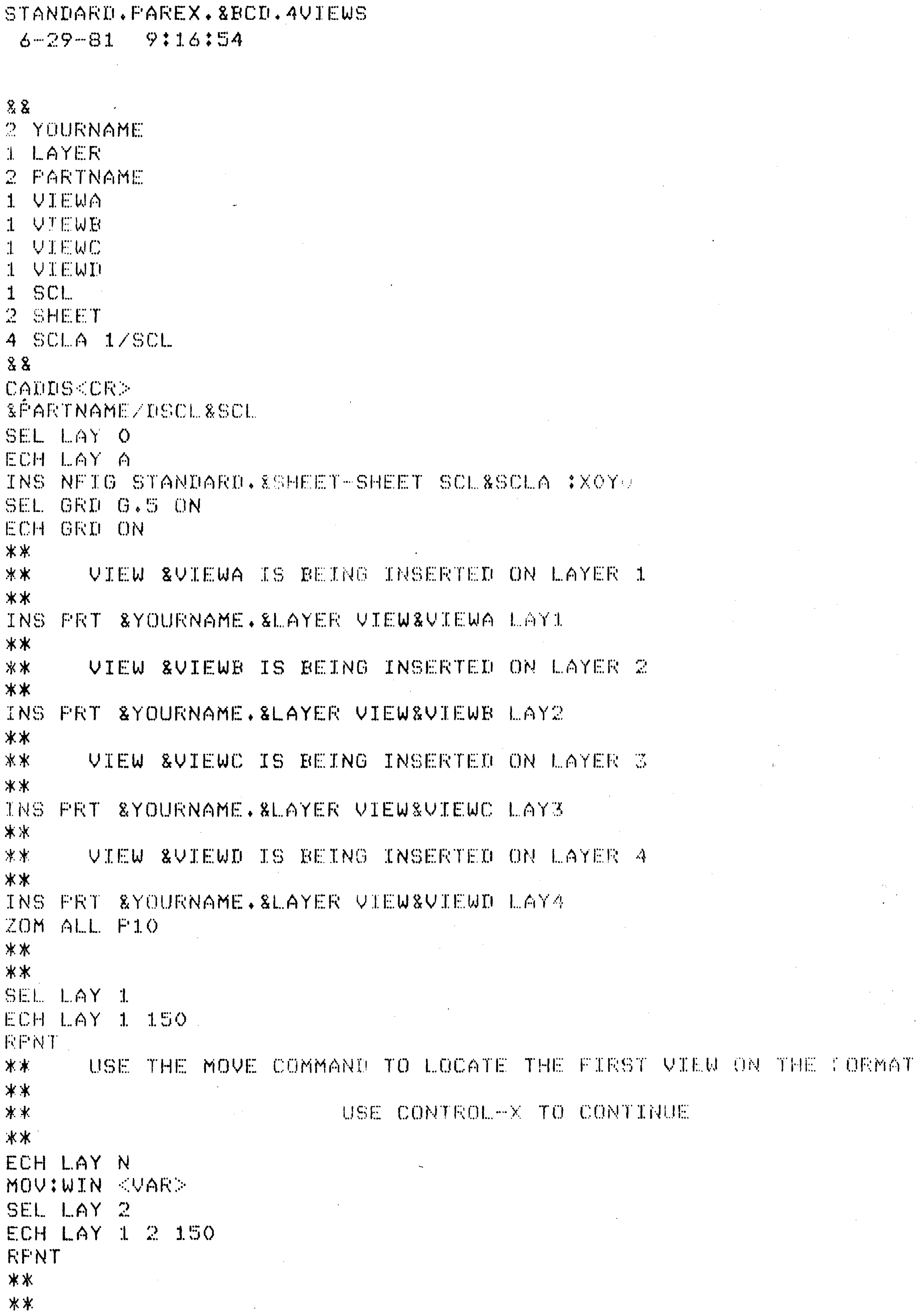
EXEC-PAREX

The following pages contain an *EXEC-PAREX FILE*, which will enable an operator with limited 3D experience to extract parts from 3D databases. This file also protects the operator from unknowingly over-writing the original 3D layout.

```
STANDARD.EXEC.&BCD.DETAIL
 6-24-81  13:01:34


**     THIS FILE AIDS THE OPERATOR IN DETAILING A PART FROM
**     A 3D LAYOUT. INPUT THE INFORMATION YOU AS YOU ARE PROMPTED.
**----------------------------------------------------------------------**
**     LAYOUT= THE NAME OF THE LAYOUT THE DETAIL IS COMING FROM
**     LAYER= THE LAYOUT LAYER NUMBER
**     YOURNAME= YOUR LAST NAME
**     PARTNAME=THE NAME OF THE FINISHED DETAIL
**     PLOT=TYPE PLOT IF YOU WANT A PLOT OTHERWISE ASTERISK
CADDS<CR>
DETAIL
RUN PAREX STANDARD.PAREX.DETAIL<CR>
0>
```

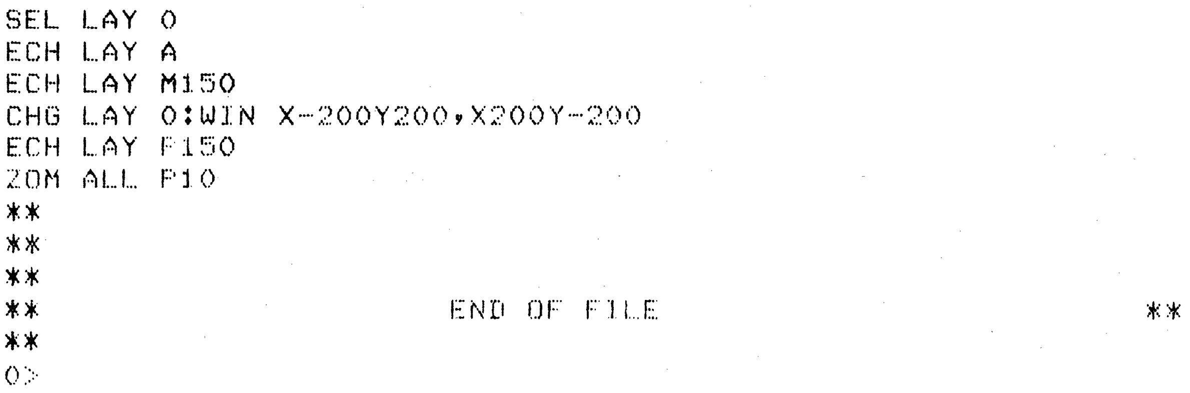
```
STANDARD.PAREX.&BCD.DETAIL
 6-29-81  8:39:55


&&
2 LAYOUT
2 LAYER
2 YOURNAME
2 PARTNAME
2 PLOT
&&
EXIT
DN
&LAYOUT
USE PORT 1
SEL LAY &LAYER APORT
ECH LAY &LAYER APORT
RPNT
CON PRT &YOURNAME.&LAYER :WIN X-200Y200,X200Y-200
EXIT
DN
&YOURNAME.&LAYER
***
***
***   IF YOU REQUIRE A PLOT, TURN ON THE VERSATEC PLOTTER     ******
***
***
REG REG
RST DIS 0
ZOM ALL P10 APORT
&PLOT   DOT SCL1
EXIT
DG
*------------------------------------------------------------------*
*                                                                  *
*                                                                  *
*  CHOOSE FROM ONE OF THE FOLLOWING OPTIONS:                       *
*                                                                  *
*     EXECUTE STANDARD.EXEC.1VIEW    PLACES ONE VIEW ON FORMAT     *
*     EXECUTE STANDARD.EXEC.2VIEWS   PLACES TWO VIEWS ON FORMAT    *
*     EXECUTE STANDARD.EXEC.3VIEWS   PLACES THREE VIEWS ON FORMAT  *
*     EXECUTE STANDARD.EXEC.4VIEWS   PLACES FOUR VIEWS ON FORMAT   *
*                                                                  *
*                                                                  *
*------------------------------------------------------------------*
XOPTION 5
EXECUTE STANDARD.EXEC.1VIEW
EXECUTE STANDARD.EXEC.2VIEWS
EXECUTE STANDARD.EXEC.3VIEWS
EXECUTE STANDARD.EXEC.4VIEWS
*** QUIT ***
&YOURNAME
&LAYER
&PARTNAME
0>
```

```
STANDARD.PAREX.&BCD.1VIEW
 6-29-81   9:05:30


&&
2 YOURNAME
1 LAYER
2 PARTNAME
1 VIEWA
1 SCL
2 SHEET
4 SCLA 1/SCL
&&
CADDS<CR>
&PARTNAME/DSCL&SCL
SEL LAY 0
ECH LAY A
INS NFIG STANDARD.&SHEET-SHEET SCL&SCLA :X0Y0
SEL GRD G.5 ON
ECH GRD ON
**
**    VIEW &VIEWA IS BEING INSERTED ON LAYER 1
**
INS PRT &YOURNAME.&LAYER VIEW&VIEWA LAY1
**
**
ZOM ALL P10
**
**
SEL LAY 1
ECH LAY 1 150
RPNT
**
**
**    NOW CHANGING ALL ENTITIES TO LAYER 0 EXCEPT THE FORMAT ENTITIES.
**
**
SEL LAY 0
ECH LAY A
ECH LAY M150
CHG LAY 0:WIN X-200Y200,X200Y-200
ECH LAY P150
ZOM ALL P10
**
**
**
**                      END OF FILE                          **
**
0>
```



```
STANDARD.PAREX.&BCD.4VIEWS
 6-29-81   9:16:54


&&
2 YOURNAME
1 LAYER
2 PARTNAME
1 VIEWA
1 VIEWB
1 VIEWC
1 VIEWD
1 SCL
2 SHEET
4 SCLA 1/SCL
&&
CADDS<CR>
&PARTNAME/DSCL&SCL
SEL LAY 0
ECH LAY A
INS NFIG STANDARD.&SHEET-SHEET SCL&SCLA :XOY
SEL GRD G.5 ON
ECH GRD ON
**
**     VIEW &VIEWA IS BEING INSERTED ON LAYER 1
**
INS PRT &YOURNAME.&LAYER VIEW&VIEWA LAY1
**
**     VIEW &VIEWB IS BEING INSERTED ON LAYER 2
**
INS PRT &YOURNAME.&LAYER VIEW&VIEWB LAY2
**
**     VIEW &VIEWC IS BEING INSERTED ON LAYER 3
**
INS PRT &YOURNAME.&LAYER VIEW&VIEWC LAY3
**
**     VIEW &VIEWD IS BEING INSERTED ON LAYER 4
**
INS PRT &YOURNAME.&LAYER VIEW&VIEWD LAY4
ZOM ALL P10
**
**
SEL LAY 1
ECH LAY 1 150
RPNT
**     USE THE MOVE COMMAND TO LOCATE THE FIRST VIEW ON THE FORMAT
**
**                              USE CONTROL-X TO CONTINUE
**
ECH LAY N
MOV:WIN <VAR>
SEL LAY 2
ECH LAY 1 2 150
RPNT
**
**
```

```
**
**
**    USE THE MOVE COMMAND TO LOCATE THE SECOND VIEW ON THE FORMAT.
**
**
**                      USE CONTROL-X TO CONTINUE
**
ECH LAY N
MOV:WIN <VAR>
SEL LAY 3
ECH LAY 1 2 3 150
RPNT
**
**
**    USE THE MOVE COMMAND TO LOCATE THE THIRD VIEW ON THE FORMAT.
**
**
**                      USE CONTROL-X TO CONTINUE
**
SEL LAY N
MOV:WIN <VAR>
SEL LAY 4
ECH LAY 1 2 3 4 150
RPNT
**
**
**    USE THE MOVE COMMAND TO LOCATE THE FORTH VIEW ON THE FORMAT.
**
**
**                      USE CONTROL-X TO CONTINUE
**
SEL LAY 4
ECH LAY N
MOV:WIN <VAR>
ECH LAY A
RPNT
SEL GRD OFF
ECH GRD OFF
**
**
**
**    USE THE ERS ALL:WIN COMMAND TO ERASE ANY UNUSED VIEWS NOT MOVED
**
**    ON TO THE FORMAT.(THE GRID HAS BEEN TURNED OFF FOR EASY DIGING!)
**
**                      USE CONTROL-X TO CONTINUE
**
SEL LAY 0
ECH LAY A
ECH LAY M150
ERS ALL:WIN<VAR>
**
**
**    NOW CHANGING ALL ENTITIES TO LAYER 0 EXCEPT THE FORMAT ENTITIES.
**
**
```



```
SEL LAY 0
ECH LAY A
ECH LAY M150
CHG LAY 0:WIN X-200Y200,X200Y-200
ECH LAY F150
ZOM ALL F10
**
**
**
**                     END OF FILE                         **
**
0>
```

NC & Manufacturing Engineering

THE USE OF CAD/CAM IN A SHORT LEADTIME PROTOTYPE APPLICATION

N.A. Morgan
Smiths Industries Limited
Vehicle Instrumentation Division
Cricklewood Works
London, N.W. 2
England

ABSTRACT

This paper describes some of the methods used to obtain prototype parts before the completion of a design. These have enabled us to reduce the repeat cycle time for prototype parts from a number of weeks to a number of hours.

Prototype parts are machined on C.N.C. Milling machines using N.C. data derived from the designer's part whilst he is still completing the detail.

The methods used to record the N.C. data, and the transfer to the machining areas, are described as well as the link required between the designer and programmer.

1. INTRODUCTION

The Vehicle Instrumentation Division of Smiths Industries Limited is a supplier of parts to the car industry. These are fitted to vehicles made by British Leyland (Austin, Morris, Jaguar, Rover, Triumph), Renault, Fiat, Volkswagen, Volvo and Hyundai.

The Engineering Department is responsible for the design, development, detail draughting and prototype manufacture. The prototypes are required for our own internal engineering development as well as for supply to our customer. The total cycle time from start of design through to volume production is often about twelve months. This cycle time has been reduced from between 18 to 24 months during the last 2-3 years by customer demand and this has forced us to find ways of shortening the engineering time. The prototypes are required very early in this cycle and often before the detail design is completed.

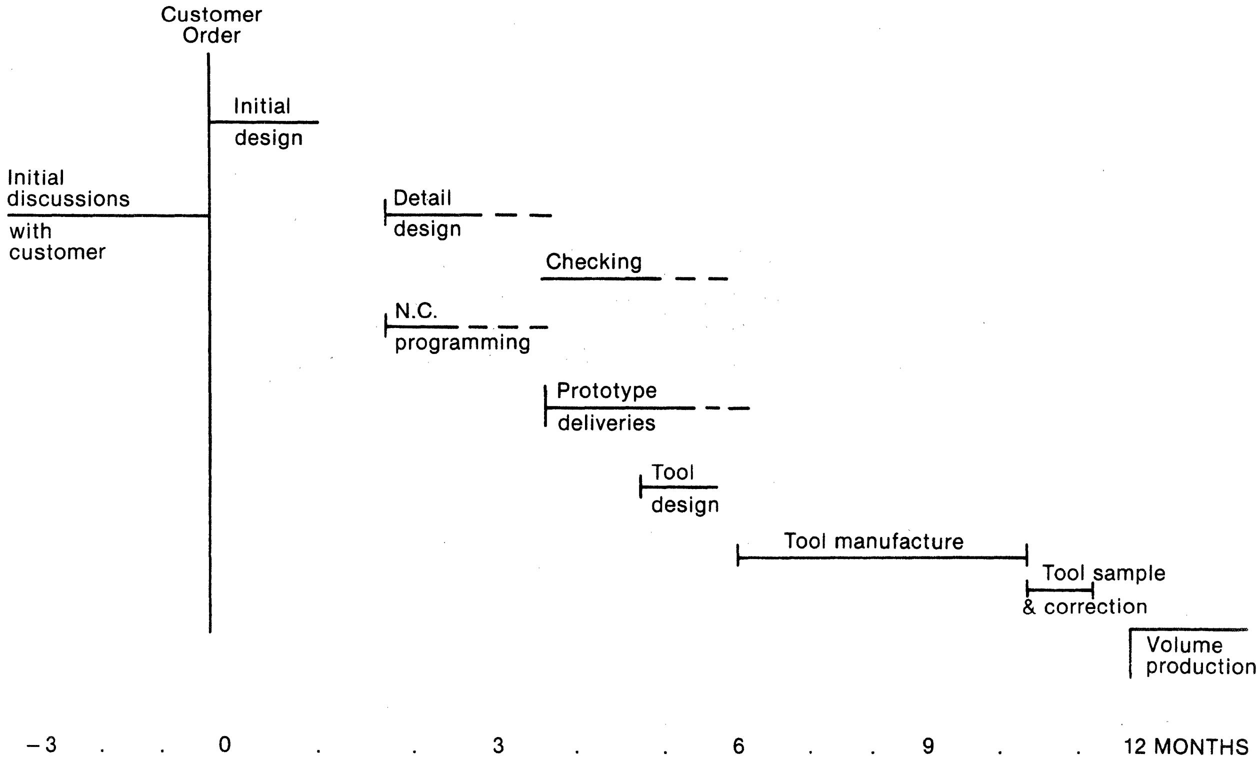


Figure 1: Timescale Showing Current Cycle Time To Volume Production

A longer cycle time was required before we acquired the C.N.C. Milling machines and the Computervision equipment. Prototypes used to be made by hand fabrication from sheets of plastic which were then glued together. This resulted in a prototype which in appearance matched the final shape of the product but was much more fragile and was not suitable for any engineering evaluation.

This method of prototype manufacture also needed a fairly complete drawing of the part to be given to the prototype parts manufacturer. The production of each subsequent prototype required almost as much time as the first one and was very expensive. Using the new method the second part follows on, a few hours, not weeks, after the first part is completed.

This paper describes how we design, programme and manufacture our prototype parts in order to meet these very tight delivery dates.

The Computervision equipment used by the Engineering department comprises a 5-graphic terminal system running on a CGP100. We also have a Benson 4-pen large drum (A0 size) plotter and an Anderson Jacobsen printer and paper tape unit. The equipment is used for a stretched single shift of about 10 hours. There are 2 Matchmaker Mill machines with Posidata controller for the C.A.M. operation.

2. DESIGN PHASE

2.1. DRAWING ACCURACY

The designer must understand that any line placed on the database of his part may be used for producing N.C.-output. In practice only the main views are used and scrap sections, part views etc. will not be used for N.C. work. The main views must therefore be produced with the lines, arcs, points etc. on nominal values where there are critical areas. This places a discipline on the designer and under no circumstances may a dimensional value be written in other than the true value. System users are not taught how to insert a dimension which does not match the database.

The designer may alter a boundary relating to an area on which the N.C. programmer has completed his work. In order to ensure that these changes are incorporated a close liaison is required between designer and programmer. A master drawing held by the programmer on which he can indicate the line or areas of the design that have been changed has been found useful to act as a reminder of program changes to be made.

There will eventually come a point where the program will be fixed and tapes created to enable prototype parts to be made. Any subsequent changes to the design will then have to be reviewed to see if the existing tapes are satisfactory for first-off prototype requirements.

2.2. CUSTOMER CHANGES

The ability to modify a design on C.A.D. is a mixed blessing. The customer, if he knows that C.A.D. is being used, can ask for a change and expect it to be implemented without the major delays that would be associated with more traditional draughting methods. This does create a problem when one is trying to produce a series of N.C. toolpaths from a design that is still fluid.

We are also actively seeking to start using the magnetic tape as a transfer medium between ourselves and our customers where they are also using the C.V. system but so far this has only been possible on a very limited basis.

2.3. DIMENSIONING

The detail dimensioning of the part may not have been done till after the programming has been completed. This obviously makes the tape testing difficult as the operator may have no dimensions on the drawing with which to check. This is why accurate toolpath plots are required as indicated later. (4.2 below).

3. **PROGRAMMING PHASE**

3.1. COMMON DATA BASE

It is essential that the programmer and designer work on the same catalogue part. In the early days we transferred a copy part to a new file for the programmer. This was done for two reasons:

a) The designer did not trust the programmer not to mess up his design,
b) The designer objected to the entity count in his part being greatly increased by the toolpaths with a resultant increase in access time and regen time.

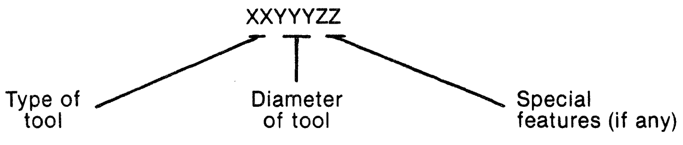
We now operate on the designer's actual part as with practice we have found that the programmer does not mess up the designer's part — or only rarely! Secondly the changes being made on the master design cannot easily be transferred without a lot of layer deletions and parts insertions. The problem of greater file size remains and the associated time to handle the files will always be slightly longer with the toolpaths included.

3.2. INITIAL PREPARATION

It is essential to prepare the sequence of machining, type of clamping and fixtures before starting the programming. For parts with multiple depths it has been found useful to prepare a colour shaded drawing with a Z depth colour indicator at the side. A box of crayons will be cheaper than the colour terminal for a few years yet!

3.3. TOOL LIBRARY

All the standard tools that are available to the programmer are held as a part (See Appendix 1). This part is inserted into the part to be machined and this saves the DEF TOOL having to be invoked. The tool library for the milling machines are held with a simple name system made up as follows:



SD16IZ for example is a *S*lot *D*rill of *16*.0mm diameter, long series.

Special tools such as angle cutters and 'D' bits are usually defined as required and given an 'A' number. They are not held on the master tool list part.

3.4. TOOLPATH CREATION

As each toolpath is created it is tagged and recorded on a programmer's reference sheet (See Appendix 2). This sheet will show the main parameters of the toolpath as well as any APT statements needed during the subsequent MACHIN operation. It also shows feeds and speeds and a brief comment or instruction for the mill operator.

The programmer's reference sheet is also held as a part which can be inserted into the part being machined. All text entry is done using text nodes on this sheet.

A typical component will contain 50–60 machining steps, each of which may be made up from a number of toolpaths or sets. Prior to the MACHIN statement the sequence of machining will be reviewed to see if the order of steps is correct and a final copy of the programmer's reference sheet will be made. This will refer to the tooling list which is created with the tools needed for the job (See Appendix 3). An indication is shown on the tooling list of the maximum depth of cuts used with each tool. This enables the machine operator to set the tools into the collets with the minimum overhang commensurate with clearing the workpiece.

3.5. FIXTURES AND CLAMPING

As the prototype sample runs are generally small (100 maximum), and batch sizes generally only up to 10, fixtures and clamps need to be kept very simple — a wooden block with dowel pins and through bolts or reach-over clamps are most commonly used. The wooden block may be machined on the CNC Mill in order to provide a nest for the component where wall sections may need some support.

Clamping is only shown on the part view when reach-over clamps are used and there is a danger of machining close to the clamps.

3.6. LINK BETWEEN DESIGNER OR PROGRAMMER

It cannot be stressed too highly that the Designer and Programmer must work closely on the project if major errors are to be avoided. The Designer should be aware of which areas the Programmer has completed. The Programmer should also determine from the Designer which areas are most likely to change and leave the programming of these areas to the end.

In order to ensure that either person sees a message left by the other a spare layer is often used to write a message. This layer is left active when the part is filed so that it will be seen next time the part is called up.

The Programmer should also agree with the Designer on a block of layers which will be reserved for his exclusive use. This will avoid any problem of overwriting.

One benefit of very early programming which we have found is that the Programmer may find errors in the part as drawn. To some extent he is acting as a preliminary checker for the Designer.

4. TAPE PREPARATION PHASE

4.1. MACHIN

This is generally considered to be the most tedious part of the programmer's work. Improvements are being investigated such as the use of EXEC programs or further grouping of SETS or SUPERSETS. Owing to pressure of demand for time on the graphics terminals, the MACHIN operation is often run from a 30 cps printer/keyboard. This further slows down the operation.

Great care is needed to ensure that one does not make a mistake during the input of data at this stage. A single typing error after one or two hours performing the MACHIN routine is very annoying — to put it politely. When can we have a more tolerant system that allows one to backtrack?

4.2. TOOLPATH PLOTS

We have found it beneficial to prepare a 1:1 size plot of the toolpath for tape tryouts. As mentioned in 2.3 above, the part may be machined before a dimensional drawing has been prepared. A standard part containing a frame is inserted into the parts being machined (See Appendix 4). This has details of the home value as well as a point showing the origin of the part for setting purposes. The first plot only shows the part to be machined. Translucent paper is then used to plot each toolpath. The Benson plotter with four pens allows us to use four colours to identify the paths. The number of each layer is also plotted so that the toolpath can be identified by its colour. The number of toolpaths should be restricted on each plot to ensure adequate clarity.

5. **PROTOTYPE PRODUCTION**

5.1. MACHINES USED

There are two C.N.C. Matchmaker Milling machines with 800mm, x, 450mm, y, and 140mm, z movement used for our prototype production. These machines have about 29.5k of useable memory. This capacity is often too small to store all the program for a complete job so the job is split into sections.

5.2. TAPE TRY OUT

The tapes, plots, programmer's reference sheet and tooling list are passed to the machine operator. Our machine operators can also manually program jobs so they are able to make changes to the tapes if required. In order to keep programs as short as possible the 'canned cycles' feature on the Mills is used for repetitive machining of such pocketed items as light boxes. The programmer will generate a MPTP path leaving the tool at the centre of each warning light box. The Milling machine operator will then write the program for the warning light box as a 'can' and call this up at each MPTP step (See Appendix 5).

The machines are run with a pointer in the chuck and 'Z inhibit' on the control panel. This enables the machine to run through the tape with the pointer — hopefully — following the toolpath shown on the plot. The operator can check the 'Z' values from the C.R.T. on the controller as the tape is tested.

Experience has shown that once the tool has got to the start of its cutting cycle the programs perform well. Problems generally arise from the programmer overlooking the line taken by the tool from the 'HOME' position to the start of the cutting.

Any errors found during tape try out are corrected by the machine operator and the N.C. data base on C.A.D. is not always up-dated. This is because the number of parts to be produced is so low and so correction of the toolpaths would often serve no useful benefit.

5.3. MATERIALS CUT

The majority of parts are cut from A.B.S. block. This material machines quite well but tools must be kept very sharp. Because the material is quite flexible (2mm wall sections) it must be cut from the outside in to the MPOC routine is the wrong way round for our use. MABS or MPRO are used and then the toolpath is copied using the 'MOVE COPY' command to a different Z depth. Support to these thin wall sections is given by nesting the part if possible.

Some parts are also machined from Aluminum or Zinc diecasting billets.

6. **OVERALL RESULTS**

The manufacture of a quantity of prototype parts in parallel with the design and detail work is now being achieved. This enables us to supply our customers with working prototypes which can be evaluated earlier than was possible before the advent of CAD/M. In some instances this enables changes to the design to be made before cutting metal on the major moulding and diecasting tools has started with the resultant saving in time and money.

Appendix 1: Standard Tooling List

Slot Drills SD

Size	L	IZ	R (Ball)	Prog. Dia.	Max. Length from collet
1.5	5	6	—	1.48	8
2	7	10	—	1.98	10
2.5	9	—	—	2.48	15
3	10	11	10	2.98	23
4	10	15	—	3.98	27
4.5	10	—	—	4.48	29
5	10	28	—	4.98	31
6	13	37	25	5.98	50
7	13	—	—	6.98	50
8	14	40	30	7.98	40
10	15	40	—	9.98	60
12	20	50	—	11.98	65
14	23	—	—	13.98	70
16	22	60	—	15.98	80
18	27	—	—	17.98	100
19	27	—	—	18.98	90
25	30	—	—	24.98	90
25.4	—	60	—	25.38	100

Centre Drills CD

Size	End Dia.	Length	Shank Dia.
1	1.0	2	4
2	1.5	2.5	4.5
3	2.25	4	6

Rippers RP

Size	L	IZ	R (Ball)	Prog. Dia.
10	30	—	—	10.0
15	40	—	—	15.0
20	40	—	—	20.0

End Mills EM

Size	L	IZ	Prog. Dia.
1/16	5	—	1.6
3/32	9	—	2.39
2.5	8	—	2.51
3	13	—	3.01
1/8	14	—	3.18
5/32	15	—	3.98
4	18	—	4.01
3/16	18	—	4.77
5	20	—	5.01
7/32	18	—	5.56
6	22	—	6.01
1/4	23	—	6.36
5/16	40	—	7.95
3/8	40	—	9.53
1/2		50	12.71
9/16		55	14.3
5/8		60	15.88
11/16		70	17.47
3/4		80	19.06

Spot Face SF

Size	L	IZ	Prog. Dia.
9.25	25	50	9.25
12.5		50	12.5

Fly Cutters FC

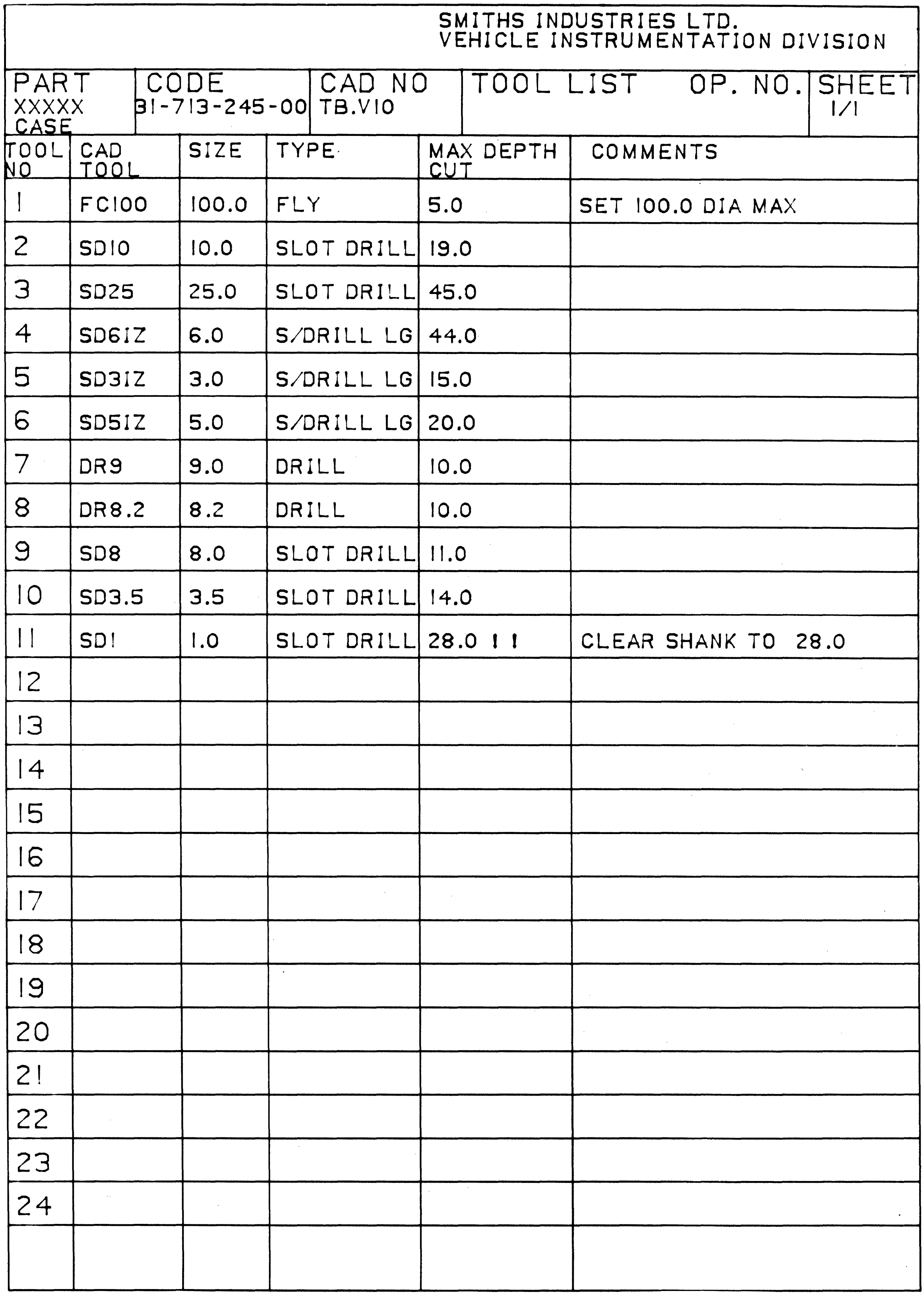
Size 50, 100, 150, 200

Appendix 2

CLFILE. VOCA1	CAD PART NAME: TB.V10	MC SET UP OP NO 1	BLOCK SIZE X 350	SETTING DATUM CENTRE OF SPEEDO HOLE IS X0 Y0. Z0 IS 9.5 ABOVE BOTTOM FACE	HOME DATUM X -450	TRANSFER X 490.1	SHEET
TAPE HEADER: VOLVO CASE OP1	PART CODE: 31-713-245-00	NPR/EJC 0059/05	Y 180 Z 60		Y 300 Z 50	Y 0 Z 0	1

OP	LAY	TAG NAME		CL PLANE	RT PLANE	ZSURF	CAD TOOL		MC TOOL		OFFSET	APT BEFORE PATH	FEEDS				SPIN SPD	COMMENTS AND INSTRUCTIONS
							NO	SIZE	NO	SIZE			1.PLU	2.RET	3.FIN	4.ROU		
1	50	MAB1		50	45	39.47 38.8	FC100	100.0	1	100.0		LOADTL/1,1 SPINDL/300,CLW COOLNT/MIST	75	125	5000		300	FLY CUT SURFACES NOTE 100 MAX DIA CUTTER
2						34.0 29.0												
3						25.5												
4	51	MAB2		50	40	35.6 32.5	SD10	9.98	2	10.0		GOHOME LOADTL/2,2	75	120	5000		1000	GENERAL CUT TO SURFACE
5						33.7 25.5												
6	52	MAB3		50	40	12.7 10.5	SD25	24.98	3	25.0		GOHOME LOADTL/3,3	75	125	5000		800	TOP W/L AREA THEN PERIMETER
7						2.1 -0.4												
8	53	MAB4		50	30	21.7 17.2	SD61Z	5.98	4	6.0		GOHOME LOADTL/4,4	75	100	5000		1200	SHARPEN CORNERS ABOVE W/L AREA
9						12.7												
10	65	PRO4		50	25	19.7 16.7	SD61Z	5.98	4	6.0		- - -	75	100	5000		1200	RECESS BY VAC. GAUGE
11						13.7												
12	54	PRO1		50	15	9.1 5.6	SD61Z	5.98	4	6.0		- - -	75	100	5000		1200	POCKET FOR SECURING SCREW
13						2.1												
14	55	MAB5		50	15	9.1 5.6	SD61Z	5.98	4	6.0		GOHOME	75	100	5000		1200	TOP SCREW RECESS (2 OFF)
15						2.1												
16	56	SET1		50	30	22.1-2.1 5 CUTS	SD61Z	5.98	4	6.0		- - -	75	100	5000		1200	LH,BOTTOM 3,RH SCREW RECESS
17																		

Appendix 3

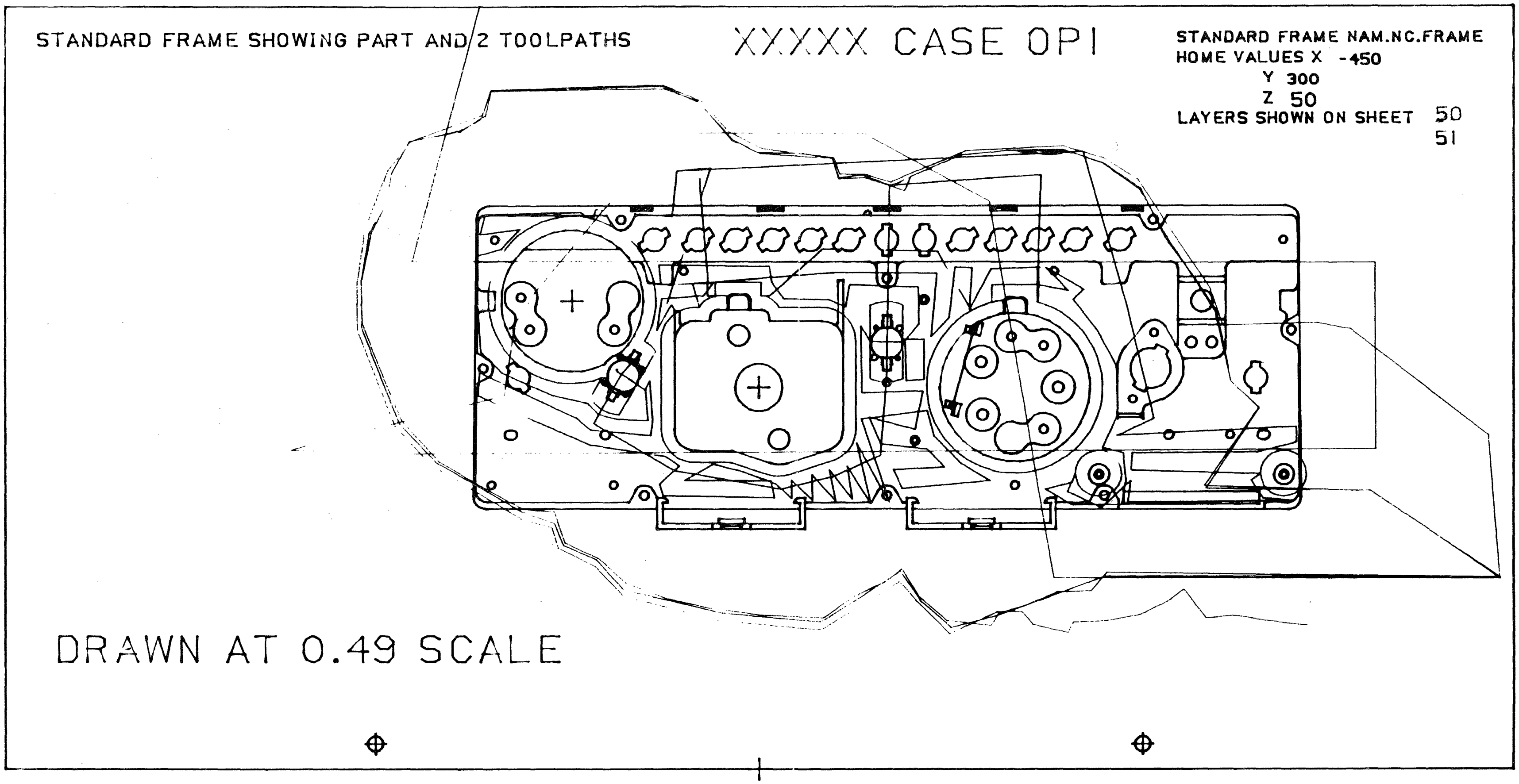


SMITHS INDUSTRIES LTD.
VEHICLE INSTRUMENTATION DIVISION

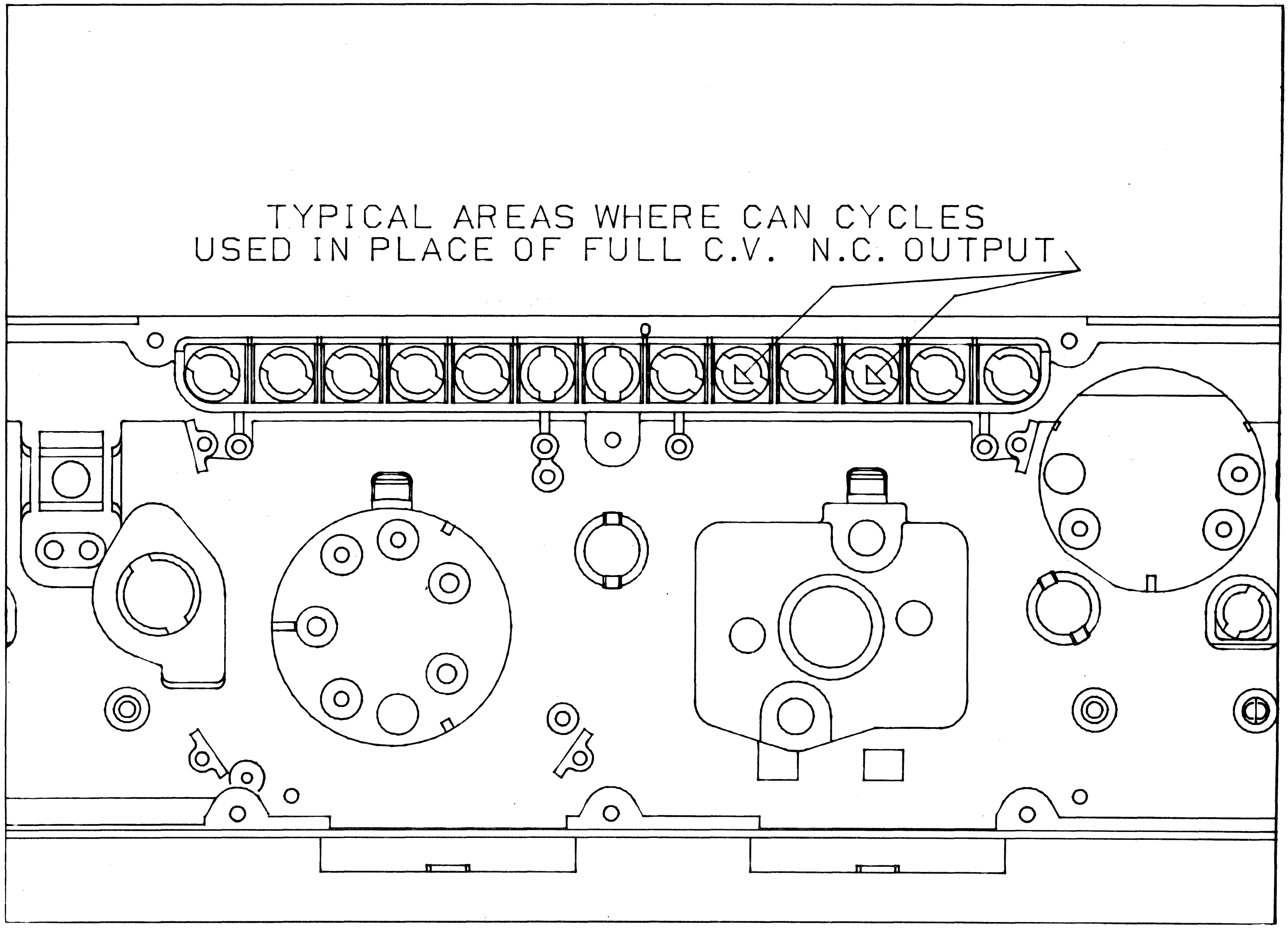
PART	CODE	CAD NO	TOOL LIST OP. NO.	SHEET
XXXXX CASE	31-713-245-00	TB.V10		1/1

TOOL NO	CAD TOOL	SIZE	TYPE	MAX DEPTH CUT	COMMENTS
1	FC100	100.0	FLY	5.0	SET 100.0 DIA MAX
2	SD10	10.0	SLOT DRILL	19.0	
3	SD25	25.0	SLOT DRILL	45.0	
4	SD6IZ	6.0	S/DRILL LG	44.0	
5	SD3IZ	3.0	S/DRILL LG	15.0	
6	SD5IZ	5.0	S/DRILL LG	20.0	
7	DR9	9.0	DRILL	10.0	
8	DR8.2	8.2	DRILL	10.0	
9	SD8	8.0	SLOT DRILL	11.0	
10	SD3.5	3.5	SLOT DRILL	14.0	
11	SD1	1.0	SLOT DRILL	28.0 ! !	CLEAR SHANK TO 28.0
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					

Appendix 4


STANDARD FRAME SHOWING PART AND 2 TOOLPATHS
XXXXX CASE OP1
STANDARD FRAME NAM.NC.FRAME
HOME VALUES X -450
Y 300
Z 50
LAYERS SHOWN ON SHEET 50
51
DRAWN AT 0.49 SCALE

Appendix 5


TYPICAL AREAS WHERE CAN CYCLES
USED IN PLACE OF FULL C.V. N.C. OUTPUT

USE OF THE CV FLAT PATTERN SOFTWARE & INTERFACE WITH MANUFACTURING

Collin McMullin
General Electric
Appliance Park
AP2-126
Louisville, KY 40225
502-452-4515

1. ABSTRACT

This paper discusses how the traditional method of transferring a blueprinted drawing to a finished part, ready for assembly, has been updated through the use of CAD/CAM.

Through the application of the Construct Flat Pattern Generator, prototype sheet metal parts may be developed quickly and accurately on CAD. This process is traced as the part information is transferred from CAD to a numerically controlled part program, through a post procesor sequence, followed by a laser cutting process, and is finalized as the modelmaker forms the sheet metal pattern into a finished product.

Through this process, the Range Model Shop Department has realized great savings in time and money, reduced the lead time in preparing range prototypes for evaluation, and directed the skills of the shop people towards increased productivity.

2. INTRODUCTION

The process of transforming a part from an idea on paper to a finished reality has changed very little in recent years. Some steps are presently unavoidable. With the advent of CAD/CAM technology, many of these older procedures are being pushed aside in favor of newer methods which directly involve CAD/CAM. In General Electric's Range Department, a fundamental change in prototype part development is occurring through the application of the Computervision software, and its interface with numerical control tools in a production environment.

Range Model Shop develops most of their parts for ranges and microwave ovens from engineering blueprints. Sheet metal parts constitute a substantial number of those parts fabricated. A modelmaker, when given a blueprint, must calculate material allowances for all bends, add this data to the part dimensions, then transfer the resulting dimensions to sheet metal. The pattern must then be cut out, excess material trimmed, holes punched, and finally formed into the finished part. It is not uncommon to find mistakes made during the process, which translates into lost time, materials, and money. This has been the classical way to fabricate prototype sheet metal parts until now.

Through the application of the Flat Pattern Generator routine, prototype sheet metal parts may be developed quickly and accurately on the CV system. This routine will enable the user to generate three-dimensional parts as two-dimensional part representations.

There are two general methods within the command structure. Method 1 requires the operator to create a set of panels representing the side and/or flanges of a part, and Method 2 requires a line representation of the cross section of a part. The part geometry will dictate which method an operator should use.

3. PART PREPARATION

To implement this process, the part must first be built up in the data base. In Method 1, the appropriate dimensions are selected from a given blueprint and used to create the necessary panels. (See Figure 1)

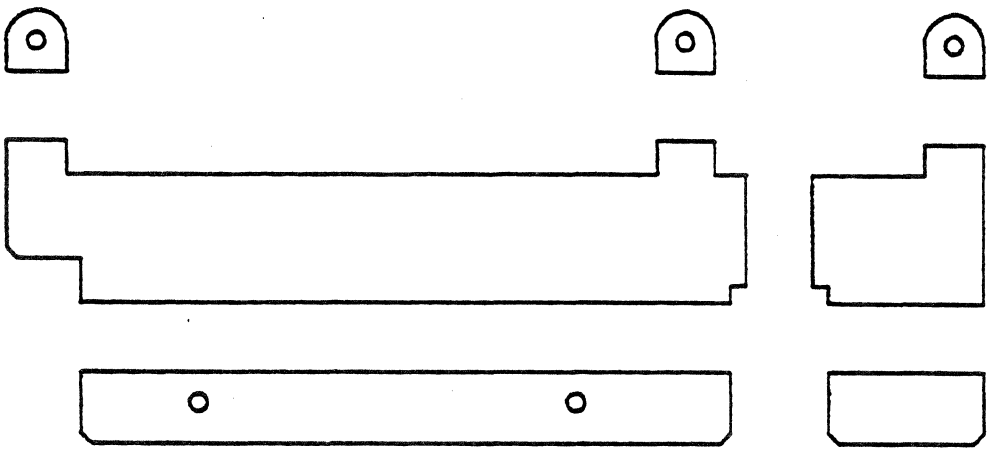


Figure 1

The size of the panels may vary according to which dimensioning modifiers the user may select. Some of the more significant modifiers include OD for outer dimensioning, ID for inner dimensioning, and RP for radial point dimensioning. These modifiers specify whether the dimensions of a panel are taken from a flat side, or from the intersections of the inner or outer surfaces of mating panels. (See Figure 2) Two other modifiers, REG or ACT, are used in conjunction with OD or ID when the angle of bend is greater than 90°. (See Figure 3) Additional modifiers, peculiar to Method 1, influence the positioning of one panel in relation to another and effect the resulting calculated bend allowance.

The DEPX.X modifier establishes the use of Method 2 and specifies the depth of the part in the Z axis.

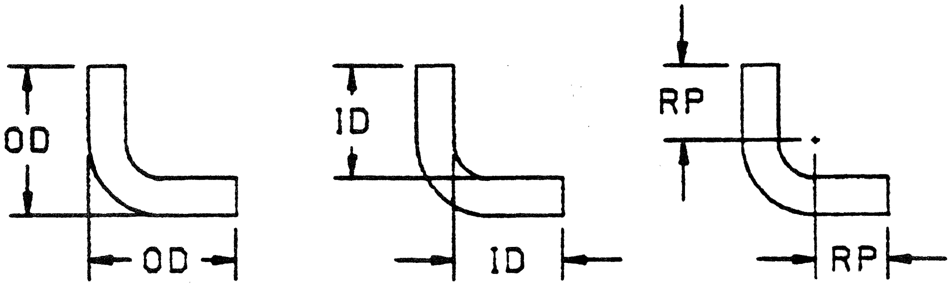


Figure 2

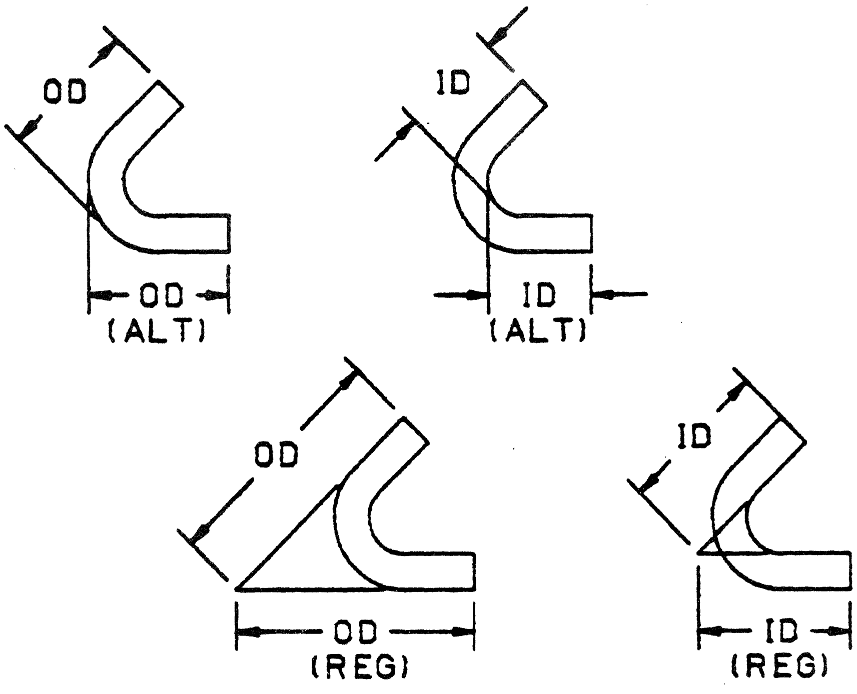


Figure 3

Modifiers common to both methods include the bend angle, the inner bend radius, material thickness, layering, and the K-factor. The K-factor may be changed by selecting a different modifier or hardness factor.

To better use system time, the panels should be sketched on paper and roughly dimensioned so that the transfer to CAD will take minimal effort. For the same reason, it is always a good idea to plan how the flat pattern will be joined together and write out the appropriate command sequences in advance. When all necessary preparations are finished, the panels can be joined together in the pre-arranged manner to produce the final flat pattern. (See Figure 4)

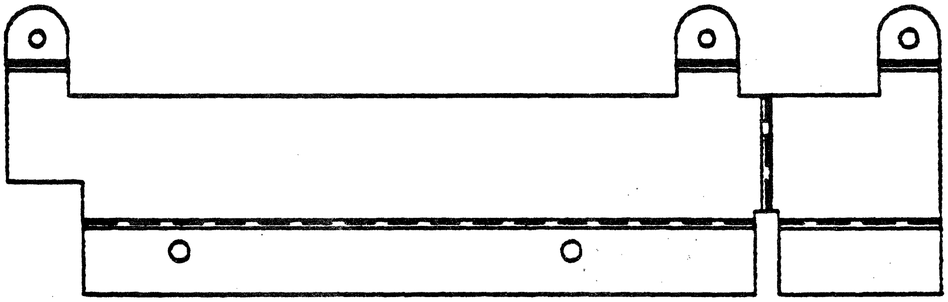


Figure 4

Upon joining two panels, the operator will find a dashed line at the bend junction. This line represents the midpoint of the bend radius. The lines representing the joined edges of the two panels are blanked out. These may be recovered and displayed or deleted if desired.

Method 2 may be used to develop a flat pattern, but its flexibility is limited. The profile view of a part, represented as a line and arc drawing, will generate its own thickness and the flat pattern upon execution of this command routine. (See Figure 5) In many instances, this method may be combined with Method 1 to create a pattern more efficiently than using only one method.

When the pattern has been completely developed it can be dimensioned for the benefit of the modelmaker. This would involve critical dimensions that provide reference for the formation of the part in the Model Shop. A hard copy or plot can then be obtained and provided to the modelmaker.

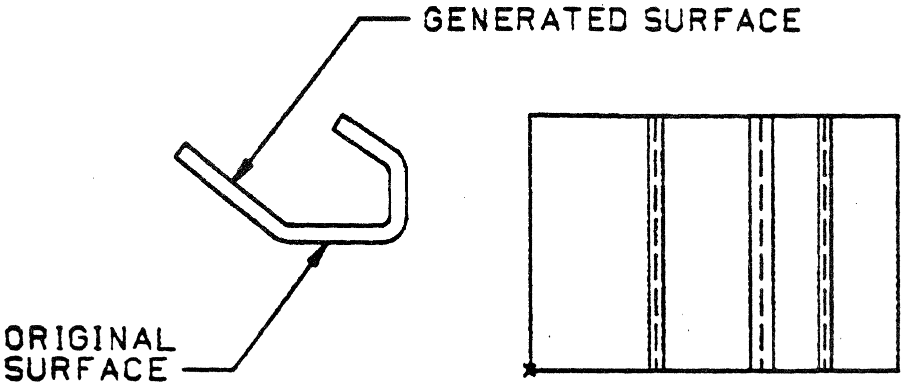


Figure 5

4. PART CREATION

Our original intent was to provide just the finished flat pattern, with the accompanying dimensions, to the shop personnel. However, it was discovered that it is possible to scribe sheet metal on our Xynetics 1101 table plotter. The Model Shop came up with a small scribe which was substituted for the ink cartridge in a pen holder. With this arrangement, as many copies of a pattern as desired can be scribed by the plotter. The modelmaker can then cut and trim the patterns, punch the holes, and bend the parts into shape. (See Figure 6)

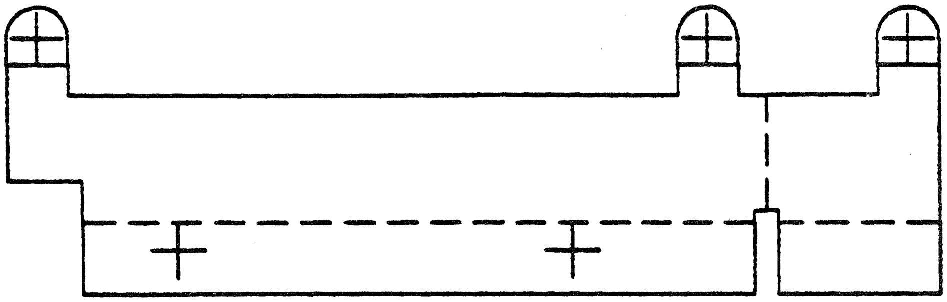


Figure 6

An analysis of the further potential of this process convinced us of the need to push further into the CAM area. As a result, the Model Shop has recently acquired two Strippit NC machines, a Laser Tool and a 75/39 Punch, to take full advantage of the CV software. Now that this NC equipment is installed, the potential for productivity gain is becoming highly significant.

With the addition of the NC equipment, it is necessary to generate NC toolpaths to cut the part patterns. This can be accomplished quickly and easily on the CV system. By applying the MPRO and MPTP routines, a simple program can be created to punch the holes, cut the profile and pockets, and scribe the bend lines of a part. (See Figure 7) Since multiple copies of a part are generally required, this one program can be stepped to cut as many parts as necessary.

To interface between CAD and CAM, a post processor is being prepared to convert the CAD NC program into language compatible to the NC controllers. This software program, when completed, will be loaded onto an in-house VAX 11/780 mainframe computer, to be accessed directly from the CV system.-The NC controllers can then interface with the VAX and extract the machine code. At present, it is necessary to send the APT source file to GE in Schenectady, through a post processor there, then back to Appliance Park where paper tapes are made and fed into the laser and punch tools. The modelmaker can then, using either method, punch and cut any many parts as required, bend them into shape, and the process is completed.

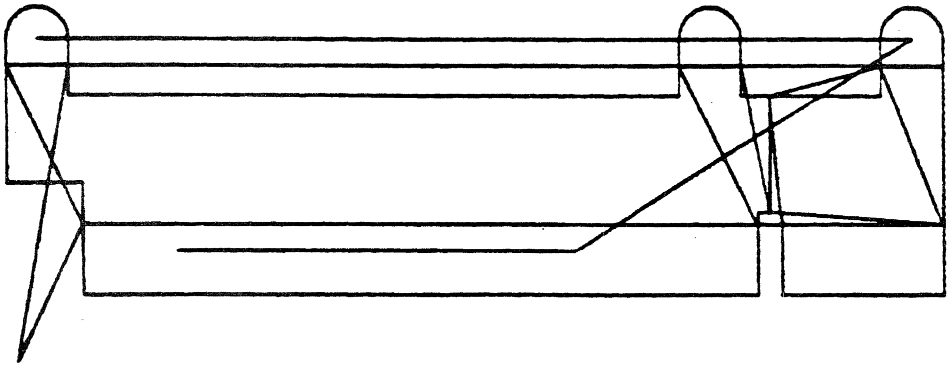


Figure 7

In addition to forming the part pattern, the laser can also be directed to scribe part data onto the part surface. This data, which may include such information as dates and project, model, part, and revision numbers, can be entered into the CV part file as nodal information. It may be transferred to the part surface by a set of subroutines and later extracted from the CV data base to provide updates of parts lists for inventory control and other purposes.

Range Engineering can take advantage of the speed and accuracy of this equipment by carrying out performance effects tests caused by the variation of the maximum and minimum tolerance limits to examine a product's construction and performance given those tolerance extremes. This can be very valuable in designing microwave ovens. Since microwaves are so sensitive to minute dimensional variations, this process can maximize the usage of a cavity and parts for total microwave cooking performance.

The initial expense for the strippits was a problem to overcome, but the return on that investment will be enormous. Current estimates predict a two-year pay back period for the NC equipment. This will enable the Model Shop to up their production rate from between 50% to 80% for flat metal parts. Further benefits can be realized by cutting nonferrous materials and performing services for other GE businesses in Appliance Park.

With the upgrading of shop facilities, the manufacturing process improvements will be extensive. Time and labor will be saved as computers do the routine work while freeing the modelmakers from mundane tasks and allowing them to use their talents in more critical areas. Human error will be largely reduced, and parts and models will be available sooner for test evaluations. In some cases, parts can be designed, produced and tested before the blueprint would ordinarily arrive in the shop.

While this technology is still fairly new to Appliance Park, it is another step towards integrating computer-aided engineering into the appliance business. This is compatible with GE's present efforts to bring the company into the forefront of the new technologies and promises a future of opportunity and modernization.

PC Circuit Design

MICROWAVE PRINTED CIRCUIT DESIGN PROCEDURE

Stuart Horwitz
Westinghouse Electric
Box 746
M/X 430
Baltimore, MD 21203
301-765-6937

ABSTRACT

Procedure has been developed for the creation of artwork for microwave printed circuits by design engineers. By the use of specially developed PEP programs, the designer is relieved of many routine tasks which would otherwise make the operation tedious and inefficient. These PEP programs take the input electrical design parameters, convert them into mechanical dimensions and configurations, and then construct the resulting two-dimensional graphical entities. At the same time, the electrical parameters are stored in the CADDs data base for later retrieval if desired. Artwork is then generated using the Gerber photoplotter.

The Operator-machine interface has been designed with the design engineer in mind, so that a minimum of computer Graphics experience is required, thus encouraging utilization by the casual user.

The procedure has proved quite valuable, both as an engineering design tool and a drafting tool. For the more commonly used microwave circuits a 10 to 1 productivity improvement is not an unreasonable expectation.

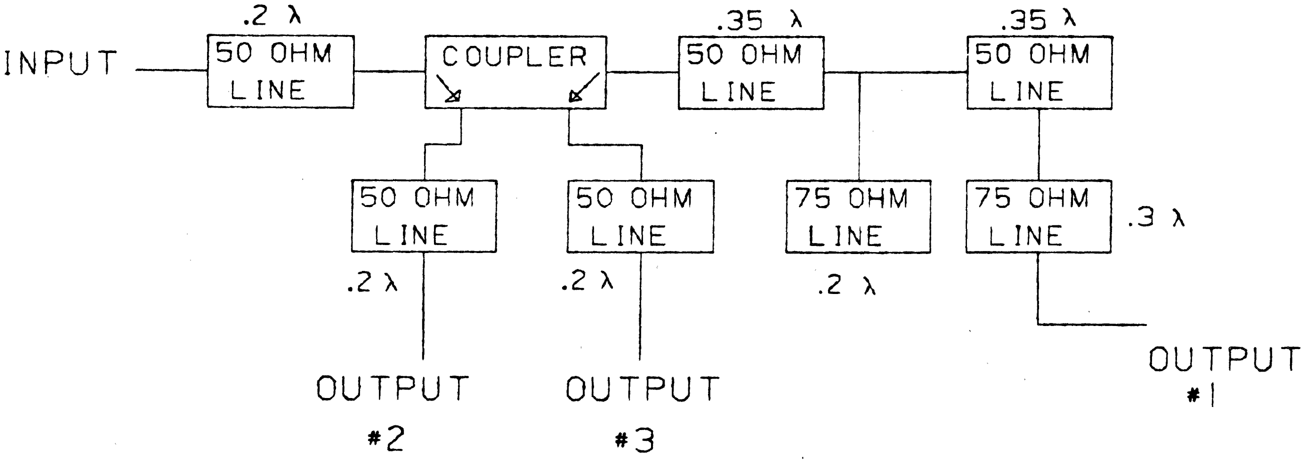
INTRODUCTION

A microwave printed-circuit board is similar to a single layer printed-circuit board in that both have a metallic conductor pattern etched onto a metallic clad dielectric material. The similarity ends there, however. Whereas, with a low frequency printed-circuit, the electrical components are the element which dictate the design, in microwave printed-circuits it is the configuration of the conductor pattern which is of primary concern. In fact, many microwave printed-circuit boards contain only a conductor pattern.

These conductor patterns form RF transmission lines of various configurations. To perform as microwave circuit elements, these lines must have an accurate and constant predetermined width, length, and separation from other conductors. Since the width must be constant, changes in direction require some sort of width compensation such as a miter.

A typical procedure for the design of a microwave printed-circuit board might be as follows:

1. The engineer does a theoretical design and produces a block diagram or schematic containing electrical parameters (see figure 1).
2. The electrical parameters are then converted to physical dimensions using procedures which require frequency and dielectric information.
3. A layout drawing of the conductor pattern and a photomask can then be made (Figure 2).
4. A device is fabricated.



TYPE OF CONSTRUCTION : STRIPLINE
CENTER FREQUENCY : 3.0 GHZ
GROUND PLANE SPACING : .125 INCHES
DIELECTRIC CONSTANT : 2.3
CONDUCTOR THICKNESS : .0015 INCHES

Figure 1: Microwave Device Block Diagram

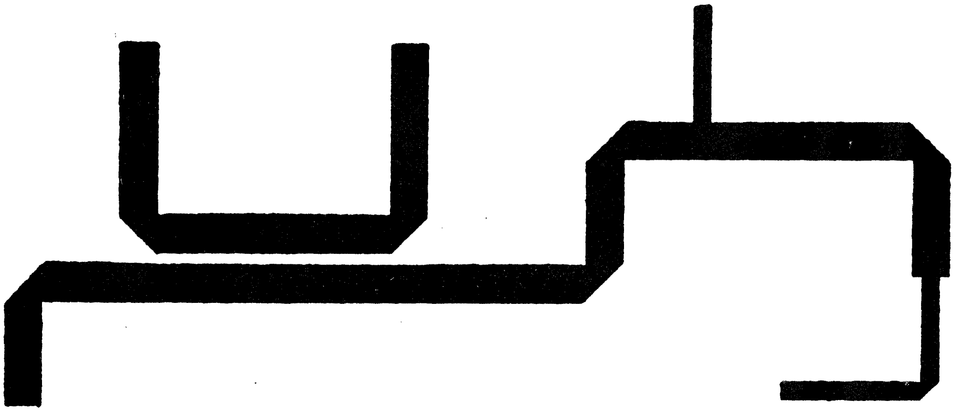


Figure 2: Photomask

At this point, the designer has a breadboard device. Evaluation of this breadboard may indicate that design changes are required. If this is so, the entire process must be repeated, although portions of the previous effort may be salvage using a cut-and-paste technique.

Moving from a working breadboard to a production design also usually requires a relayout of the circuit, to conform to mechanical constraints. Again, the entire process must be repeated. At each iteration, there is often a degradation in dimensional accuracy. During all these operations, the electrical parameters are buried in the designers notebook.

The object of the procedure described in this paper is to streamline the design process, while maintaining dimensional accuracy, by using the Interactive Graphics System to perform steps 2 and 3. At the same time, electrical design parameters are stored so as to be useful in future design iterations.

DESCRIPTION OF THE PROCEDURE

This procedure utilizes the Gerber photoplotter to produce the etching mask, from information generated by the Gerber post-processing program, although this procedure would be applicable to other artwork generation devices with minor modifications. The post-processor will only process closed strings or nodal lines, which tends to indicate that the conductor pattern layout should be performed using these entities only. It was found to be much more efficient, however, to create the patterns using lines and then to use the "Generate String" command to create the required strings from these line figures. Care must be taken when creating the line pattern so that the generated strings will be acceptable to the Gerber post-processor.

The key to the procedure is the use of a series of PEP language programs which do the actual graphical construction. PEP was chosen partly because of its similarity to FORTRAN, which made it possible to write programs having a high degree of user interaction, and partly because the documentation appeared to be more complete than PAREX or VARPRO.

These PEP programs can be divided into three groups: Initialization, RF transmission line construction, and RF component construction. The first two groups contain only one program at present, while several have been written which fall into the third group.

To understand the difference between the program groups, let us first look at the properties which affect the conversion of electrical parameters into physical dimensions. These are:

1. Type of microwave circuit construction. Only two types are presently allowed, microstrip (single ground plane) or stripline (two ground planes).
2. Dielectric constant of the board material.
3. Center frequency of the device.
4. Ground plane spacing or board thickness.
5. Thickness of the conductor metalization.
6. Miter definition (how much of a corner is cut away?).

Usually these properties, once determined, do not change for a given part. Therefore, it is desirable to enter them only once during construction and to have them stored within the data base. This is the function of the initialization program.

This program requests that the operator enter the above information. The data is checked for validity and then scaled to produce numbers between 0 and 1. These numbers become coordinates for two graphical points which are constructed on an otherwise unused layer (200 in this case). These points are labelled so that their coordinates can be obtained and decoded by other PEP programs. To use the remaining programs in this procedure, this program must be run. Also, it can be run only once for a given part.

The second program group, RF transmission line construction, contains the most versatile of the PEP programs. The program currently in use is primarily intended to construct graphical representation of RF transmission lines, given the electrical parameters of impedance and wavelengths. However, it also serves as a utility program to perform several other functions necessary to the overall procedure.

Before describing the program's operation, it is necessary to define the graphical figure which will be produced. This figure, an RF transmission line figure or RFIG is defined as follows:

A. It consists of a collection of lines connected end-to-end to form a closed figure (no overlaps or breaks).
B. No two adjacent lines are co-linear.
C. Lines are constructed so that the end point of one is connected to the start point of the next.
D. An identifying point containing length and width (or impedance if applicable) data is located at the center point of the first segment of the RFIG.
E. One or more points to be used for connection to other RFIG's is located on layer 100.
F. All lines and the identifying point are on the same layer and are grouped with the connect points.

The first three properties are required so that the string generated from the RFIG will be acceptable to the Gerber post-processor. The last two properties are aids for use during graphical construction. The remaining property is intended for storing electrical parameters of the RFIG in the data base. A typical RFIG is shown in figure 3.

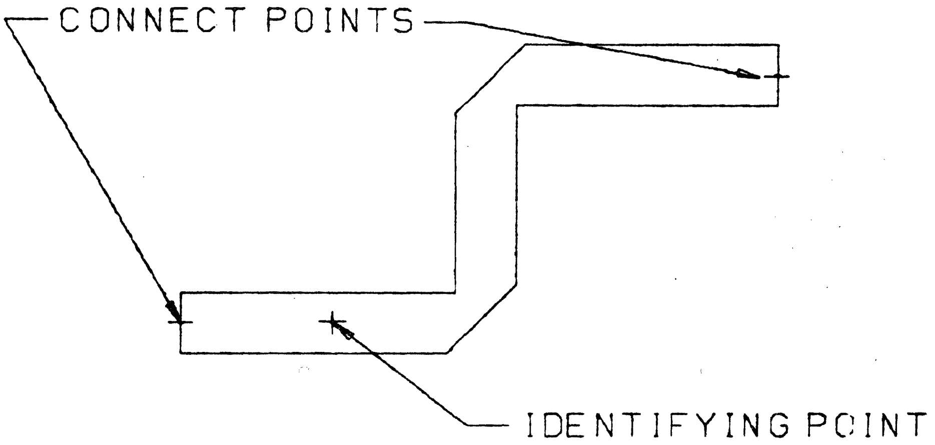
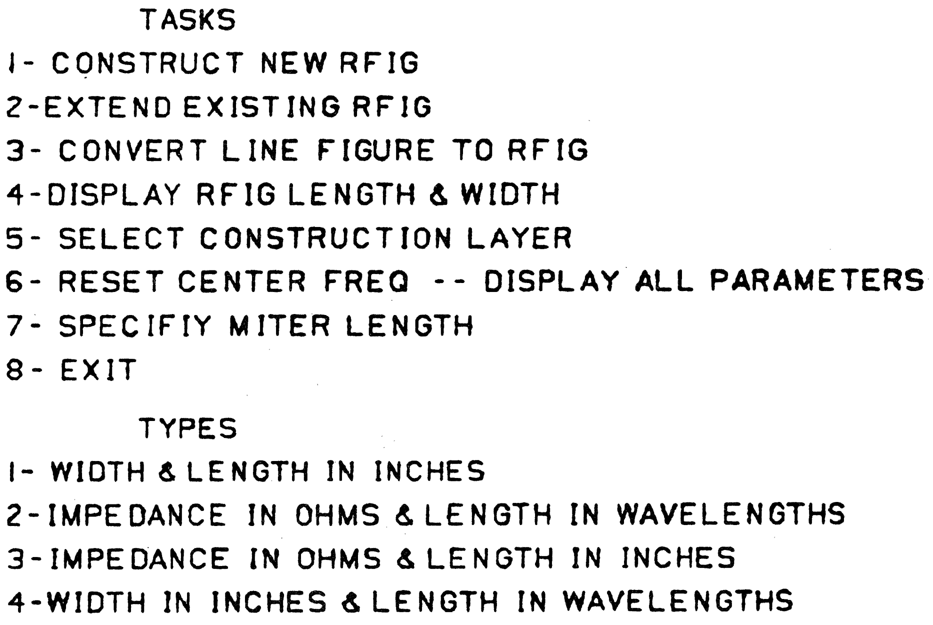


Figure 3: RF Transmission Line Figure (RFIG)

The functions of the construction program can be seen from figure 4. The program is divided into a number of "Tasks", one of which must be selected by the user. The user must also select a "TYPE" number to define the form of the RFIG length and width data. This program creates RFIG's having vertical and horizontal segments only, with a mitered corner at each bend.



```
            TASKS
1- CONSTRUCT NEW RFIG
2-EXTEND EXISTING RFIG
3- CONVERT LINE FIGURE TO RFIG
4-DISPLAY RFIG LENGTH & WIDTH
5- SELECT CONSTRUCTION LAYER
6- RESET CENTER FREQ -- DISPLAY ALL PARAMETERS
7- SPECIFIY MITER LENGTH
8- EXIT
            TYPES
1- WIDTH & LENGTH IN INCHES
2-IMPEDANCE IN OHMS & LENGTH IN WAVELENGTHS
3-IMPEDANCE IN OHMS & LENGTH IN INCHES
4-WIDTH IN INCHES & LENGTH IN WAVELENGTHS
```

Figure 4: RF Transmission Line Construction Program

The third type of program, RF component construction, requires the least amount of operator intervention. There is a separate program of each type of component, such as directional couplers, power dividers, filters, etc. In each case, the conductor pattern is pre-defined, with the actual dimensions being calculated from the data obtained from the initialized parameters and occasionally operator input. All programs require selection of the construction layer and location of the origin of the figure. A typical component is shown in figure 5.

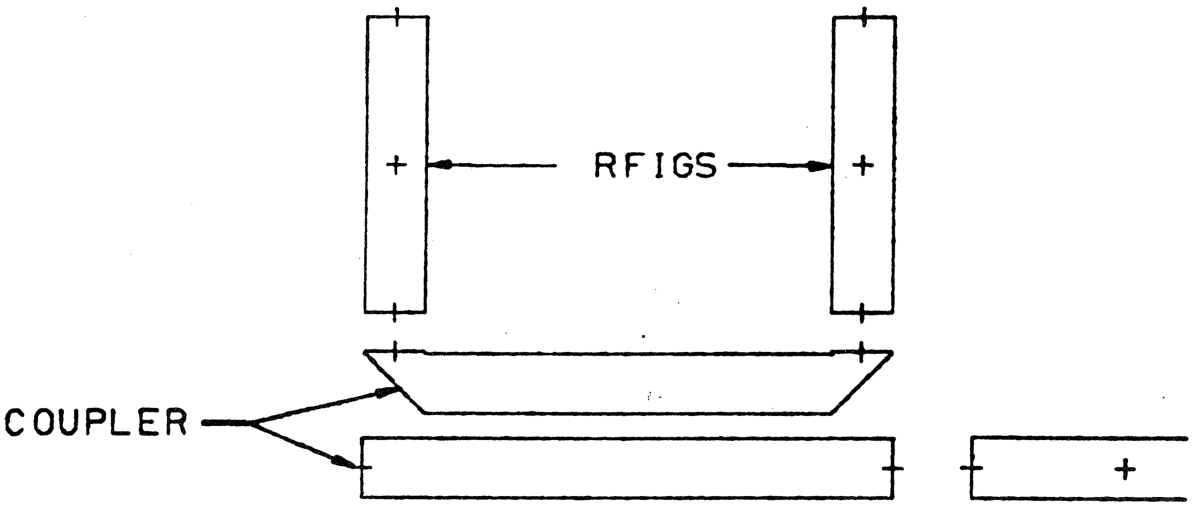


Figure 5: Microwave Component & Interconnects

PART CONSTRUCTION PROCEDURE

To generate artwork for the circuit of figure 1, the procedure might be as follows:

1. Run the initialization program, entering the parameters shown in figure 1.
2. Run the RF transmission line program. Select the desired construction layer.
3. Select Task 1, Type 2.
4. Starting from the right side of figure 1, begin RFIG construction.
   a.) Enter impedance and length in response to program requests.
   b.) Digitize the location of the center points of the RFIG. When the required length has been reached, the RFIG will be generated and the task terminated.
5. Generate additional RFIG's sequentially, using connect points to locate the origin of each succeeding one. After the first four RFIG's have been constructed, the configuration might appear as shown in figure 6.
6. Exit from the program.
7. Construct the directional coupler by executing the appropriate component construction program. Select the construction layer, enter the coupling value and digitize the origin location.
8. Construct the remaining RFIG's as before. The complete circuit is shown in figure 7.
9. Using the "Generate String" command with the "Group" entity mask, create a copy of the circuit in which all RFIG's are replaced with closed strings.
10. Execute the Gerber post-processor and obtain a photoplot.

At this point, the user has the finished artwork and can proceed to fabricate and evaluate his device. Modification of this conductor pattern is done by deleting, re-constructing, and translating RFIG's and strings. In this way, the conductor is shuffled around to meet the desired electrical and mechanical requirements. At no time during this operation is there a loss of dimensional accuracy. Also, at any time, the designer may retrieve any or all of the electrical parameters from the data base by using tasks 4, 6, or 7 of the RF line construction program.

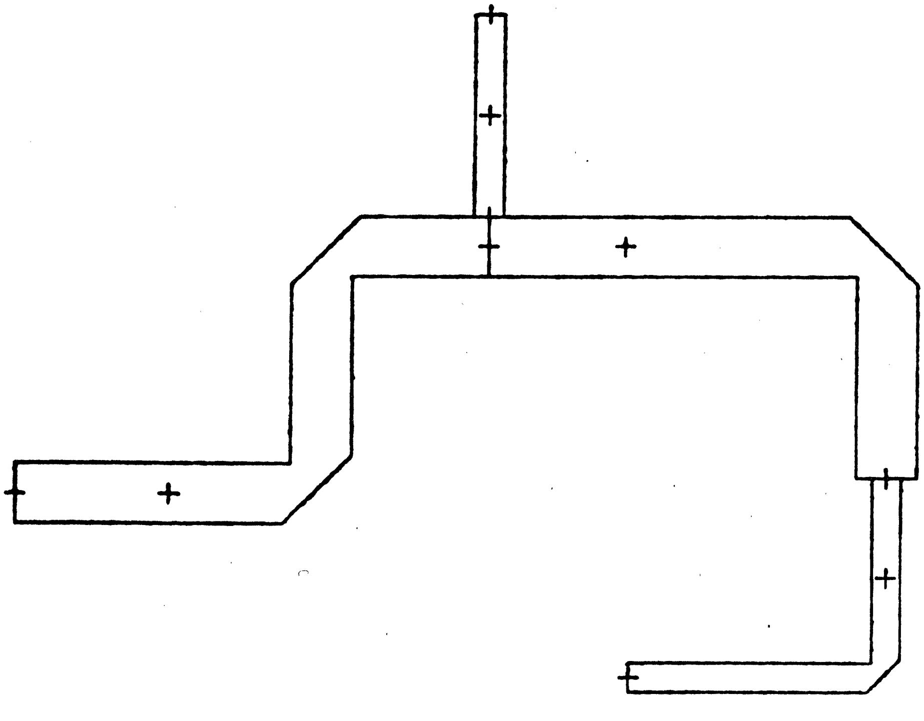


Figure 6: Typical RFIG Interconnects

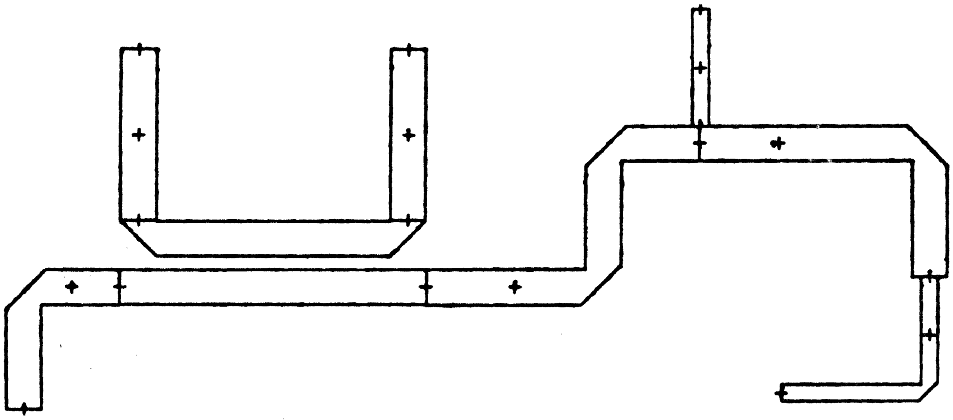


Figure 7: Completed Microwave Circuit

CONCLUSION

It can be seen from the foregoing that the entire operation can be performed with the use of only about a dozen graphical commands. This factor is important for the casual user who is not interested in becoming a graphics expert, but requires an expedient method of fabricating microwave parts.

The effectiveness of this procedure can be illustrated using the circuit of figure 1. To generate artwork manually to the required accuracy would take three hours of drafting effort and one hour of engineering. Using the computer graphics procedures, a design engineer could have finished artwork with less than 30 minutes of effort.

ELECTRO-MECHANICAL PACKAGING OF ELECTRONIC PRODUCTS

Dr. Itaru Aramaki
Manager, Products Design
Systems & Electronics Division
Sumitomo Electric Industries, Ltd.
Osaka, Japan

1. INTRODUCTION

In the field of systems and electronics, Sumitomo Electric Industries, Ltd. has an established reputation as a leading manufacturer of vehicular traffic control systems, especially as the builder of the world's largest traffic control and surveillance system for Tokyo.

Recently we have also been gaining acceptance as the foremost manufacturer of various other electronic products such as bus location systems, industrial crane control systems, and hand-scan optical character readers.

In order to design and manufacture these electronic products assuring higher quality, on shorter schedules, and at lower costs, we have successfully built a CAD/CAM system around a stand-alone interactive graphics system.

Our CAD/CAM system:

(1) Semi-automates electronic and structural design of our electronic products.
(2) Automatically generates production data for the production lines of printed circuit boards and enclosures such as housing cabinets, rack-mounted chassis, operator consoles, and instruction panels.

In the past our engineers and draftsmen worked long hours with paper and pencil, hunched over drawing boards.

Now they proceed with their work quickly in front of CRT displays, giving instructions to and being prodded by the system.

When the design stage is over, the system generates essentially automatically manufacturing drawings, punched paper tapes, and magnetic tapes for numerically controlled machines, as well as bills of materials and other documents necessary for procurement purposes.

Since data on typical electronic components and structural parts are stored in the computer as a common data base, engineers and draftsmen can use them in common, bringing them easily to the CRT display. They can thus finish their design work accurately and on shorter schedules without constantly referring to detailed component brochures, without making individualistic choices, and without making transcriptional mistakes while picking up dimensions from the brochures.

2. MANUFACTURING DESIGN OF PRINTED CIRCUIT BOARDS

2.1 Compiling A Circuit Diagram

The engineer expresses his idea of how to realize a specified function in one or more rough sketches, using various functional blocks which indicate either integrated circuit components or some standardized functional blocks.

The rough sketches have to be prepared manually on the engineer's desk at the present state of our CAD/CAM system.

When the rough sketches are ready, they are put into the system by means of a digitizer/plotter.

After the rough sketches have been stored in the data base, the technician works on them to complete one detailed circuit diagram corresponding to one printed circuit board at a CRT work station.

He works interactively with the system, editing, modifying, and calling up standardized functional circuit blocks from the data base to complete a detailed circuit diagram for one printed circuit board.

The detailed circuit diagram is now stored in the data base.

Our next job is to check the validity of this circuit diagram: if it really functions logically as the engineer intended.

2.2 Testing A Logic Circuit Diagram

The detailed circuit diagram compiled in the previous stage, which in our case is mostly digital, is taken out of the data base in a form suitable for a logic function tester on a punched paper tape and logic simulation is carried out.

Corrections of mistakes can be done easily on the CRT with resultant automatic changes in the data base.

Prior to the introduction of our CAD/CAM system, logical mistakes could not be found until after physical printed circuit boards were manufactured and actually switched on.

When the mistakes are found and corrected, the data base is updated to store the final, clean error-free circuit diagram.

2.3 Detailed Manufacturing Design of A Printed Circuit Board

The conversion of the circuit diagram into manufacturing data starts now.

As the first step, the functional elements in the circuit diagram are semi-automatically allocated to physical, commercially available integrated circuits and other electronic components.

When all the functional elements of the circuit diagram have been allocated to actual electronic components, the system determines how to connect them electrically on the printed circuit board.

2.4 Producing Manufacturing Data for A Printed Circuit Board

Our CAD/CAM system now produces a complete set of manufacturing data for the printed circuit board.

Aside from the manufacturing data for the machines, various documents are also automatically generated, including a list of electronic components.

Last but not least, a complete circuit diagram is produced which incorporates all the modifications which have become necessary in the process of conversion from a conceptual circuit diagram into detailed manufacturing data as well as the actual names and pin numbers of the electronic components on the just completed physical printed circuit board.

3. STRUCTURAL DESIGN OF ELECTRONIC PRODUCTS

So far we have looked at what our CAD/CAM system can do in converting functional ideas into printed circuit boards.

The other important aspect of our CAD/CAM system is the structural design of our electronic products.

3.1 Conceptual and Detailed Structural Design in Three Dimensions

Fig. 1 shows a traditional two-dimensional manual drawing of a rack-mounted chassis, in which printed circuit boards are mounted.

Starting from this drawing, a great number of detailed drawings had to be worked out for all the parts which constitute this chassis.

This was before our CAD/CAM system came into play.

The structural design of our electronic products is done in three dimensions on the CRT screen as shown in Fig. 2.

Since typical components like the connectors and pushbuttons in Fig. 2 are stored in the computer as a common data base, they can be easily brought to the CRT screen, without constantly referring to detailed component catalogs.

Perhaps the most important feature here is that our engineers can now clearly visualize in three dimensions what they are trying to accomplish.

The engineer's job is essentially over when he has created a three dimensional figure like the one shown in Fig. 2 and stored it in the data base.

All the manufacturing data are semi-automatically produced from the three dimensional data base.

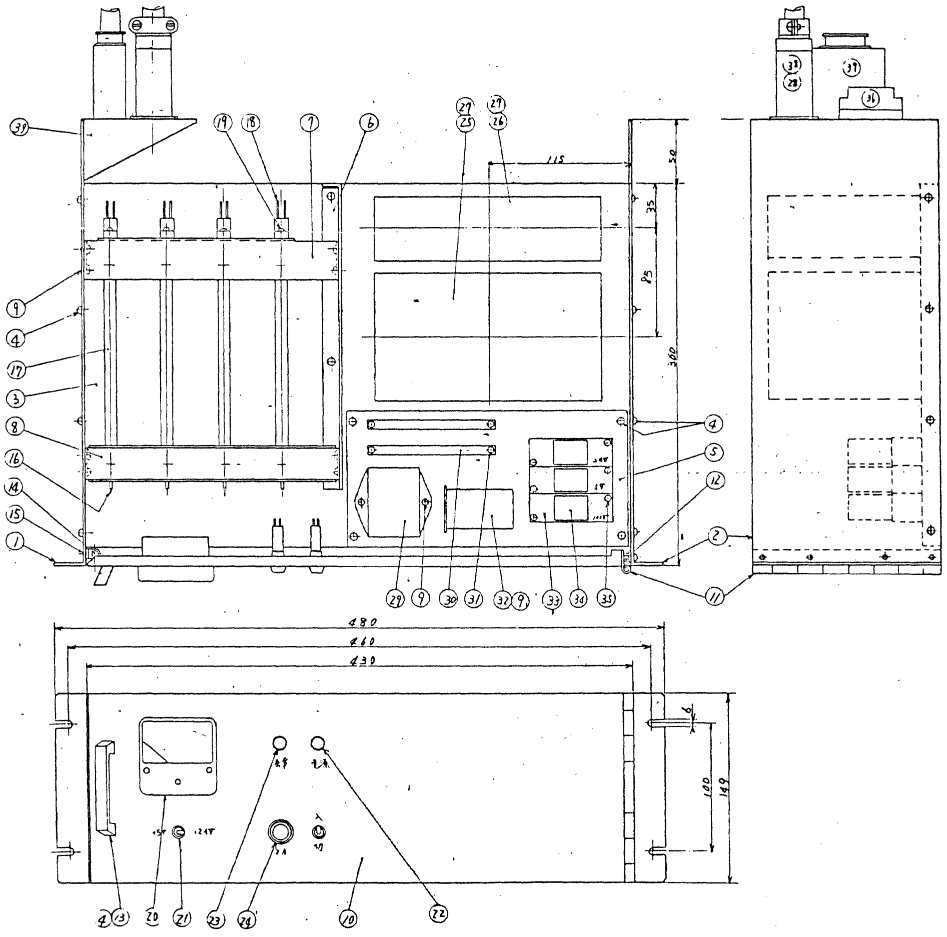


Figure 1: A traditional 2D manual drawing of a chassis

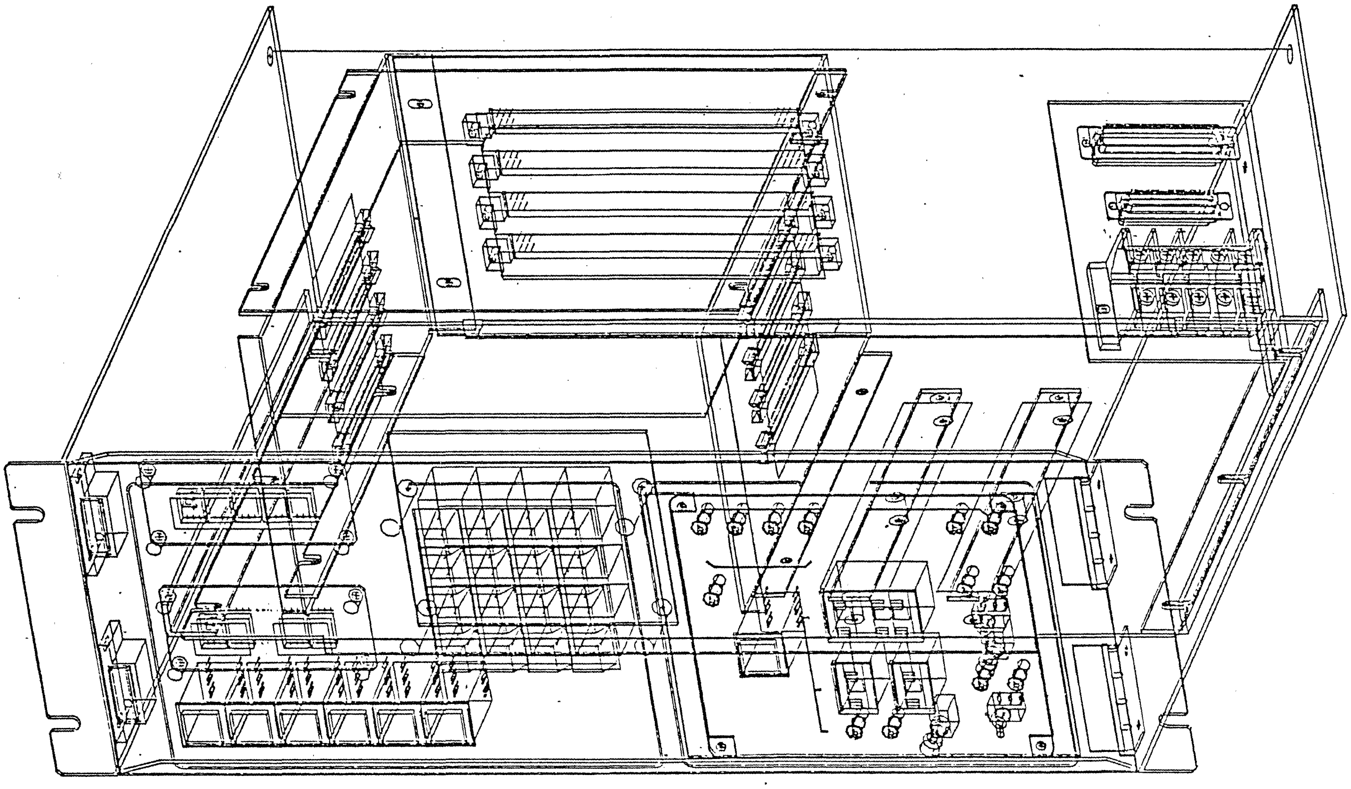


Figure 2: The structural design of a box in 3D

3.2 Producing Manufacturing Data

When the three dimensional structural design is completed and stored in the data base, the technician takes over the job from the engineer.

He is now ready to generate manufacturing data from the data base prepared by the engineer.

For example, a three dimensional piece in Fig. 3 is reduced to a manufacturing drawing at the bottom of the figure which directly corresponds to a punched paper tape. The cost of manufacturing this piece is also automatically tallied, counting the number of bendings, studs, taps, spot weldings, and so forth.

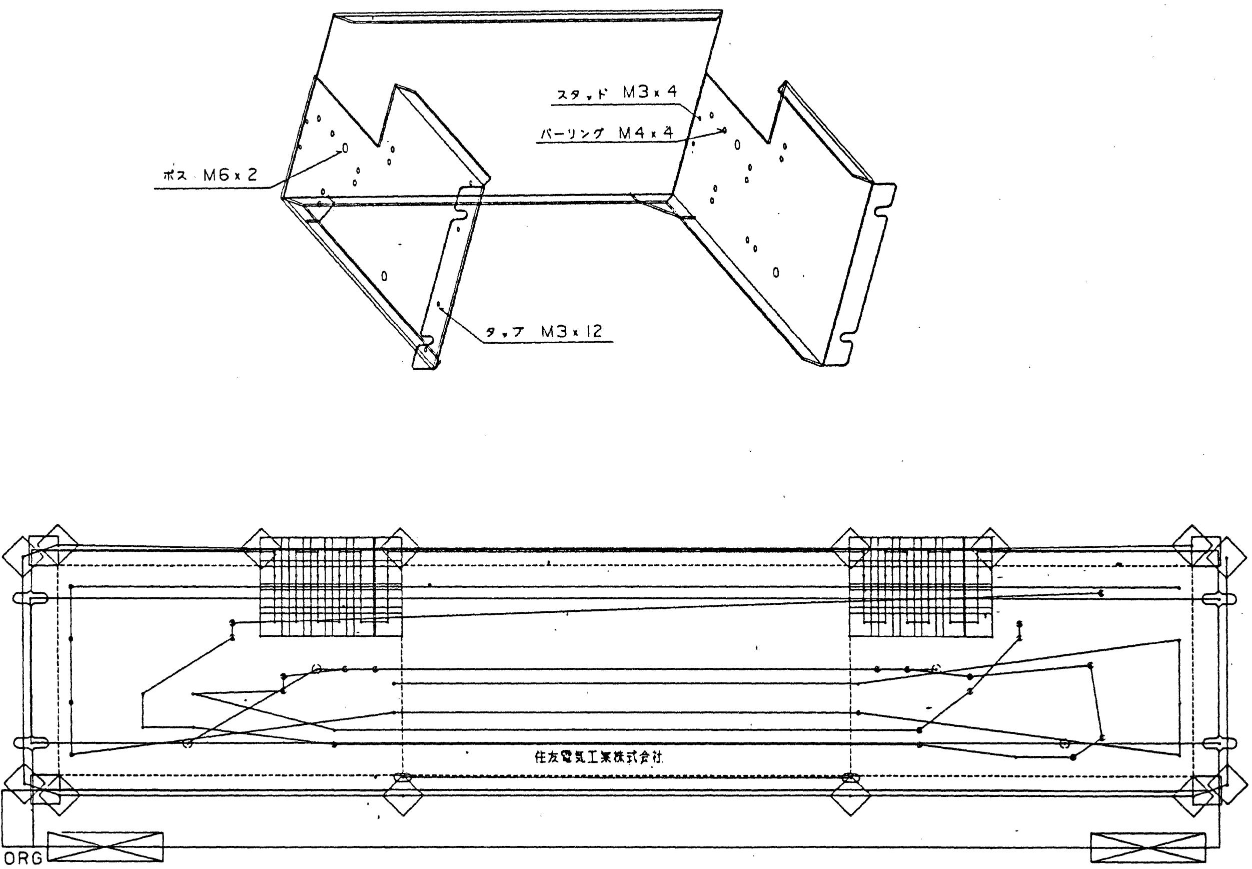


Figure 3: A 3D piece is reduced to a manufacturing drawing

4. THE SYSTEM CONFIGURATION OF OUR OWN CAD/CAM SYSTEM

Fig. 4 shows the entire system configuration of our CAD/CAM system. It's a kind of hodge-podge of things American, Japanese, in-house and outside.

The IAG is the interactive graphics computer.

The lower half, as it were, represents CAD, while the upper half represents CAM.

Our CAD/CAM system has three major features:

(1) Based on a stand-along system, our "application-ware" has completely adapted the off-the-shelf software to our needs.
(2) Structural design of electronic products is done in three dimensions, consistently from the conceptual design level down to the flat pattern layouts of metallic sheets.
(3) Starting from rough sketches, electronic circuit diagrams are edited, checked, and converted into a complete set of production data for printed circuit boards in a consistent and semi-automatic flow.

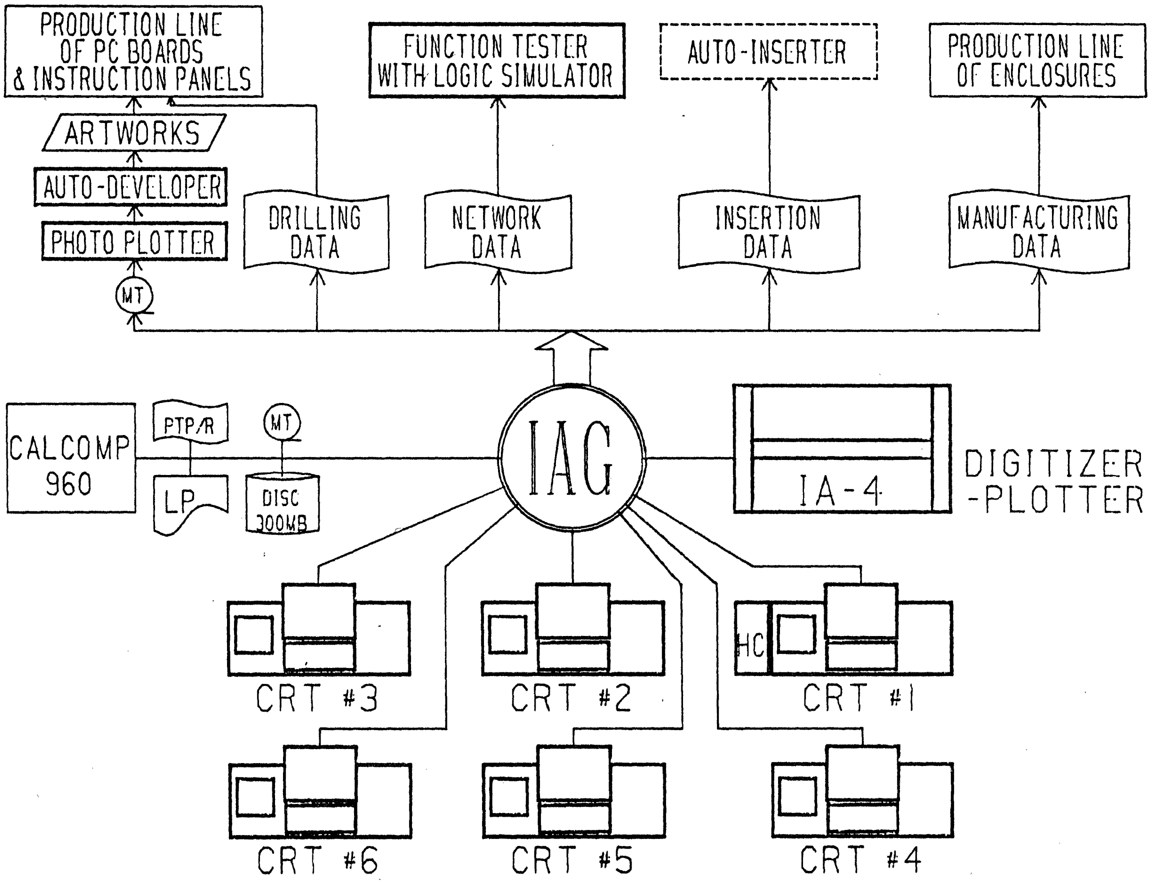


Figure 4: The entire system configuration

5. CONCLUSIONS

Although our CAD/CAM system has been in full operation for only seven months, the benefits seem to surpass our initial expectation.

Among other things, first, design errors have drastically decreased. Second, design and production schedules are getting shorter. Third, design standardization through the accumulated data base is fast becoming a reality.

In fact, our motto, "By CAD/CAM out of labor-intensive work" seems to be amply justified even in the first seven months.

We are determined to further computerize our design and manufacturing activities so that we will be able to supply still better electronic products at lower prices and on shorter schedules.

WIRE-WRAP BOARD DESIGN USING A COMPUTERVISION CADDS 3 SYSTEM

Anthony Minniti
Xerox Corporation
800 Phillips Road
Bldg. 207A
Webster, NY 14580
716-422-8494

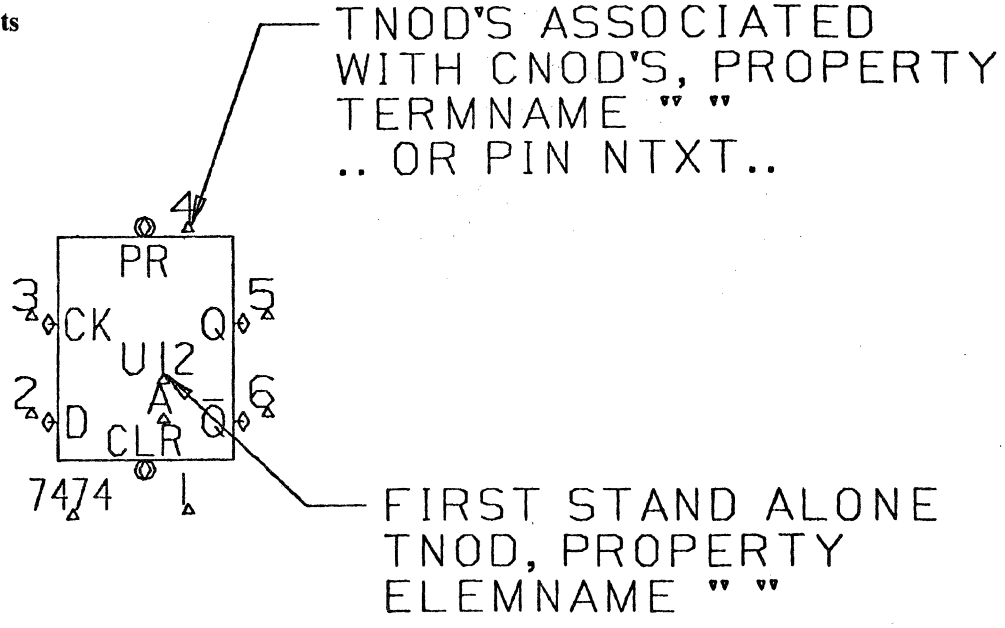
ABSTRACT

Labor cost savings of 10 to 1 have been achieved in the assembly and wiring of wire-wrapped boards. The technique utilizes a custom-made Computervision post-processor that merges schematic and board assembly information to create an N/C tape for a semi-automatic wire-wrap machine.

This paper will discuss the board assembly and schematic structure, voltage and ground busing, and component hook-up techniques. A method for obtaining a cut wire bin table and a wire from — to listing of the N/C tape that is used on the wire-wrap machine will also be presented.

Design Requirements

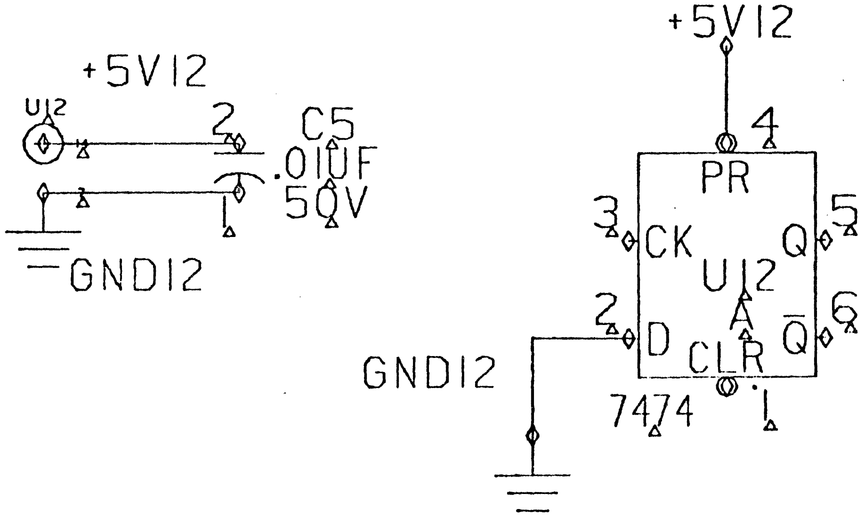
Schematic Symbols



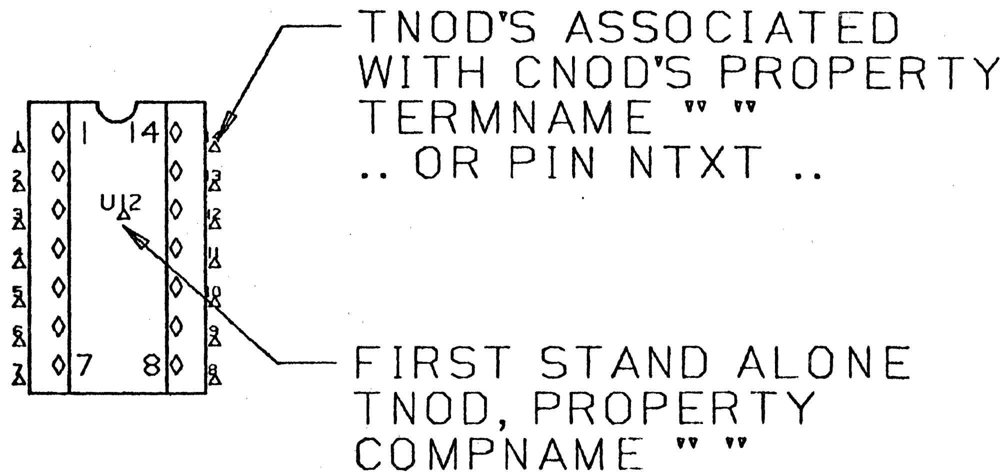
Design Requirements

Schematic Drawing

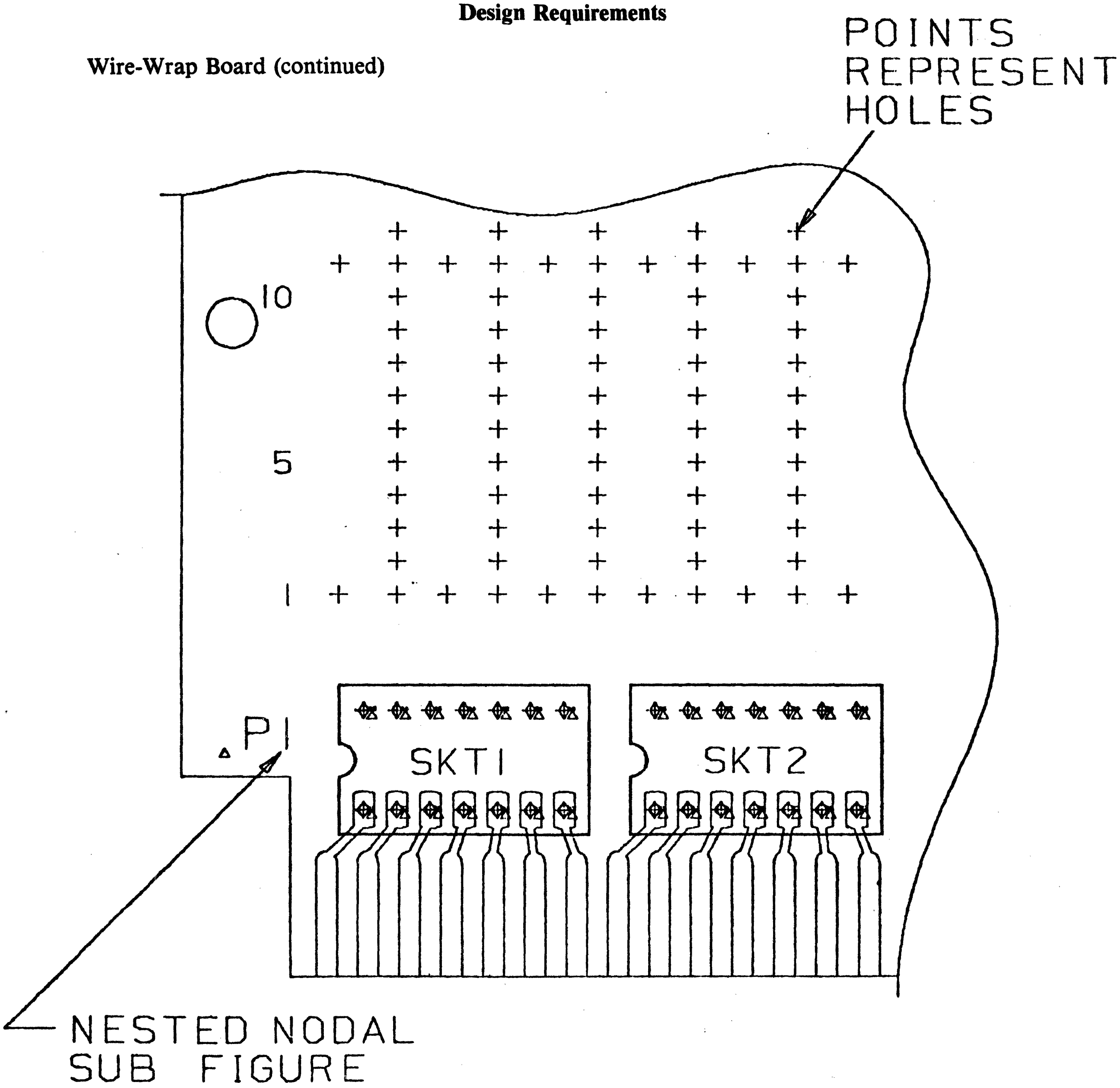
A. Stand alone TNOD, property sheet " ".
B. Network NTXT 20 characters max. (no spaces).
C. Busing techniques.



Wire-Wrap Symbols

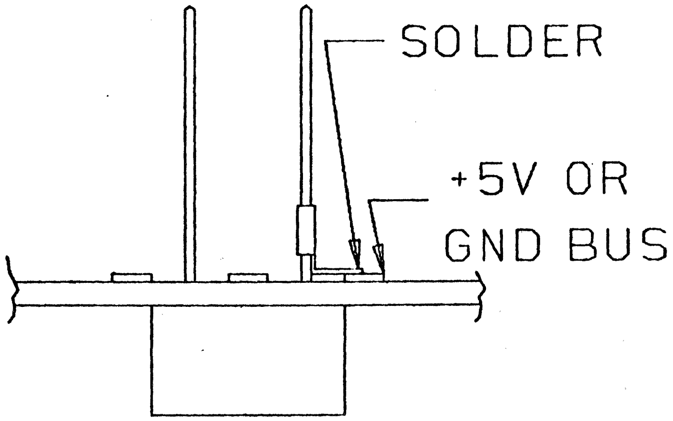

1
14
U12
7
8
1
2
3
4
5
6
7
14
13
12
11
10
9
8
TNOD'S ASSOCIATED WITH CNOD'S PROPERTY TERMNAME " " .. OR PIN NTXT ..
FIRST STAND ALONE TNOD, PROPERTY COMPNAME " "

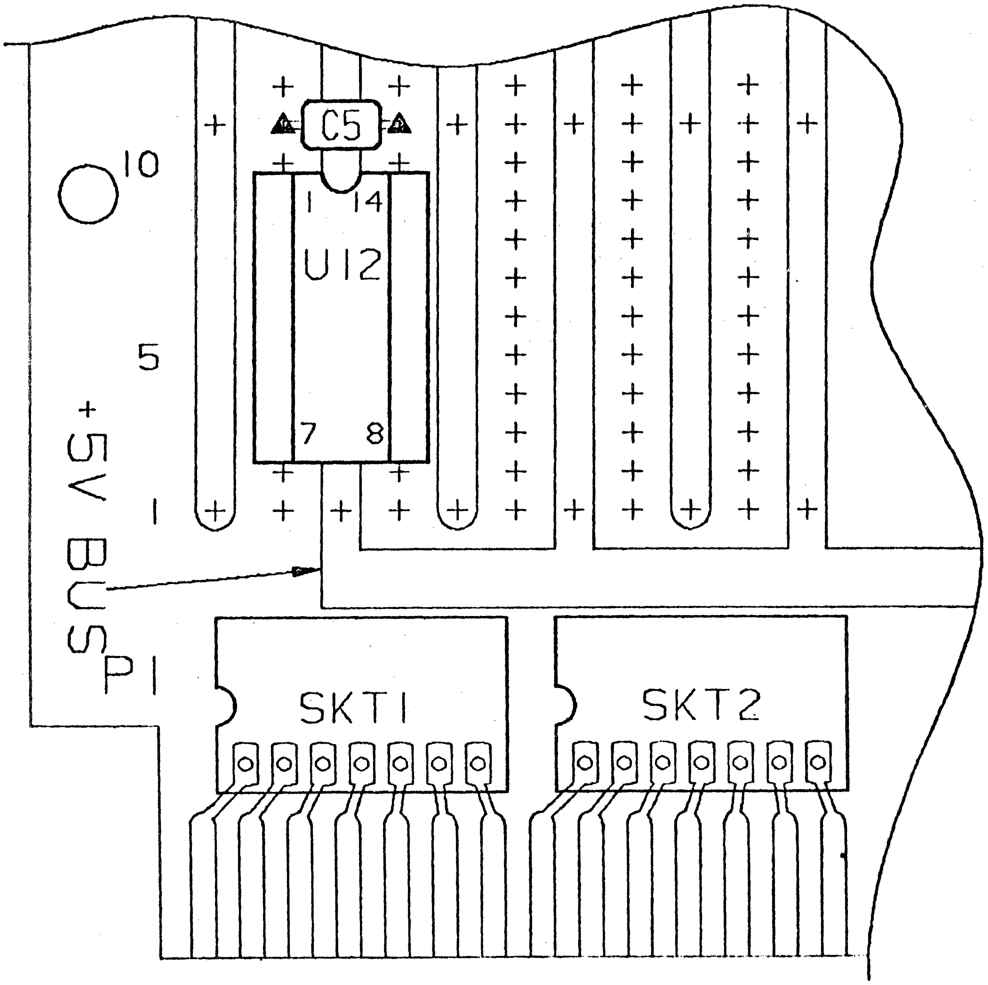
Wire-Wrap Board (continued)



Wire-Wrap Board (cont.)

A. XOYO location upper left (preferred).
B. Locate wire-wrap pin at XOYO if possible.
C. Use of bus strips.





Other Requirements

Hardware

A. Paper tape punch.

Software

A. PAREX
B. Data extract
C. Format net
D. Sort file
E. Merge file
F. PUT wire
G. CVTECO
H. PEP

File Structure

A. Schematic drawing sheets EL.99T00221-C101.SHIA thru SHXA
B. 1 to 1 wire-wrap board EL.99T221-C101.WWBDA

Use of PAREX Program
(Fig. 1)

Responses to 'Params'

A. DWGNO	99T00221-C101
B. REV	A
C. WWDWG	99T221-C101
D. SHTS	SH2A,SH3A
E. WIRE	#28

Explanation of program (Fig. I)

A. A multi sheet extract of the schematic is made (see Fig. 2).
B. A formatted net list is derived from 'A' above (see Fig. 3 and II).
C. A X-Y coordinate extract of the wire-wrap board is made (see Fig. 4 and 10).
D. The formatted net list in 'B' above is merged into the wire-wrap board and the PUT wire command is given.
E. CVTECO and PEP are used to clean up the machine code, obtain a cut wire bin listing, and a from-to listing of the machine code (see Figs. 5 through 9).

Checking the Results

Files to print

A. EL.WW99T00221-C101A.TRACE
B. EL.99T221-C101A.TRACE
C. EL.WW99T00221-C101A and EL.WW99T00221-C101AA (net list and formatted net list of schematic)
D. EL.99T221-C101.HFO (errors during merge)
E. EL.99T221-C101.HF (machine code to be output)
F. EL.99T221-C101.BIN (cut wire bin listing)
G. EL.99T221-C101.PEPI (from-to listing of drive tape)

Processing the Results

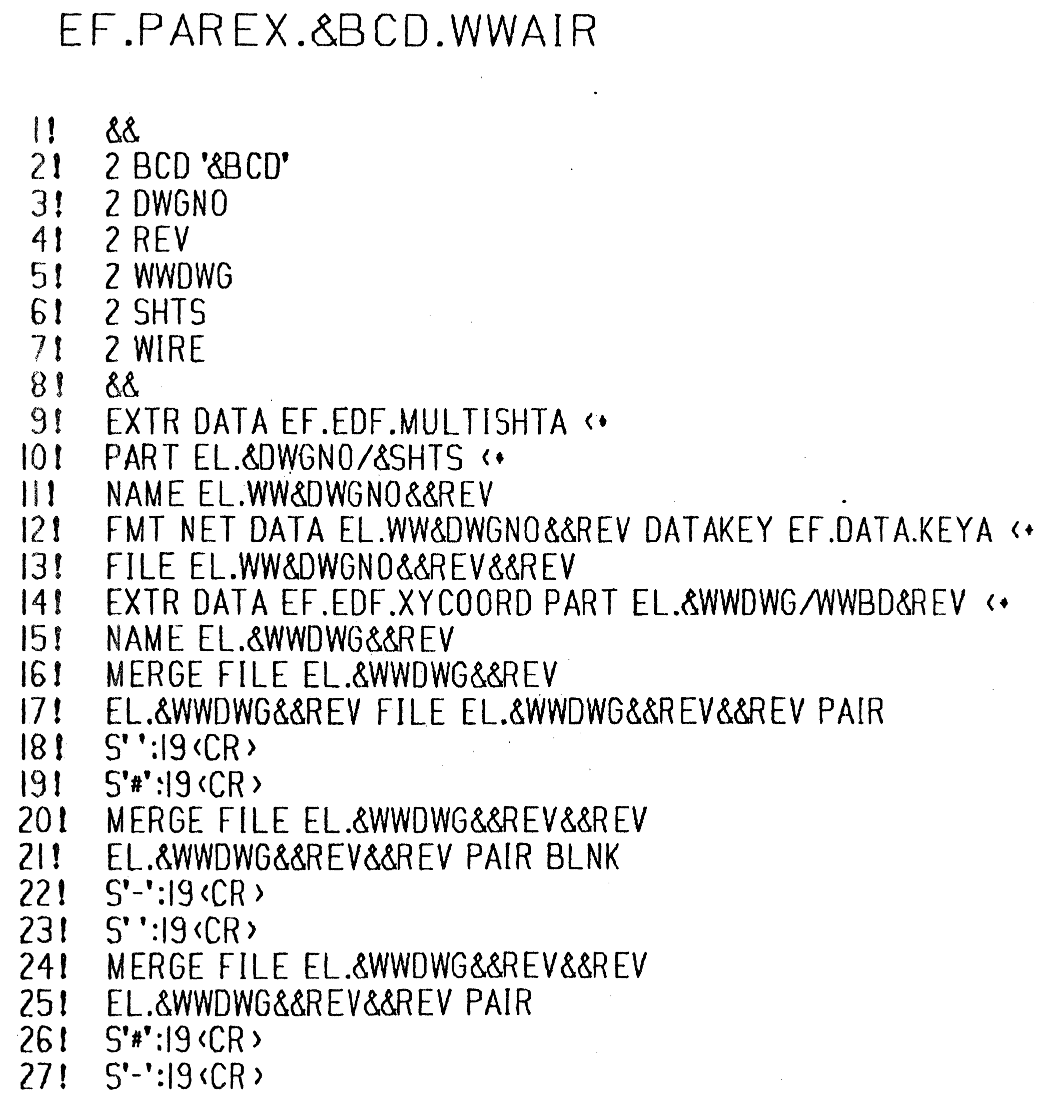
A. PRNT EL.99T221-C101.BIN/NOHEAD,DIGITS=0 (see Fig. 7)
B. PRNT EL.99T221-C101.PEPI/NOHEAD,DIGITS=0 (see Fig. 8)
C. EXECUTE EL.PUNCH99T221-C101 (punch new tape)
D. Delete the following files after final tape-off.
DELFIL EL/99T221-C101AA
DELFIL EL/99T221-C101
DELFIL EL.&BCD/WW99T00221-C101AA
DELFIL EL.&BCD/99T221-C101AA
DELFIL EL.&BCD/99T00221-C101A

Acknowledgements

George Cook — Manager, Tool Engineering and Services, Xerox Corporation.

Tony Affuso — Manager, Electrical Test Equipment and Computer Graphics System, Xerox Corporation.

Charles Schelemer — Application Engineer, Contact Systems Inc., Danbury, Conn. 06810



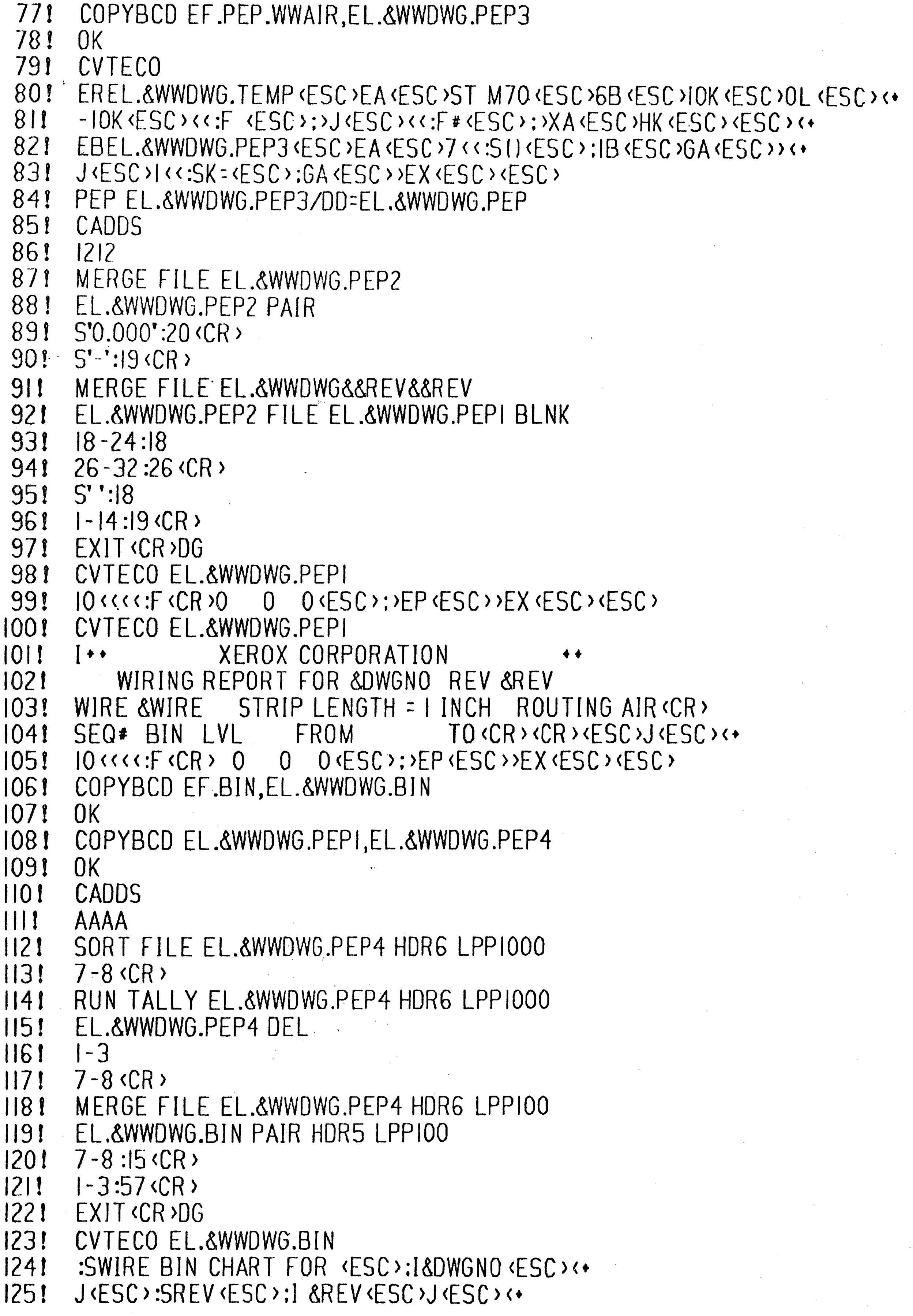
```
EF.PAREX.&BCD.WWAIR

&&
2 BCD '&BCD'
2 DWGNO
2 REV
2 WWDWG
2 SHTS
2 WIRE
&&
EXTR DATA EF.EDF.MULTISHTA <•
PART EL.&DWGNO/&SHTS <•
NAME EL.WW&DWGNO&&REV
FMT NET DATA EL.WW&DWGNO&&REV DATAKEY EF.DATA.KEYA <•
FILE EL.WW&DWGNO&&REV&&REV
EXTR DATA EF.EDF.XYCOORD PART EL.&WWDWG/WWBD&REV <•
NAME EL.&WWDWG&&REV
MERGE FILE EL.&WWDWG&&REV
EL.&WWDWG&&REV FILE EL.&WWDWG&&REV&&REV PAIR
S' ':19<CR>
S'#':19<CR>
MERGE FILE EL.&WWDWG&&REV&&REV
EL.&WWDWG&&REV&&REV PAIR BLNK
S'-':19<CR>
S' ':19<CR>
MERGE FILE EL.&WWDWG&&REV&&REV
EL.&WWDWG&&REV&&REV PAIR
S'#':19<CR>
S'-':19<CR>
```

Figure 1

(continued)

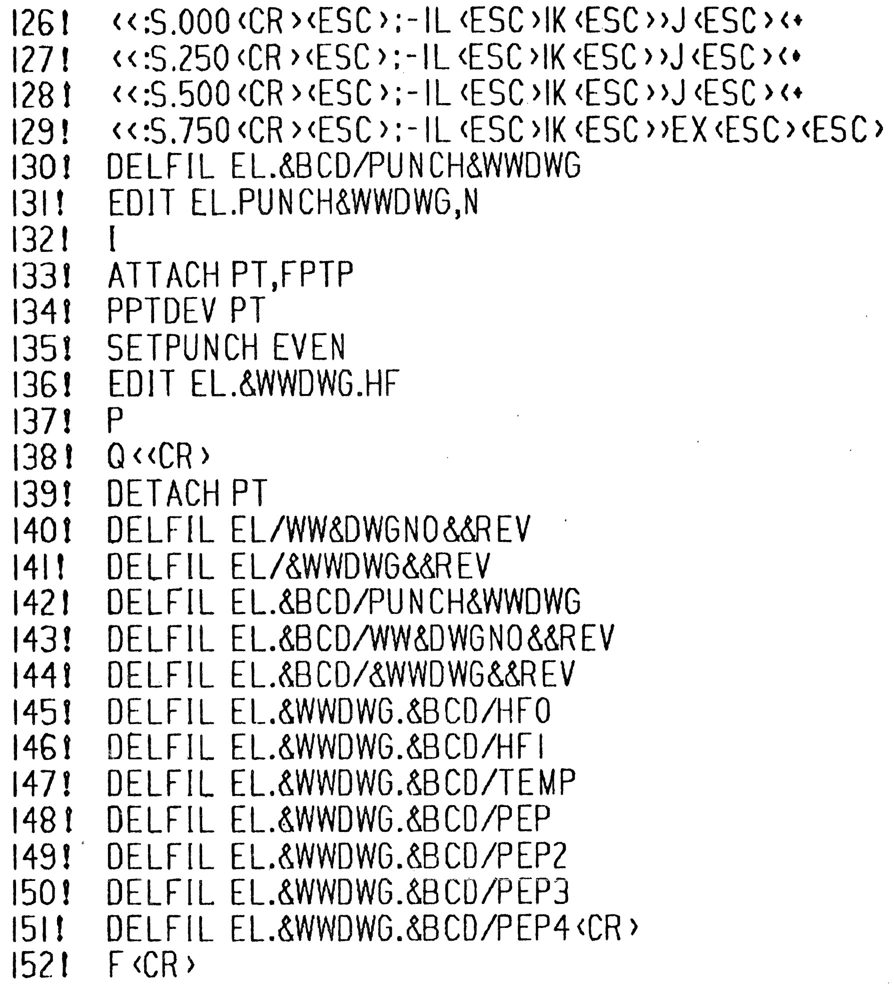
```
28! MERGE FILE EL.&WWDWG&&REV&&REV
29! EL.&WWDWG&&REV&&REV PAIR
30! S' 0.000':27‹CR›
31! S'-':27‹CR›
32! EXIT‹CR›DG
33! HARDFILE EL.&WWDWG.HF
34! OK
35! CADDS
36! EL.&WWDWG.WWBD&REV
37! SEL LAY 0
38! ECH LAY A
39! MERGE NET EL.WW&DWGNO&&REV&&REV :X10Y5
40! ATTACH PT FPTP
41! PUT WIRE MACH CS ZERO 25 DEV PT PRINT
42! Y
43! DETACH PT
44! EXIT‹CR›DG
45! HARDFILE
46! COPYBCD EL.&WWDWG.HF,EL.&WWDWG.HF0
47! OK
48! CVTECO EL.&WWDWG.HF
49! IPA‹ESC›3‹‹‹‹:ST$‹ESC›;0L‹ESC›-1000K‹ESC›-IPA‹ESC›0;›%%A:EY›‹ESC›‹ESC›‹+
50! IPA‹ESC›10‹‹J‹ESC›‹‹:F  T$‹CR›  ‹ESC›;›J‹ESC›‹‹:F  T$‹ESC›;1K‹ESC››‹+
51! J‹ESC›‹‹:S  M70‹ESC›;5B‹ESC›2D‹ESC››J‹ESC›‹‹:F$‹ESC›;›J‹ESC›‹+
52! ‹‹:SM70‹ESC›;3B‹ESC›IT  ‹ESC›L‹ESC››J‹ESC›‹+
53! ‹‹:F‹CR›‹CR› SECOND LEVEL OF WIRE‹CR›‹ESC›;›‹+
54! ‹‹:SR‹ESC›;3B‹ESC›F   ‹ESC›L‹ESC›1000K‹ESC›EP‹ESC›EF‹ESC›‹+
55! -IPA‹ESC›0;›%%A;EP›EQ‹ESC›‹ESC›
56! CVTECO
57! EREL.&WWDWG.HF‹ESC›EA‹ESC›EWEL.&WWDWG.HF1‹ESC›‹ESC›‹+
58! IPA‹ESC›10‹‹‹‹:FT M70M97‹CR›‹ESC›;›J‹ESC›‹‹:F M70M97‹ESC›;›J‹ESC›‹+
59! ‹‹:S‹CR›‹ESC›;1B‹ESC›1 0000‹ESC›1L‹ESC››‹‹:SR‹ESC›;0L‹ESC›K‹ESC›‹+
60! EP‹ESC›EF‹ESC›-IPA‹ESC›0;›%%A;EP‹ESC››EQ‹ESC›‹ESC›
61! CVTECO EL.&WWDWG.HF
62! IPA‹ESC›10‹‹‹‹:FX+0000‹ESC›;›J‹ESC›‹‹:FY+0000‹ESC›;›‹+
63! ‹‹:SR‹ESC›;-IPA‹ESC›0;›%%A;EP‹ESC››EX‹ESC›‹ESC›
64! HARDFILE EL.&WWDWG.TEMP
65! OK
66! EDIT EL.&WWDWG.HF1
67! T100000
68! Q
69! HARDFILE
70! DELFIL EL.&WWDWG/PEP2
71! DELFIL EL.&WWDWG/PEP
72! EDIT EL.&WWDWG.PEP,N
73! I
74! 011 EL.&WWDWG.HF1
75! 012 EL.&WWDWG.PEP2‹CR›
76! F
```

Figure 1 (continued)



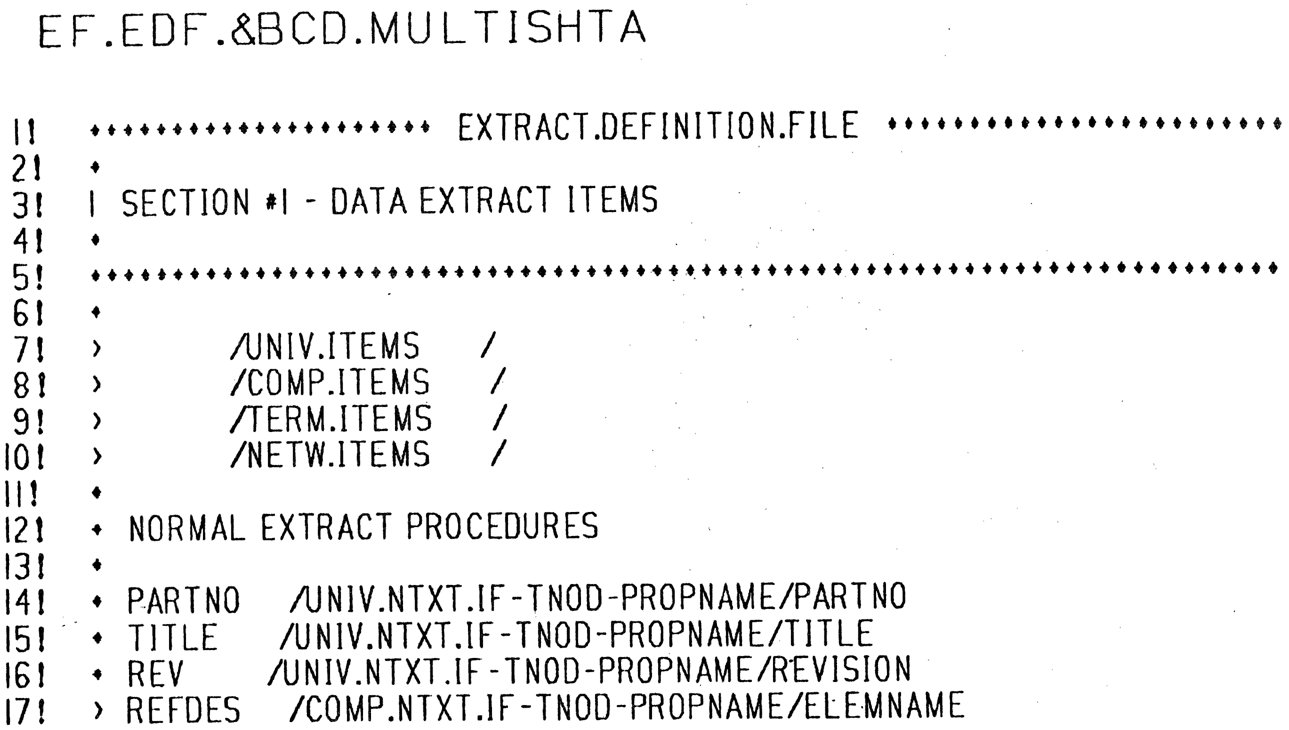
```
 77!  COPYBCD EF.PEP.WWAIR,EL.&WWDWG.PEP3
 78!  OK
 79!  CVTECO
 80!  EREL.&WWDWG.TEMP‹ESC›EA‹ESC›ST M70‹ESC›6B‹ESC›10K‹ESC›0L‹ESC›‹•
 81!  -10K‹ESC›‹‹:F ‹ESC›;›J‹ESC›‹‹:F#‹ESC›;›XA‹ESC›HK‹ESC›‹ESC›‹•
 82!  EBEL.&WWDWG.PEP3‹ESC›EA‹ESC›7‹‹:S()‹ESC›;IB‹ESC›GA‹ESC››‹•
 83!  J‹ESC›I‹‹:SK=‹ESC›;GA‹ESC››EX‹ESC›‹ESC›
 84!  PEP EL.&WWDWG.PEP3/DD=EL.&WWDWG.PEP
 85!  CADDS
 86!  1212
 87!  MERGE FILE EL.&WWDWG.PEP2
 88!  EL.&WWDWG.PEP2 PAIR
 89!  S'0.000':20‹CR›
 90!  S'-':19‹CR›
 91!  MERGE FILE EL.&WWDWG&&REV&&REV
 92!  EL.&WWDWG.PEP2 FILE EL.&WWDWG.PEP1 BLNK
 93!  18-24:18
 94!  26-32:26‹CR›
 95!  S' ':18
 96!  1-14:19‹CR›
 97!  EXIT‹CR›DG
 98!  CVTECO EL.&WWDWG.PEP1
 99!  10‹‹‹‹:F‹CR›0    0   0‹ESC›;›EP‹ESC››EX‹ESC›‹ESC›
100!  CVTECO EL.&WWDWG.PEP1
101!  I••            XEROX CORPORATION            ••
102!     WIRING REPORT FOR &DWGNO  REV &REV
103!  WIRE &WIRE    STRIP LENGTH = 1 INCH   ROUTING AIR‹CR›
104!  SEQ# BIN  LVL     FROM          TO‹CR›‹CR›‹ESC›J‹ESC›‹•
105!  10‹‹‹‹:F‹CR› 0    0   0‹ESC›;›EP‹ESC››EX‹ESC›‹ESC›
106!  COPYBCD EF.BIN,EL.&WWDWG.BIN
107!  OK
108!  COPYBCD EL.&WWDWG.PEP1,EL.&WWDWG.PEP4
109!  OK
110!  CADDS
111!  AAAA
112!  SORT FILE EL.&WWDWG.PEP4 HDR6 LPP1000
113!  7-8‹CR›
114!  RUN TALLY EL.&WWDWG.PEP4 HDR6 LPP1000
115!  EL.&WWDWG.PEP4 DEL
116!  1-3
117!  7-8‹CR›
118!  MERGE FILE EL.&WWDWG.PEP4 HDR6 LPP100
119!  EL.&WWDWG.BIN PAIR HDR5 LPP100
120!  7-8 :15‹CR›
121!  1-3:57‹CR›
122!  EXIT‹CR›DG
123!  CVTECO EL.&WWDWG.BIN
124!  :SWIRE BIN CHART FOR ‹ESC›;I&DWGNO‹ESC›‹•
125!  J‹ESC›:SREV‹ESC›;I &REV‹ESC›J‹ESC›‹•
```

Figure 1 (cont.)



```
<<:S.000<CR><ESC>:-IL<ESC>IK<ESC>>J<ESC><*
<<:S.250<CR><ESC>:-IL<ESC>IK<ESC>>J<ESC><*
<<:S.500<CR><ESC>:-IL<ESC>IK<ESC>>J<ESC><*
<<:S.750<CR><ESC>:-IL<ESC>IK<ESC>>EX<ESC><ESC>
DELFIL EL.&BCD/PUNCH&WWDWG
EDIT EL.PUNCH&WWDWG,N
I
ATTACH PT,FPTP
PPTDEV PT
SETPUNCH EVEN
EDIT EL.&WWDWG.HF
P
Q<<CR>
DETACH PT
DELFIL EL/WW&DWGNO&&REV
DELFIL EL/&WWDWG&&REV
DELFIL EL.&BCD/PUNCH&WWDWG
DELFIL EL.&BCD/WW&DWGNO&&REV
DELFIL EL.&BCD/&WWDWG&&REV
DELFIL EL.&WWDWG.&BCD/HF0
DELFIL EL.&WWDWG.&BCD/HF1
DELFIL EL.&WWDWG.&BCD/TEMP
DELFIL EL.&WWDWG.&BCD/PEP
DELFIL EL.&WWDWG.&BCD/PEP2
DELFIL EL.&WWDWG.&BCD/PEP3
DELFIL EL.&WWDWG.&BCD/PEP4<CR>
F<CR>
```

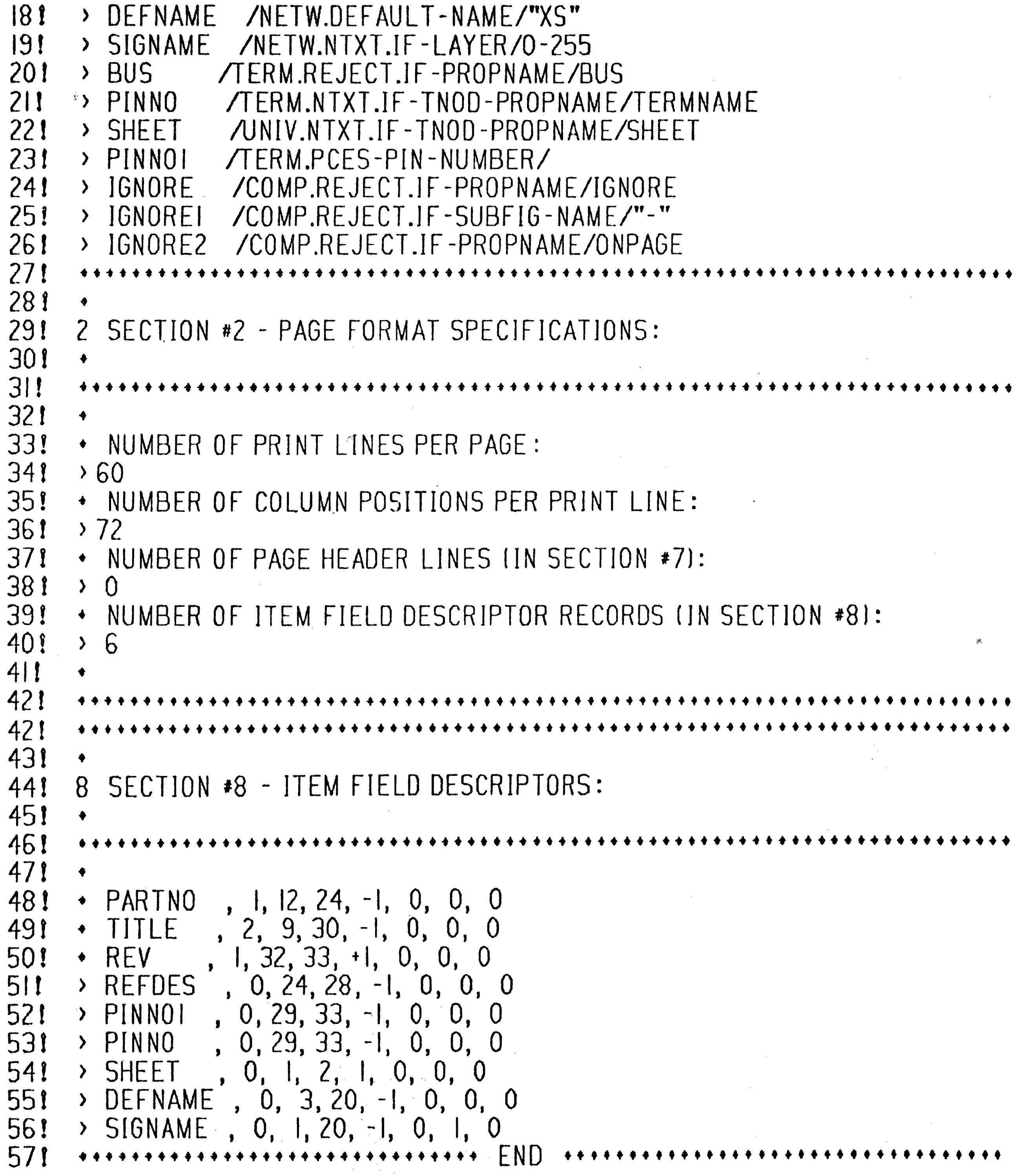
Figure 1 (continued)



EF.EDF.&BCD.MULTISHTA

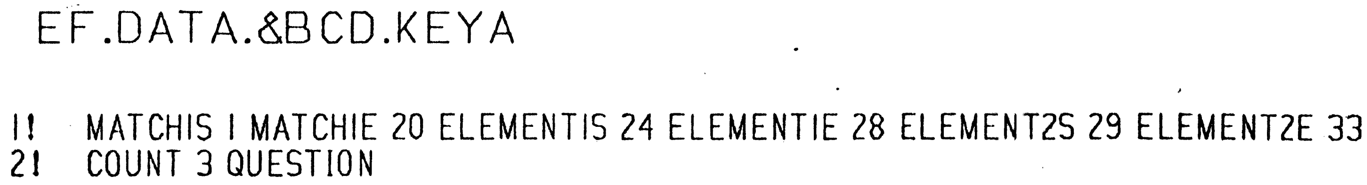
```
********************** EXTRACT.DEFINITION.FILE ************************
*
| SECTION #1 - DATA EXTRACT ITEMS
*
***********************************************************************
*
>       /UNIV.ITEMS   /
>       /COMP.ITEMS   /
>       /TERM.ITEMS   /
>       /NETW.ITEMS   /
*
* NORMAL EXTRACT PROCEDURES
*
* PARTNO   /UNIV.NTXT.IF-TNOD-PROPNAME/PARTNO
* TITLE   /UNIV.NTXT.IF-TNOD-PROPNAME/TITLE
* REV   /UNIV.NTXT.IF-TNOD-PROPNAME/REVISION
> REFDES   /COMP.NTXT.IF-TNOD-PROPNAME/ELEMNAME
```

Figure 2 *(continued)*



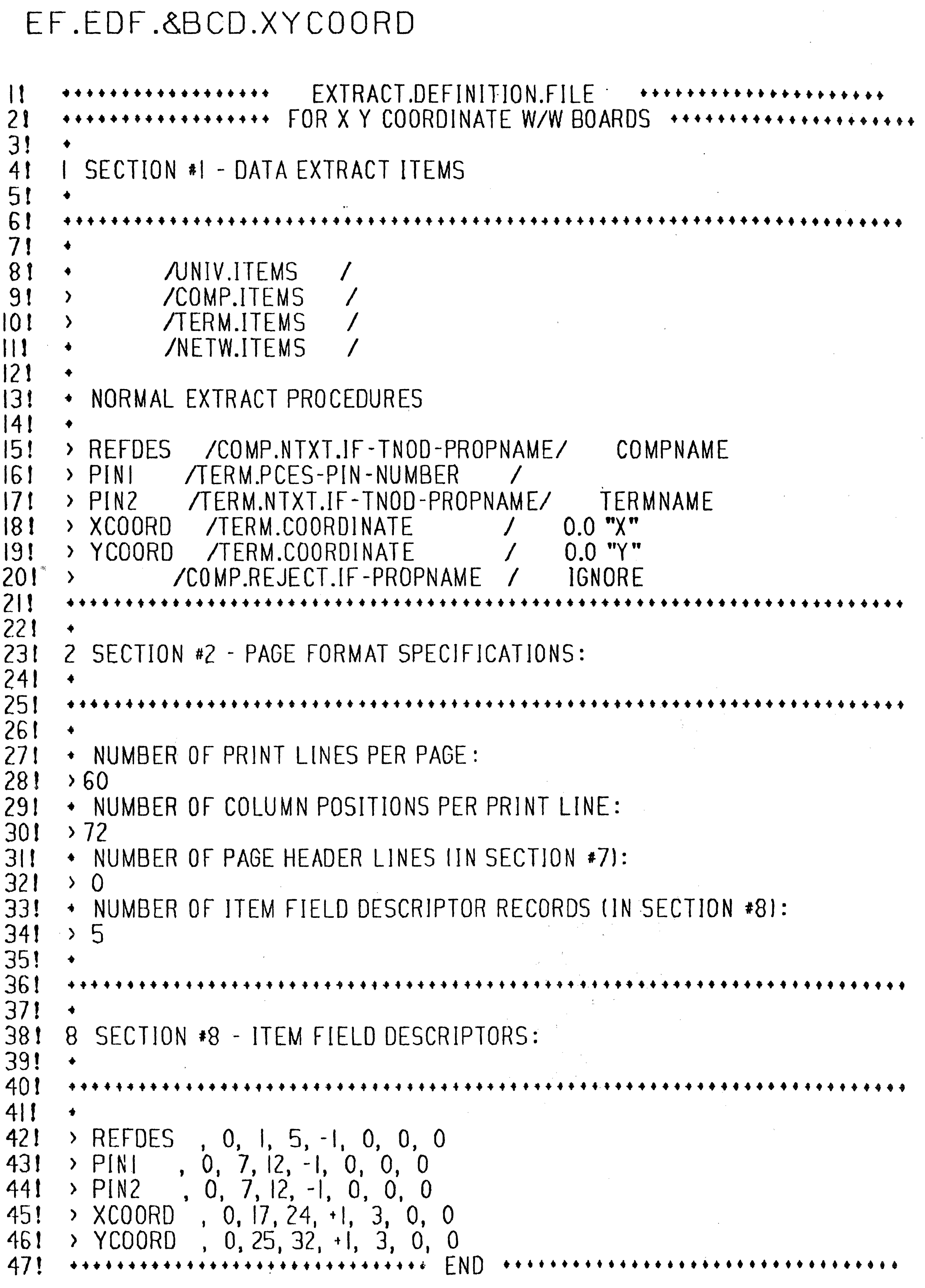
```
> DEFNAME   /NETW.DEFAULT-NAME/"XS"
> SIGNAME   /NETW.NTXT.IF-LAYER/0-255
> BUS      /TERM.REJECT.IF-PROPNAME/BUS
> PINNO    /TERM.NTXT.IF-TNOD-PROPNAME/TERMNAME
> SHEET    /UNIV.NTXT.IF-TNOD-PROPNAME/SHEET
> PINNO1   /TERM.PCES-PIN-NUMBER/
> IGNORE    /COMP.REJECT.IF-PROPNAME/IGNORE
> IGNORE1   /COMP.REJECT.IF-SUBFIG-NAME/"-"
> IGNORE2   /COMP.REJECT.IF-PROPNAME/ONPAGE
****************************************************************
*
2 SECTION #2 - PAGE FORMAT SPECIFICATIONS:
*
****************************************************************
*
* NUMBER OF PRINT LINES PER PAGE:
>60
* NUMBER OF COLUMN POSITIONS PER PRINT LINE:
>72
* NUMBER OF PAGE HEADER LINES (IN SECTION #7):
> 0
* NUMBER OF ITEM FIELD DESCRIPTOR RECORDS (IN SECTION #8):
> 6
*
****************************************************************
****************************************************************
*
8 SECTION #8 - ITEM FIELD DESCRIPTORS:
*
****************************************************************
*
* PARTNO   , 1, 12, 24, -1,  0,  0,  0
* TITLE    , 2,  9, 30, -1,  0,  0,  0
* REV      , 1, 32, 33, +1,  0,  0,  0
> REFDES   , 0, 24, 28, -1,  0,  0,  0
> PINNO1   , 0, 29, 33, -1,  0,  0,  0
> PINNO    , 0, 29, 33, -1,  0,  0,  0
> SHEET    , 0,  1,  2,  1,  0,  0,  0
> DEFNAME  , 0,  3, 20, -1,  0,  0,  0
> SIGNAME  , 0,  1, 20, -1,  0,  1,  0
****************************** END ******************************
```

Figure 2 (continued)



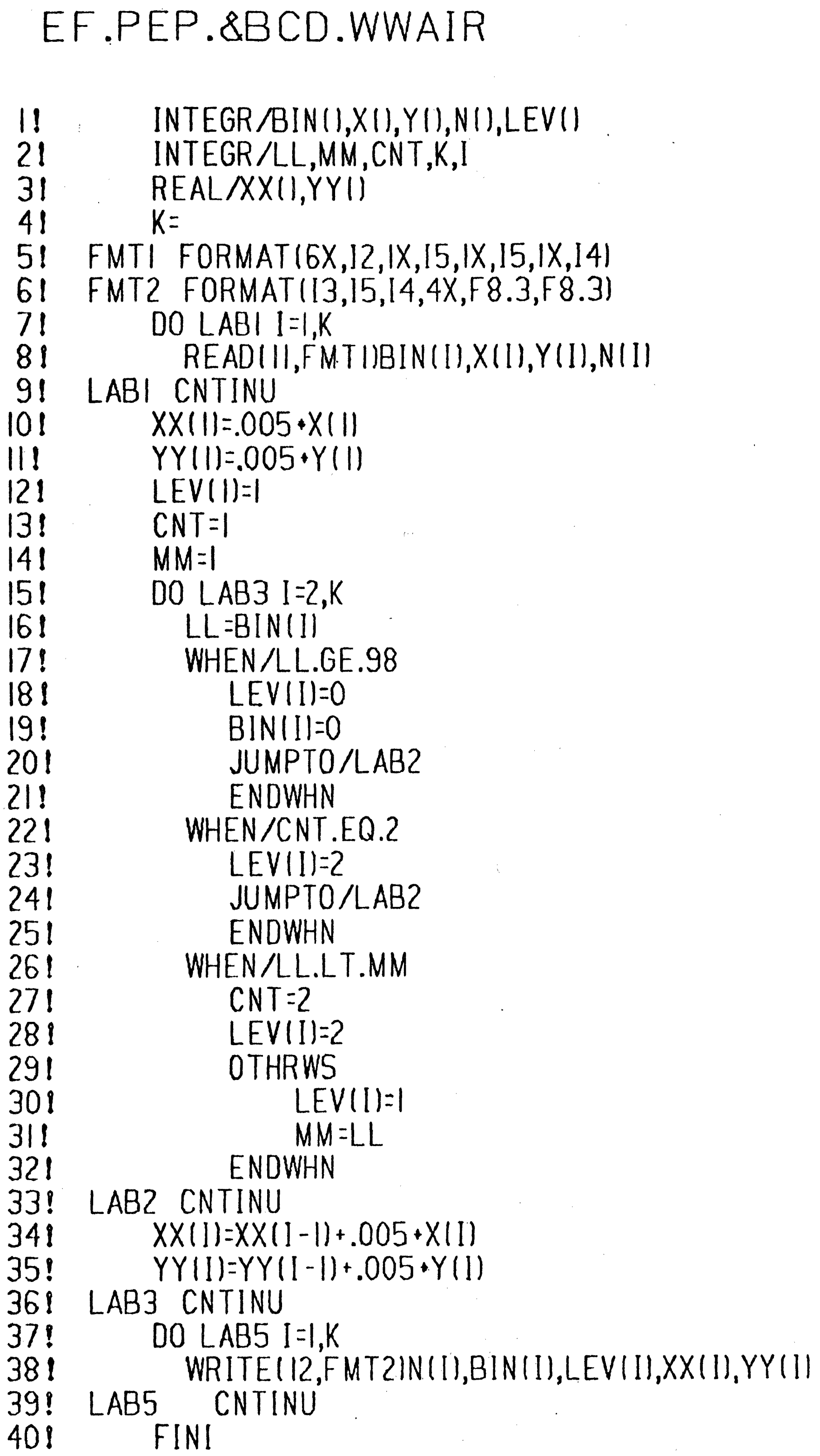
```
EF.DATA.&BCD.KEYA

MATCH1S 1 MATCH1E 20 ELEMENT1S 24 ELEMENT1E 28 ELEMENT2S 29 ELEMENT2E 33
COUNT 3 QUESTION
```

Figure 3



```
EF.EDF.&BCD.XYCOORD

******************    EXTRACT.DEFINITION.FILE    ********************
****************** FOR X Y COORDINATE W/W BOARDS ********************
*
I SECTION #I - DATA EXTRACT ITEMS
*
*********************************************************************
*
*        /UNIV.ITEMS   /
>        /COMP.ITEMS   /
>        /TERM.ITEMS   /
*        /NETW.ITEMS   /
*
* NORMAL EXTRACT PROCEDURES
*
> REFDES   /COMP.NTXT.IF-TNOD-PROPNAME/    COMPNAME
> PINI     /TERM.PCES-PIN-NUMBER     /
> PIN2     /TERM.NTXT.IF-TNOD-PROPNAME/    TERMNAME
> XCOORD   /TERM.COORDINATE          /    0.0 "X"
> YCOORD   /TERM.COORDINATE          /    0.0 "Y"
>          /COMP.REJECT.IF-PROPNAME  /    IGNORE
*********************************************************************
*
2 SECTION #2 - PAGE FORMAT SPECIFICATIONS:
*
*********************************************************************
*
* NUMBER OF PRINT LINES PER PAGE:
>60
* NUMBER OF COLUMN POSITIONS PER PRINT LINE:
>72
* NUMBER OF PAGE HEADER LINES (IN SECTION #7):
> 0
* NUMBER OF ITEM FIELD DESCRIPTOR RECORDS (IN SECTION #8):
> 5
*
*********************************************************************
*
8 SECTION #8 - ITEM FIELD DESCRIPTORS:
*
*********************************************************************
*
> REFDES  , 0,  I,  5, -I,  0,  0,  0
> PINI    , 0,  7, I2, -I,  0,  0,  0
> PIN2    , 0,  7, I2, -I,  0,  0,  0
> XCOORD  , 0, I7, 24, +I,  3,  0,  0
> YCOORD  , 0, 25, 32, +I,  3,  0,  0
****************************** END ********************************
```

Figure 4



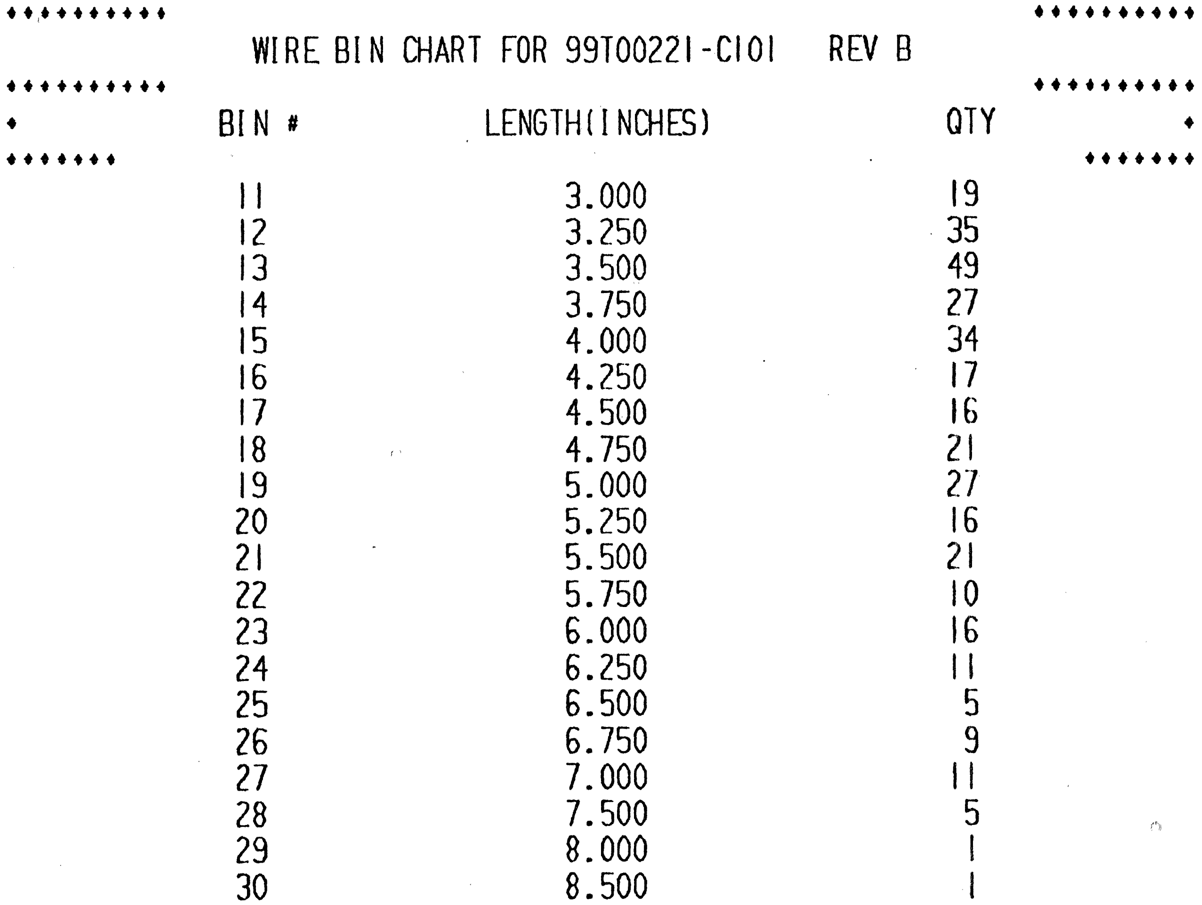
```
EF.PEP.&BCD.WWAIR

      INTEGR/BIN(),X(),Y(),N(),LEV()
      INTEGR/LL,MM,CNT,K,I
      REAL/XX(),YY()
      K=
FMT1  FORMAT(6X,I2,1X,I5,1X,I5,1X,I4)
FMT2  FORMAT(I3,I5,I4,4X,F8.3,F8.3)
      DO LAB1 I=1,K
        READ(11,FMT1)BIN(I),X(I),Y(I),N(I)
LAB1  CNTINU
      XX(1)=.005*X(1)
      YY(1)=.005*Y(1)
      LEV(1)=1
      CNT=1
      MM=1
      DO LAB3 I=2,K
        LL=BIN(I)
        WHEN/LL.GE.98
           LEV(I)=0
           BIN(I)=0
           JUMPTO/LAB2
           ENDWHN
        WHEN/CNT.EQ.2
           LEV(I)=2
           JUMPTO/LAB2
           ENDWHN
        WHEN/LL.LT.MM
           CNT=2
           LEV(I)=2
           OTHRWS
                LEV(I)=1
                MM=LL
           ENDWHN
LAB2  CNTINU
      XX(I)=XX(I-1)+.005*X(I)
      YY(I)=YY(I-1)+.005*Y(I)
LAB3  CNTINU
      DO LAB5 I=1,K
        WRITE(12,FMT2)N(I),BIN(I),LEV(I),XX(I),YY(I)
LAB5    CNTINU
      FINI
```

Figure 5

```
EF.&BCD.BIN

 1!  **********                                        **********
 2!             WIRE BIN CHART FOR    REV
 3!  **********                                        **********
 4!  *         BIN #        LENGTH(INCHES)             QTY          *
 5!  *******                                            *******
 6!              1             0.500
 7!              2             0.750
 8!              3             1.000
 9!              4             1.250
10!              5             1.500
11!              6             1.750
12!              7             2.000
13!              8             2.250
14!              9             2.500
15!             10             2.750
16!             11             3.000
17!             12             3.250
18!             13             3.500
19!             14             3.750
20!             15             4.000
21!             16             4.250
22!             17             4.500
23!             18             4.750
24!             19             5.000
25!             20             5.250
26!             21             5.500
27!             22             5.750
28!             23             6.000
29!             24             6.250
30!             25             6.500
31!             26             6.750
32!             27             7.000
33!             28             7.500
34!             29             8.000
35!             30             8.500
36!             31             9.000
37!             32             9.500
38!             33            10.000
39!             34            10.500
40!             35            11.000
41!             36            11.500
42!             37            12.000
43!             38            12.500
44!             39            13.000
45!             40           100.000
```

Figure 6

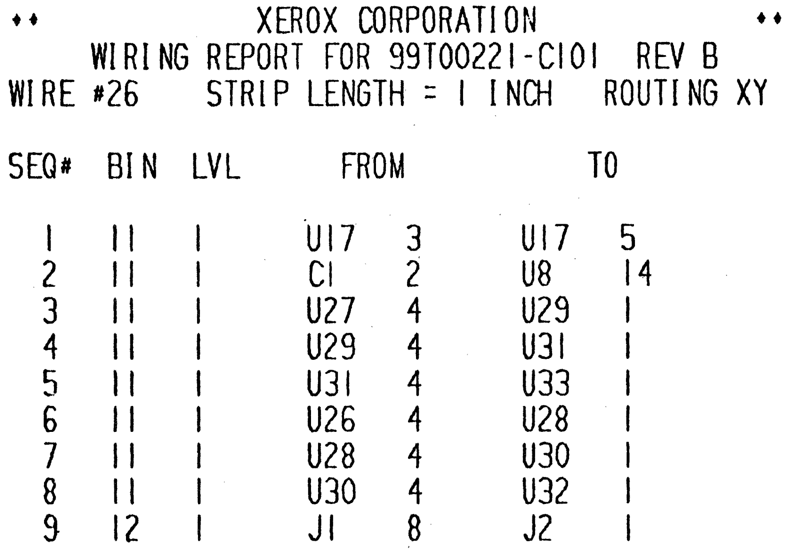
EL.99T221-C101.BIN



```
**********                                                        **********
               WIRE BIN CHART FOR 99T00221-C101    REV B
**********                                                        **********
*             BIN #            LENGTH(INCHES)            QTY               *
*******                                                              *******
              11                   3.000                 19
              12                   3.250                 35
              13                   3.500                 49
              14                   3.750                 27
              15                   4.000                 34
              16                   4.250                 17
              17                   4.500                 16
              18                   4.750                 21
              19                   5.000                 27
              20                   5.250                 16
              21                   5.500                 21
              22                   5.750                 10
              23                   6.000                 16
              24                   6.250                 11
              25                   6.500                  5
              26                   6.750                  9
              27                   7.000                 11
              28                   7.500                  5
              29                   8.000                  1
              30                   8.500                  1
```

Figure 7

EL.99T221-C101.PEP1



```
**              XEROX CORPORATION                **
    WIRING REPORT FOR 99T00221-C101  REV B
WIRE #26    STRIP LENGTH = 1 INCH   ROUTING XY

SEQ#  BIN  LVL      FROM           TO

  1   11    1      U17   3      U17   5
  2   11    1      C1    2      U8   14
  3   11    1      U27   4      U29   1
  4   11    1      U29   4      U31   1
  5   11    1      U31   4      U33   1
  6   11    1      U26   4      U28   1
  7   11    1      U28   4      U30   1
  8   11    1      U30   4      U32   1
  9   12    1      J1    8      J2    1
```

Figure 8

(continued)

```
10   12   1      U19    1      U19   4
11   12   1      U21    1      U21   4
12   12   1      U23    1      U23   4
13   12   1      U25    1      U25   4
14   12   1      U18    1      U18   4
15   12   1      U20    1      U20   4
16   12   1      U22    1      U22   4
17   12   1      U24    1      U24   4
18   12   1      FC1    23      FC1   18
19   12   1      R2     9      R3    16
20   12   1      U8     1      U8    4
21   12   1      U8     13      U8    10
22   12   1      U7     1      U7    4
23   12   1      U7     13      U7    10
24   12   1      U14    1      U14   4
25   12   1      U14    13      U14   10
26    0   0      X0Y0   1
27   12   1      U13    1      U13   4
28   12   1      U13    13      U13   10
29   12   1      U10    1      U10   4
30   12   1      U10    13      U10   10
31   12   1      U9     1      U9    4
32   12   1      U9     13      U9    10
33   12   1      U12    1      U12   4
34   12   1      U12    13      U12   10
35   12   1      U11    1      U11   4
36   12   1      U11    13      U11   10
37   13   1      R5     1      U17   1
38   13   1      U26    7      R3    1
39   13   1      U26    6      R3    2
```

Figure 8 (continued)

```
EL.99T221-C101.HF

     T M70M97
     T M70M11X-0400Y-0280N0001N
     T M70M98Y-0040
     T M70M97
     T M70M11X+0260Y+0320N0002N
     T M70M98Y-0040
     T M70M11X-0420Y-0420N0003N
     T M70M98Y-0040
     T M70M11Y-0060N0004N
     T M70M98Y-0040
     T M70M11Y-0060N0005N
     T M70M98Y-0040
```

Figure 9

(continued)

```
T M70M11X-0160Y+0240N0006N
T M70M98Y-0040
T M70M11Y-0060N0007N
T M70M98Y-0040
T M70M11Y-0060N0008N
T M70M98Y-0040
T M70M97
T M70M12X+0320Y+0140N0009N
T M70M98Y-0060
T M70M97
T M70M12X-0160Y+0620N0010N
T M70M98Y-0060
T M70M12Y-0040N0011N
T M70M98Y-0060
T M70M12Y-0040N0012N
T M70M98Y-0060
T M70M12Y-0040N0013N
T M70M98Y-0060
T M70M12X-0160Y+0360N0014N
T M70M98Y-0060
T M70M12Y-0040N0015N
T M70M98Y-0060
T M70M12Y-0040N0016N
T M70M98Y-0060
T M70M12Y-0040N0017N
T M70M98Y-0060
T M70M12X-0800Y-0080N0018N
T M70M96X-0020
T M70M98Y+0060
T M70M12X+0600Y+0020N0019N
T M70M98Y-0060
T M70M12X+0860Y+0380N0020N
T M70M98Y-0060
```

Figure 9 (continued)

```
EL.99T221-C101BB


    CE   36           0.300  -0.200
    CE   1            0.400  -0.200
    CE   2            0.400  -0.300
    CE   3            0.400  -0.400
    CE   4            0.400  -0.500
    CE   5            0.400  -0.600
    CE   6            0.400  -0.700
    CE   7            0.400  -0.800
    CE   8            0.400  -0.900
    CE   9            0.400  -1.000
```

Figure 10

(continued)

CE	10	0.400	-1.100
CE	11	0.400	-1.200
CE	12	0.400	-1.300
CE	13	0.400	-1.400
CE	14	0.400	-1.500
CE	15	0.400	-1.600
CE	16	0.400	-1.700
CE	17	0.400	-1.800
CE	18	0.400	-1.900
CE	19	0.400	-2.000
CE	20	0.400	-2.100
CE	21	0.400	-2.200
CE	22	0.400	-2.300
CE	23	0.400	-2.400
CE	24	0.400	-2.500
CE	25	0.400	-2.600
CE	26	0.400	-2.700
CE	27	0.400	-2.800
CE	28	0.400	-2.900
CE	29	0.400	-3.000
CE	30	0.400	-3.100
CE	31	0.400	-3.200
CE	32	0.400	-3.300
CE	33	0.400	-3.400
CE	34	0.400	-3.500
CE	35	0.400	-3.600
CE	70	0.300	-3.600
CE	69	0.300	-3.500
CE	68	0.300	-3.400
CE	67	0.300	-3.300

Figure 10 (continued)

EL.WW99T00221-C101BB

001	2XS0001	U3-6	U25-2
002	2XS0003	U1-8	U19-3
003	2XS0005	U2-10	U23-2
004	2XS0007	U2-4	U21-3
005	2XS0009	U2-8	U22-3
006	2XS0011	U3-2	U24-2
007	2XS0013	U2-12	U23-3
008	2XS0015	U3-4	U24-3
009	2XS0017	U3-8	U25-3
010	2XS0019	U1-6	U19-2
011	2XS0021	U1-12	U20-3
012	2XS0023	U1-4	U18-3
013	2XS0025	U1-2	U18-2

Figure 11

(continued)

```
014 2XS0027          U2-2   U21-2
015 2XS0029          U1-10   U20-2
016 2XS0031          U2-6   U22-2
017 2XS0034          U17-11   U17-9   U17-5   U17-3   U17-1   R5-1
                     J1-1
018 2XS0062          J2-1   J1-8
019 3XS0002          U4-5   R3-1   U26-7
020 3XS0004          U4-9   R3-2   U26-6
021 3XS0006          U4-11   R3-3   U27-7
022 3XS0008          U5-1   R3-5   U28-7
023 3XS0010          U6-13   R4-16   U33-6
024 3XS0012          U5-11   R4-9   U30-7
025 3XS0014          U6-1   R4-11   U31-7
026 3XS0016          U5-5   R3-7   U29-7
027 3XS0018          U6-9   R4-14   U32-6
028 3XS0020          U5-13   R4-10   U30-6
029 3XS0022          U5-9   R3-8   U29-6
030 3XS0024          U6-3   R4-12   U31-6
031 3XS0026          U5-3   R3-6   U28-6
032 3XS0028          U4-13   R3-4   U27-6
033 3XS0030          U6-11   R4-15   U33-7
034 3XS0032          U6-5   R4-13   U32-7
```

Figure 11 (continued)

INTRODUCTION OF A CAD SYSTEM FOR PRINTED CIRCUIT BOARD DESIGN BASED ON COMPUTERVISION SYSTEMS

by
H. Rohrbacher, G. Martin
Manager, Support System Development
Standard Elektrik Lorenz AG
Stuttgart, West Germany

1. ABSTRACT

The present article covers the process of introduction of computer aids for PCB's into an electronic industry, the analysis of the present status and the future aspects.

For the particular industry in question the best approach was a combination of commercially available dedicated hardware and software and internally developed software running on general purpose computers.

2. INTRODUCTION

All stages in the life of an electronic system — from the functional design through detailed design, board layout, manufacturing, acceptance testing to field or system integration testing — will eventually be controlled or supported by computers. During the last decade computer aids have been offered for each of these steps.

Normally these systems are independent of each other and most of the data transfer and consistency check of shared information is done by people.

Consequently when companies start to introduce a new set of computer aids, much effort and time is spent to integrate it into the existing and used systems. The established system has to be supported and upgraded by an internal system group.

In the following chapters the development of the CAD system for printed board layout and document generation is presented.

3. CAD SYSTEMS FOR PRINTED BOARD LAYOUT

In the area of printed circuit board design three major ways of computer aided processes have been established. These ways and the hardware systems used are shown in figure 1.

3.1 Computer Aided Digitizing and Plotting (CDP)

The first step into CAD was the introduction of a system consisting of a digitizer and a photoplotter in 1970. This system replaced the "manual taping" process followed by photographic reduction.

The only documents which can be produced are films for the artwork. Our company installed two AEG digitizers, type RPE and one photoplotter GERBER 1232 in 1970. On the digitizer the x,y coordinates of the tracks and holes are taken from the layout sketch and punched on papertape together with the commands for the photoplotter.

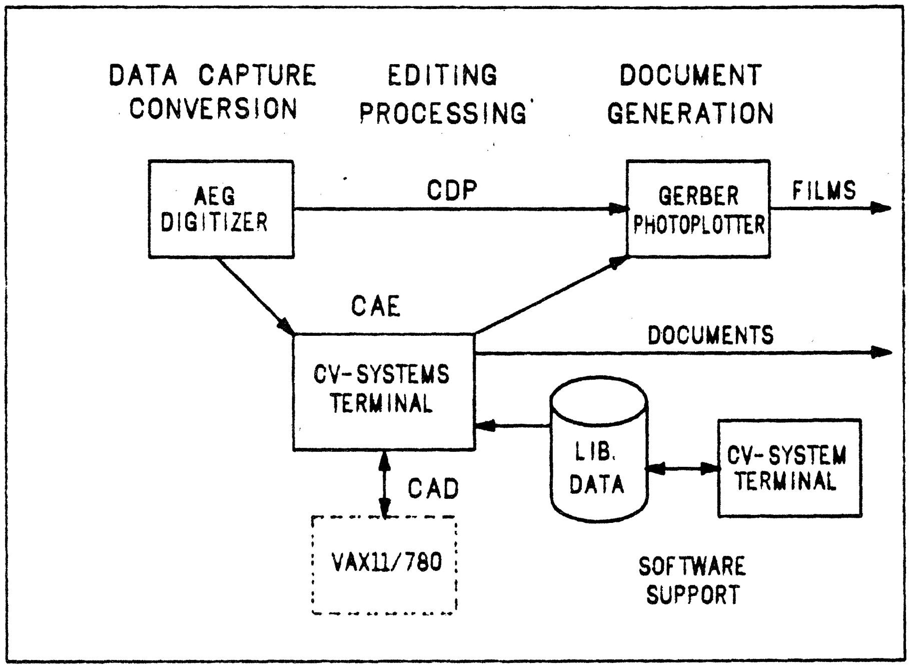


Figure 1: Hardware configuration

Digitizing errors are corrected directly on the papertape.

This system is still operational and used for board layouts, and small changes. The benefits are very fast turn around time and small machine costs.

Achieved results are:
Cost reduction (in 1976): ≈ 2 : 1 (manual taping : CDP)
Processing time : 7 h } double sided, 4×6.5 inch
Elapsed time : 3 days }

Documents produced:
Artwork film side 1
Artwork film side 2
Drilling film
(Solder resist mark)

3.2 Computer Aided Editing (CAE)

A more comfortable and flexible CAD system is achieved by the use of an Interactive Graphic System. The additional equipment consisting of a computer, graphic terminals and a penplotter enables to check the digitized data and edit them if necessary.

Further on not only layout artwork will be produced but also checkplots of electrical schematics and component mounting diagrams using element libraries.

The previous system has been extended by Computervision Systems with Nova 1200 processors and Design Terminals in 1975.

The CAE process is used for more complex boards, backpanels and since 1978 for digitizing electrical schematics from a sketch.

The achieved results are (for layouts only):
Cost reduction (in 1978): ≈ 1.3 : 1 (CAE : CDP)
Processing time : 20 h
Elapsed time : 5 days (without layout sketch)

Additional documents which are generated:
Circuit schematics
Assembly drawing
White print mask
Solder resist mask
Assembly parts list
Net list for In-circuit Tester
Drill tape on request

3.3 Computer Aided Design (CAD)

The addition of more computer power to the system gives the capability to replace some of the manual activities by algorithmic programs, i.e. gate assignment, placement, routing, logic simulation and automatic test program generation.

The programs for gate assignment and component placement run on a VAX 11/780. Logic simulation and automatic test program generation systems are installed on the IBM 3033.

For routing the standard program on the Computervision System is used. The difference to the CAE process is that digitizing of the layout can be omitted. Normally the algorithmic programs give a 90–95% solution which is completed by CAE.

The full CAD process is at present in the phase of pilot production.

Results achieved so far:

Cost comparison (1980)	:	1 : 1 CAE : CAD
expected	:	2 : 1
Processing time	:	1 : 1
expected	:	3 : 1
Elapsed time	:	14 days
expected	:	6 days
Time to complete layout	:	1 hour/open line

4. ORGANIZATION

For the SEL Telecommunications Group the CAD activities are allocated in one central organization.

4.1 Operating and Service

The CAD equipment is installed centrally in one area. The centralization enables exact job accounting, good terminal capacity forecast, backup during downtimes of one machine, and quick service for users' requirements of tapes and disks.

The digitizers and photoplotters are managed in closed shop technique which is natural for those systems.

The CV Systems and the VAX are organized in open shop and are available in one slightly extended shift. To enable the PCB layouters the use of the system, an 8 day training course and permanent support on the job is offered. Approximately 40% of the potential users have been trained up to now.

4.2 Software Support

From the very beginning the decision was made to start with a turnkey system but to do own programming:

The reasons were:

Non-availability of announced software
Long run times of some standard software
Special interfaces for in-house purposes.

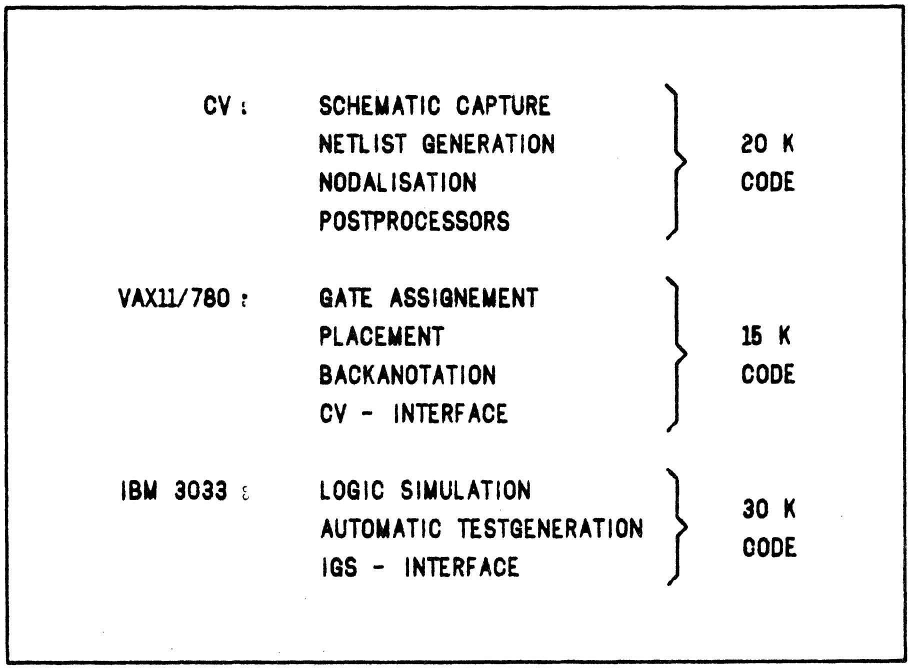
Own software developed by SEL or fundamental changes in the CV software are listed below:

Data capture and transformation:
Interface from AEG digitizer to CV — special
Net list generator
Nodalizer (adds nodal information) } — faster than CV
Libraries (physical and schematic symbols) — special

Algorithmic processes:
Component packaging and placement
Logic simulation
Automatic test program generation } — not available from CV

Documentation process:
Artwork output for Gerber — special

The amount of own software is given in figure 2.



CV :	SCHEMATIC CAPTURE NETLIST GENERATION NODALISATION POSTPROCESSORS	20 K CODE
VAX11/780 :	GATE ASSIGNEMENT PLACEMENT BACKANOTATION CV - INTERFACE	15 K CODE
IBM 3033 :	LOGIC SIMULATION AUTOMATIC TESTGENERATION IGS - INTERFACE	30 K CODE

Figure 2: Software developed in-house

The Software Support Group is also responsible for installation and test of new revisions of operating systems and graphic software, for trial application of new software modules and users' documentation and training. The effort is split up in figure 3.

5. FUTURE ASPECTS

In the past the CAD systems have proved to be an economic tool for PCB design and documentation. At present the installed machine capacity is still too small to process all PCB design on the systems. In the next four years three activities will continue:

— installation of additional capacity
— increase the capability of the algorithmic programs
— additional types of documents produced.

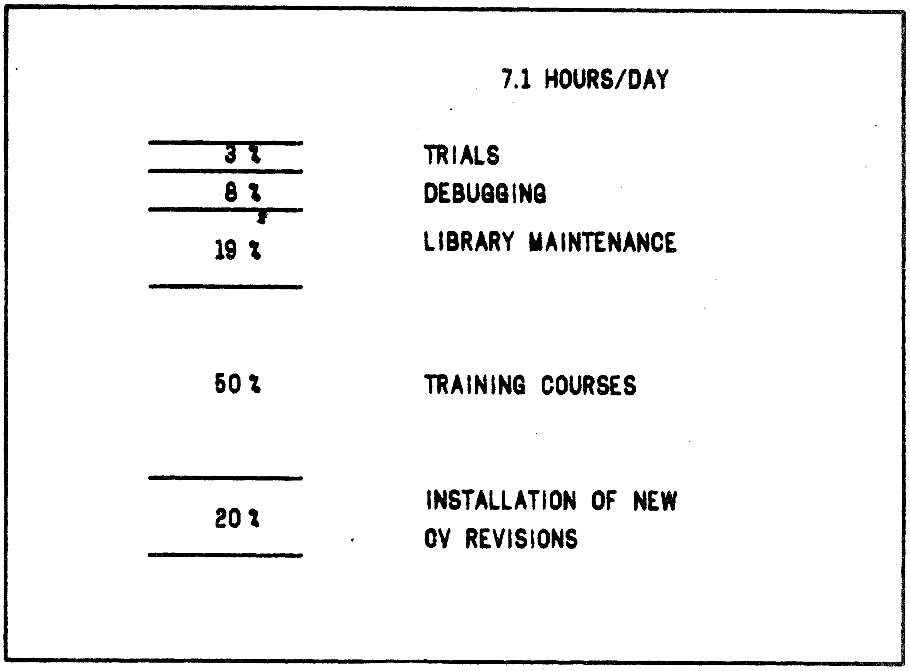


Figure 3: Terminal capacity for support

The goal is to come up with a system capacity to handle all PCB's developed in SEL from 1985 onwards.

The trends are shown in figure 4.

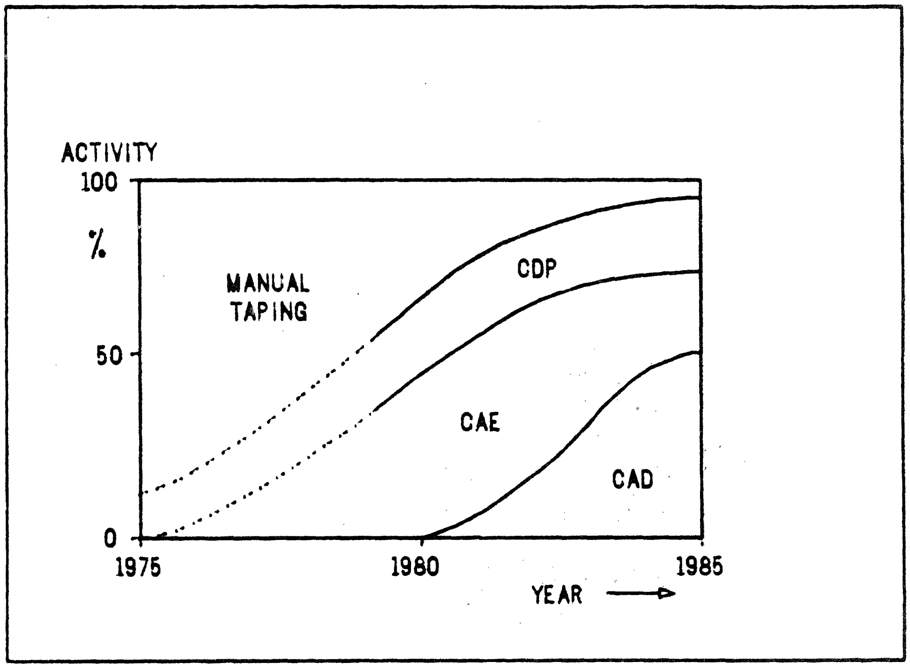


Figure 4: Plans for the next years

6. ACKNOWLEDGEMENT

The authors are very grateful for the discussion and technical contributions to this article provided by S. Grützmann of SEL CADEM department.

7. REFERENCES

(1) D. L. Earll
The CAD/CAM Handbook, p. 239–255 (1980)
Computervision Corporation Bedford, Mass.

(2) J. Vliestra
An Overview of Computer Aided Printed Circuit Board Design
in an Electronic Industry
Computer in Industry 1 (1979), p. 41–58
North-Holland Publishing Company

CVLCAP
COMPUTERVISION LOGIC CIRCUIT ANALYSIS PROGRAM

Allen Brown
Computervision Centre
1040 Uxbridge Rd.
Hayes, Middlesex
England
1-561-2626

ABSTRACT

CVLCAP is a computer program that analyzes electrical networks of logic gates. It runs on any CV system which has at least one COMDEV (system console or designer work station), disk storage and the CADDS/PEP applications software. CVLCAP is modeled on concepts put forth in the reference listed below.

The program is written in Computervision PEP and has provisions for selecting either of two modes of running. The "learner/auto-tutorial" mode provides detailed instructions to the user at each step in the program. For those familiar with the program's man/machine dialog requirements there is a much less verbose "expert user" mode.

INTRODUCTION

CVLCAP provides the design engineer with a useful design tool to verify that a network will function as logically expected. Seven types of gates, AND, OR, XOR, NAND, NOR, NXOR and INVERTER (or NOT), are recognized by the program. A gate may have up to eight input nodes but only one output node.

A description of the logic circuit to be analyzed is input and any or all of five tasks are performed on it. Briefly, the tasks are as follows:

1. The user assigns a unique number to each gate of interest and states its type. Unique numbers are also assigned to the input and output nodes of these gates.
2. The user defines the state of these nodes.
3. CVLCAP exercises the logic circuit until stability of nodal values is attained.
4. The CV computer prints out the state of the nodes.
5. The CV computer summarizes the circuit's logic elements.

The designer can analyze the output from task four to determine if the logic circuit will perform as he had expected.

While theoretically, this program and the CV computer can handle networks of up to 1200 gates and 4800 nodes, from a practical standpoint CVLCAP will be most used with logic circuits of more than 10 gates and up to the point where chips rather than discrete logic constitute the network.

GATES AND STATES

A gate by definition, is a device which produces a single logical output state as a function of one or more input states. A binary logic state is a value of either Ø or 1. If a state is unknown, CVLCAP provides for a random or "undefined" state which is represented by an asterisk (*).

Figure 1 shows a set of truth tables for AND, OR and exclusive OR (XOR) gates with two inputs and for NOT which, by definition, has only one input. The truth tables for NAND, NOR and XNOR can be obtained by simply complementing the tabular values in the AND, OR and XOR tables respectively.

The asterisk (*) is defined for CVLCAP purposes to be its own complement. Although it is possible to represent the "undefined complement" with its own symbol, this makes no sense if there is more than one undefined state in the circuit. Clearly, two undefined states could have opposite values. Therefore, the ORing of an "undefined" with an "undefined-complement" would not necessarily yield a value of 1.

AND			
	Ø	1	*
Ø	Ø	Ø	Ø
1	Ø	1	*
*	Ø	*	*

OR			
	Ø	1	*
0	Ø	1	*
1	1	1	1
*	*	1	*

XOR			
	Ø	1	*
Ø	Ø	1	*
1	1	Ø	*
*	*	*	*

NOT	
Ø	1
1	Ø
*	*

Figure 1

USING CVLCAP

The CVLCAP program will perform any of the following five tasks:

TASK	FUNCTION
1.	Accept the logic circuit description input interactively or from a data file.
2.	Accept assignment of node states interactively or from a data file.
3.	Exercise the circuit until its nodal value stability is attained.
4.	Output to the list device or to a data file, an ordered list of the defined node state array.
5.	Output to the list device or to a data file, and ordered list of all gates in the circuit.

The network of gates and nodes (lines) which represent the logic circuit to be investigated must be specified. The rules for describing the features of the network are outlined below.

Each gate must have only one output node. An INVERTER (NOT) gate has only one input, as well. XOR and XNOR have two inputs, while AND, OR, NAND and NOR gates may have up to eight.

Each node will have one of three values, logic Ø, logic 1 or "undefined" (*). If a nodal value is not explicitly declared, CVLCAP assigns a default value of "undefined".

Each gate must be labeled with an integer sequence number and so must each node. The order in which gates or nodes are labeled is irrelevant but each gate must have a unique label. Each node must also have a unique number but a gate may have the same number as a node since CVLCAP maintains separate lists for each. Normally, these integer labels will be the standard CV symbol sequence numbers.

RUNNING CVLCAP

To invoke CVLCAP, the user needs only to enter the command:

```
XPEP CVE.CVLCAP
```

into the CV computer within CADDS.

The clerical details of how the circuit description and the nodal states are input is completely auto-tutorial in actual program usage. The graphical representation of the circuit to be analyzed does not necessarily need to be in the CADDS data base. It can be in any other graphical or tabular form but most probably it would originate from an on-line display of a logic program with CV sequence numbers at the CV workstation.

CVLCAP is a complete hand holding program which will give the user *total* guidance at all times unless the user deliberately chooses the "expert user" mode. In that case the detailed explanations are shortened.

Normally, the sequence of operations is:

1.	Input the circuit description	(Task 1)
2.	Verify the circuit description	(Task 5)
3.	Assign nodal values	(Task 2)
4.	Verify the nodal values	(Task 4)
5.	Exercise the circuit	(Task 3)
6.	Verify final nodal values	(Task 4)
7.	Modify circuit description	(Task 1)
8.	Verify the circuit description	(Task 5)
9.	Change nodal values	(Task 2)
10.	Verify the nodal values	(Task 4)
11.	Exercise the circuit	(Task 3)
12.	etc.	. . . . . .

A simple logic circuit is shown in Figure 2. Figures 3 and 4 respectively show a circuit description and gate status description after executing CVLCAP for this circuit.

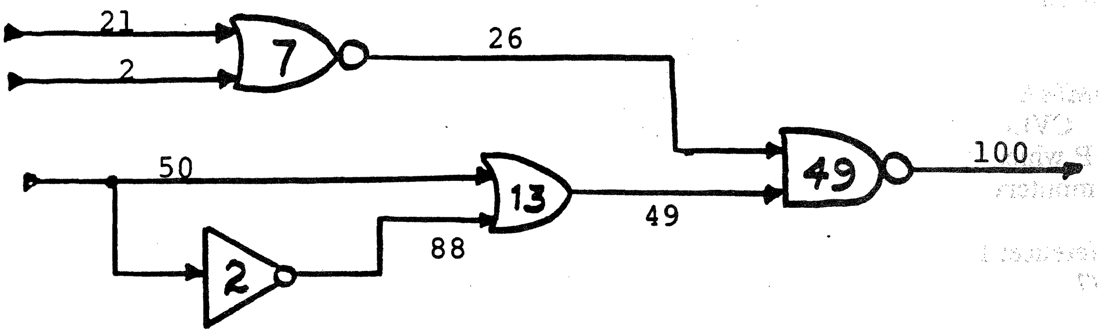


Figure 2

```
                  CIRCUIT GATE LIST
GATE  TYPE  OUT# IN# IN# IN#  IN#  IN#  IN#  IN#  IN#
   2  INV    88  50
   7  NOR    26  21   2
  13  OR     49  88  50
  49  NAND  100  26  49
```

Figure 3

```
                              NODE VALUES
NODE V NODE V NODE V NODE V NODE V NODE V NODE V NODE V NODE V NODE V
   2 Ø   21 1   26 Ø   49 1  1ØØ 1
```

Figure 4

HOW CVLCAP WORKS

CVLCAP is written in PEP and is invoked from a PEP object file, CVE.CVLCAP. The program assumes that it is dealing with a novice user unless it is told otherwise. It is a forgiving program, i.e. the user is allowed to respecify any input item and he is warned before any input data is overwritten.

If the user indicates in response to a prompt, that he is specifying a new circuit, all node state data is set to "undefined" and all gate descriptions are set to "null". The user then changes these conditions with his input.

If the user indicates that he is not specifying a new circuit, conditions already entered remain and he can change those nodes and/or gates that he wishes.

The heart of CVLCAP is the circuit analysis (Task 3). It is a simple and straightforward process. Each gate is examined. If it is a "null" gate, i.e. not defined, it is ignored. If the gate is defined, its output node value is computed from its inputs. If the computed output node value is different from the current output node value, the latter is changed and the "node change" flag is set. After every node has been analyzed, the node change flag is examined. If it is set, the entire circuit analysis process is repeated. Theoretically, it is possible that an oscillating type circuit may never reach stability and iterate forever. However, CVLCAP intervenes after the user-specified maximum number of iterations is reached and presents the user with an opportunity to continue execution or to request another task.

If a gate remains undefined, it is never listed by the circuit description part of the program (Task 5). If a node is still in an undefined state, it is never listed by the nodal description part of the program (Task 4).

SUMMARY

CVLCAP is a completely auto-tutorial interactive program written in Computervision PEP which allows a user to analyze networks of logic gates. It is available through normal Computervision software distribution channels.

Reference: LOGAN . . . A Circuit Analysis Program, Robert J. Bishop, Interface Age, May 1977.

"BOOSTING YOUR PRODUCTIVITY THROUGH CUSTOM SOFTWARE & THE ZERO SALARY SHIFT"

by
Marta Maddy & Paul Minkin
Honeywell Inc.
Parkdale Plaza Bldg.
1660 S. Highway 100
St. Louis Pk, MN 55416
612-542-5133

ABSTRACT

Here are some simple economics of CADCAM: Make a Capital Investment in equipment, and training and prepare an expense budget for system overhead. We expect a good return on our investment — but how? Simply offset the depreciation and operating costs with operators utilizing the system at a productivity rate that is higher than manual methods. Obviously the systems pay for themselves when the increased productivity of the operators balances the costs. To speed up the payback CAD users will either be improving their productivity figures, or they will be using multiple shifts. This paper will explore various methodologies to increase the operators' efficiency. Specific concentration will be on custom software that typically set up execute files that can be run at night. This type of operation is not new and is typically referred to as the batch shift; but a more appropriate title is "The Zero Salary Shift". This title refers to the use of execute files on the CV system without an operator. This is using the computer as a robot and involves no salary overhead, hence the name — Zero Salary Shift.

PRODUCTIVITY BOOSTERS ON A CAD SYSTEM

Because of the nature of CAD systems, they elicit a higher than manual productivity figure for an operator that has mastered the interaction between himself and the computer. The obvious things here are Auto-Routing, Auto-Placement for Printed Circuit Board layout, Dimensioning, 3-D work, and Mass Property Analysis for Mechanical Design to name a few. This mastery occurs through training and experience. It is crucial that each site has trained operators — otherwise the natural form of payoff is weakened and is delayed longer than necessary. Another very effective way to increase system productivity is by setting up jobs for the Zero Salary Shift. At Honeywell Residential our goal was to find the processing that could be set up in an execute file and then to create these execute files with a minimum amount of effort by the operators and with a minimum amount of training.

SPECIFIC AREAS TO BOOST OUR PRODUCTIVITY

Since we felt fairly confident that we could implement a Zero Salary Shift, the next step was to define the areas where the operator had a lot of lag time at the terminal or command sequences that could be greatly simplified by some outside processing. An example of lag time is a particular command that would take enough time for the operator to have a coffee break, read the sports page, or start thinking about lunch. Under this lag time category we felt we could gain time by preprocessing Photoplotting and Auto-routing, ending up with execute files to accomplish the task. Under the complicated command structure category, we felt Library Management and Part Releasing could use some smoothing out. The following chart indicates the areas where we put our efforts:

GAIN A PRODUCTIVITY INCREASE	SMOOTH OUT COMPLICATED PROCESSES
PHOTOPLOTTING	LIBRARY MANAGEMENT
AUTOROUTE	PART RELEASING

IMPLEMENTATION

The methods an operator can use to set up execute files for batch runs are:

1. Have the operator write the execute files.
2. Run a Parex or Var Pro file then input appropriate parameters, or
3. Write custom software to drive a question and answer session with the operator which would: loop with questions, handle and report input errors, and make operator-computer interface very easy, simple, and friendly.

Because of our stated goal of a minimal amount of effort going into the operator's interaction with the computer we opted to write some custom software to interactively handle questions and answers, report errors, repeat sessions, and to optionally produce execute files which can be run on the Zero Salary Shift. This interface is to be easy, simple, friendly and look clean to the operator [1,5]. With our software we are able to provide step by step instructions should they be necessary. Ultimately these front end software routines will update a disk file or produce one or more execute files that can be run on our Zero Salary Shift.

CHOICE OF PROGRAMMING LANGUAGE

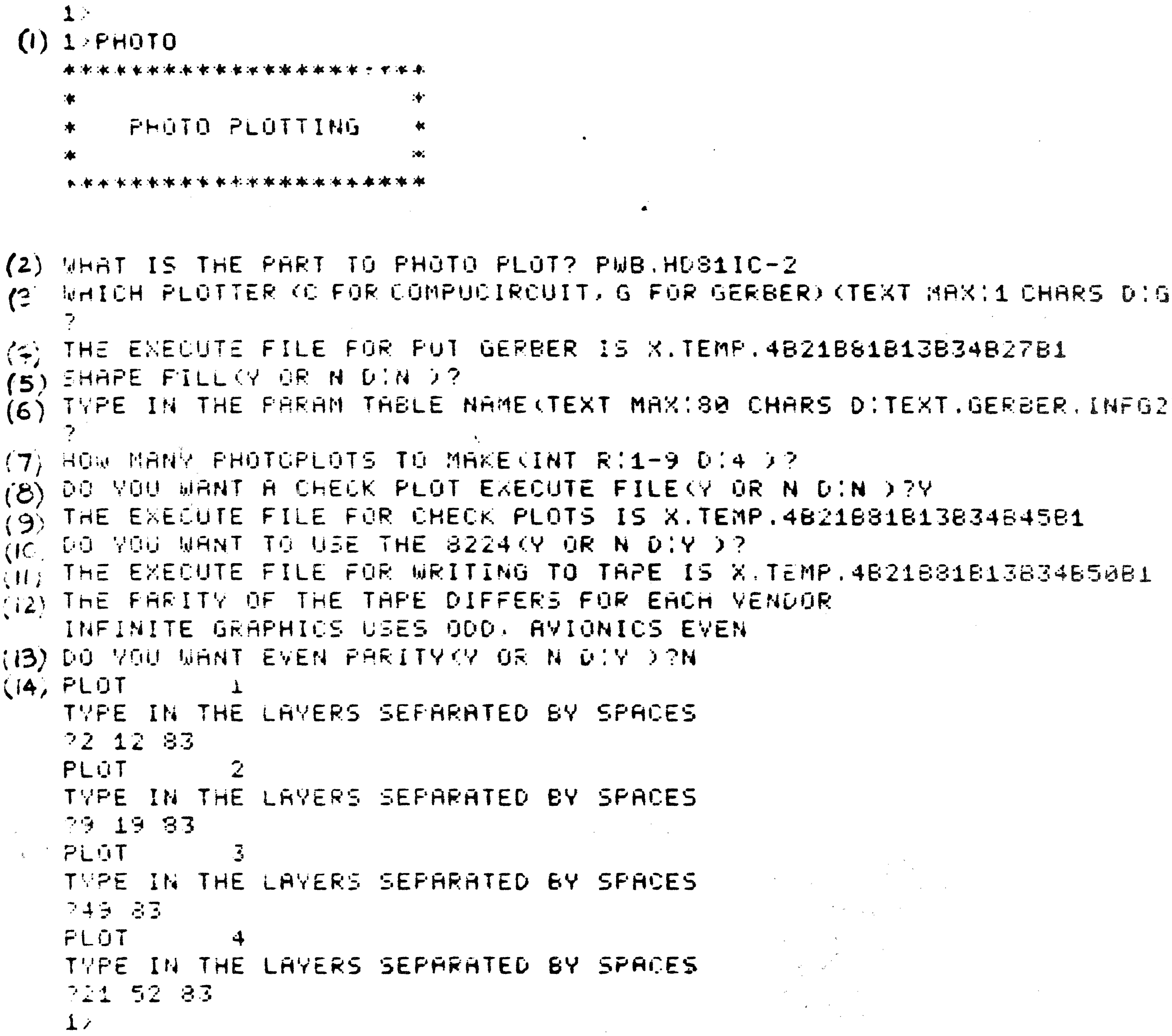
Programming languages have been constantly maturing during the past 25 years. Modern languages have become easier to code, more comprehensible to an outside group, and more reliable. These modern languages also give more computer power per language construct than older, more poorly designed languages. A modern programming language will support a style of programming called structured programming. Structured programming is a systematic programming method developed around a small set of fundamental program control structures, and a small set of data structures that are used to create a program that is clearly laid out, easy to follow, easy to analyze, and easy to modify [4].

Computervision's CGOS 200/10 allows a choice between TPL and FORTRAN-S for any custom code that is run from the system level (we did not consider doing any CADDS programming because of the overhead of CADDS, and the problem of maintaining the software from Rev. to Rev.) [2,3,6]. Although TPL does not support structured programming techniques as well as some other languages (notably Pascal, ADA), it certainly supports this enlightened form of programming more than FORTRAN. In the words of one of the many Computer Science Prophets: FORTRAN is a dinosaur, we must allow it to settle into ancient history. Thus we chose TPL. Our choice was also influenced by some other Software Engineering considerations: Along with the supply of system subroutines that comes with CGOS 200/10, we also had access to TPL subroutines that were developed by Honeywell's Avionics Division. We also found adequate people resources for TPL software development within CV's Application Engineering crew, and also within our Avionics Division.

EXAMPLES:

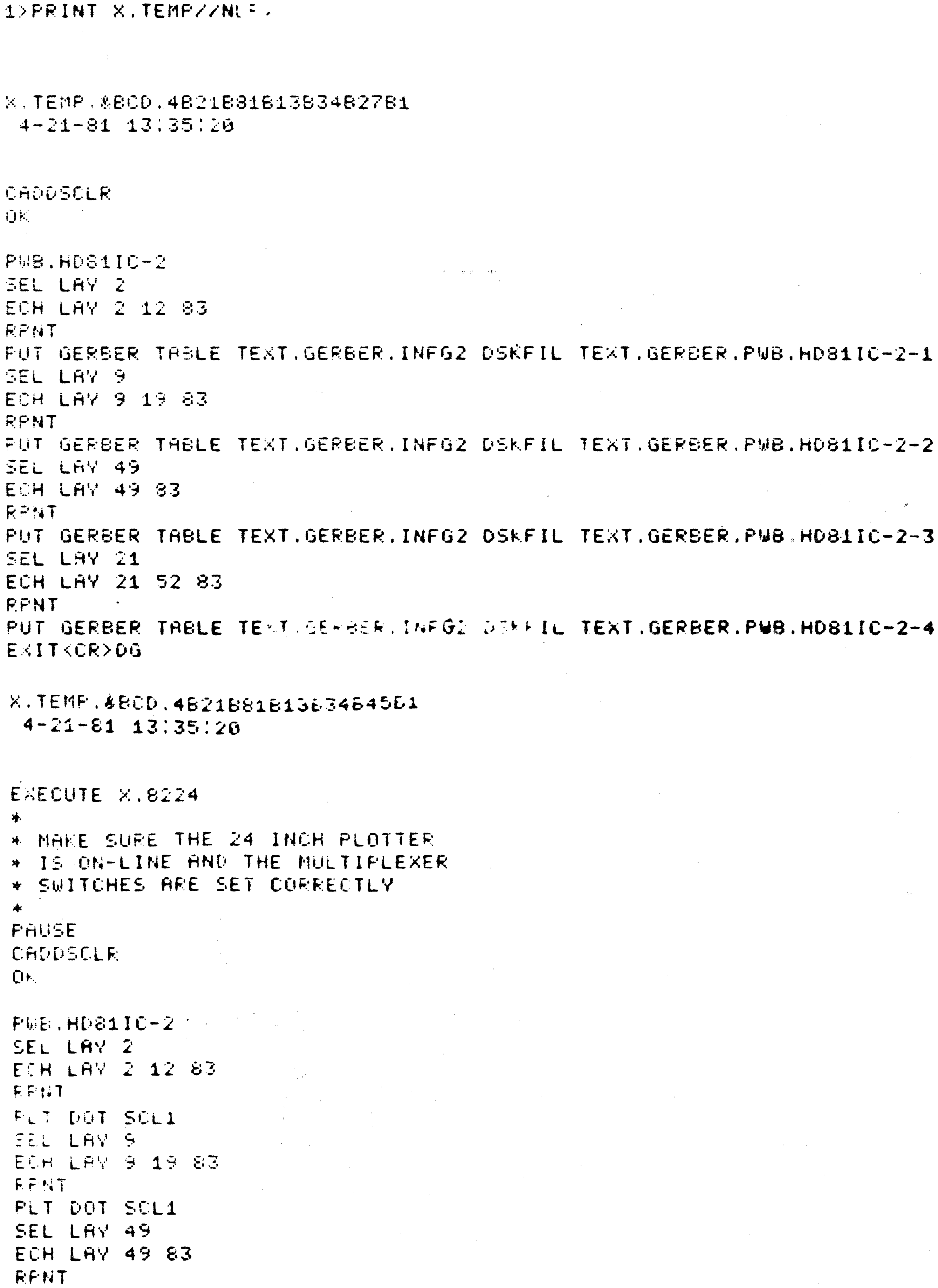
PHOTOPLOTTING

The first software written was to produce execute files that would create photoplot files for the Gerber or Compucircuit photoplotter. We use an offsite photoplotter and transfer our data to the plotter via tape. The following is the interaction that the operator has with the software:

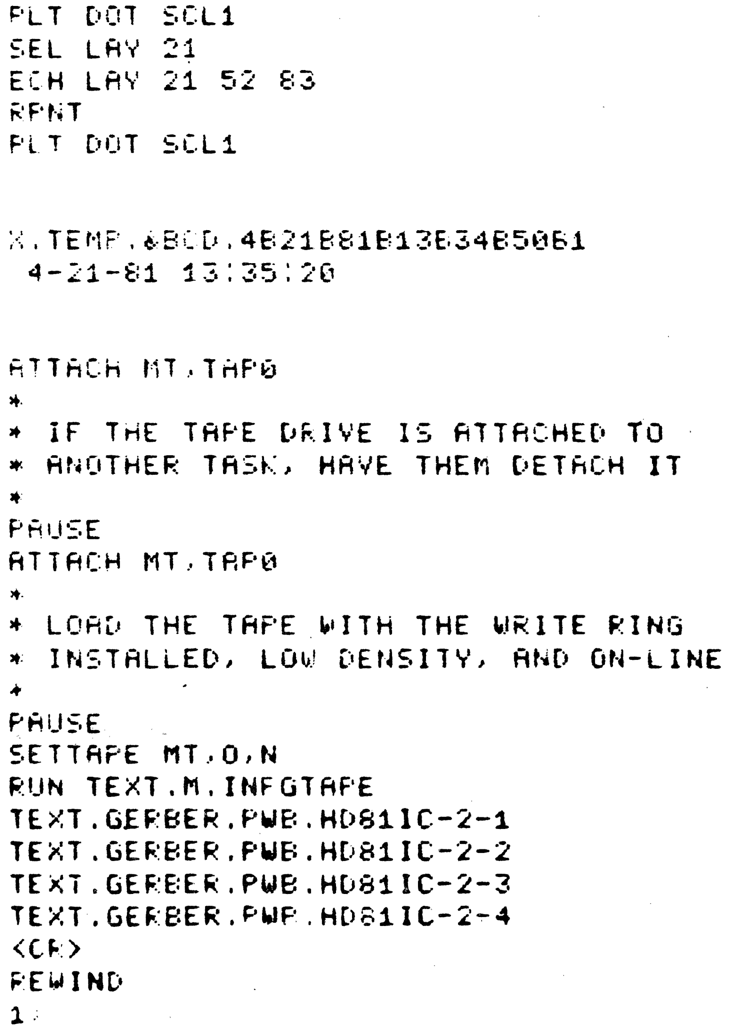


```
     1>
(1)  1>PHOTO
     ************************
     *                      *
     *    PHOTO PLOTTING    *
     *                      *
     ************************


(2)  WHAT IS THE PART TO PHOTO PLOT? PWB.HD81IC-2
(3)  WHICH PLOTTER (C FOR COMPUCIRCUIT, G FOR GERBER)(TEXT MAX:1 CHARS D:G)
     ?
(4)  THE EXECUTE FILE FOR PUT GERBER IS X.TEMP.4B21B81B13B34B27B1
(5)  SHAPE FILL(Y OR N D:N )?
(6)  TYPE IN THE PARAM TABLE NAME(TEXT MAX:80 CHARS D:TEXT.GERBER.INFG2)
     ?
(7)  HOW MANY PHOTOPLOTS TO MAKE(INT R:1-9 D:4 )?
(8)  DO YOU WANT A CHECK PLOT EXECUTE FILE(Y OR N D:N )?Y
(9)  THE EXECUTE FILE FOR CHECK PLOTS IS X.TEMP.4B21B81B13B34B45B1
(10) DO YOU WANT TO USE THE 8224(Y OR N D:Y )?
(11) THE EXECUTE FILE FOR WRITING TO TAPE IS X.TEMP.4B21B81B13B34B50B1
(12) THE PARITY OF THE TAPE DIFFERS FOR EACH VENDOR
     INFINITE GRAPHICS USES ODD, AVIONICS EVEN
(13) DO YOU WANT EVEN PARITY(Y OR N D:Y )?N
(14) PLOT        1
     TYPE IN THE LAYERS SEPARATED BY SPACES
     ?2 12 83
     PLOT        2
     TYPE IN THE LAYERS SEPARATED BY SPACES
     ?9 19 83
     PLOT        3
     TYPE IN THE LAYERS SEPARATED BY SPACES
     ?49 83
     PLOT        4
     TYPE IN THE LAYERS SEPARATED BY SPACES
     ?21 52 83
     1>
```

This program is run from the system level (1), the user types in the part (2), and the plotter to use (3). The operator at (3) is informed that text input is requested, with a maximum of 1 character, and the default is "G". The operator is informed of the execute file name at (4) followed by a parameter table request (5,6). At (7) the number of photoplots is determined: the input is integer, the range is between 1 and 9 plots, and the default is 4 plots. The operator is then interrogated if a check plot execute file is needed (this will produce the commands to plot the board on either the 24 or 42 inch Versatec plotters (9,10)). The execute file that writes the photoplot output from the Put Gerber commands to tape is annunciated and the parity for the tape drive is determined (12,13,14). In (14) each plot layer scheme is determined. The following is a listing of each execute file produced.



```
1>PRINT X.TEMP//NLF,


X.TEMP.&BCD.4B21B81B13B34B27B1
 4-21-81 13:35:20


CADDSCLR
OK

PWB.HD81IC-2
SEL LAY 2
ECH LAY 2 12 83
RPNT
PUT GERBER TABLE TEXT.GERBER.INFG2 DSKFIL TEXT.GERBER.PWB.HD81IC-2-1
SEL LAY 9
ECH LAY 9 19 83
RPNT
PUT GERBER TABLE TEXT.GERBER.INFG2 DSKFIL TEXT.GERBER.PWB.HD81IC-2-2
SEL LAY 49
ECH LAY 49 83
RPNT
PUT GERBER TABLE TEXT.GERBER.INFG2 DSKFIL TEXT.GERBER.PWB.HD81IC-2-3
SEL LAY 21
ECH LAY 21 52 83
RPNT
PUT GERBER TABLE TEXT.GERBER.INFG2 DSKFIL TEXT.GERBER.PWB.HD81IC-2-4
EXIT<CR>DG

X.TEMP.&BCD.4B21B81B13B34B45B1
 4-21-81 13:35:20


EXECUTE X.8224
*
* MAKE SURE THE 24 INCH PLOTTER
* IS ON-LINE AND THE MULTIPLEXER
* SWITCHES ARE SET CORRECTLY
*
PAUSE
CADDSCLR
OK

PWB.HD81IC-2
SEL LAY 2
ECH LAY 2 12 83
RPNT
PLT DOT SCL1
SEL LAY 9
ECH LAY 9 19 83
RPNT
PLT DOT SCL1
SEL LAY 49
ECH LAY 49 83
RPNT
```



```
PLT DOT SCL1
SEL LAY 21
ECH LAY 21 52 83
RPNT
PLT DOT SCL1


X.TEMP.&BCD.4B21B81B13B34B50B1
 4-21-81 13:35:20


ATTACH MT.TAP0
*
* IF THE TAPE DRIVE IS ATTACHED TO
* ANOTHER TASK, HAVE THEM DETACH IT
*
PAUSE
ATTACH MT.TAP0
*
* LOAD THE TAPE WITH THE WRITE RING
* INSTALLED, LOW DENSITY, AND ON-LINE
*
PAUSE
SETTAPE MT.O,N
RUN TEXT.M.INFGTAPE
TEXT.GERBER.PWB.HD81IC-2-1
TEXT.GERBER.PWB.HD81IC-2-2
TEXT.GERBER.PWB.HD81IC-2-3
TEXT.GERBER.PWB.HD81IC-2-4
<CR>
REWIND
1
```

In the last execute file produced, the line RUN TEXT.M.INFGTAPE is a homegrown program that transfers the disk file photoplot commands to tape in the correct character set and the correct tape blocking format.

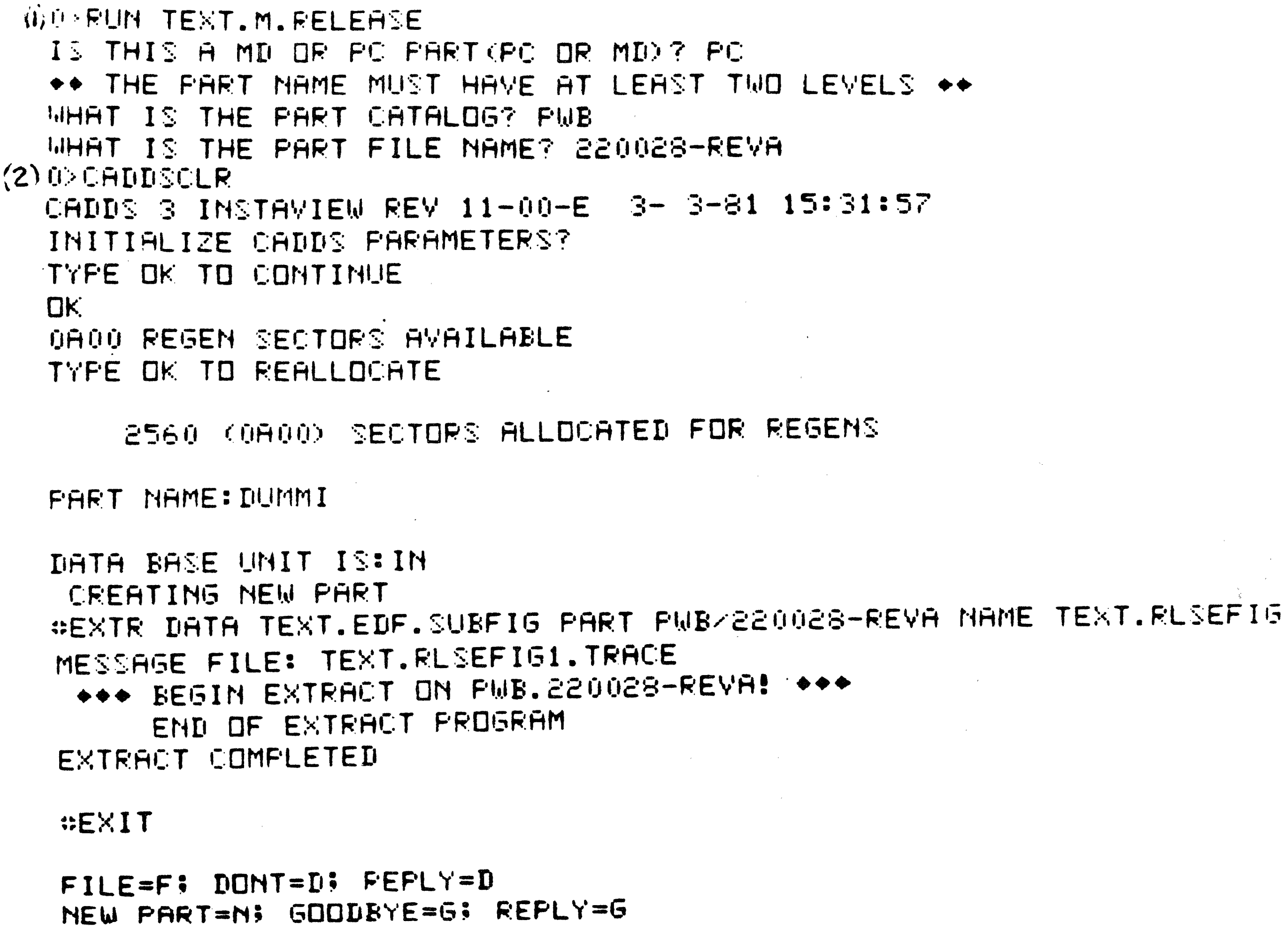
With this technique of generating custom execute files for each job, we will save at least 100 hours per year of valuable system time. I think the casual CADDS user can see the unmeasurable benefits of having the TPL software produce these execute files rather than the operator himself.

AUTOROUTING

Similarly we have produced software to preprocess the ROUTE BRD command. As in photoplotting, the operator is asked about each autoroute parameter (with an optional explanation) and an execute file is set up with the ROUTE command and the user defined parameters (or a default if a return was hit to a parameter request). This file is then executed on our Zero Salary Shift with the operator receiving the routed board in the morning. We estimate a savings of 100 to 200 hours per year per 8 hour shift of system time.

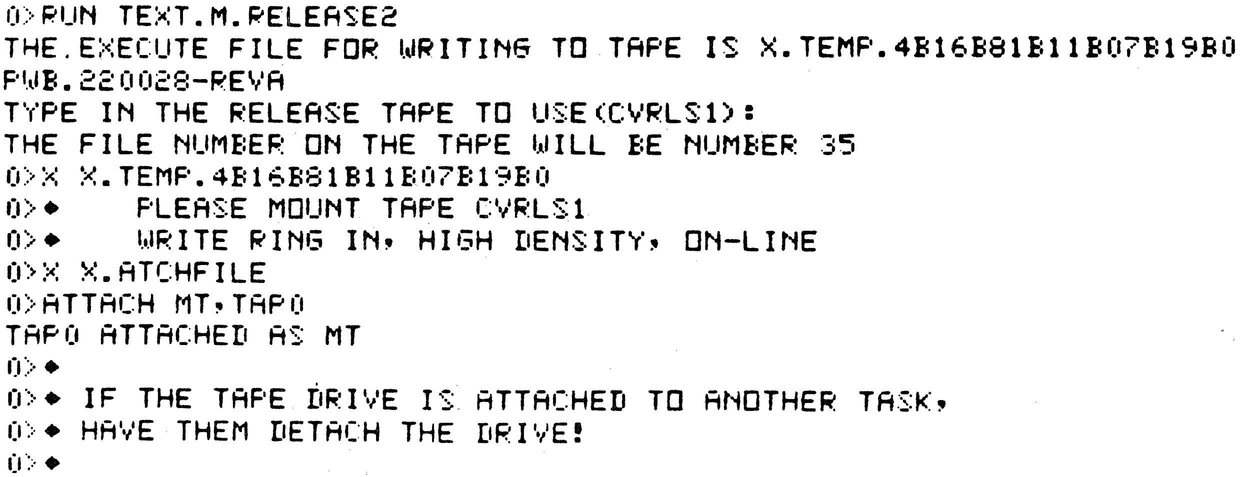
PART RELEASING & RESTORING

Releasing and Restoring parts as official engineering documentation can be very messy. Our definition of a released part consists of the part and any subfigures used on the part. This package is put onto mag tape and deleted from the disk. We also require that the subfigures put on tape with each part be renamed to avoid any overwrite of library figures with the same name that may have changed since the part was released. To put all this messing around in software has greatly simplified our operations. The software is run as follows:

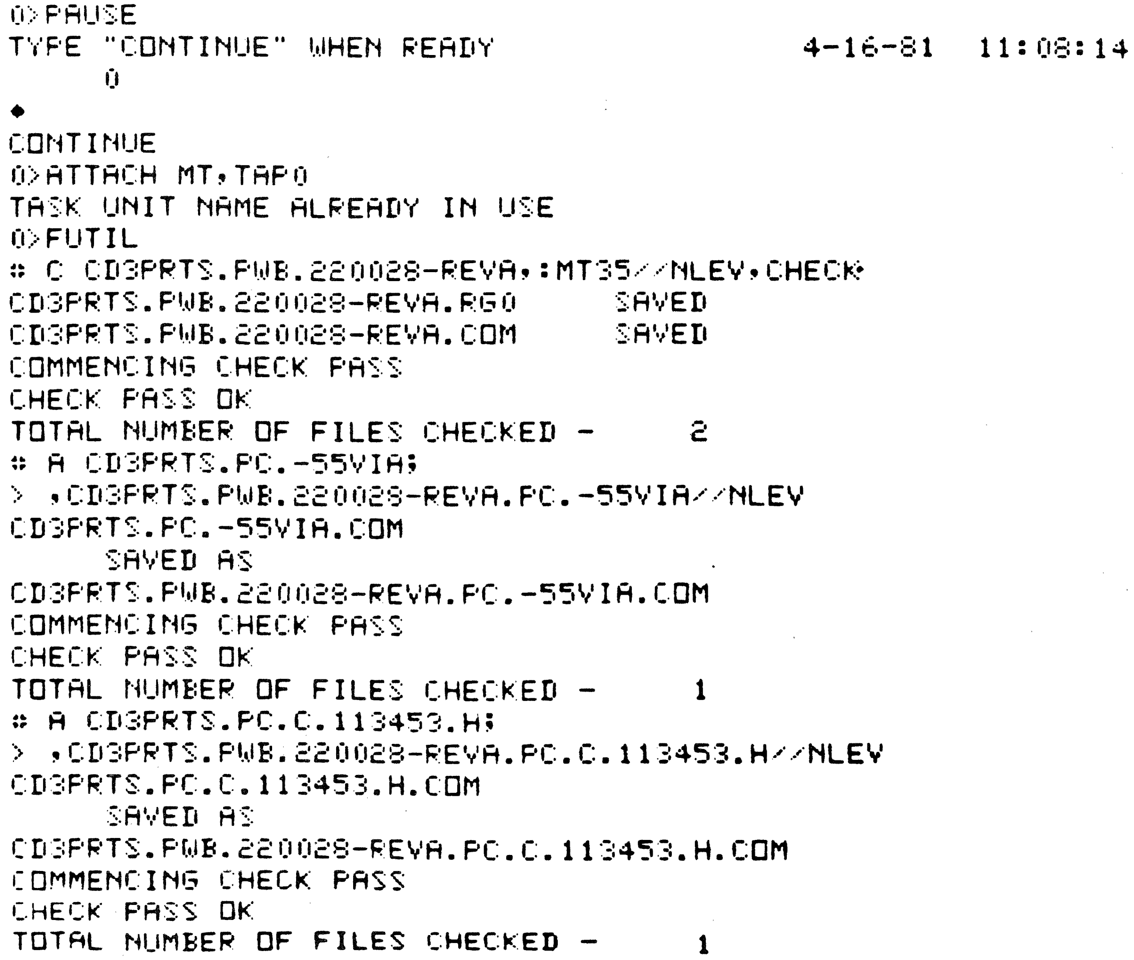


```
(1)0>RUN TEXT.M.RELEASE
  IS THIS A MD OR PC PART(PC OR MD)? PC
  ◆◆ THE PART NAME MUST HAVE AT LEAST TWO LEVELS ◆◆
  WHAT IS THE PART CATALOG? PWB
  WHAT IS THE PART FILE NAME? 220028-REVA
(2)0>CADDSCLR
  CADDS 3 INSTAVIEW REV 11-00-E  3- 3-81 15:31:57
  INITIALIZE CADDS PARAMETERS?
  TYPE OK TO CONTINUE
  OK
  0A00 REGEN SECTORS AVAILABLE
  TYPE OK TO REALLOCATE

       2560 (0A00) SECTORS ALLOCATED FOR REGENS

  PART NAME:DUMMI

  DATA BASE UNIT IS:IN
   CREATING NEW PART
  :EXTR DATA TEXT.EDF.SUBFIG PART PWB/220028-REVA NAME TEXT.RLSEFIG
  MESSAGE FILE: TEXT.RLSEFIG1.TRACE
   ◆◆◆ BEGIN EXTRACT ON PWB.220028-REVA! ◆◆◆
       END OF EXTRACT PROGRAM
  EXTRACT COMPLETED

  :EXIT

  FILE=F; DONT=D; REPLY=D
  NEW PART=N; GOODBYE=G; REPLY=G
```

At (1) the program TEXT.M.RELEASE is run to start an execute file at (2) that extracts subfigures from the released board. At this point another program is run from the execute file that determines the release tape to use, the file number on the tape, and analyzes the extracted subfigure names to construct an execute file that copies all pertinent files to tape and changes the name of the subfigures:

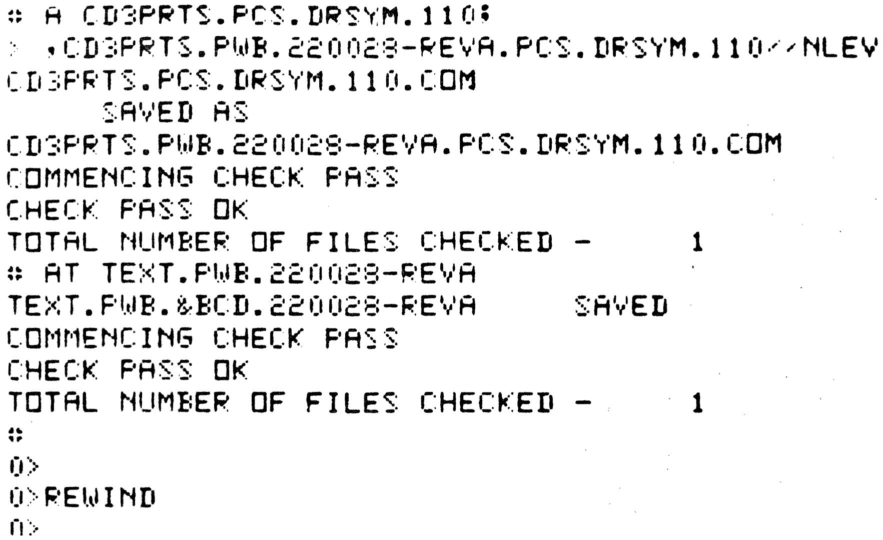


```
0>RUN TEXT.M.RELEASE2
THE.EXECUTE FILE FOR WRITING TO TAPE IS X.TEMP.4B16B81B11B07B19B0
PWB.220028-REVA
TYPE IN THE RELEASE TAPE TO USE(CVRLS1):
THE FILE NUMBER ON THE TAPE WILL BE NUMBER 35
0>X X.TEMP.4B16B81B11B07B19B0
0>◆    PLEASE MOUNT TAPE CVRLS1
0>◆    WRITE RING IN, HIGH DENSITY, ON-LINE
0>X X.ATCHFILE
0>ATTACH MT,TAP0
TAP0 ATTACHED AS MT
0>◆
0>◆ IF THE TAPE DRIVE IS ATTACHED TO ANOTHER TASK,
0>◆ HAVE THEM DETACH THE DRIVE!
0>◆
```



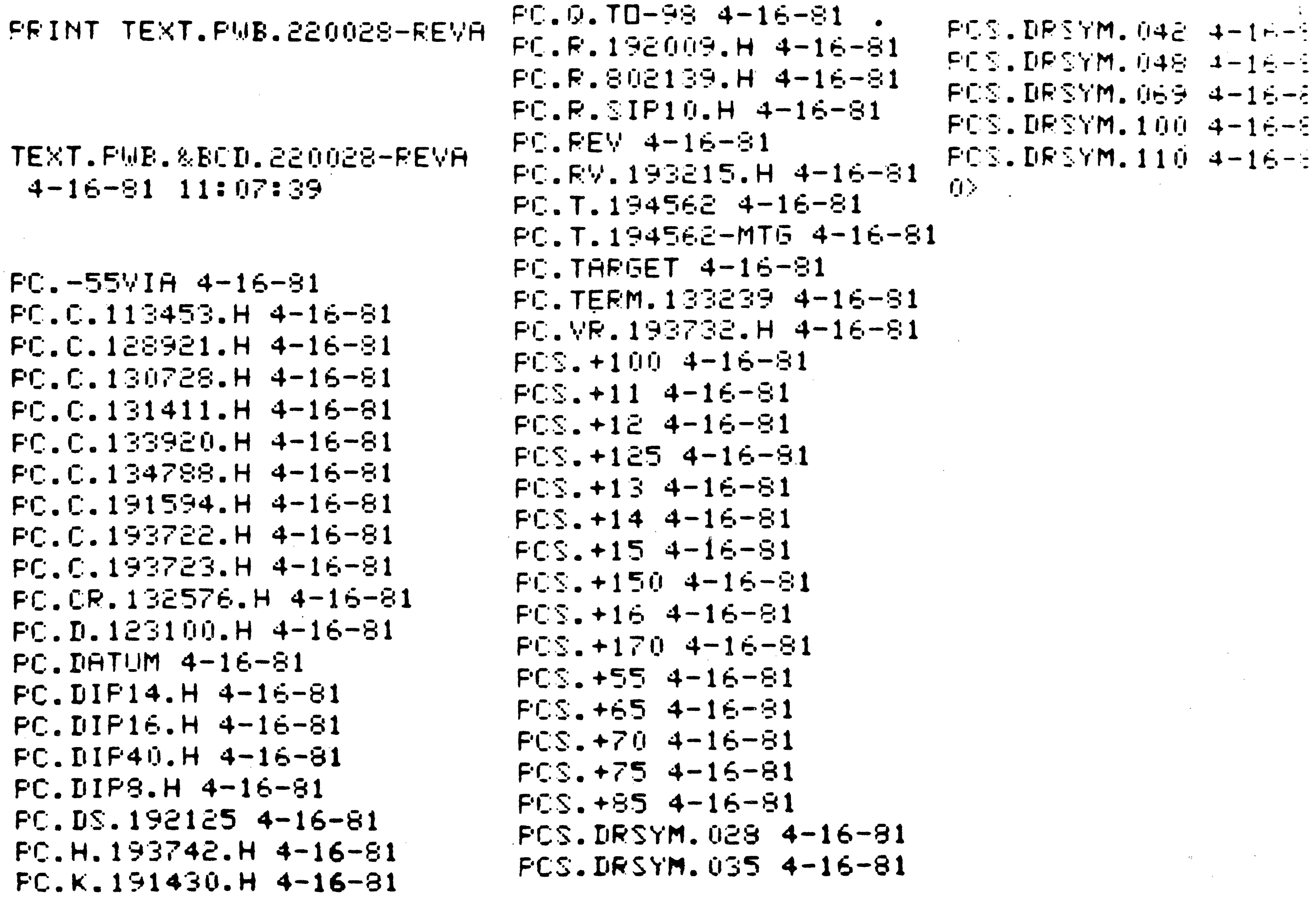
```
0>PAUSE
TYPE "CONTINUE" WHEN READY                    4-16-81   11:08:14
     0
◆
CONTINUE
0>ATTACH MT,TAP0
TASK UNIT NAME ALREADY IN USE
0>FUTIL
: C CD3PRTS.PWB.220028-REVA,:MT35//NLEV,CHECK,
CD3PRTS.PWB.220028-REVA.RG0       SAVED
CD3PRTS.PWB.220028-REVA.COM       SAVED
COMMENCING CHECK PASS
CHECK PASS OK
TOTAL NUMBER OF FILES CHECKED -      2
: A CD3PRTS.PC.-55VIA;
> ,CD3PRTS.PWB.220028-REVA.PC.-55VIA//NLEV
CD3PRTS.PC.-55VIA.COM
     SAVED AS
CD3PRTS.PWB.220028-REVA.PC.-55VIA.COM
COMMENCING CHECK PASS
CHECK PASS OK
TOTAL NUMBER OF FILES CHECKED -      1
: A CD3PRTS.PC.C.113453.H;
> ,CD3PRTS.PWB.220028-REVA.PC.C.113453.H//NLEV
CD3PRTS.PC.C.113453.H.COM
     SAVED AS
CD3PRTS.PWB.220028-REVA.PC.C.113453.H.COM
COMMENCING CHECK PASS
CHECK PASS OK
TOTAL NUMBER OF FILES CHECKED -      1
```

and this continues for all subfigures ending as follows:



```
: A CD3PRTS.PCS.DRSYM.110;
> ,CD3PRTS.PWB.220028-REVA.PCS.DRSYM.110//NLEV
CD3PRTS.PCS.DRSYM.110.COM
     SAVED AS
CD3PRTS.PWB.220028-REVA.PCS.DRSYM.110.COM
COMMENCING CHECK PASS
CHECK PASS OK
TOTAL NUMBER OF FILES CHECKED -      1
: AT TEXT.PWB.220028-REVA
TEXT.PWB.&BCD.220028-REVA      SAVED
COMMENCING CHECK PASS
CHECK PASS OK
TOTAL NUMBER OF FILES CHECKED -      1
:
0>
0>REWIND
0>
```

The last file added, TEXT.& BCD.PWB22028-REVA, contains a list of subfigures on the board and the release date of each subfigure. This file is used to produce a report when restoring the board:



```
PRINT TEXT.PWB.220028-REVA


TEXT.PWB.&BCD.220028-REVA
 4-16-81 11:07:39


PC.-55VIA 4-16-81
PC.C.113453.H 4-16-81
PC.C.128921.H 4-16-81
PC.C.130728.H 4-16-81
PC.C.131411.H 4-16-81
PC.C.133920.H 4-16-81
PC.C.134788.H 4-16-81
PC.C.191594.H 4-16-81
PC.C.193722.H 4-16-81
PC.C.193723.H 4-16-81
PC.CR.132576.H 4-16-81
PC.D.123100.H 4-16-81
PC.DATUM 4-16-81
PC.DIP14.H 4-16-81
PC.DIP16.H 4-16-81
PC.DIP40.H 4-16-81
PC.DIP8.H 4-16-81
PC.DS.192125 4-16-81
PC.H.193742.H 4-16-81
PC.K.191430.H 4-16-81
```

```
PC.Q.TD-98 4-16-81 .
PC.R.192009.H 4-16-81
PC.R.802139.H 4-16-81
PC.R.SIP10.H 4-16-81
PC.REV 4-16-81
PC.RV.193215.H 4-16-81
PC.T.194562 4-16-81
PC.T.194562-MTG 4-16-81
PC.TARGET 4-16-81
PC.TERM.133239 4-16-81
PC.VR.193732.H 4-16-81
PCS.+100 4-16-81
PCS.+11 4-16-81
PCS.+12 4-16-81
PCS.+125 4-16-81
PCS.+13 4-16-81
PCS.+14 4-16-81
PCS.+15 4-16-81
PCS.+150 4-16-81
PCS.+16 4-16-81
PCS.+170 4-16-81
PCS.+55 4-16-81
PCS.+65 4-16-81
PCS.+70 4-16-81
PCS.+75 4-16-81
PCS.+85 4-16-81
PCS.DRSYM.028 4-16-81
PCS.DRSYM.035 4-16-81
```

```
PCS.DRSYM.042 4-16-
PCS.DRSYM.048 4-16-
PCS.DRSYM.069 4-16-
PCS.DRSYM.100 4-16-
PCS.DRSYM.110 4-16-
0>
```

The software to restore a part will produce an execute file to copy the part and subfigures from tape to disk, and produce a report of any changes that have occurred to a library figure since the part was released.

LIBRARY MANAGEMENT

We have attempted to simplify the management of our libraries by some utility software that manages a cross reference of part number to library figure name. One program will insert new entries in the index and report if the part number already exists. Another utility will report the library name when the operator inputs the part number. By automating the cross reference we have shortened our board layout time and stopped the proliferation of notebooks around the computer area.

PROBLEMS

Whenever software is developed on a system the maintenance costs of the software must be considered. These small utilities generally require little maintenance but sometimes there is some retooling to do when a new release of software comes from CV or our own internal requirements change. Depending on what changes, this retooling can be costly. At this point we have experienced stability from rev. to rev. but we recognize that things might change in the future. (This is especially true when we migrate from CADDS III to CADDS IV).

RESULTS AND CONCLUSIONS

As mentioned earlier, we will be saving at least 300 hours per year by using the Zero Salary Shift, and an undetermined payback from simplifying complex commands. Our operators have enthusiastically been using the software as they want to finish their jobs as soon as possible and move on to the next one. The user really likes the speed also. Compared to some CADDS commands, the operators feel that custom software run from the system level is lightning quick.

In the 80's a strong emphasis of software development will be on demystifying its use. The user of a software project will find the software friendly, easy to use, and problem free. By doing some custom TPL software we have cleaned up Photoplotting, Auto-route parameters, Library Management, and Part Archiving. But this is just a start, we will continue to identify those areas that we can improve productivity by use of our Zero Salary Shift or those areas that are complicated, hard to learn and used by multiple operators. These areas will save hundreds of hours per year of valuable system time and will continue to validate this valuable computer system tool.

REFERENCES

1. Bergland, G.D., "Structured Design Methodologies", *Tutorial: Software Design Strategies; IEEE*. pp. 162-181.
2. Computervision Corp., *CGOS 200/10 Operator Manual.*
3. Computervision Corp., *CGOS 200/10 Programming Languages (FORTRAN/TPL).*
4. Zelkowitz, Marvin V., "Perspectives on Software Engineering", *ACM Computing Survey 10*, 2 June 1978, pp. 197-216.
5. Irby, Thomas S., "Human Factors and Psychology Contributors to Improved End User Interfaces", *Proc. Honeywell's 5th Annual Software Engineering Conference*, pp. HH1-HH10.
6. Minkin, Paul A., and Luck, James, "The Roommates: CALMA and Computervision", *Proc. Honeywell's 2nd Annual CADCAM Workshop*, pp. CADX-1-CADX-16.

INTERACTIVE GRAPHIC VIBRATION CHARACTERIZATION OF PRINTED CIRCUIT BOARD ASSEMBLIES USING VOUGHT DEVELOPED CAE SOFTWARE ON COMPUTERVISION

By
Jerry Hegg
Vought Corporation
Unit 2-54120
P.O. Box 225907
Dallas, TX 75265
214-266-4957

1.0 INTRODUCTION

1.1 As an experienced military printed circuit board designer, I readily recognize and appreciate the value in being able to "ballpark" a PC board sinusoidal vibration response during the initial design stages. Vibration problems found later in the design process or during testing typically require expensive rework for their resolution. Board design parameters such as size, weight, thickness, PC substrate characteristics, and mounting methods all contribute to its vibration response.

1.2 In the past, simple metal plate equations were used assuming various boundary conditions, size, weight, etc., and rough hand calculations made. Now with the advent of the computer, and specifically computer graphics, these "ballpark" calculations are made right at the CRT. Single degree-of-freedom response graphs may be plotted on the CRT and on paper. Response curves may be superimposed on the CRT to analyze effects of parameter variations.

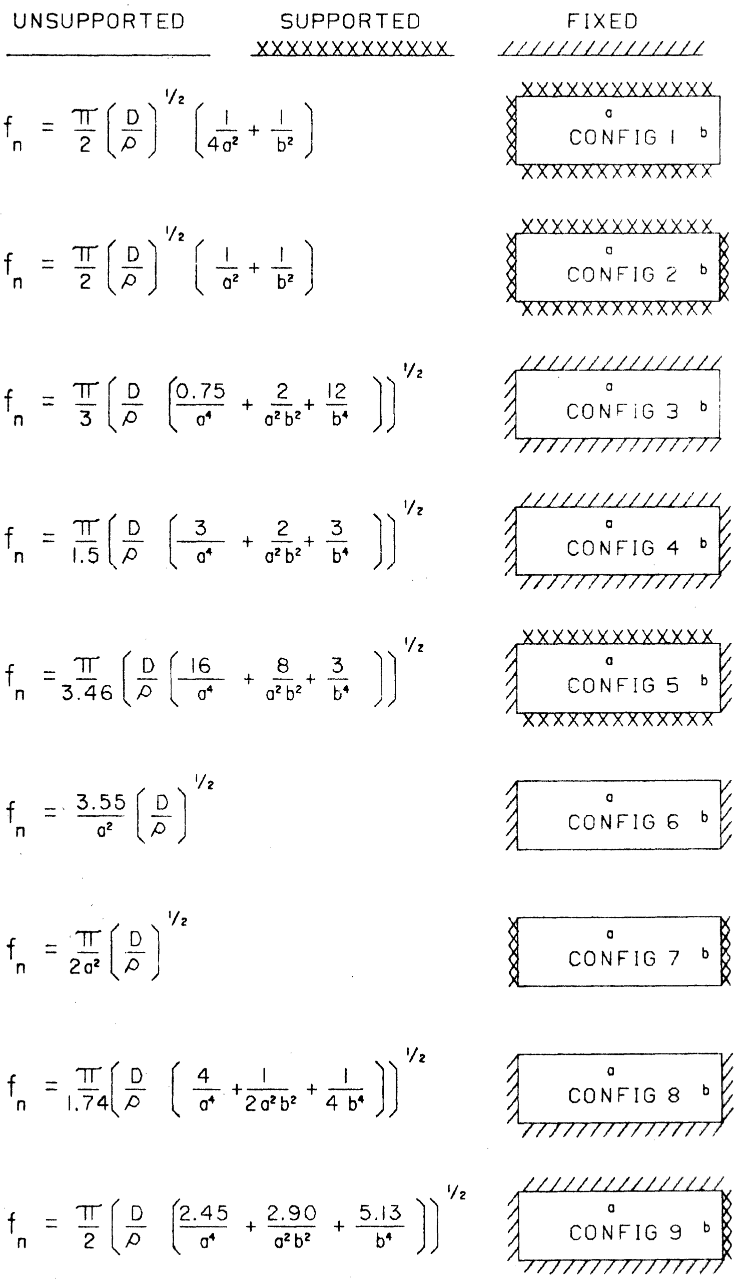
1.3 Experience has shown that use of simple equations for rectangular plates with evenly distributed mass and with maximum deflection at the center of the board provides adequate analysis capability for preliminary design. These analyses produce a reasonable measure of amplitude and natural frequency, within approximately +/−15% that could be expected based upon typical first cut estimates of masses and dimensional values.

1.4 It is recognized that this approach has limitations because of variations in characteristics of glass epoxy board materials and the accuracies of the estimates on other parameters — i.e., component weight. Component weight is specified by manufacturers as a maximum or "not to exceed" quantity. Therefore, to obtain more accurate results, exact component weight data is required. A limitation of these equations has to do with the nonlinear response of the standardly used glass epoxy PC substrate. Therefore, reliable response data cannot be evaluated for vibration inputs in excess of 3 g's (only relative response data can be evaluated). There are more sophisticated methods which may be used for detailed analysis where required but this paper addresses only the simplified first approximation type equations.

1.5 However, with this tool a PC designer can evaluate his design's vibration response quickly without spending detailed modeling time and expensive mainframe computer time using software analysis tools such as NASTRAN.

2.0 VIBRATION EQUATIONS

2.1 The equations utilized are derived as shown in Dave Steinberg's book published by McGraw-Hill entitled, "Vibration Analysis for Electronic Equipment", therefore an attempt at derivation will not be made here. Each equation is used to calculate the fundamental resonant frequency at the PC board center. The equations and edge conditions are shown in Figure 1.



$$f_n = \frac{\pi}{2}\left(\frac{D}{\rho}\right)^{1/2}\left(\frac{1}{4a^2}+\frac{1}{b^2}\right)$$

$$f_n = \frac{\pi}{2}\left(\frac{D}{\rho}\right)^{1/2}\left(\frac{1}{a^2}+\frac{1}{b^2}\right)$$

$$f_n = \frac{\pi}{3}\left(\frac{D}{\rho}\left(\frac{0.75}{a^4}+\frac{2}{a^2b^2}+\frac{12}{b^4}\right)\right)^{1/2}$$

$$f_n = \frac{\pi}{1.5}\left(\frac{D}{\rho}\left(\frac{3}{a^4}+\frac{2}{a^2b^2}+\frac{3}{b^4}\right)\right)^{1/2}$$

$$f_n = \frac{\pi}{3.46}\left(\frac{D}{\rho}\left(\frac{16}{a^4}+\frac{8}{a^2b^2}+\frac{3}{b^4}\right)\right)^{1/2}$$

$$f_n = \frac{3.55}{a^2}\left(\frac{D}{\rho}\right)^{1/2}$$

$$f_n = \frac{\pi}{2a^2}\left(\frac{D}{\rho}\right)^{1/2}$$

$$f_n = \frac{\pi}{1.74}\left(\frac{D}{\rho}\left(\frac{4}{a^4}+\frac{1}{2a^2b^2}+\frac{1}{4\,b^4}\right)\right)^{1/2}$$

$$f_n = \frac{\pi}{2}\left(\frac{D}{\rho}\left(\frac{2.45}{a^4}+\frac{2.90}{a^2b^2}+\frac{5.13}{b^4}\right)\right)^{1/2}$$

Figure 1

2.2 The parameters used in the equations are defined as:

f_n = Natural frequency of PC board
a = Board length
b = Board width
W = Board weight (including components)
g = 386 in/sec^2 (gravity)
h = Board thickness
E = Modulus of elasticity of board material
(2×10^6 lb/in^2 for glass epoxy)
u = Poisson's ratio (.12 for glass epoxy)

$$D = \frac{Eh^3}{12(1-u^2)}$$

$$\rho = \frac{\text{mass}}{\text{area}} = \frac{W}{gab}$$

2.3 The "Q" or transmissibility of the board is approximated as follows:
$Q_m = 1.5\sqrt{f_n}$
2.4 Maximum G response to a sinusoidal input at f_n is:
$G_{out} = Q_m G_{in}$
2.5 The three sigma response to a random vibration input is as follows:
$G_{out} = 3\sqrt{\frac{\pi}{2} D f_n Q}$ RMS
2.6 The transmissibility at any frequency is as follows:

$$Q = \frac{1 + \left(\frac{f}{f_n Q_m}^2\right)}{\left(1 - \frac{f^2}{f_n^2}\right)^2 + \left(\frac{f}{f_n Q_m}\right)^2}$$

2.7 Using the above equations the natural frequency f_n, the maximum g response G_{out}, and a transmissibility Q at any frequency can be calculated. A plot of the transmissibility curve starting at f = 0 can also be accomplished.

3.0 INPUT FILE

3.1 The input parameter file for the solution is structured as follows:

1) Mount Cnfg	= 1
2) Length	= XXXXXX.XXX inches
3) Width	= XXXXXX.XXX inches
4) Thickness	= .062 inches
5) Weight	= XXXXXX.XXX pounds
6) Elasticity	= 2000000.000 lbs/in$_2$
7) MY Factor	= .120
8) G input	= XXXXXX.XXX GRMS SIN
9) P.S. Density	= XXXXXX.XXX G$_2$/HZ

3.2 In order to call up the program in CADDS, the command "FREQ" is entered through the keyboard. The input file above is automatically displayed and the designer enters the line number to be edited and then the parameter value, followed by a carriage return. When the input file has been created, the option to "CALCULATE", "RE-DISPLAY" and "EDIT" the input file, or "QUIT" are available. If calculate is selected the system immediately prints out the natural frequency f_n, the maximum amplitude in g's, sine and random; maximum amplitude in inches, sine and random as illustrated in Figure 2.

```
1) MOUNT CNFG        =         6
2) LENGTH            =         8.500 INCHES
3) WIDTH             =         4.752 INCHES
4) THICKNESS         =          .062 INCHES
5) WEIGHT            =         0.335 POUNDS
6) ELASTICITY        = 2000000.000 LBS/IN2
7) MU FACTOR         =         0.125
8) G-INPUT           =         5.750 GRMS SIN
9) P.S. DEN          =         1.259 G2/HERTZ
ENTER LINE NUMBER TO CHANGE, "D" TO RE-DISPLAY, "C" TO COMPUTE
BOARD FREQUENCY, OR "Q" TO QUIT
C
NATURAL FREQUENCY =       67.335 HERTZ
QN                =       12.309
SAMP SIN          =        0.15297613
SAMP RAN          =        0.26252122
GOUT-SIN          =       70.775 GRMS SIN
GOUT-RAN          =      121.456 GRMS RANDOM
TYPE OK TO PLOT
OK
```

Figure 2

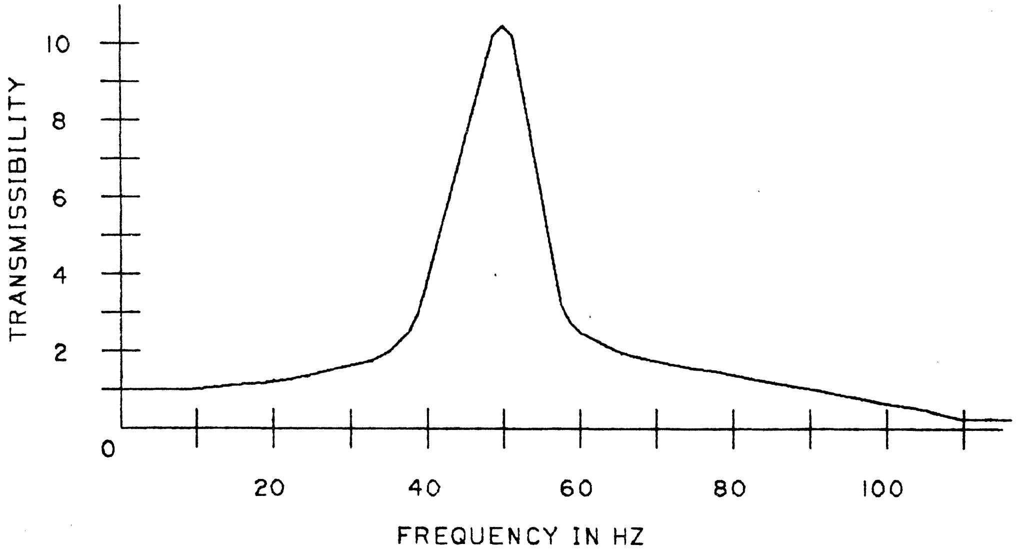
4.0 OUTPUT FILE

4.1 When "CALCULATE" has been selected and calculated response data displayed, options available then are "RE-DISPLAY" the input file, "EDIT", "QUIT", or "PLOT". In order to plot the response curve, a basic graph outline CDC3PRT must be on the screen. The response curve is plotted quickly as illustrated in figure 3. Another "CALCULATE" and "PLOT" can then be superimposed as many times as desired. Individual plots or composites may be stored on disk or tape as standard CD3PRT's.

5.0 CONCLUSIONS

5.1 This software developed at Vought is a handy tool for "ballparking" a PC board single-degree-of-freedom vibration response to a sinusoidal input during the initial layout stage of a design right at the CRT. It is not a tool to replace finite element modeling and analysis but may be effectively used to compare designs and point out questionable designs for more detailed analysis.

5.2 Possible future enhancements for this tool would be to use the system's data extract capabilities to measure the size of the designed board, extract weight of each component, mounting configuration, etc., and automatically build the input file. In addition, more detailed analysis approaching NASTRAN may be possible with 3D graphical output similar to Figure 4. This type of output would depict the g-force amplitude profile and indicate the exact location of the point(s) of maximum response rather than assuming a uniform distribution of mass and maximum displacement at the PC board center.


TRANSMISSIBILITY
10
8
6
4
2
0
20
40
60
80
100
FREQUENCY IN HZ

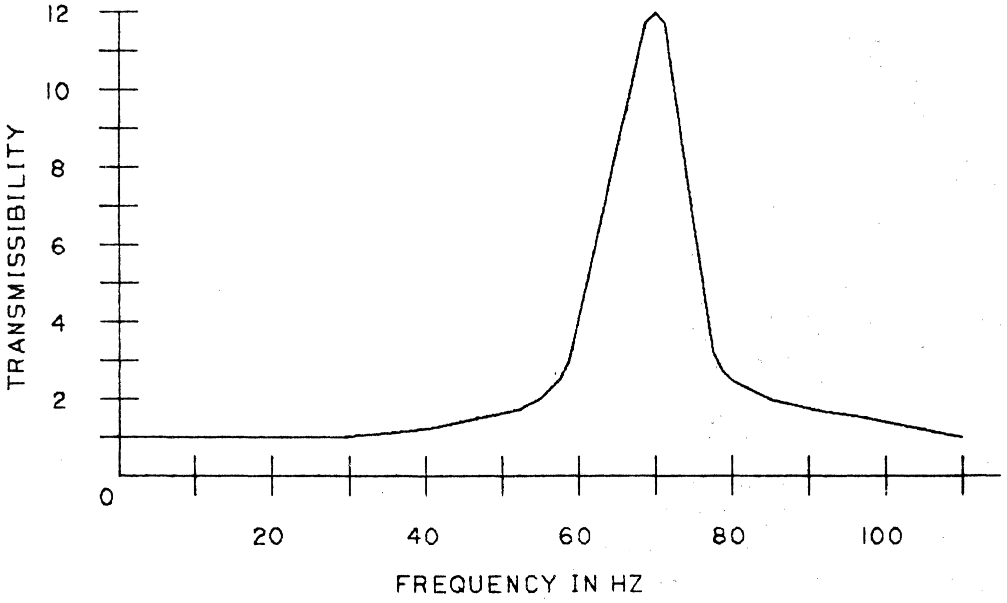

TRANSMISSIBILITY
12
10
8
6
4
2
0
20
40
60
80
100
FREQUENCY IN HZ

Figure 3

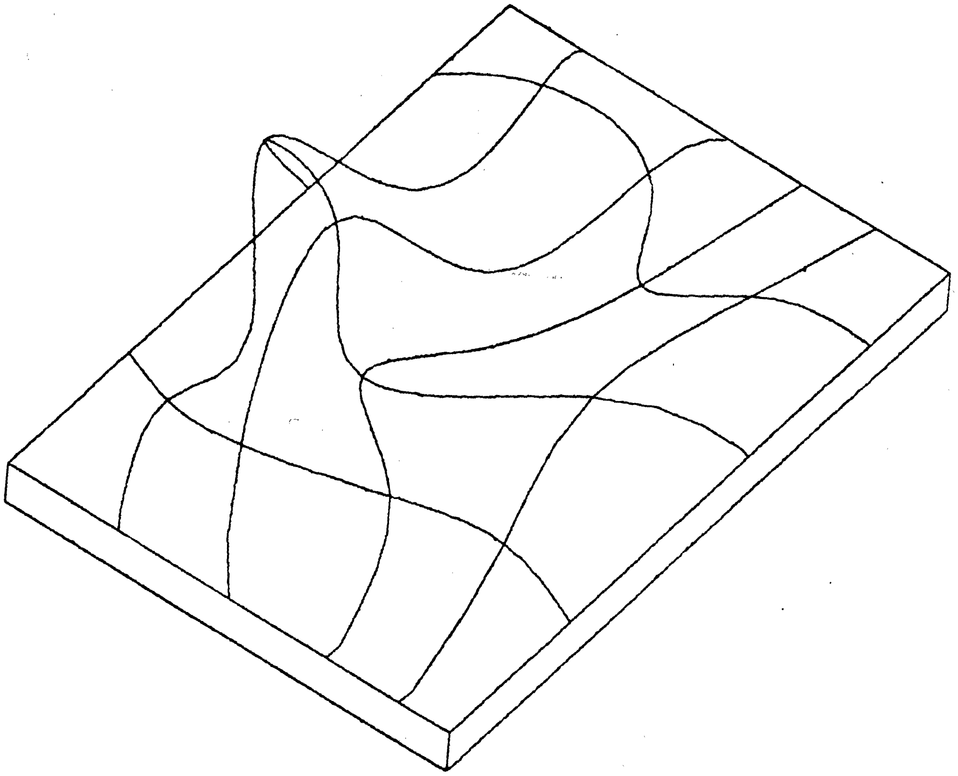


Figure 4

STRIPLINE ARTWORKS IN CADDS III
A CLOSE TOLERANCE,
HIGH FREQUENCY CIRCUIT
ON A SPECIAL SUBSTRATE

Gary Goertzen
Rockwell International
CTG
P.O. Box 10462
Dallas, TX 75207
214-690-5000

APPLICATION REQUIREMENTS

0 ACCURATE ARTWORK (±.0005)

0 NONSTANDARD FEATURES

0 TEST REAL CIRCUIT

0 FAST ARTWORK GENERATION

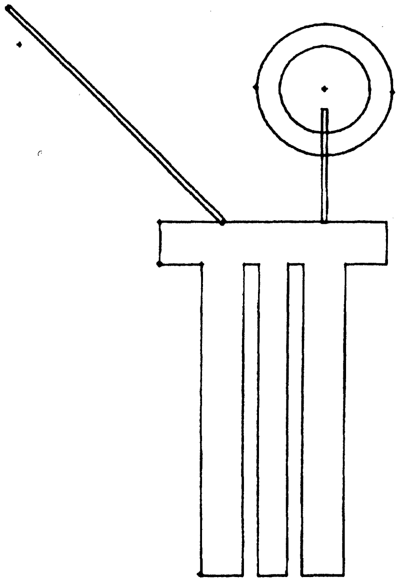
0 QUICK CHANGES (MULTI GENERATIONS)

0 EASY METHOD OF INTEGRATING CIRCUITS

ACCURATE ARTWORK (±.0005)

- GERBER OUTPUT

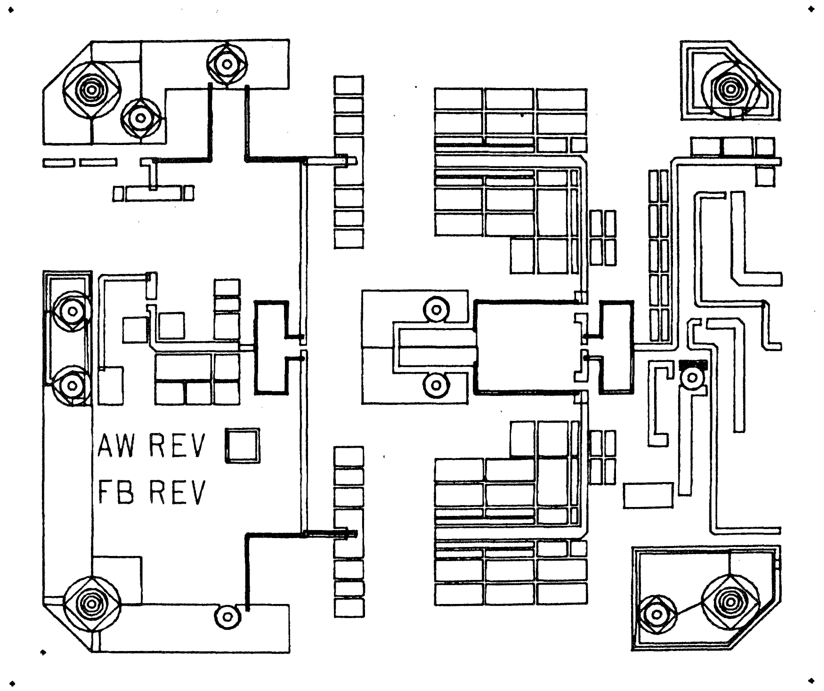
- SCALE OF 1-1 OR GREATER

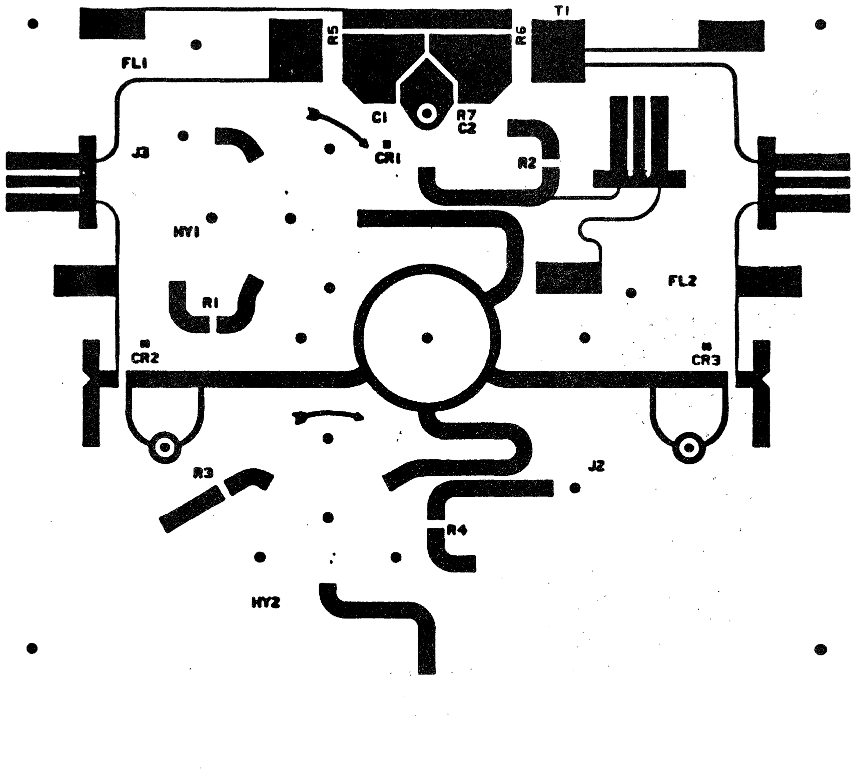


NONSTANDARD FEATURES

- **CIRCUITS NOT MADE WITH STANDARD LINES AND PADS**

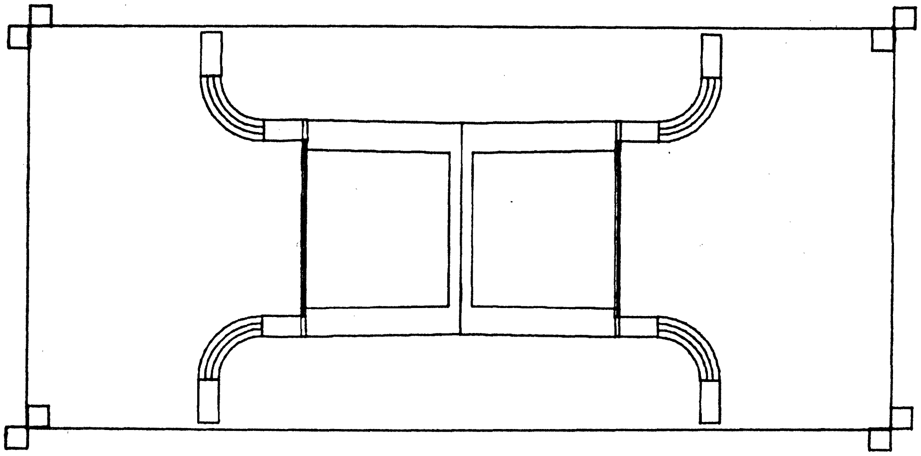
- **MANY DIFFERENT GEOMETRICAL SHAPES, ARCS, AND CURVES**


AW REV
FB REV


FL1
R5
R6
T1
C1
R7
C2
J3
CR1
R2
HY1
FL2
R1
CR2
CR3
R3
J2
R4
HY2

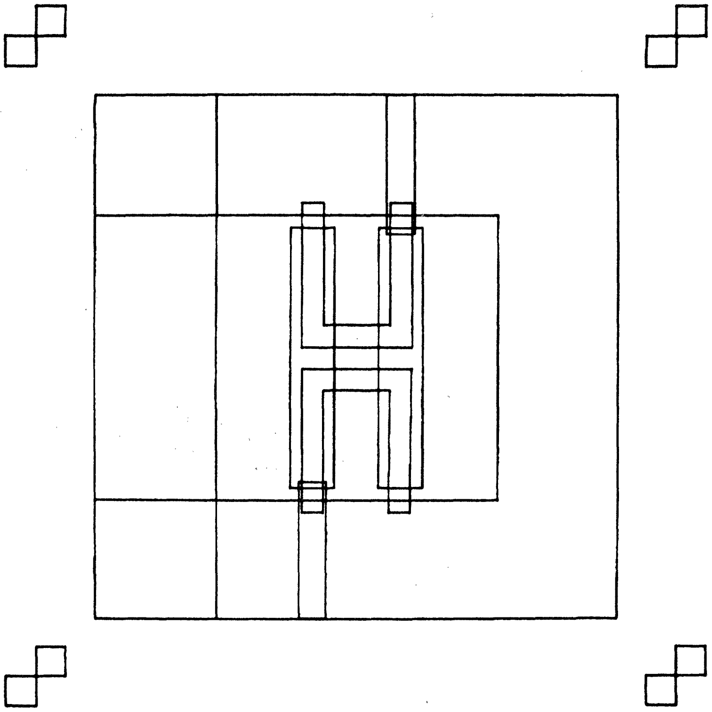
TEST REAL CIRCUIT

- CAN BE DONE EACH TIME IN CADDS III
- CAN ONLY BE DONE ON ACCURATE ARTWORK



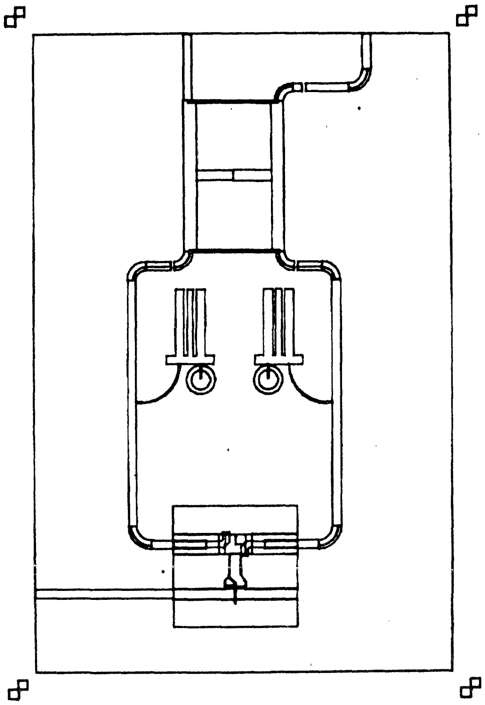
FAST ARTWORK GENERATION

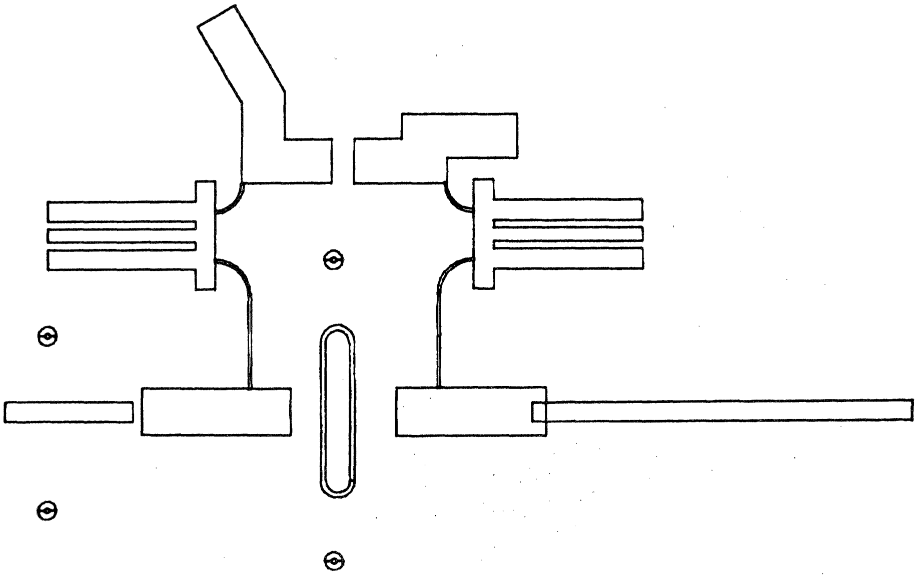
- IN CADDS III (NEW WAY)
- MANUAL CUT AND PEEL (COORD.) (OLD WAY)
- SOME CLAD CUT AND SCRAPE (VERY INACCURATE)



QUICK CHANGES (MULTI GENERATIONS)

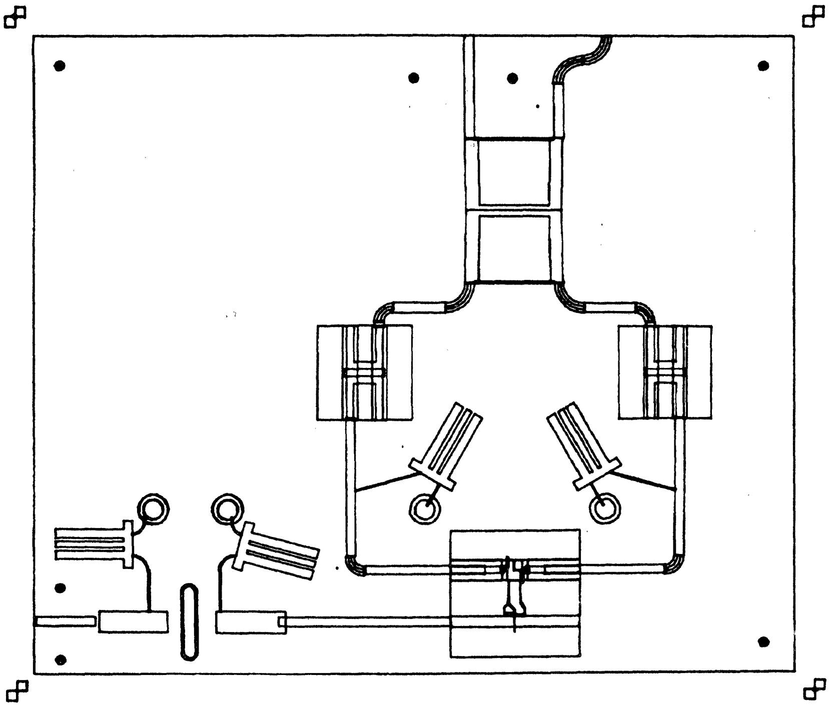
- MADE EASILY IN CADDS III
- CUT NEW ART OR FIX OLD
- SOME CLAD CUT AND SCRAPE





EASY METHOD OF INTEGRATING CIRCUITS

- INSERT PARTS IN CADDS III
- MAKE NEW LAYOUT AND CUT NEW ART



Electrical Diagrams

CAD GENERATED WIRELISTS

Larry Shuhy
Harris Corporation
Publication Press Division
P.O. Box 515
Westerly, RI 02891
203-599-1500 x409
May 20, 1981

ABSTRACT

Data extract/data merge is a useful tool for retrieving information from drawings generated on the Computervision system. Although the emphasis has been on organizing printed circuit information for autoroute, the basic technique may be adapted to a variety of processes which require large quantities of data to be retrieved and organized. With information properly applied on an electrical schematic, a true to/from list may be extracted, sorted and printed in batch mode.

BACKGROUND

The generation of wirelists to be used in building a printing press has in the past been a tedious process. Due to the competitive nature of this business, there is much customized design. Printers themselves strive for greater productivity and have unique requirements for their products. As a result, no two presses go out looking alike. Every change affects the electrical design of the press and every press is wired differently. This requires that a new wirelist be generated each time, with thousands of wires being identified.

In the past, press electrical documentation consisted of schematics and back panel wiring. There were three or four general schematics each containing a number of subsystems. A change/addition/deletion of any part on any subsystem required a whole new schematic. The backpanel wiring contained the to/from information and changes required a new drawing here too. There was, as a result, a unique drawing package for each customer. Creating this package required up to 1,500 hours of engineers' time, and the possibilities for error were tremendous.

ANALYSIS

Analyzing this process in the interest of increased productivity, it was determined that if the total electrical system was broken down into subsystems which didn't change, the pieces could be put together to create a customer order without creating or modifying any drawings. Orders would look more uniform and the level of expertise required to process the order would go down. Because little or nothing changed, the error rate would go down, and the only variable would be the wirelist which would be automatically generated by extracting schematics.

A program was undertaken and a presentation was given to management explaining the details of the new technique using the CAD system for order processing. Because there was significant up front time involved in setting up the system, it had to be justified on cost savings downstream.

The overall objective of the program was to shorten the time required to process the electrical portion of a customer order through re-organization of the work and use of CAD. Success would be measured in the time required to produce the necessary documentation for the next customer order. The up front time to regroup, redraw and check the drawings, as well as the time to create a schematic library of parts, would not be counted towards the first order, although it was estimated that all this would take less time than a typical manual order. Expectations were that a typical order would be cut from 1,500 hours down to less than 40 hours. Payback would be realized on the second order.

DETAILS

The tasks required to achieve this objective included creating a library of schematic parts which conformed to ANSI standards, creating a set of schematics which represented the press electrical system and devising a program which would extract the information and sort it into the proper form.

Creating a library of schematics was not as large a project as originally anticipated. Parts were taken from drawings as required and organized in a catalog. Once created, each symbol was checked graphically, by layer, and by property to assure conformance to standards. Once a part is used as a nodal figure, it may not be altered without affecting past drawings. Care must be taken to get them right to avoid this problem later. About 150 parts were identified and cataloged (see Figure 1).

Creating the set of electrical schematics for the press system represented the bulk of the program. The existing drawings had to be separated into logical subgroups which represented electrical options. Each drawing required additional information which could not come from the original drawings. This is information not normally included on a pure schematic. This included color, gauge, signal name and cable number. Each drawing had to be checked and extracted to be sure that all information was in the proper location on the drawings. This information included reference designators, pin numbers, to/from sequence and wire identification.

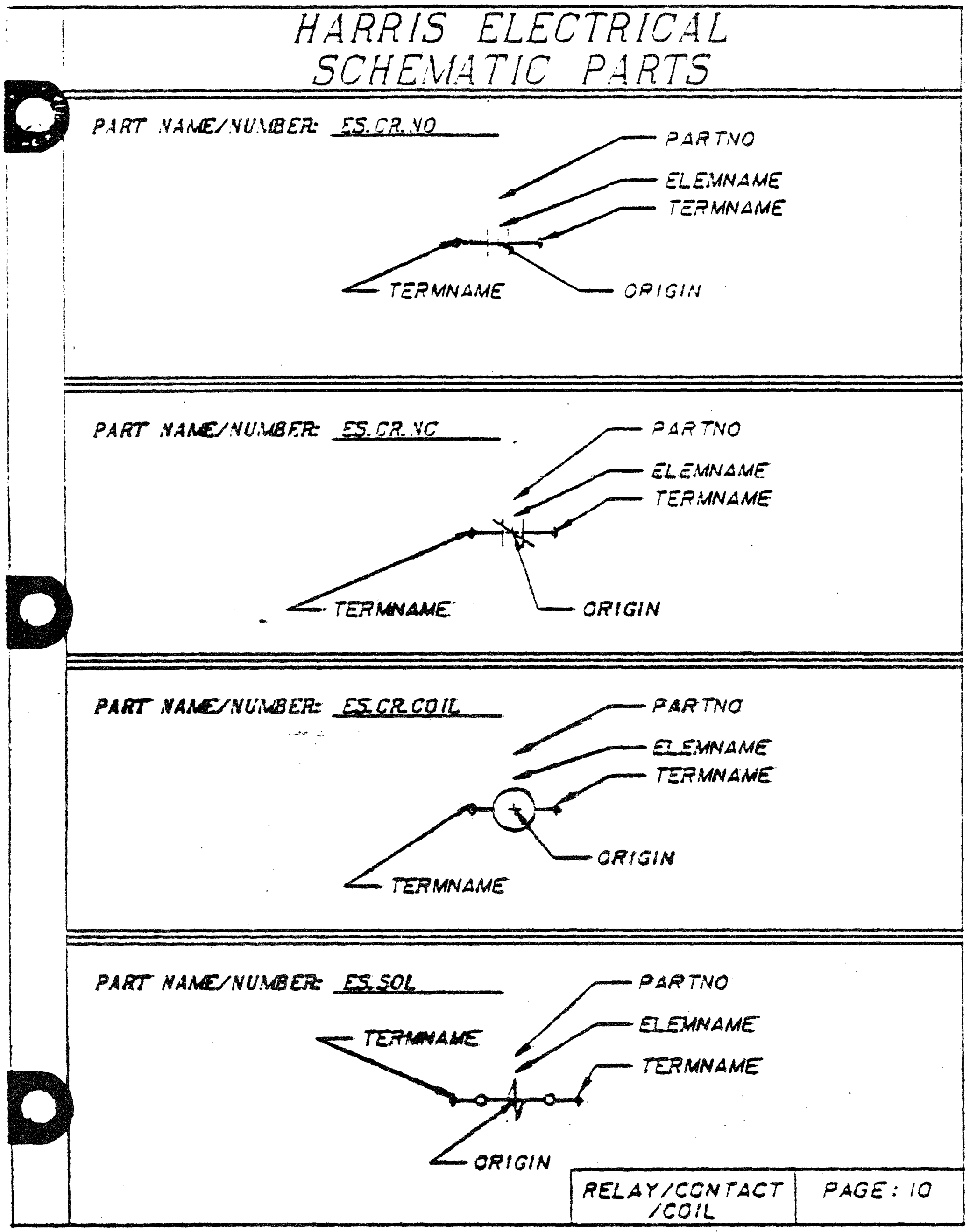
A drawing created in this manner is not really a schematic any more. In order to achieve a true to/from list, every "line" on a schematic must truly represent the wire it identifies. A wire may have only two ends and dots may not be used as intersection points, but only as graphical entity. Wires must be drawn continuous from one terminal to another. This requires knowledge of the physical location of parts to make reasonable sense of the wire routing.

Devising a program which properly extracted and sorted the schematic information had to go hand-in-hand with schematic creation. The form of the final output had to be agreed upon (see Figure 2). The form desired was not a natural output of the extract process. A wirelist with the "from" and the "to" on the same line was most useful for wiring purposes. Lists also had to be separated by enclosure or conduit to allow wiring of each section individually. By using extract, merge, rearrange, sort, and CVTECO, the final output was achieved.

ADVANTAGES

There are several advantages derived from the system as it was implemented. Some of them were directly measurable and others were more nebulous. Some of them were directly attributed to use of the Computervision system and others a result of the way the work was organized.

1. Because of the way the drawings are broken up, there is a set of twenty-two drawings which completely described the electrical system. All available options are included in this package. These drawings do not normally change, and may be supplied to field service as a comprehensive package. Greater familiarity may be gained with these drawings because they are standard. They have standard layouts and standard symbols and have been proven correct through repeated usage.



HARRIS ELECTRICAL SCHEMATIC PARTS

PART NAME/NUMBER: ES.CR.NO

PART NAME/NUMBER: ES.CR.NC

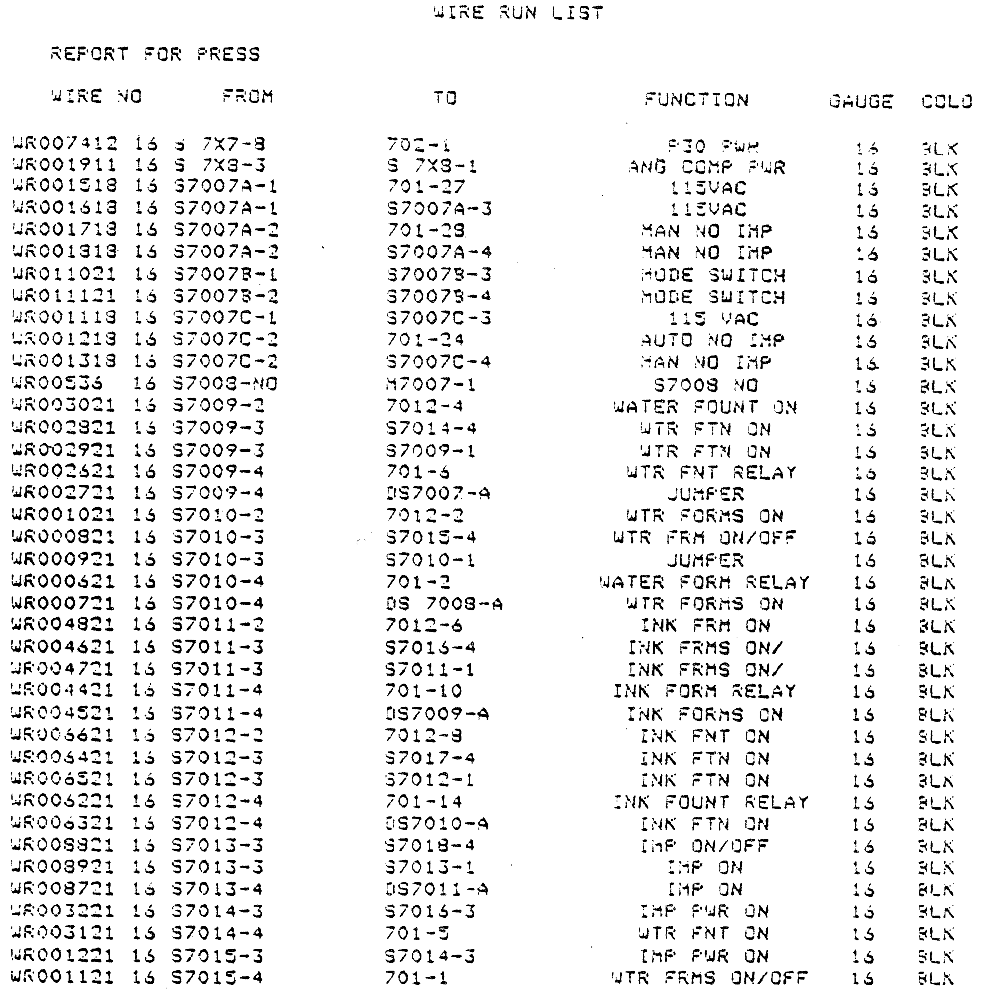
PART NAME/NUMBER: ES.CR.COIL

PART NAME/NUMBER: ES.SOL

RELAY/CONTACT /COIL	PAGE: 10

Figure 1: Page from Schematic Library Book

2. Having a standard symbol library does a number of things. First, it facilitates drawing on the system by having parts available for application to a drawing. Second, it restricts operators to using known, approved symbols rather than personal artistic endeavors. Third, it provides a position on each symbol for insertion of the properties required for meaningful extract. Fourth, it confines the choice of components to those which are available, keeping inventories down. New parts may be created on the system as required, but their layering and construction is checked before addition to the formal library.
3. Whenever a nodal line is put on a schematic, it is extracted. Wires which do not represent actual wires (i.e., PC clad) are not drawn as nodal lines. This means that every connection is accounted for. Every wire is listed along with its properties. This provides an error check on the final design. A tally and sort is also done on the terminations to determine whether any have been omitted or duplicated.



```
                                 WIRE RUN LIST
     REPORT FOR PRESS

     WIRE NO         FROM                 TO                    FUNCTION           GAUGE   COLO

WR007412 16 S 7X7-8               702-1                  P30 PWR              16     BLK
WR001911 16 S 7X8-3               S 7X8-1              ANG COMP PWR           16     BLK
WR001518 16 S7007A-1              701-27                  115VAC              16     BLK
WR001618 16 S7007A-1              S7007A-3                115VAC              16     BLK
WR001718 16 S7007A-2              701-28                MAN NO IMP            16     BLK
WR001818 16 S7007A-2              S7007A-4              MAN NO IMP            16     BLK
WR011021 16 S7007B-1              S7007B-3              MODE SWITCH           16     BLK
WR011121 16 S7007B-2              S7007B-4              MODE SWITCH           16     BLK
WR001118 16 S7007C-1              S7007C-3                115 VAC             16     BLK
WR001218 16 S7007C-2              701-24                AUTO NO IMP           16     BLK
WR001318 16 S7007C-2              S7007C-4              MAN NO IMP            16     BLK
WR00536  16 S7008-NO              M7007-1                S7008 NO             16     BLK
WR003021 16 S7009-2               7012-4              WATER FOUNT ON          16     BLK
WR002821 16 S7009-3               S7014-4               WTR FTN ON            16     BLK
WR002921 16 S7009-3               S7009-1               WTR FTN ON            16     BLK
WR002621 16 S7009-4               701-6               WTR FNT RELAY           16     BLK
WR002721 16 S7009-4               DS7007-A                JUMPER              16     BLK
WR001021 16 S7010-2               7012-2              WTR FORMS ON            16     BLK
WR000821 16 S7010-3               S7015-4            WTR FRM ON/OFF           16     BLK
WR000921 16 S7010-3               S7010-1                 JUMPER              16     BLK
WR000621 16 S7010-4               701-2             WATER FORM RELAY          16     BLK
WR000721 16 S7010-4               DS 7008-A           WTR FORMS ON            16     BLK
WR004821 16 S7011-2               7012-6               INK FRM ON             16     BLK
WR004621 16 S7011-3               S7016-4             INK FRMS ON/            16     BLK
WR004721 16 S7011-3               S7011-1             INK FRMS ON/            16     BLK
WR004421 16 S7011-4               701-10            INK FORM RELAY            16     BLK
WR004521 16 S7011-4               DS7009-A            INK FORMS ON            16     BLK
WR006621 16 S7012-2               7012-8               INK FNT ON             16     BLK
WR006421 16 S7012-3               S7017-4              INK FTN ON             16     BLK
WR006521 16 S7012-3               S7012-1              INK FTN ON             16     BLK
WR006221 16 S7012-4               701-14            INK FOUNT RELAY           16     BLK
WR006321 16 S7012-4               DS7010-A             INK FTN ON             16     BLK
WR008821 16 S7013-3               S7018-4             IMP ON/OFF              16     BLK
WR008921 16 S7013-3               S7013-1                IMP ON               16     BLK
WR008721 16 S7013-4               DS7011-A               IMP ON               16     BLK
WR003221 16 S7014-3               S7016-3              IMP PWR ON             16     BLK
WR003121 16 S7014-4               701-5                WTR FNT ON             16     BLK
WR001221 16 S7015-3               S7014-3              IMP PWR ON             16     BLK
WR001121 16 S7015-4               701-1             WTR FRMS ON/OFF           16     BLK
```

Figure 2: Wire Run List Report

4. Schematics are drawn by the engineering department. Because the wire routing must now be specified at the schematic level (as well as wire size and color), the manufacturing department no longer dictates these parameters. This documents the wire runs for the next build of the same product and maintains the engineering integrity of the design.
5. Each enclosure of a press may be built at a different facility and the final assembly done at the customer site. Each needs a list for wiring that portion of the product. Each line on the schematic, therefore, must be identified by the list on which it will appear. A table was compiled of the appropriate lists and their corresponding property identification. This value was entered into the cable property of each wire and later sorted. These lists are then sorted alpha-numerically by component number for ease of wiring.
6. The technique of wiring consoles with a wirelist and backpanel assembly is a deviation from our regular procedure. Normally, the wiring information is all contained on the asembly/wiring drawing which has a pigtail on each connection with text indicating the destination of that wire. Changes in this system required modification to that drawing and rechecking. Using the wirelist method, the wirelist is the only variable. Wiring the console entails running down the list, crossing off wires as they are done. This provides additional savings in the program, plus, again, less chance for error.
7. As the project progressed, we used a Versatec 11 inch plotter for check plots of the schematic. These plots became so popular that we have adopted them as the standard field service document. They have been arranged in a book with foldout pages. These

books are easy to handle, both for field service and the customer. Since the Computervision data base is the original, there need not be a D-size piece of paper to represent a schematic. These 11×22 sheets serve the purpose of a final document and revision levels are handled through the CAD system.

8. An extract of each schematic is stored in the system under an identifiable name and is updated only when the drawing is changed. The data extract process takes significant system time. During normal operation of the system, a schematic with 100 wires can take 30 minutes to extract. Twenty-two of these done in batch would then take 11 hours! Since they need not be redone unless they change, the extract text files for these schematics are strung together and sorted when the order is processed. The schematics, therefore, need not be cluttering up the storage area unless they are to be worked on.

THE PROCESS

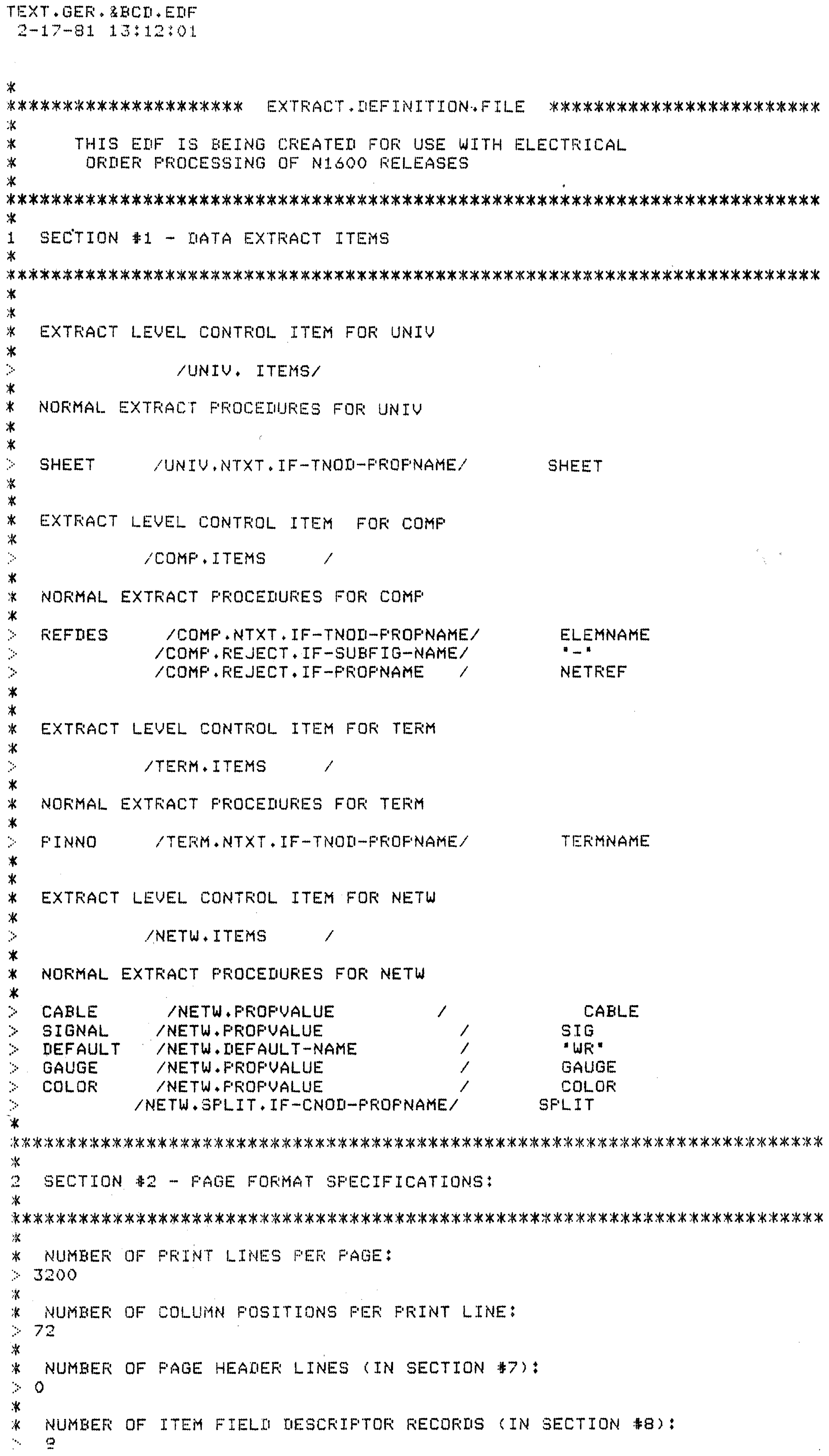
Library schematic figures as they were created, must include properties, Elemname "____", Tername "____", Part no "____", just as they would with a PC part. They may also have a Valvea "____", Volta "____", Watta "____", Currenta "____", if that particular part needs to be further defined. These parts are created and checked with execute programs which insures that all required information exists on the proper layers. Each connect node must have the property "split". These parts are inserted onto a drawing as nodal figures as required. A book has been assembled with the name and graphical representation of each identified as well as the location of tnods for the various properties and the origin for ease of insertion. All these parts are created on the same grid spacing and are checked to JIC standards where applicable. The parts are then protected from operator access. Once parts are used as nodal figures they may not be altered or deleted without affecting all drawings on which they appear.

With the components placed on the drawing, nodal lines are used to connect their terminals. Wires or nodal lines must be inserted only between the actual terminals connected. They must run the way the wire runs. To graphically approximate a real schematic, lines are run directly over each other and dots used to depict their intersection. The split property insures that each line has only two ends.

As lines are entered they must all be given a number of properties which will later be extracted. These include 1) color "____", 2) gauge "____", 3) cable "____", 4) sig "____". Properties may be selected and filled in as the lines are entered. Default values may be used and selectively changed later such as color "blk" and gauge "16", or whole drawings may be given properties by window. (Be sure to specify NLIN modifier!)

Text is inserted on the components for terminal numbers and reference designators. Justification and height of text has been predetermined for normal orientation and reasonable drawing density. Care must be taken when modifying this text that the desired tnod and property are attached.

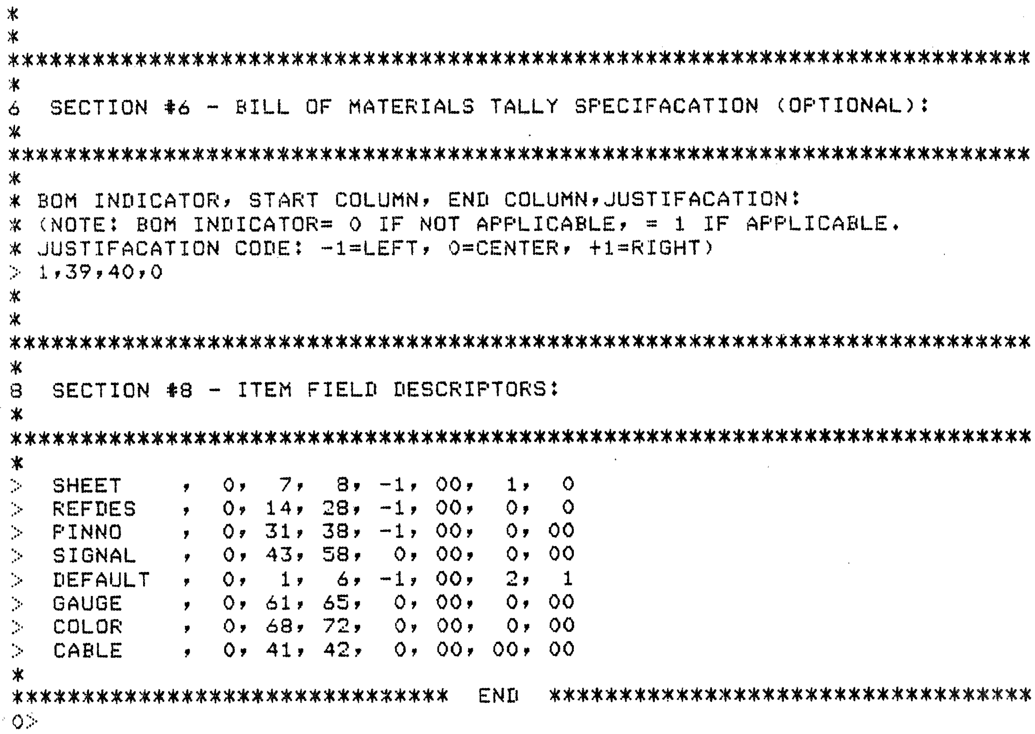
An extract program was written and compiled which derived all the desired information from the schematic and organized it into a report (see Figure 3). In order to sort this report into the required lists without destroying the integrity of the ends of the wires, each wire had to be resident on a single line. Normally the extract gives a two-line representation of a wire. This conversion was accomplished by doing a tally on the wire number and keying in on the tally to eliminate every other line. A format net was also performed, the normal report form has staggered fields (i.e., not columnized) as a result of variable field length in the reference designator — terminal number identification. A short TECO program was written to place the second field into a known column for sorting. Another file was created with all the information common to the extract output and FMT NET output using DATA MERGE. A run rearrange was done to take this scattered information and columnize it for the final report. Two sorts were then done, one by "cable number" and one by "froms" for wiring purposes. The output of this program is done on a Decwriter.



```
TEXT.GER.&BCD.EDF
 2-17-81 13:12:01


*
*********************  EXTRACT.DEFINITION.FILE  ************************
*
*     THIS EDF IS BEING CREATED FOR USE WITH ELECTRICAL
*      ORDER PROCESSING OF N1600 RELEASES
*
************************************************************************
*
1  SECTION #1 - DATA EXTRACT ITEMS
*
************************************************************************
*
*
*  EXTRACT LEVEL CONTROL ITEM FOR UNIV
*
>               /UNIV. ITEMS/
*
*  NORMAL EXTRACT PROCEDURES FOR UNIV
*
*
>  SHEET     /UNIV.NTXT.IF-TNOD-PROPNAME/       SHEET
*
*
*  EXTRACT LEVEL CONTROL ITEM  FOR COMP
*
>            /COMP.ITEMS     /
*
*  NORMAL EXTRACT PROCEDURES FOR COMP
*
>  REFDES     /COMP.NTXT.IF-TNOD-PROPNAME/       ELEMNAME
>            /COMP.REJECT.IF-SUBFIG-NAME/        '-'
>            /COMP.REJECT.IF-PROPNAME   /        NETREF
*
*
*  EXTRACT LEVEL CONTROL ITEM FOR TERM
*
>            /TERM.ITEMS     /
*
*  NORMAL EXTRACT PROCEDURES FOR TERM
*
>  PINNO     /TERM.NTXT.IF-TNOD-PROPNAME/        TERMNAME
*
*
*  EXTRACT LEVEL CONTROL ITEM FOR NETW
*
>            /NETW.ITEMS     /
*
*  NORMAL EXTRACT PROCEDURES FOR NETW
*
>  CABLE      /NETW.PROPVALUE          /           CABLE
>  SIGNAL    /NETW.PROPVALUE             /       SIG
>  DEFAULT   /NETW.DEFAULT-NAME          /       'WR'
>  GAUGE     /NETW.PROPVALUE             /       GAUGE
>  COLOR     /NETW.PROPVALUE             /       COLOR
>           /NETW.SPLIT.IF-CNOD-PROPNAME/      SPLIT
*
************************************************************************
*
2  SECTION #2 - PAGE FORMAT SPECIFICATIONS:
*
************************************************************************
*
*  NUMBER OF PRINT LINES PER PAGE:
> 3200
*
*  NUMBER OF COLUMN POSITIONS PER PRINT LINE:
> 72
*
*  NUMBER OF PAGE HEADER LINES (IN SECTION #7):
> 0
*
*  NUMBER OF ITEM FIELD DESCRIPTOR RECORDS (IN SECTION #8):
>  9
```

Figure 3

(continued)



```
*
*
************************************************************************
 *
6  SECTION #6 - BILL OF MATERIALS TALLY SPECIFACATION (OPTIONAL):
*
************************************************************************
*
* BOM INDICATOR, START COLUMN, END COLUMN,JUSTIFACATION:
* (NOTE: BOM INDICATOR= 0 IF NOT APPLICABLE, = 1 IF APPLICABLE.
* JUSTIFACATION CODE: -1=LEFT, 0=CENTER, +1=RIGHT)
> 1,39,40,0
*
*
************************************************************************
*
8  SECTION #8 - ITEM FIELD DESCRIPTORS:
*
************************************************************************
*
>  SHEET    ,  0,  7,  8, -1, 00,  1,  0
>  REFDES   ,  0, 14, 28, -1, 00,  0,  0
>  PINNO    ,  0, 31, 38, -1, 00,  0, 00
>  SIGNAL   ,  0, 43, 58,  0, 00,  0, 00
>  DEFAULT  ,  0,  1,  6, -1, 00,  2,  1
>  GAUGE    ,  0, 61, 65,  0, 00,  0, 00
>  COLOR    ,  0, 68, 72,  0, 00,  0, 00
>  CABLE    ,  0, 41, 42,  0, 00, 00, 00
*
*******************************  END  **********************************
0>
```

Figure 3 (continued)

RESULTS

All schematics at Harris Corporation, Publication Press Division, are done on CAD. Using the Computervision system and the schematic library parts, a two-to-one increase in productivity was realized on schematics where wirelists are not to be extracted. Where all information must be included for wirelists, depending upon the availability of that information, the schematic phase tends to take as long as it might off the system, however, once this information is present, the process of deriving a wirelist is a factor of 50–100 faster with fewer errors. The elimination of wiring on the backpanel drawing and plotting these layouts on CAD using a program which generates front, back, and detail drawing from a customer proposal, has cut that process by a factor of 50 also. Panel library parts are available which encompass the proposal representation, the cutout detail and dimensions, and the backpanel representation all inserted at once using tag points. Multiple languages are layered onto the library parts for engraving drawings.

Gaining acceptance for the idea, and then the process was a major consideration. Fitting the procedure into the business cycle was also a concern. Different individuals had to be used for entering schematics and for order processing. The program was set up to require minimal knowledge of Computervision graphics commands. Data sheets are filled out using the customer contract which supply all the information for schematics required. A text tile is edited and an execute command performed to generate the wirelists.

Instead of feeling that their jobs are threatened, people must be made to realize that they will do better work, and more of it, as a result of eliminating the tedium of checking and sorting. More time and manpower will be available for imaginative engineering instead of clerical work. Results on a project of this type must be demonstrated early and be specific to gain confidence in the usefulness and ease of implementation. A solid program is imperative to acceptance. This was accomplished in our project.

APPENDIX A

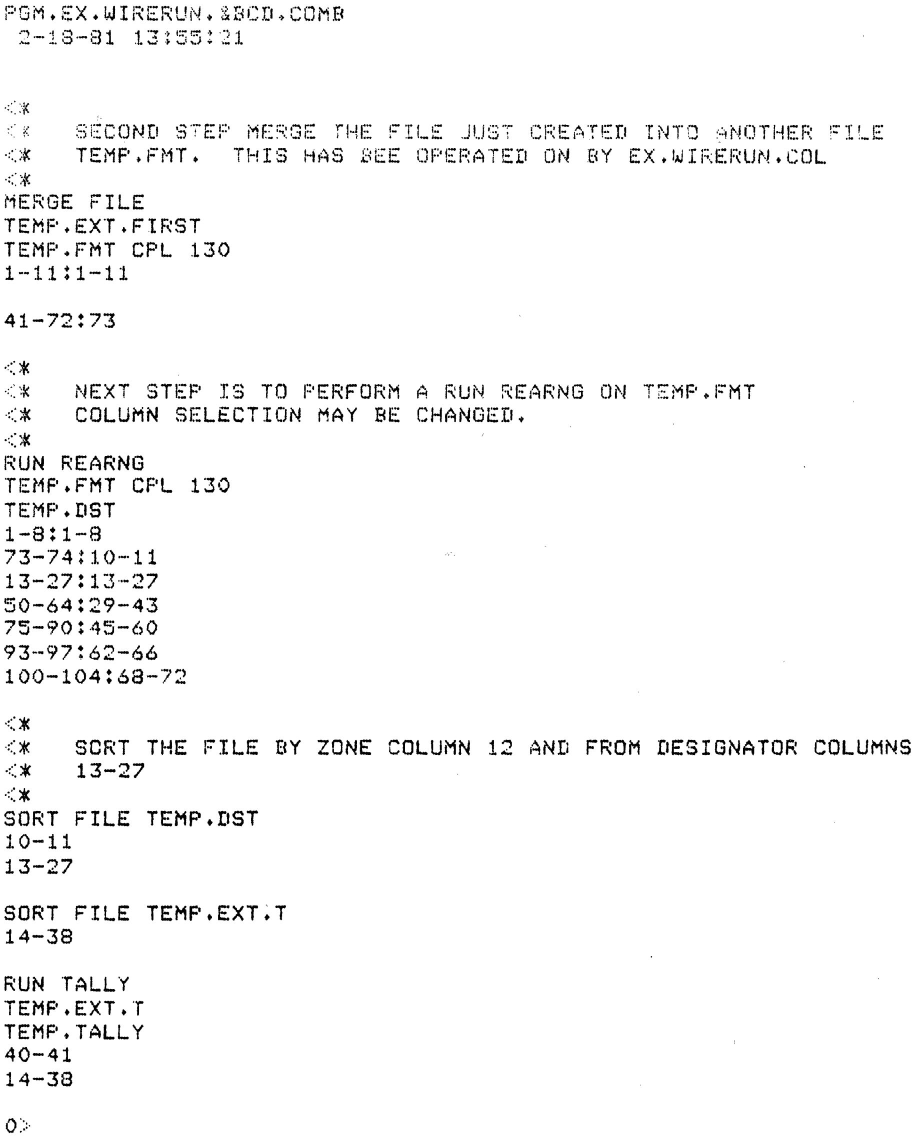
1. Copy of program
2. Copy of menu
3. Sample schematic

```
PGM.PAREX.&BCD.RUN
 2-18-81 12:00:46


&&
2 RPTFL
2 DKEY
&&
EXIT
DG
DELFIL TEMP/EXT
DELFIL TEMP.&&BCD/TALLY/LIST
DELFIL TEMP.&&BCD/DST/LIST
DELFIL TEMP.&&BCD/FMT/LIST
DELFIL TEMP.&&BCD/FMT2
COPYBCD TEXT.EXTRACT.INTERIM,TEMP.EXT.FIRST
CADDSCLR
OK<CR>
<CR>
DDD
FMT NET DATA TEMP.EXT.FIRST DATAKEY &DKEY FILE TEMP.FMT
EXIT
DG
COPYBCD TEMP.EXT.FIRST,TEMP.EXT.T
COPYBCD TEMP.FMT,TEMP.FMT2
EXECUTE EX.WIRERUN.RPT
DELFIL &RPTFL/&&BCD
COPYBCD TEMP.TALLY,&RPTFL.TALLY
COPYBCD TEMP.DST,&RPTFL.RPT
PRNT &RPTFL.RPT
PRNT &RPTFL.TALLY
O>
```

```
PGM.EX.WIRERUN.&BCD.RPT
 1-28-81 16:10:34

<*
<*   THIS REPORT CALLS ALL THE NECESSARY EXECUTE FILES TO
<*   PREPARE THE NECESSARY WIRERUN REPORTS.  SEE SPECIFICATION
<*   FOR DETAILS ABOUT THE REPORT FORMAT.
<*
<*
<*
<*   WIRERUN.COL IS A TECO PROGRAM THAT REFORMATS THE OUTPUT
<*   OF A FMT NET COMMAND .  THE RESULTS OF THIS COMMAND ARE ASSUMED
<*   TO BE IN A FILE CALLED TEMP.FMT
<*
EXECUTE EX.WIRERUN.COL
<*
<*   NOW EXECUTE TEH ROGRAM WHICH WILL TAKE THE RESULTS OF AN EXTR DATA
<*   COMMAND (ASUMED TO BE IN A FILE CALLED TEMP.EXT.1)
<*   AND BLANK OUT ALL SUBSEQUENT OCCURENCES OF THE WIRE NAME
<*
<*
<*   EX.WIRERUN.BLNK AND COMB OPERATE FROM CADDS LEVEL
<*
<*   ENTER CADDS
CADDSCLR
OK<CR>
<*   EXTRA CARRIAGE RETURN IN CASE TMP FILES EXIST
<CR>
DDD
EXECUTE EX.WIRERUN.BLNK
<*
<*
<*   THE NEXT STEP IS TO COMBINE THE INFORMAT IN TEMP.EXT.1 AND TEMP.FMT
<*   INTO A SINGLE FILE.    THIS FILE WILL BE SORTED IN ORDER BY
<*   ZONE, AND WITHIN ZONE, BY FROM INDICATOR.
<*  THIS FILE IS CALLED TEMP.DST
EXECUTE EX.WIRERUN.COMB
<*
<*   NEXT FROM THIS ONE FILE, A SEPARATE FILE FOR EACH ZONE WILL
<*   BE CREATED THESE ARE TEMP.EXT.A,B,C,D,E,F,BLNK FOR THE RESPECTIVE
<*   ZONES.  THIS IS DONE BY EX.WIRERUN.TECO
<*
<*   MUST RETURN TO SYSTEMS LEVEL TO FINISH
EXIT DG
<*EXECUTE EX.WIRERUN.TECO
O>
```

```
PGM.EX.WIRERUN.&BCD.COL
 1-22-81 16:44:07


CVTECO TEMP.FMT
<*
<*   READ IN WHOLE FILE     JUMP TO BEGINNING OF BUFFER
EA J<ESC><ESC>
<*   ENTIRE PROGRAM WILL BE IN AN ITERATIVE LOOP TO BE TERMINATED
<*   WHEN THE POINTER POSITION EQUALS THE END OF FILE
<*
<<:<CTRL-T>;
<*
<*   SAVE THE CURRENT POINTER POSITION IN REGISTER 1 FOR
<*   LATER USE
<*
.P1<ESC>
<*   BECAUSE WR NAME ENDS IN COLUMN 11 POSITION TO NEXT COLUMN
<*
12C<ESC>
<*
<*   SEARCH FOR THE NEXT OCCURENCE OF THREE BLANKS  THIS IS THE
<*   SPACE BETWEEN THE FROM AND TO ENTRIES
<*   NOTE:  IF THIS SPACING IS EVER CHANGED, THE SEARCH STRING MUST
<*   ALSO BE CHANGED
<*
S   <ESC>
<*
<*   WANT TO POSITION THIS FIELD IN COLUMN 50.  CALCULATE THE
<*   NUMBER OF BLANKS THAT MUST BE INSERTED AS :
<*   50 - (THE CURRENT POINTER POSITION - THE POINTER POSITION AT
<*   THE BEGINNING OF THE LINE,IE., STORED IN REG 1)-1
<*   USE THIS AS AN ITERATION FACTOR FOR AN INSERT COMMAND
50-.+Q1-1<<I <ESC>>
<*
<*
<*   DOWN ONE LINE
<*
1L<ESC>
<*
<*   CLOSE LOOP
<*
> EX<ESC><ESC>
0>
```

```
PGM.EX.WIRERUN.&BCD.BLNK
 1-22-81 17:42:03


<*
<*   THIS PROGRAM WILL MERGE A FILE ON ITSELF.  IF COLUMNS 39:40 ARE
<*  BLANK, COLUMNS 1-11 WILL BE BLANKED.  USED TO FORMAT THE
<*  OUTPUT OF A DATE EXTRACT RUN, USING A BOM TALLY.  IN THIS CASE
<*  ONLY THE FIRST OCCURRENCE OF A WIRE NAME WILL BE PRINTED
<*  ALL SUBSEQUENT LINES WILL BE BLANKED (IN COLUMNS 1-11)
<*
MERGE FILE
TEMP.EXT.FIRST
TEMP.EXT.FIRST BLNK
S '  ':39-40

S '           ':1-11
<CR>
0>
```

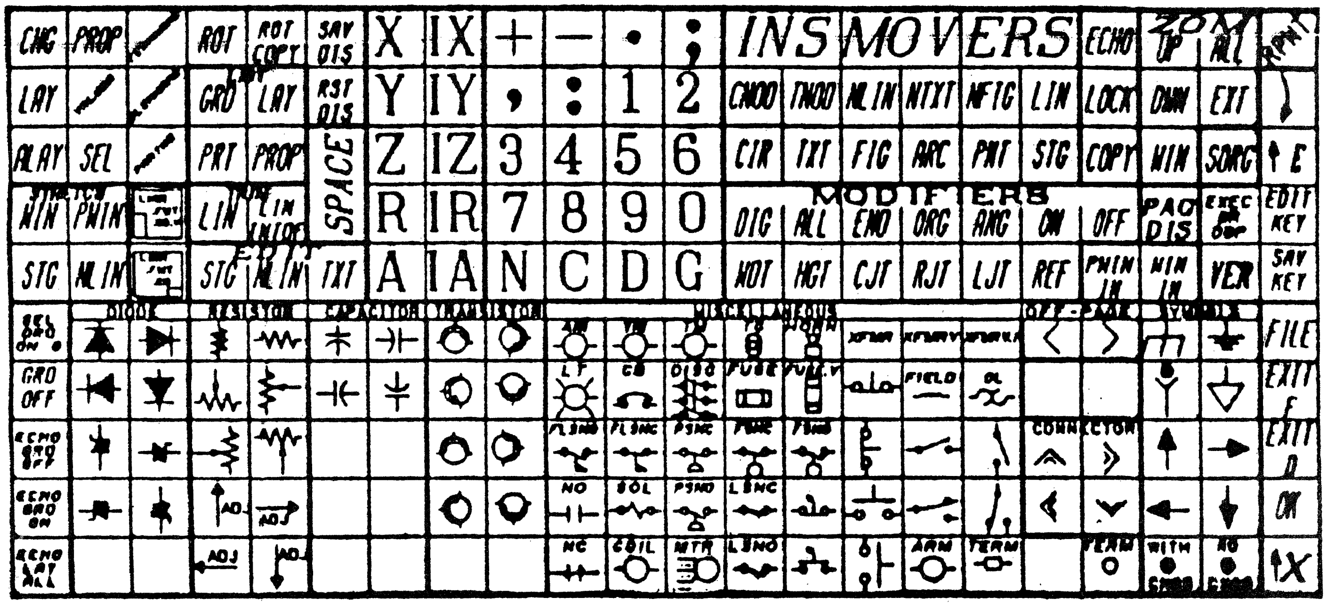


```
PGM.EX.WIRERUN.&BCD.COMB
 2-18-81 13:55:21


<*
<*   SECOND STEP MERGE THE FILE JUST CREATED INTO ANOTHER FILE
<*   TEMP.FMT.  THIS HAS SEE OPERATED ON BY EX.WIRERUN.COL
<*
MERGE FILE
TEMP.EXT.FIRST
TEMP.FMT CPL 130
1-11:1-11

41-72:73

<*
<*   NEXT STEP IS TO PERFORM A RUN REARNG ON TEMP.FMT
<*   COLUMN SELECTION MAY BE CHANGED.
<*
RUN REARNG
TEMP.FMT CPL 130
TEMP.DST
1-8:1-8
73-74:10-11
13-27:13-27
50-64:29-43
75-90:45-60
93-97:62-66
100-104:68-72

<*
<*   SORT THE FILE BY ZONE COLUMN 12 AND FROM DESIGNATOR COLUMNS
<*   13-27
<*
SORT FILE TEMP.DST
10-11
13-27

SORT FILE TEMP.EXT.T
14-38

RUN TALLY
TEMP.EXT.T
TEMP.TALLY
40-41
14-38

O>
```

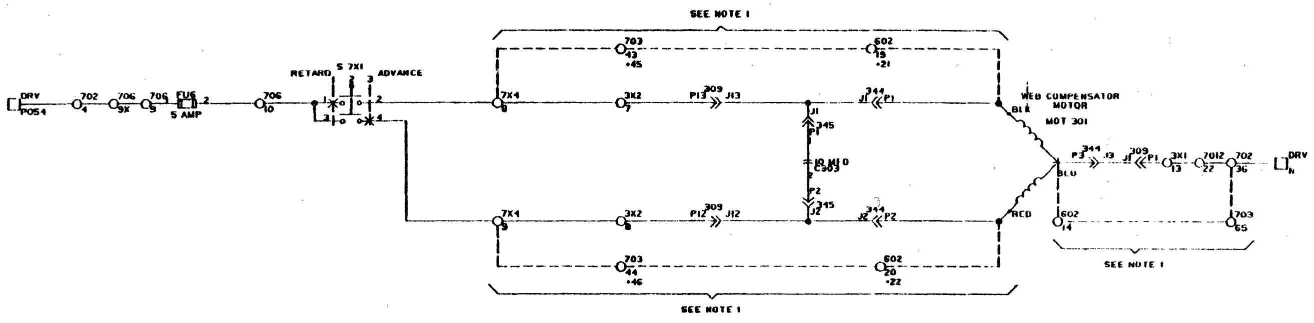
```
7>CVTECO TEMP.FMT
TECO Version 4.02
#EA J$$
#
<:&T
.P1$
12C$
S   $
50-.+Q1-1<I $>
1L$
> EX$$
7>
```

SCHEMATIC KEYFILE 2101.2

LIBR.MENU.2101.2



ECN NO	ZONE	DATE	BY	REVISIONS



1.FOLDER TERMINALS ARE USED ON UNIT NEXT TO FOLDER.

• 703-45 ,703-46 & 602-21 ,602-22 } FOR L.H. UNITS

LEGEND HARDWARE TERMINATION	
0010	DRIVE CONTROL
100	REEL STAND
200	INFEED
300	PRINTING UNIT (3X)
400	DRYER
500	CHILL ROLL
600	FOLDER
700	PRESS CONSOLE
800	FOLDER CONSOLE
900	MISC.(TELECOLOR)

SHEET 7
DATA EXTRACT : COMPLETE
ECH SORG OFF
ECH CNOD OFF
ECH TNOD OFF
DWG SIZE D PLOT SCALE 1.0 =1.0
THICK PEN LAYERS : 20
THIN PEN LAYERS :0-8,15

NO	DRAWING NO	FILE	

REFERENCE DRAWINGS

SYMBOL H-1600 FILE D

NAME WEB COMPENSATOR
SCHEMATIC -L.H 4R.H.
CV-P16.ES.448548.4

DATE 1-27
SCALE

DWN BY RLM CHK BY AT APPR BY AT REV

448548

SHEET OF

APPLICATION OF "DATA EXTRACT"
TO
AEROSPACE VEHICLE WIRING

Robert D. Cook
Vought Corporation
P.O. Box 225907
Unit 2-51700
Dallas, TX 75265
214-330-4601 X372

I. INTRODUCTION

Vought has been active in the field of Computer Aided Design for over 15 years. The company is a prime contractor for military aircraft and missiles. This paper describes the application of Computer Aided Design, particularly ComputerVision (CV) Data Extract to the process of design, documentation and fabrication of the electrical systems and wiring installed in the aircraft, missile and other products of Vought Corporation.

The wiring in the product is broken up into wiring harnesses for fabrication and installation of wiring into the end product (see fig. 1). The systems are described by wiring (interconnection) diagrams which both document and aid in the checkout of the system (see fig. 2). Customer documentation requires us to produce both wiring diagrams and wiring harnesses.

In May 1975, Vought acquired its first ComputerVision (CV) equipment. This consisted of two Interact IV tables, an Interact II, one CRT and a Nova 1200 mini-computer. We immediately began using this system to generate wiring diagrams using CADDS I. The harness drawing data was still hand-coded from the wiring diagrams and batch loaded into a mainframe.

The system configuration has changed considerably since then. There are now two separate systems consisting of five or six CRT display consoles, a CGP-100, and a tape drive. Each system also has one INTERACT IV table and a TI Silent 700 terminal. A third system has raster graphic terminals on a CGP-200. A UT200 interface has been established between the CYBER 175 mainframe and each of the systems. The systems share a Versatec dot plotter for an output device.

II. GOALS

Since the wire harness data is hand-coded from the wiring diagrams, several opportunities for error exist. Therefore, a major goal is to automate the coding of this data. Eventually, the initial coding is to be done by computer extraction of the data from the wiring diagrams, thus eliminating manual coding and keypunch and reducing the span time between program go-ahead and manufacturing the harnesses. It is also very desirable to be able to edit the extracted data table and have the changes automatically incorporated into the graphics (back-annotation).

III. PROTOTYPE PHASE

A. Study

Because of the above goal, an investigation was made into the use of Data Extract, a CV software package. Essentially Data Extract pulls off desired non-graphical data (i.e., text) using pre-defined properties and associations. The information required to support this wire harness data is shown in Table I. Approximately a man-month was spent on this study, and a working prototype was developed for test purposes on two different aircraft programs.

The use of Data Extract meant that all diagrams generated on these two programs would have to be done using nodal entities.Therefore, a new part library was required that would contain all the associations needed to support Data Extract. A new library was created instead of modifying the existing library so that the existing drawings would not be impacted by the Data Extract capability.

The major program on which Data Extract was used was the initial design of a military aircraft. Due to schedule contraints, the wiring diagrams had to be digitized after the wiring harnesses were completed. Therefore, we were not able to use the Data Extract process to replace the coding but we were able to use it to compare the wiring diagrams and the wiring harnesses.

B. Results

It was immediately apparent that it took twice as long to digitize the drawings using nodal entities as opposed to digitizing without using nodal entities.This result had been anticipated however, and determined to be acceptable if the overall span time was shortened by reducing subsequent coding or checking.

Once a few of the diagrams had been digitized, the Extract procedure was executed, and the resulting data transmitted via the UT200 to the CYBER 175 computer where it was processed into the same form as the existing harness data. A comparison was then made between this data and the harness data and a discrepancy report generated. The comparison resulted in the discovery of a few problems which were corrected. This meant that the harnesses were now freer from error than those generated by hand with manual checking.

C. Problems

Several problems were encountered in the course of this study phase. One problem was the limited documentation available concerning Data Extract. A large portion of the trouble we had in getting Data Extract to work is directly attributable to that fact.

A second problem was discovered in putting all the information into the diagram. Although we originally intended to use nodal text insertion to get all the data into the diagram, we switched to using default properties for that information which did not have to be visible on the diagram.

A third problem was discovered in trying to extract shielding information from the diagram. We have decided to sidestep this problem by coding the shield entries by hand and adding them to the data base. We are continuing to work on a solution to the problem. This problem is partially due to the format of our shield data. That is, the relationship between the graphic presentation and the tabular presentation is not straightforward.

A fourth problem was that post-processing of the extracted file inhibited the back-annotation process. A satisfactory solution to this problem has not yet been found.

Once all the problems were identified in the study phase, solution or work arounds were developed to correct them. This resulted in changes to our approach, including the development of the following new digitizing techniques.

1. A menu to support Data Extract.
2. A parametric execute (PAREX) file to insert wire numbers.
3. A PAREX file to run the extract process including the required data manipulatron.
4. A PAREX file to transfer the data to the mainframe.
5. Text annotation techniques to insert pin text.

IV. PRODUCTION PHASE

The first use of this extract procedure was in the design of a missile by Vought engineering. Basically, one engineer designed the electrical wiring required for the systems on the product, laid out the diagrams, digitized the diagrams using Data Extract-compatible figures and properties, extracted the data from the diagrams and transferred the data to the CYBER 175 mainframe where he was able to process it as if he had coded it by hand (see fig. 3). Table II describes the savings data we accumulated during the course of the design.

V. CONCLUSIONS AND RECOMMENDATIONS

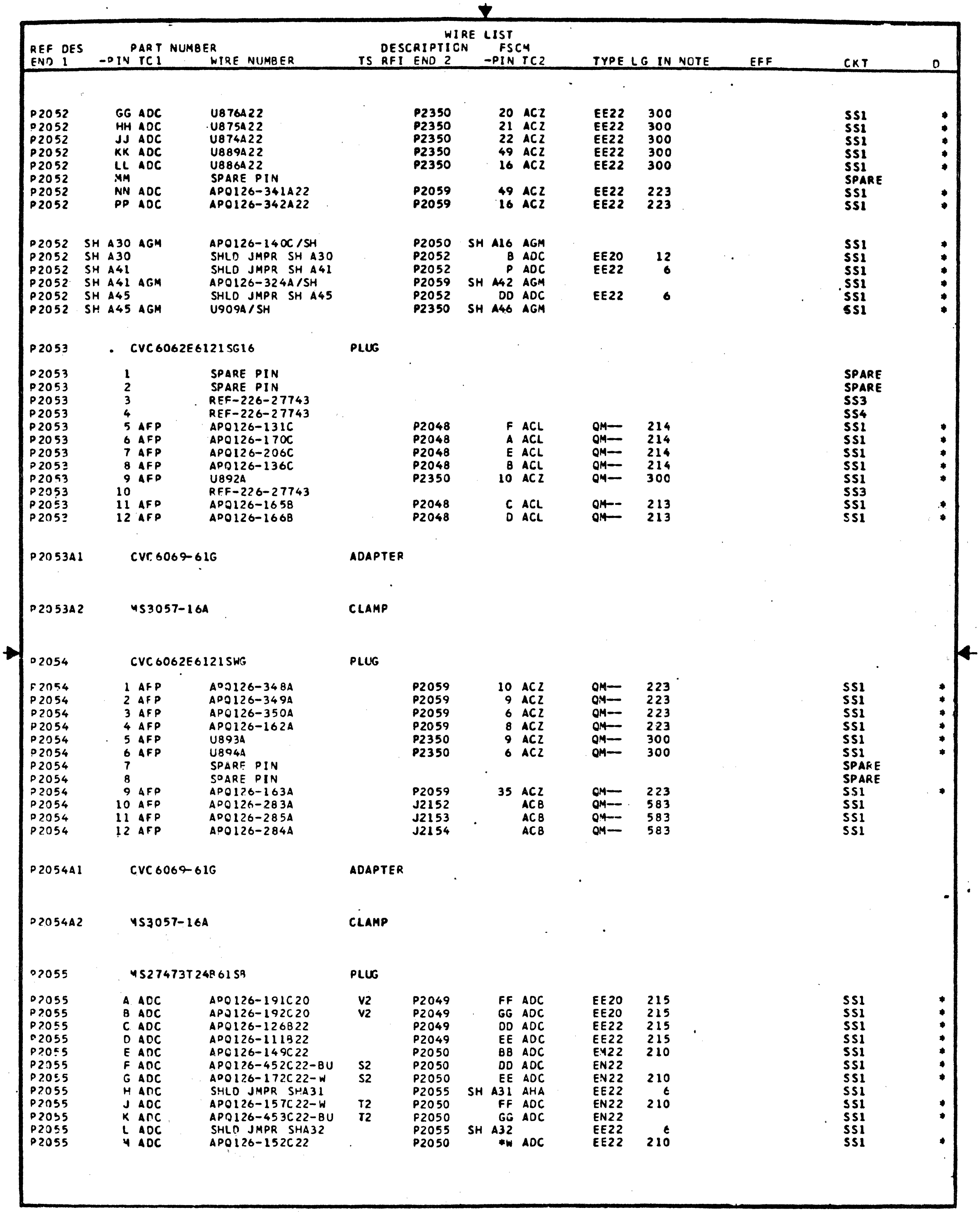
The savings encountered in both the prototype and production phases of the development process prove that we have achieved a viable approach to interfacing the wiring diagram with the wire harness. A large portion of the labor involved in generating the wire harness can be eliminated by the use of Data Extract, but more importantly the duplication of effort required in digitizing the wiring diagram and adding the same information for the wire harnesses and the possibility of introducing errors is eliminated.

One thing that would make this process more efficient is the ability to have the wiring diagrams and the wire harness data system on the same computer. This will be a reality if CV develops a computer the size of a VAX 11/780 with a COBOL compiler. We feel that this is a definite goal that will allow CV equipment to handle the entire electrical documentation problem.

Using Data Extract to retrieve the interconnection data does have some disadvantages. Each diagram requires more space for on-line or archival storage. The process also requires that all the diagrams be digitized before the harnesses can be worked on since a harness is usually split up among several diagrams.

One area that still needs to be studied is back-annotation of changes into the diagrams. We have found that manipulation of the data necessary to support the wire harness system destroys the capability to back-annotate the changes.

The major benefit of this process is that it allows the design task to be accomplished using fewer people by placing more of the burden on the machine. A second major benefit is the ability to incorporate changes to an existing design in a shorter time span. This benefit could be more important than the first, since sometimes reduced schedules are more important than reduced costs.



```
                                                       WIRE LIST
REF DES        PART NUMBER                        DESCRIPTION     FSCM
END 1      -PIN TC1         WIRE NUMBER         TS RFI END 2      -PIN TC2     TYPE LG IN NOTE      EFF          CKT        D


P2052        GG ADC         U876A22                    P2350        20 ACZ     EE22    300                       SS1        *
P2052        HH ADC         U875A22                    P2350        21 ACZ     EE22    300                       SS1        *
P2052        JJ ADC         U874A22                    P2350        22 ACZ     EE22    300                       SS1        *
P2052        KK ADC         U889A22                    P2350        49 ACZ     EE22    300                       SS1        *
P2052        LL ADC         U886A22                    P2350        16 ACZ     EE22    300                       SS1        *
P2052        MM             SPARE PIN                                                                            SPARE
P2052        NN ADC         APQ126-341A22              P2059        49 ACZ     EE22    223                       SS1        *
P2052        PP ADC         APQ126-342A22              P2059        16 ACZ     EE22    223                       SS1        *


P2052    SH A30 AGM         APQ126-140C/SH             P2050    SH A16 AGM                                       SS1        *
P2052    SH A30             SHLD JMPR SH A30           P2052         B ADC     EE20     12                       SS1        *
P2052    SH A41             SHLD JMPR SH A41           P2052         P ADC     EE22      6                       SS1        *
P2052    SH A41 AGM         APQ126-324A/SH             P2059    SH A42 AGM                                       SS1        *
P2052    SH A45             SHLD JMPR SH A45           P2052        DD ADC     EE22      6                       SS1        *
P2052    SH A45 AGM         U909A/SH                   P2350    SH A46 AGM                                       SS1        *

P2053        . CVC6062E6121SG16              PLUG

P2053         1             SPARE PIN                                                                            SPARE
P2053         2             SPARE PIN                                                                            SPARE
P2053         3             REF-226-27743                                                                        SS3
P2053         4             REF-226-27743                                                                        SS4
P2053         5 AFP         APQ126-131C                P2048         F ACL     QM—     214                       SS1        *
P2053         6 AFP         APQ126-170C                P2048         A ACL     QM—     214                       SS1        *
P2053         7 AFP         APQ126-206C                P2048         E ACL     QM—     214                       SS1        *
P2053         8 AFP         APQ126-136C                P2048         B ACL     QM—     214                       SS1        *
P2053         9 AFP         U892A                      P2350        10 ACZ     QM—     300                       SS1        *
P2053        10             REF-226-27743                                                                        SS3
P2053        11 AFP         APQ126-165B                P2048         C ACL     QM--    213                       SS1        *
P2053        12 AFP         APQ126-166B                P2048         D ACL     QM—     213                       SS1        *

P2053A1        CVC6069-61G                   ADAPTER


P2053A2        MS3057-16A                    CLAMP


P2054          CVC6062E6121SWG               PLUG

P2054         1 AFP         APQ126-348A                P2059        10 ACZ     QM—     223                       SS1        *
P2054         2 AFP         APQ126-349A                P2059         9 ACZ     QM—     223                       SS1        *
P2054         3 AFP         APQ126-350A                P2059         6 ACZ     QM—     223                       SS1        *
P2054         4 AFP         APQ126-162A                P2059         8 ACZ     QM—     223                       SS1        *
P2054         5 AFP         U893A                      P2350         9 ACZ     QM—     300                       SS1        *
P2054         6 AFP         U894A                      P2350         6 ACZ     QM—     300                       SS1        *
P2054         7             SPARE PIN                                                                            SPARE
P2054         8             SPARE PIN                                                                            SPARE
P2054         9 AFP         APQ126-163A                P2059        35 ACZ     QM—     223                       SS1        *
P2054        10 AFP         APQ126-283A                J2152           ACB     QM—     583                       SS1
P2054        11 AFP         APQ126-285A                J2153           ACB     QM—     583                       SS1
P2054        12 AFP         APQ126-284A                J2154           ACB     QM—     583                       SS1

P2054A1        CVC6069-61G                   ADAPTER


P2054A2        MS3057-16A                    CLAMP


P2055          MS27473T24B61SB               PLUG

P2055         A ADC         APQ126-191C20       V2     P2049        FF ADC     EE20    215                       SS1        *
P2055         B ADC         APQ126-192C20       V2     P2049        GG ADC     EE20    215                       SS1        *
P2055         C ADC         APQ126-126B22              P2049        DD ADC     EE22    215                       SS1        *
P2055         D ADC         APQ126-111B22              P2049        EE ADC     EE22    215                       SS1        *
P2055         E ADC         APQ126-149C22              P2050        BB ADC     EM22    210                       SS1        *
P2055         F ADC         APQ126-452C22-BU    S2     P2050        DD ADC     EN22                              SS1        *
P2055         G ADC         APQ126-172C22-W     S2     P2050        EE ADC     EN22    210                       SS1        *
P2055         H ADC         SHLD JMPR SHA31            P2055    SH A31 AHA     EE22      6                       SS1
P2055         J ADC         APQ126-157C22-W     T2     P2050        FF ADC     EN22    210                       SS1        *
P2055         K ADC         APQ126-453C22-BU    T2     P2050        GG ADC     EN22                              SS1        *
P2055         L ADC         SHLD JMPR SHA32            P2055    SH A32         EE22      6                       SS1
P2055         M ADC         APQ126-152C22              P2050        *W ADC     EE22    210                       SS1        *
```

Figure 1: Sample Wiring Harness

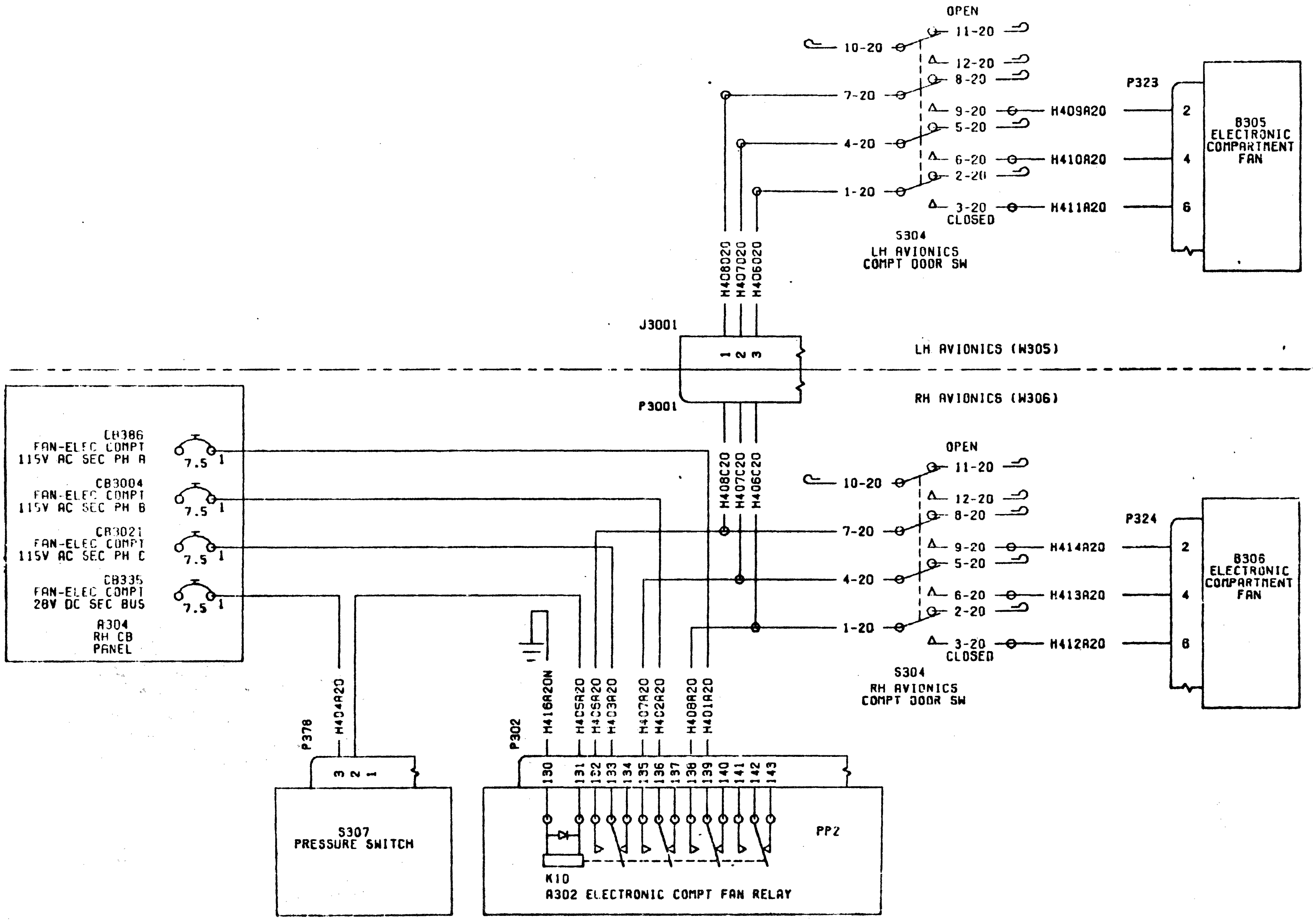
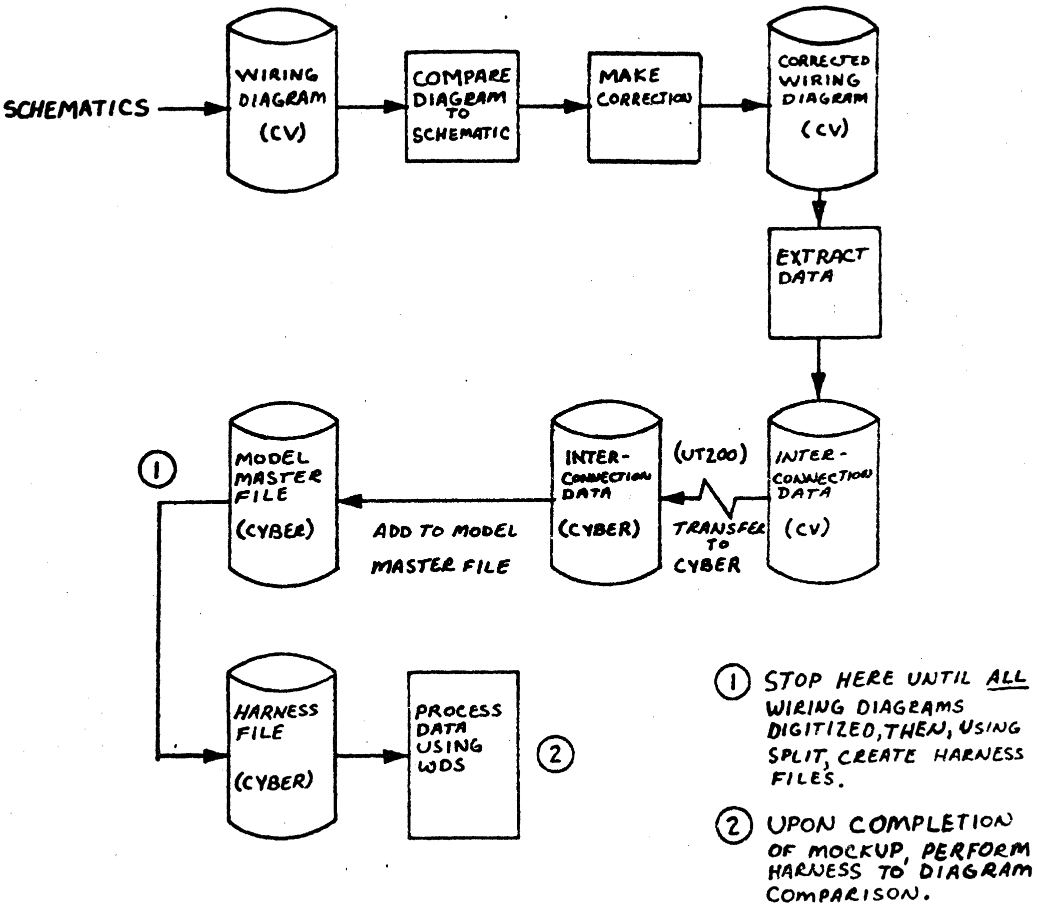


Figure 2: Sample Wiring Diagram



1 STOP HERE UNTIL ALL WIRING DIAGRAMS DIGITIZED, THEN, USING SPLIT, CREATE HARNESS FILES.

2 UPON COMPLETION OF MOCKUP, PERFORM HARNESS TO DIAGRAM COMPARISON.

Figure 3: Graphic System Interface

Table I:
Wire Harness Data Elements

ELEMENT NAME	DEFINITION
ASSEMBLY NUMBER	WIRE HARNESS ASSEMBLY INTO WHICH A WIRE SHALL BE INSTALLED.
REFERENCE DESIGNATOR (END 1)	UNIQUE IDENTIFIER WITHIN A PRODUCT WHICH IS ASSOCIATED WITH A PARTICULAR PIECE OF HARDWARE.
PIN NUMBER (END 1)	UNIQUE END POINT OR CONTACT FOR A GIVEN REFERENCE DESIGNATOR
TERMINATION CODE (END 1)	UNIQUE CODE DESCRIBING THE HARDWARE AND VOUGHT SPECIFICATION REQUIRED TO TERMINATE A WIRE AT THIS END.
WIRE NUMBER	A NUMBER THAT UNIQUELY IDENTIFIES A PIECE OF WIRE IN A GIVEN PRODUCT.
REFERENCE DESIGNATOR (END 2)	UNIQUE IDENTIFIER WITHIN A PRODUCT WHICH IS ASSOCIATED WITH A PARTICULAR PIECE OF HARDWARE.
PIN NUMBER (END 2)	UNIQUE END POINT OR CONTACT FOR A GIVEN REFERENCE DESIGNATOR.
TERMINATION CODE (END 2)	UNIQUE CODE DESCRIBING THE HARDWARE AND VOUGHT SPECIFICATION REQUIRED TO TERMINATE WIRE AT THIS END.
WIRE TYPE/GAUGE	A UNIQUE CODE THAT IDENTIFIES A PIECE OF WIRE WITH A PARTICULAR WIRE PART NUMBER
WIRE LENGTH	ACTUAL OR ESTIMATED CUT LENGTH OF A PIECE OF WIRE
NOTES	TWO 2-DIGIT FIELDS THAT REFERENCE A NOTE CONTAINED IN THE PARTS LIST FOR A GIVEN ASSEMBLY.
WIRING DIAGRAM	A REFERENCE TO THE SYSTEM INTERCONNECTION WIRING DRAWING.
EFFECTIVITY	A MEANS OF IDENTIFYING THE USE OF THIS WIRE IN A GIVEN SINGLE PRODUCT OR BLOCK
TWIST SYMBOL	A UNIQUE-TO-THE-ASSEMBLY MEANS OF IDENTIFYING TWISTED OR MULTI-CONDUCTER CABLES.

Table II:
Program Cost Data

PROCESS	MANUAL	CV (NON-NODAL)	CV (NODAL)
Diagram Cost* Per Diagram	28	20	30
TOTAL	560	400	600
Harness Coding W/O Data EXTR	666	666	-
W/ Data EXTR	-	-	150
TOTAL	1226	1066	750
% Savings Over Manual	-	13.1%	38.8%
Over CV (non-nodal)	-	-	29.6%

*Includes diagram layout time, 10 hrs, 10 hrs and 12 hrs respectively.

Communications

AN OVERVIEW OF CVNET

Ken Johnson
Engineering Computer Services
STG Division
Westinghouse Electric Corporation
700 Braddock Avenue
East Pittsburgh, Pa. 15112
412-256-2191

ABSTRACT

This paper gives an overview of Computervision's data communications package which is called CVNET. Included is a definition of CVNET, its purpose, the hardware and software necessary to implement it, and some instructions on its use.

The more commonly used commands are described, but not in any great detail. The protection flags and CVNET log files are also touched upon. The paper is concluded with some curves of data transmission time versus sector size.

Presentation is done with a set of foils and an overhead projector. This document consists of a copy of each foil preceded by a page of supporting text.

INTRODUCTION

Foil #1

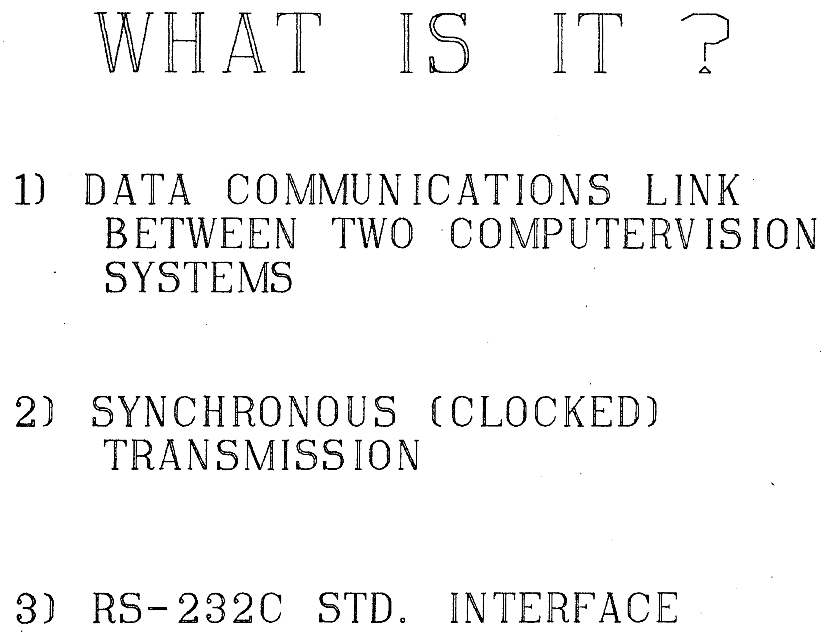
Our experience with CVNET comes from a CADDS3, CGP/100, rev. 10 system, and the material presented here is from that frame of reference. The advent of CADDS3 rev. 11, CADDS4, and Computervision's networking facility affects change in some areas of this presentation, and the user needs to keep that in mind.

CVNET

Foil #2

CVNET is a data communications link that can be used between any two Computervision systems having compatible hardware and software.

It uses synchronous transmission and the standard RS-232 interface.



WHAT IS IT ?

1) DATA COMMUNICATIONS LINK BETWEEN TWO COMPUTERVISION SYSTEMS

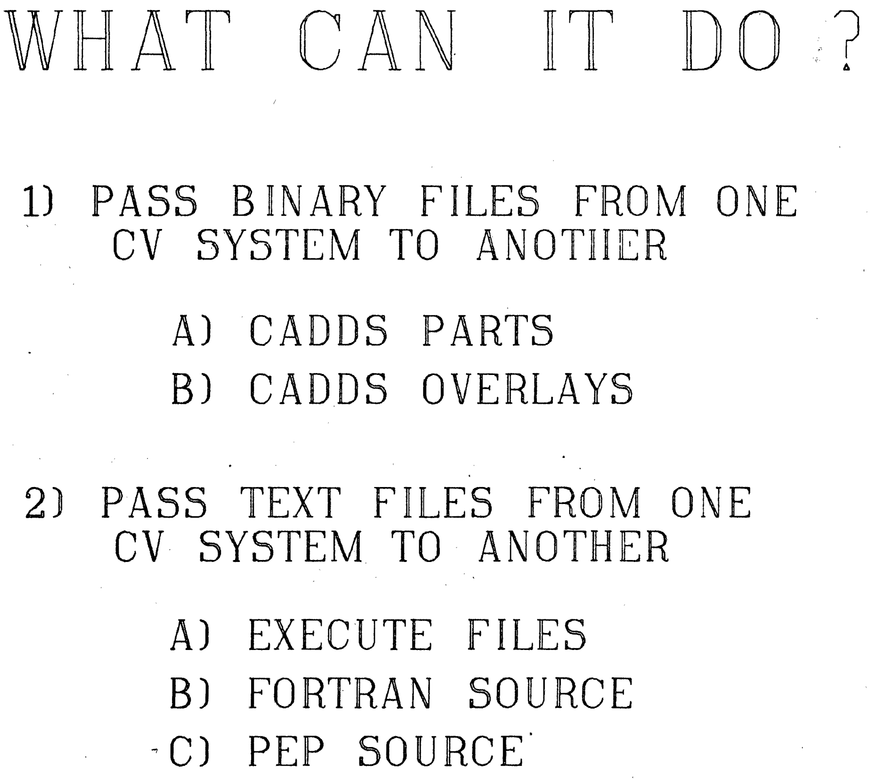
2) SYNCHRONOUS (CLOCKED) TRANSMISSION

3) RS-232C STD. INTERFACE

Foil #3

With this data communications link, you can pass binary files back and forth between systems. CADDS parts and CADDS object code overlays are examples of binary files.

Text files can also be passed over CVNET. Execute files, FORTRAN source programs, and PEP source programs are examples of text files. Any file created with the text editor is a text file.



WHAT CAN IT DO ?

1) PASS BINARY FILES FROM ONE CV SYSTEM TO ANOTIIER

A) CADDS PARTS

B) CADDS OVERLAYS

2) PASS TEXT FILES FROM ONE CV SYSTEM TO ANOTHER

A) EXECUTE FILES

B) FORTRAN SOURCE

C) PEP SOURCE

Foil #4

To implement CVNET, you need the CVNET software, you need a communications controller which can be either a SLU board or PCU board, and you need some kind of a communications link between the systems.

WHAT DO I NEED ?

1) CVNET SOFTWARE PACKAGE

2) COMMUNICATIONS CONTROLLER BOARD

A) SLU (SYNCHRONOUS LINE UNIT)

B) PCU (PROGRAMMABLE COMMUNICATIONS UNIT)

3) DATA COMMUNICATIONS LINK

A) CHOICE OF THREE TYPES

Foil #5

There are three different types of communications links available.

For long distances there are dial-up modems that use the telephone lines. Maximum line speed is 9,600 baud.

Short-haul modems can be used over distances of up to 7000 feet. The line speed is 19,200 baud.

For local links of up to 50 feet, two systems can be cabled together directly without the use of modems.

CVNET will operate with any device that has an RS-232C interface at speeds up to 56,000 bps. However, the maximum transfer rate of the PCU is 19,200 bps.

LINK OPTIONS

A) DIAL-UP LINES AND MODEMS

B) PRIVATE LINES WITH DEDICATED MODEMS (SHORT HAUL)

C) HARDWIRED LINK

Foil #6

These are some of the more common links used. The first three are dial-up modems, the next two are short-haul modems and the last one is the hardwired link.

MODEM	BAUD RATE	LINE TYPE	MAX. LENGTH
BELL 208B	4,800	DIAL-UP	
BELL 209A	9,600	PRIVATE LINE	
TELE-DYNAMICS 7208B	4,800	DIAL-UP	
SPECTRON DT-401	19,200	4-WIRE	7,000 FT.
PRENTICE SLD 19.2M	19,200	4-WIRE	7,000 FT.
HARDWIRED (NO MODEM)	40K	SPECIAL CV CABLE	50 FT.

Foil #7

Your configuration has to contain the auxiliary units for the PCU or SLU controller. These two are for the PCU. Both are needed for one PCU.

CONFIGURATION (PCU)

```
AUX UNIT 6: PCUC
  PARAMETERS:
     1. NAME1 (BCD) = PC
     2. NAME2 (BCD) = 00
     3. DEVNO IN (HEX) = 0010
     4. DEVNO OUT (HEX) = 0011
     5. UNCHWRD1 (HEX) = 0000
     6. UNCHWRD2 (HEX) = 0000
     7. UNCHWRD3 (HEX) = 0000

AUX UNIT 7: PCUG
  PARAMETERS:
     1. NAME1 (BCD) = PC
     2. NAME2 (BCD) = 01
     3. DEVNO IN (HEX) = 001C
     4. DEVNO OUT (HEX) = 001D
     5. UNCHWRD1 (HEX) = 3330
     6. UNCHWRD2 (HEX) = 0000
     7. UNCHWRD3 (HEX) = 0008
```

Foil #8

This is the way the configuration would look for a SLU controller.

CONFIGURATION (SLU)

```
AUX UNIT 8: TBIO
  PARAMETERS:
     1. NAME1 (BCD) = CV
     2. NAME2 (BCD) = NO
     3. DEVNO IN (HEX) = 002A
     4. DEVNO OUT (HEX) = 002B
     5. UNCHWRD1 (HEX) = 0000
     6. UNCHWRD2 (HEX) = 0000
     7. UNCHWRD3 (HEX) = 0000
```

Foil #9

Of course, before your system will understand any CVNET commands, you need to activate the CVNET command table.

Now we have everything we need to run CVNET.

COMMAND TABLE

```
0> SYSCMTB CMTB.CVNET <CR>
```

Foil #10

Before we can pass any data, we need to bring CVNET up on a task. This needs to be done at both the sending system and receiving system.

Most modems have an "automatic answering" option. This option allows you to bring CVNET up on a task and leave it active, then any remote system that dials you up will be connected automatically without operator intervention. For this luxury, however, you need to dedicate a task to CVNET.

HOW DO I USE IT ?

1) NEED TO BRING UP CVNET
   ON A TASK AT BOTH SYSTEMS

Foil #11

If your communication board is a SLU, simply type in CVNET and the device code which is usually CVNO. If you have only one communications device on your system, you need only type in CVNET and return.

USING SLU

0>CVNET [device code]<CR>

Foil #12

If you're using a PCU board, you must type in one of three options.

1) CVNETR and the device code for dial-up modems.
2) CVNET and the device code for hardwired and PCU boards in both systems.
3) CVNETS and the device code for hardwired and the other system is using a SLU board.

The device code is usually PC00. Again, you don't have to type in the device code if there is only one communications board in the system.

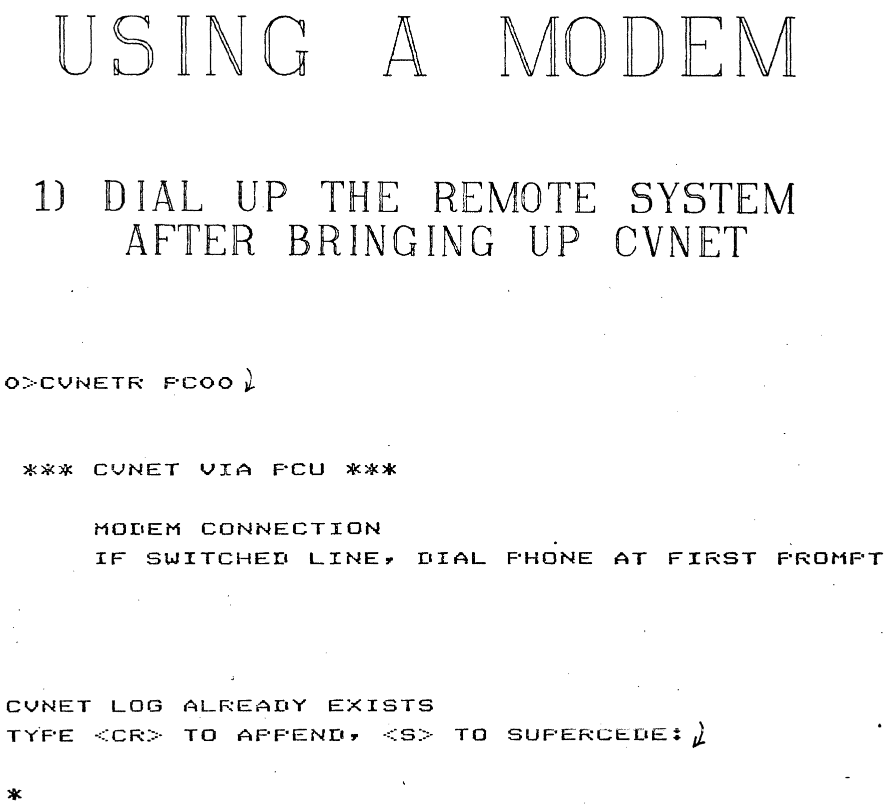
USING PCU

A) ONE OF THREE OPTIONS

1) MODEM LINK:	CVNETR
2) LOCAL PCU TO PCU:	CVNET
3) LOCAL PCU TO SLU:	CVNETS

Foil #13

When using dial-up modems, bring CVNET up first, then dial the other system. Notice that CVNETR is used.



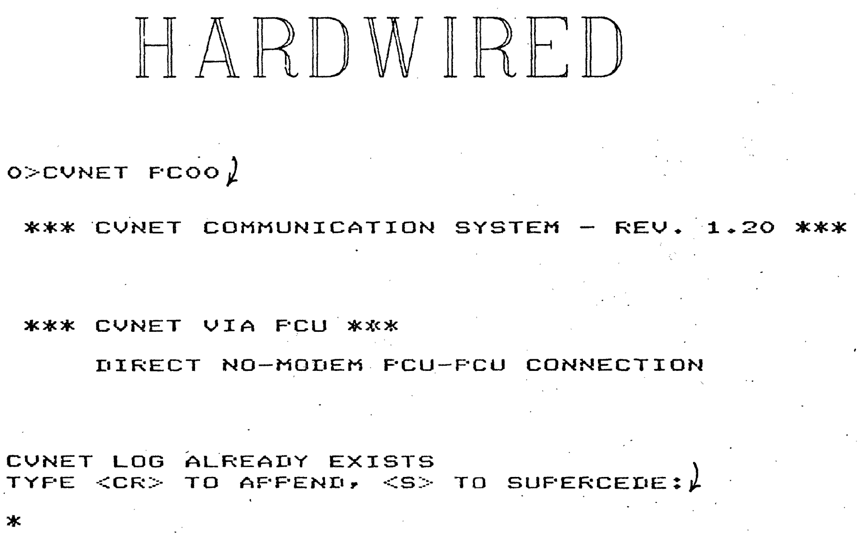
USING A MODEM

1) DIAL UP THE REMOTE SYSTEM AFTER BRINGING UP CVNET

```
O>CVNETR PCOO ↲

 *** CVNET VIA PCU ***

     MODEM CONNECTION
     IF SWITCHED LINE, DIAL PHONE AT FIRST PROMPT


CVNET LOG ALREADY EXISTS
TYPE <CR> TO APPEND, <S> TO SUPERCEDE: ↲

*
```

Foil #14

This is an example of bringing up CVNET for a hardwired PCU to PCU connection. I'll explain the CVNET LOG later.

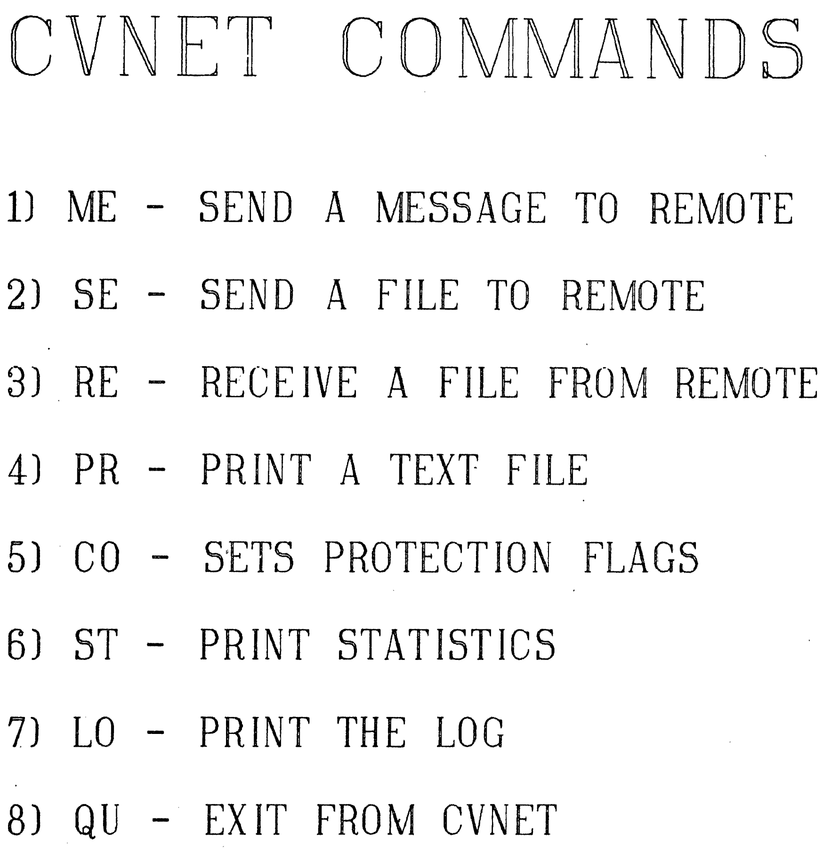


HARDWIRED

```
O>CVNET PCOO ↲

 *** CVNET COMMUNICATION SYSTEM - REV. 1.20 ***


 *** CVNET VIA PCU ***
     DIRECT NO-MODEM PCU-PCU CONNECTION


CVNET LOG ALREADY EXISTS
TYPE <CR> TO APPEND, <S> TO SUPERCEDE: ↲
*
```

Foil #15

There are a number of CVNET commands that can be used once CVNET has been brought up on a task. This is a list of the more common ones. All of them can be abbreviated to the first two characters as they are shown here. I'll go over each of these in a little more detail.



CVNET COMMANDS

1) ME - SEND A MESSAGE TO REMOTE

2) SE - SEND A FILE TO REMOTE

3) RE - RECEIVE A FILE FROM REMOTE

4) PR - PRINT A TEXT FILE

5) CO - SETS PROTECTION FLAGS

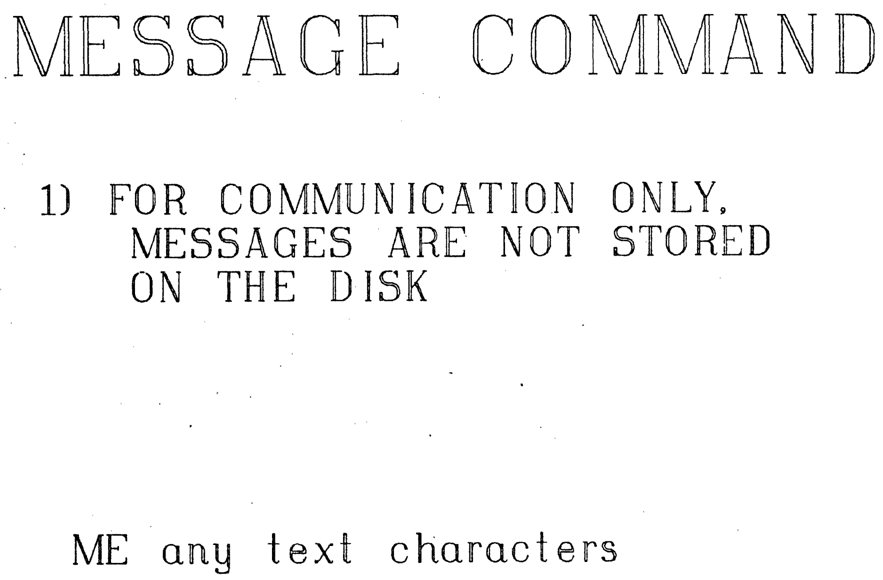
6) ST - PRINT STATISTICS

7) LO - PRINT THE LOG

8) QU - EXIT FROM CVNET

Foil #16

The message command is used only for carrying on a dialog between yourself and an operator at the remote system. Messages are not stored on the disk as binary and text files are. They only appear on the receiving taks output device.



MESSAGE COMMAND

1) FOR COMMUNICATION ONLY,
MESSAGES ARE NOT STORED
ON THE DISK

```
ME any text characters
```

Foil #17

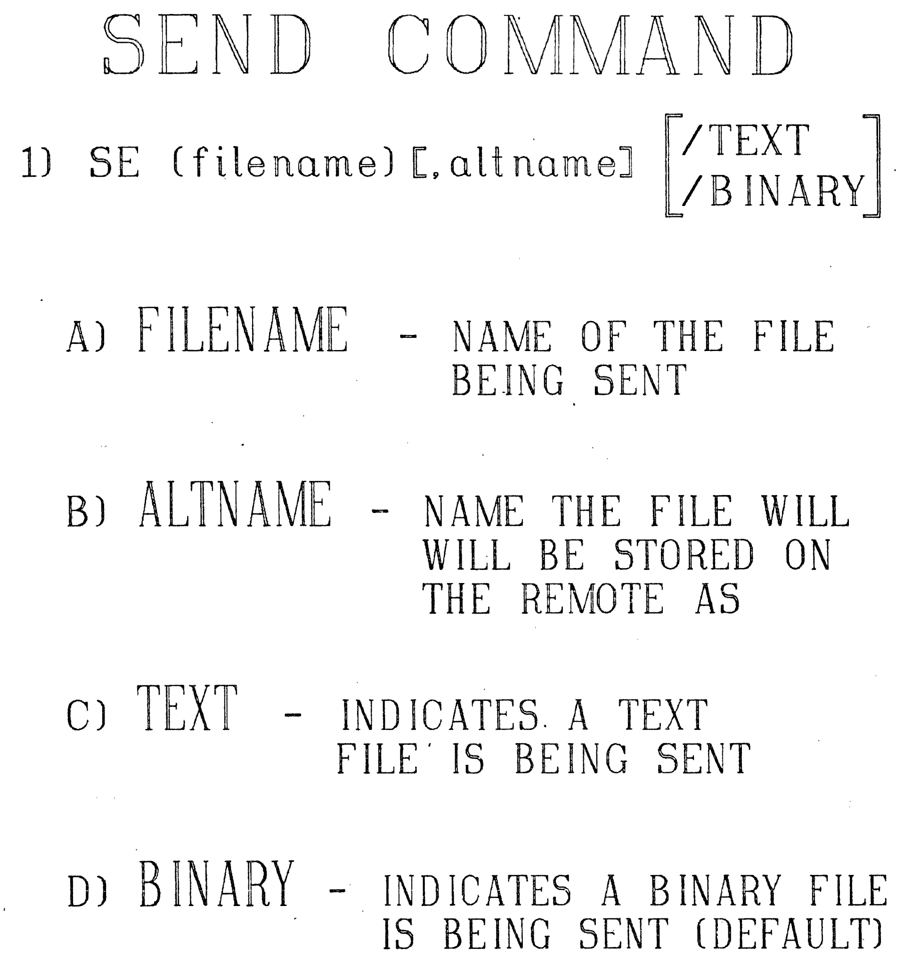
The SEND command is used to send a file from your system to some other system.

FILENAME is the full name of the file you want to read from your disk and pass to another system via CVNET.

ALTNAME is an alternate name you can attach to your file as you send it to the receiving system. The file will be stored on the receiving system's disk under the alternate name.

TEXT or TE indicates to CVNET that a text file is being sent.

BINARY or BI indicates a binary file. This need not be typed because binary is the default.



SEND COMMAND

1) SE (filename) [,altname] [/TEXT /BINARY]

A) FILENAME - NAME OF THE FILE BEING SENT

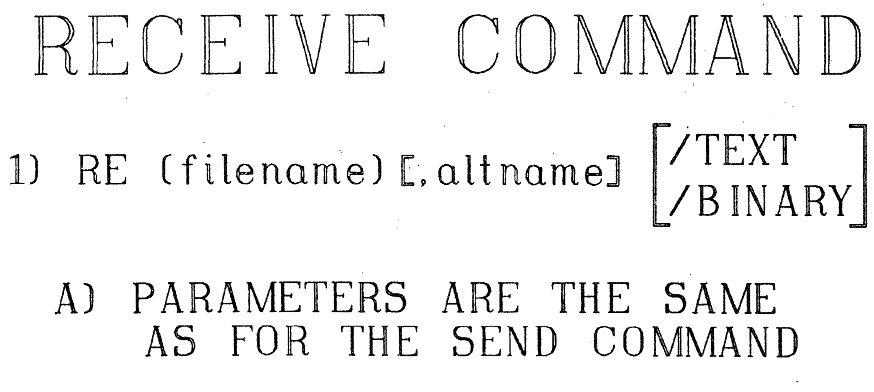
B) ALTNAME - NAME THE FILE WILL WILL BE STORED ON THE REMOTE AS

C) TEXT - INDICATES A TEXT FILE IS BEING SENT

D) BINARY - INDICATES A BINARY FILE IS BEING SENT (DEFAULT)

Foil #18

The RECEIVE command is the same as the SEND command except it is initiated from the receiving system. With the RECEIVE command you reach out and pluck a file from the remote system's disk. All the parameters have the same meaning as they did for the SEND command.



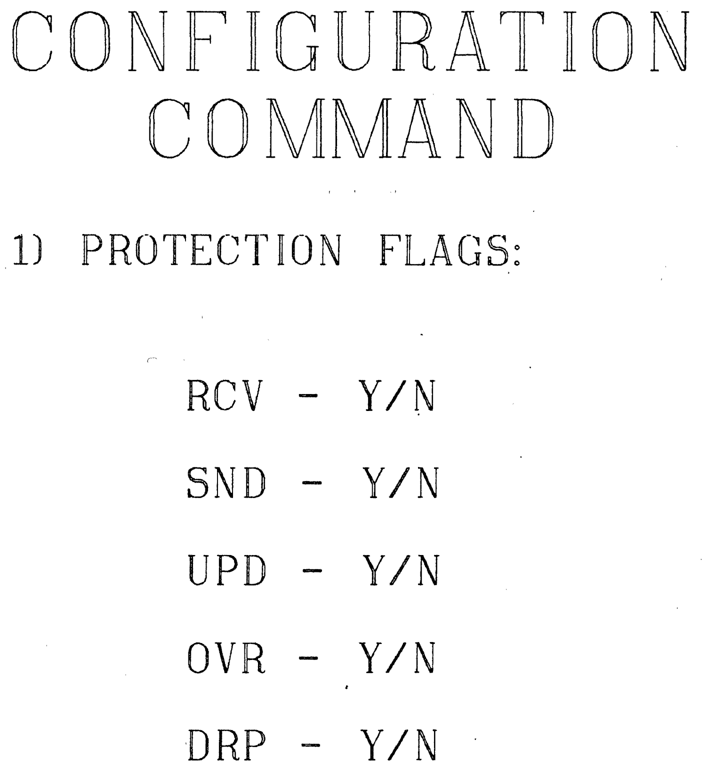
RECEIVE COMMAND

1) RE (filename) [,altname] [/TEXT /BINARY]

A) PARAMETERS ARE THE SAME AS FOR THE SEND COMMAND

Foil #19

Now, you might be concerned about unauthorized people having access to your disk. CVNET gives the user the option of setting some protection flags that will give him a measure of control. He can prevent outside users from writing files on his disk, copying files from his disk, and overwriting existing files.



CONFIGURATION COMMAND

1) PROTECTION FLAGS:

RCV - Y/N

SND - Y/N

UPD - Y/N

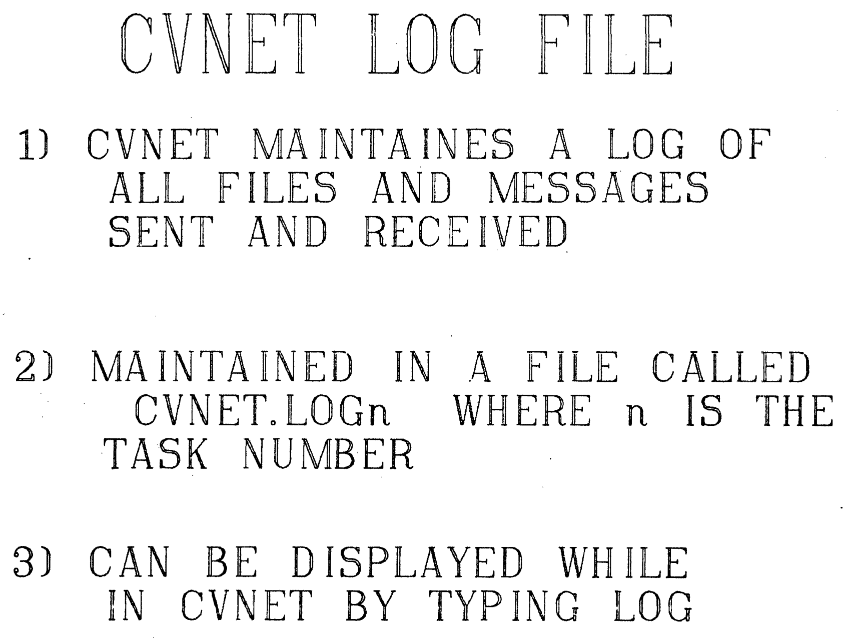
OVR - Y/N

DRP - Y/N

Foil #20

A log of each CVNET session is maintained on the system in a file called CVNET.LOGn where n is the task number.

This log can be displayed while CVNET is up by typing in LOG or LO. At systems level, you simply print the file CVNET.LOGn.



CVNET LOG FILE

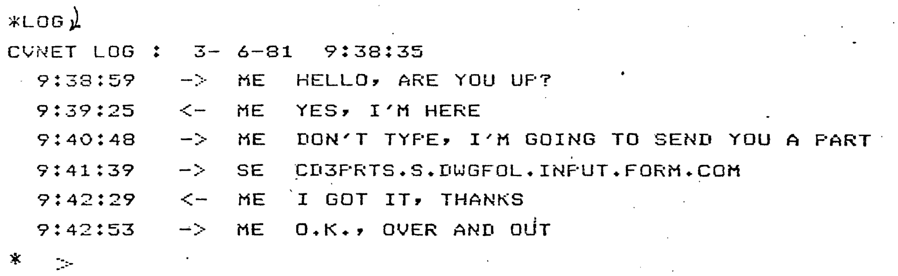
1) CVNET MAINTAINES A LOG OF
ALL FILES AND MESSAGES
SENT AND RECEIVED

2) MAINTAINED IN A FILE CALLED
CVNET.LOGn WHERE n IS THE
TASK NUMBER

3) CAN BE DISPLAYED WHILE
IN CVNET BY TYPING LOG

Foil #21

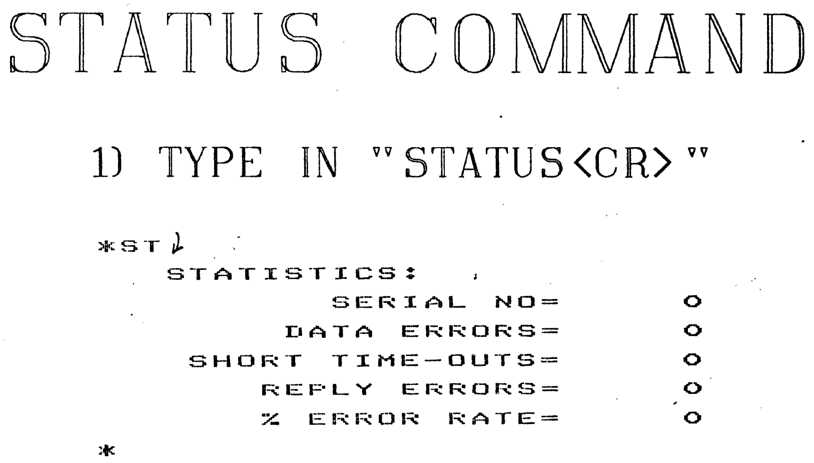
This is a printout of a log file. Each CVNET session is appended to the end of the file unless you elect to supersede the file. Each new session is indicated with the date and time. The arrows indicate the direction of communication.



```
*LOG↲
CVNET LOG :  3- 6-81  9:38:35
  9:38:59   ->  ME  HELLO, ARE YOU UP?
  9:39:25   <-  ME  YES, I'M HERE
  9:40:48   ->  ME  DON'T TYPE, I'M GOING TO SEND YOU A PART
  9:41:39   ->  SE  CD3PRTS.S.DWGFOL.INPUT.FORM.COM
  9:42:29   <-  ME  I GOT IT, THANKS
  9:42:53   ->  ME  O.K., OVER AND OUT
*  >
```

Foil #22

The STATUS command is intended to display statistics that were compiled during the CVNET SESSION. Typing in STATUS or ST causes the table to be printed out.



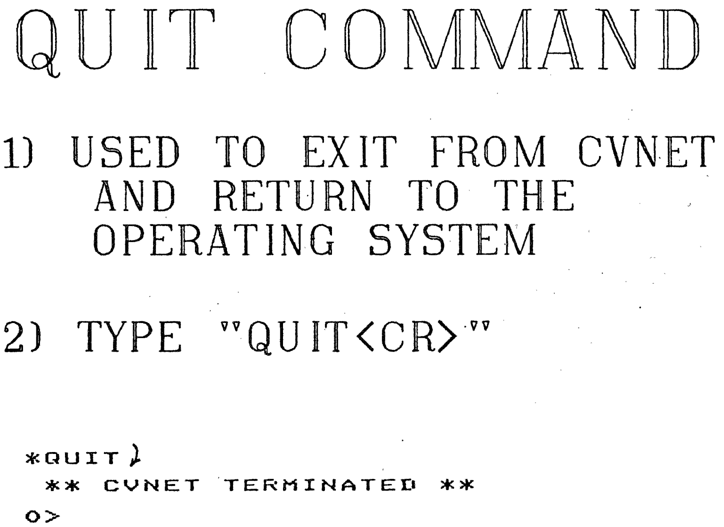
STATUS COMMAND

1) TYPE IN "STATUS<CR>"

```
*ST↲
     STATISTICS:
                    SERIAL NO=        0
                  DATA ERRORS=        0
           SHORT TIME-OUTS=           0
                 REPLY ERRORS=        0
                 % ERROR RATE=        0
*
```

Foil #23

The last command we are going to look at is the QUIT command. The QUIT command terminates CVNET and returns the task to the operating system. This command usually takes 15 or 20 seconds to execute.



QUIT COMMAND

1) USED TO EXIT FROM CVNET
   AND RETURN TO THE
   OPERATING SYSTEM

2) TYPE "QUIT<CR>"

```
*QUIT↲
 ** CVNET TERMINATED **
O>
```

Foil #24

CVNET provides the user with the ability to develop execute files of CVNET commands. These files can then be executed during a CVNET session. All one needs to do is type in the name of the execute file, and execution will begin.

EXECUTION FILES

1) TEXT FILES CONTAINING CVNET COMMANDS CAN BE EXECUTED FROM CVNET

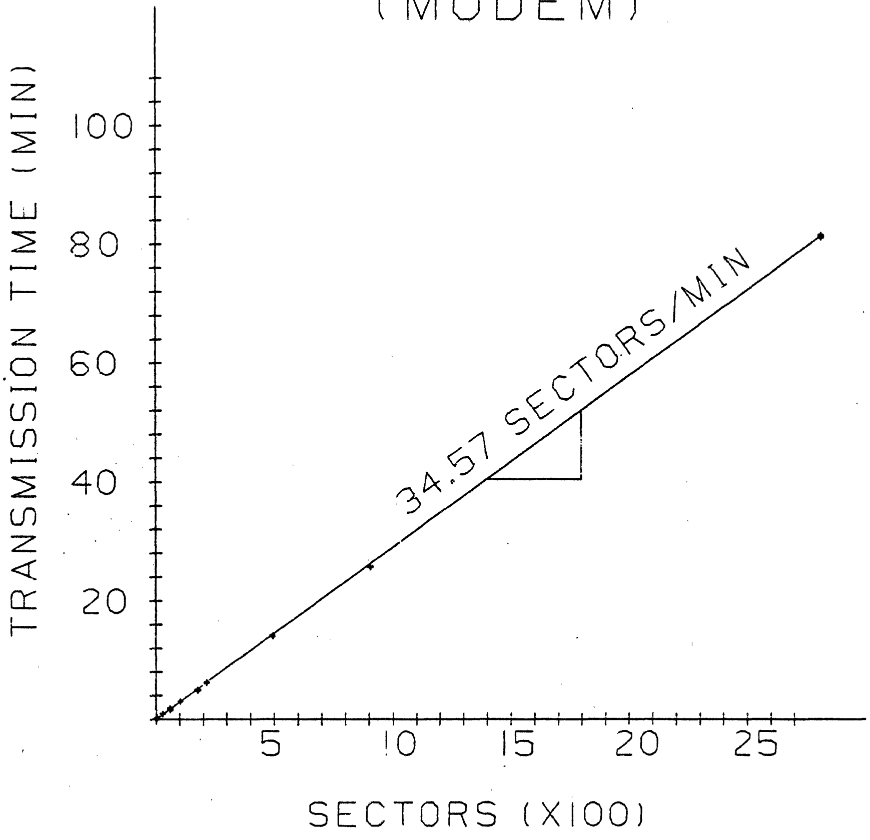
2) SIMPLY TYPE IN THE FILE NAME

Foil #25

These last three slides give you an idea of how long it takes to send data via CVNET.

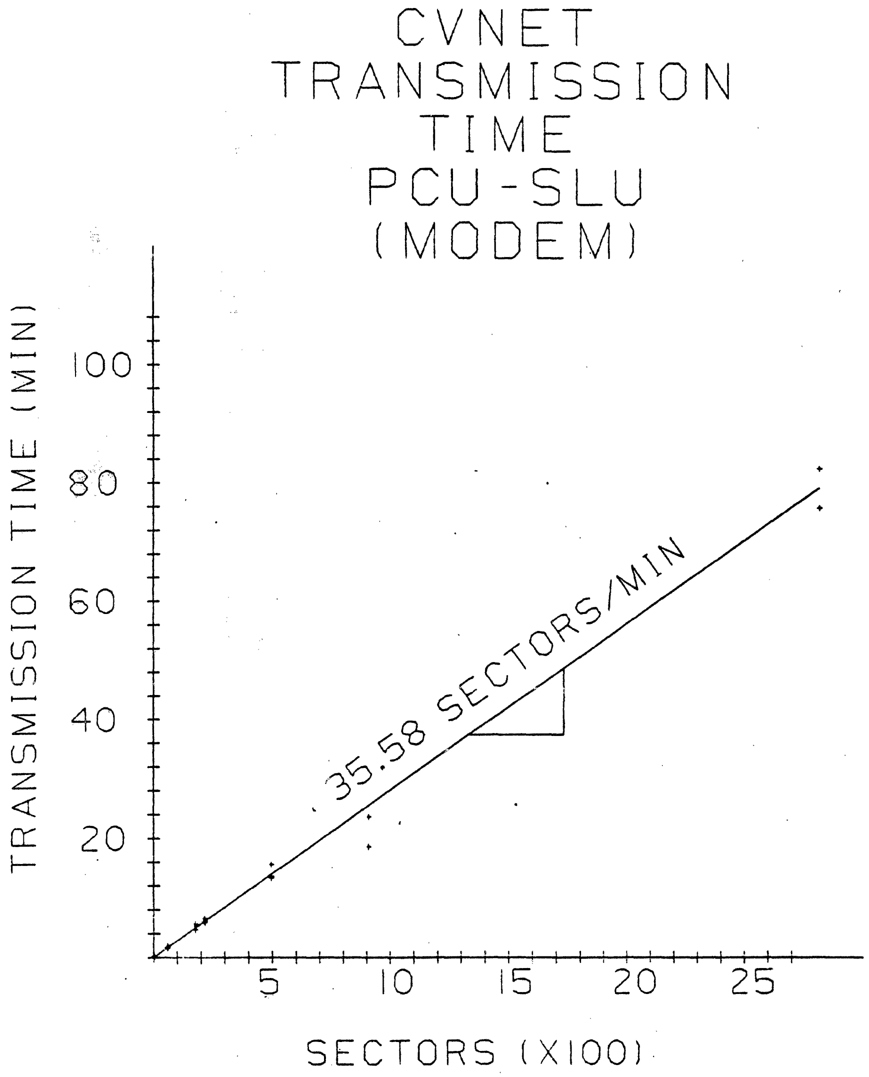
These times were taken using dial-up modems and 4800 baud lines. Teach system had a PCU board.

CVNET TRANSMISSION TIME PCU-PCU (MODEM)



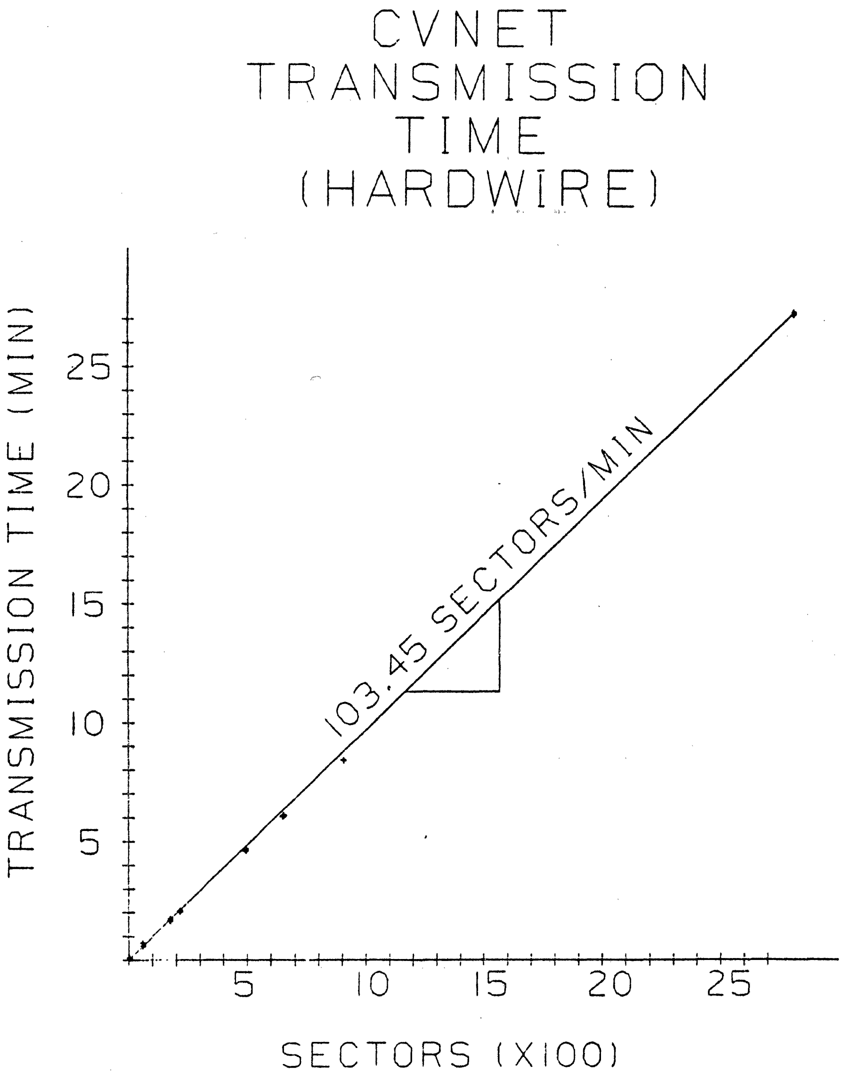
Foil #26

This slide is the same as the first except one of the systems has a SLU board in it.



Foil #27

The last slide was taken from two systems hardwired together. The line speed is limited by the PCU board to 19,200 baud.



CV HASP

A.H. Killinger
Drafting Interactive Graphics
NTD Division
Westinghouse Electric Corporation
P.O. Box 355
Pittsburgh, PA 15230
412-373-4147

ABSTRACT

This paper provides the Computervision (CV) user with some general information about CV HASP and some specific examples showing how to use the CV system as a terminal to a host mainframe.

The examples show the CV system operating as a terminal to both IBM and CDC mainframes and the transfer of a drawing data base between CV systems using an IBM as the common link.

This presentation also uses a set of slides and an overhead projector.

INTRODUCTION

CV HASP (Houston Automatic Spooling Program) is a synchronous communication package which allows the CV system to access the computing power and other facilities of a host mainframe.

Access to a mainframe computer provides the opportunity to perform complex analysis on data derived directly from the drawing data base. When a fast mainframe response is available, geometric definition and analysis could be accomplished at one session.

The examples used for this paper are simple and are only intended to illustrate the process.

CV HASP

CV HASP—What is it? HASP is an integrated hardware and software synchronous communication package which will provide access to a host mainframe. The CV system thus becomes a terminal of the host computer.

What do you need to use HASP?

1. A "PCU" board and software, both available from CV.
2. Sign-on and sign-off records. The CV user's guide* and your local computer user advisor will provide the correct format.
3. Communications link (telephone, modems, etc.)

Types of Communication Links.

1. Dial-up telephone and modems.
2. Private line with dedicated modems.
3. Direct hardwired cable (50 ft. max.)

*See "CV HASP Communications User Guide" (Order number 001-00221) for more detailed description.

EQUIPMENT

The equipment at the *W* MNC facility is as shown in Figure 1. It consists of two (2) CGP 200/10 systems each equipped with a PCU board and HASP software and both systems running 11-E Operating Systems. One CV system is hardwired to the CDC-7600 facility via a 9600 baud line while the other CV system uses a 4800 dial-up line to either the CDC facility or the IBM facility.

The hardwired line was obtained so that an engineer at a work station remote from the CGP could connect to the CDC by simply typing in HASP. Without the hardwire, the engineer had to type in @ CONNECT ↲ at his work station within a time limit after dialing the phone at the CGP.

The CV supplied commands used to control the terminal operation are shown in Figure 2. These commands facilitate the CONNECTing, OPENING and CLOSing of streams, SENDing, PRINTing, and QUITing of HASP operations.

In addition to the CV supplied commands, there are other commands which provide the CV HASP terminal operator with control of the mainframe and its support devices. On the IBM, these are called JES3 (Job Entry System) commands. On the CDC, the commands are for INTERCOM.

It is necessary to have a working knowledge of these commands in order to start and stop printers, to define and start and stop card readers, move files, etc. Your local computer advisor will be able to provide the information relevant to your site.

WHAT DID WE WANT TO DO AND WHY?

1. In order to make the computing capacity of the IBM available to the CV system, it was desired to be able to create an IBM job on the CV system, execute the IBM job on the IBM and receive the output at the CV terminal.

2. In order to make use of the specific programs on the CDC system (such as finite element analysis) it was desired to be able to create a CDC job on the CV system, transmit the job to the CDC, execute the job on the CDC and receive the output at the CV terminal. The CDC job (created on the CV system) would contain data derived from a CV drawing or a CV PEP or Fortran program. It is also desired to use the CDC system as the common link to other special purpose computers which will construct the appropriate grid on a geometry (PRIME) prior to finite element analysis and which will perform structural analysis (Data General). However, the communications with the special purpose computers is not reported on in this presentation.

3. Many of the Westinghouse CV sites will have PCU boards and HASP software which will provide access to a central Westinghouse IBM system. Since HASP provides more flexibility and stability than CVNET, it was desired to be able to use HASP and the IBM system as the vehicle to transfer drawings between CV systems.

WHAT HAS BEEN DONE?

1. An IBM job has been created on the CV system, transmitted to and executed on the IBM and the output printed on the CV terminal. The IBM job consisted of a set of six (6) Job Control Cards followed by a very small Fortran program followed by two (2) additional Job Control Cards (see Figure 3). The IBM job was transmitted to the IBM as shown in Figure 4. This is the actual printout from the typewriter and as such is very busy and although Figure 4 is included in the written report, it is not shown here. Figure 5 shows the commands which were typed in by the operator and deletes the system responses. After CONNECTing, the reader is OPENed, the file containing the IBM job is sent (SEND) and the reader is

CLOSED. Closing the reader signals, to the IBM, the completion of the batch job input. The printer is started and after execution of the IBM job, the output is transmitted to the CV system and the print stream is closed. The printout may be printed and/or displayed on a CRT as desired.

2. A CDC job has been created on the CV system, transmitted to and executed on the CDC and the output printed on the CV system. To do this, three text files were created on the CV system. These files are shown in Figure 6. The first text file contains five (5) lines of CDC Control Cards. The second text file contains a very short Fortran program. The third file contains a line with a hex 3F in the first character. This is the CDC "end of section" card which is needed to terminate the CDC Control Cards. As there is no key on the CV keyboard which wil generate a hex 3F, I used the PACHFIL command to generate the required character. The CDC job was transmitted as shown in Figures 7, 8 and 9. Figures 7 and 8 are the actual typewriter printouts which include system responses and are very cluttered. Figure 9 (which has most of the system responses deleted) shows the sequence of commands used to CONNECT, LOGIN, DEFINE, control the card reader and line printer, SEND the CDC job and receive the printout after execution of the CDC job. The printout may be printed and/or displayed on a CRT as desired.

3. A drawing has been transferred from one CV system to another CV system using HASP and the IBM as the common link. To do this, a text file which contains the required IBM job was created on the "SENDING" CV system. This text file and the HASP commands used to send the text file and the drawing is shown in Figure 10. Here we see that the CV text file contains nine (9) Job Control Cards and uses the IBM utility IEBGENER. IEBGENER will accept a data file, which is the drawing, and will put that data file in the punch queue of the "RECEIVING" CV terminal. The punch file will be in a "HOLD" condition.

Also shown in Figure 10 are the HASP commands used to SEND the IBM job and the drawing data base to the IBM. After execution of the IBM job, the drawing data base will be in the punch queue of the "RECEIVING" CV terminal. To retrieve the drawing data base and generate the drawing, the "RECEIVING" CV terminal would proceed as shown in Figure 11. This figure shows the CV system being CONNECTed and a JES3 command to list all data sets which are in a HOLD condition. Also shown is the JES3 command to delete the "LACE" cards from the beginning and end of the punch data. Next is the JES3 command to release the punch file from the HOLD condition which will allow the file to be transmitted to the "RECEIVING" CV terminal. After the file has been transmitted to the CV terminal, the file must be copied into the CD3PRTS catalog before the drawing may be called up on the CRT in the normal manner as also shown in Figure 11.

SAMPLE OF A HASP LOG PRINTOUT.

Figure 12 shows a sample of a HASP log printout. The log of the previous HASP session remains in the system until the next HASP session has been completed at which time the previous log is replaced with the current log.

PROBLEMS

Problems do exist. The problems appear to differ from site to site, at least in severity.

The main problem I have observed is that sometimes after the "@ CONNECT" command has been issued, the system will hang and will not respond until the operator enters a "CTRL-B". All commands after this point appear to be ignored and a "CTRL-SHIFT-POINT" (or "CTRL-C-C" on REV 11-E) must be entered to terminate the task. HASP on this task will not be usable until the system has been booted.

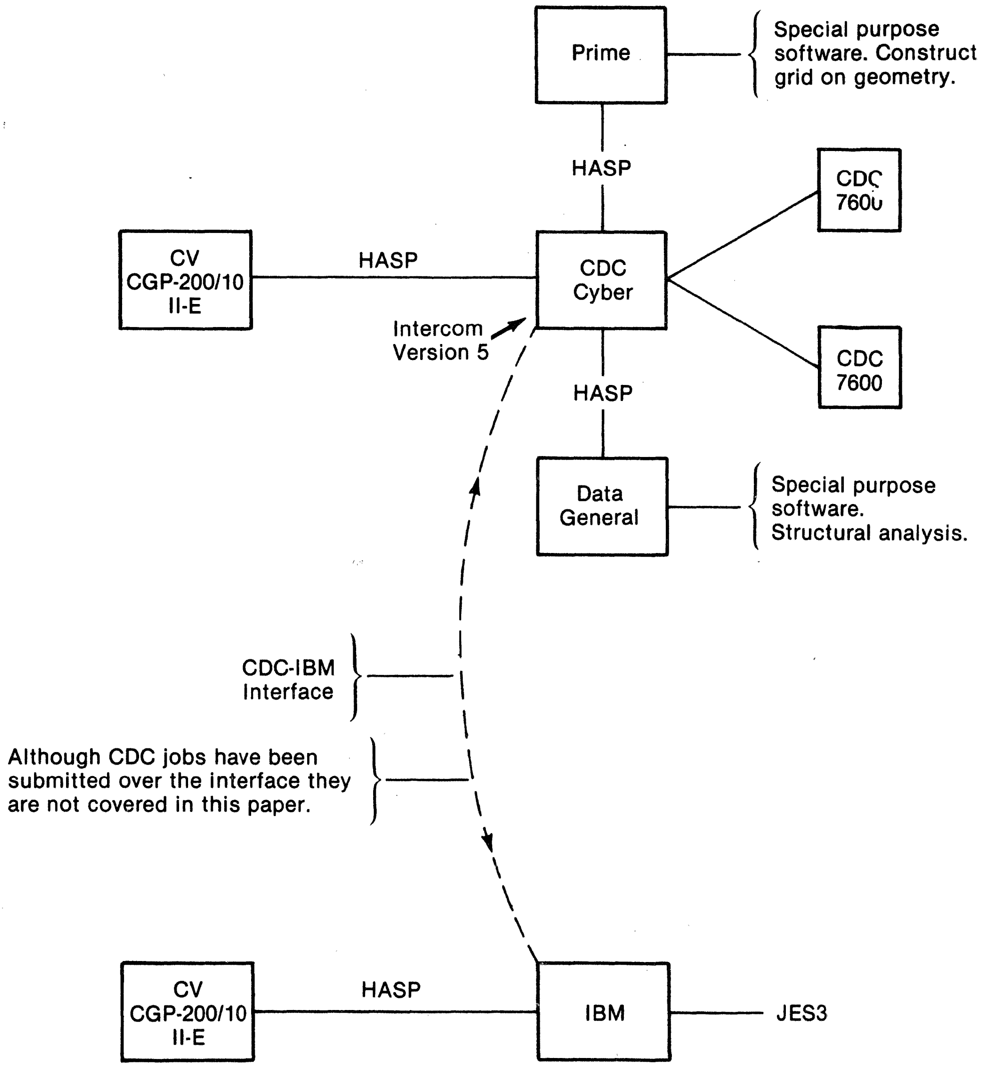
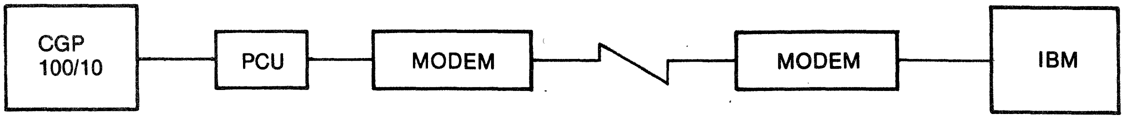

Prime
Special purpose software. Construct grid on geometry.
HASP
CDC 7600
CV CGP-200/10 II-E
HASP
CDC Cyber
Intercom Version 5
HASP
CDC 7600
Data General
Special purpose software. Structural analysis.
CDC-IBM Interface
Although CDC jobs have been submitted over the interface they are not covered in this paper.
CV CGP-200/10 II-E
HASP
IBM
JES3

Figure 1

CONNECT	CONNECT TERMINAL TO MAINFRAME.
OPEN<stream id>	OPEN READER.
SEND<stream id>: <filename> [/<flags>]	SEND FILES TO MAINFRAME.
CLOSE<stream id>	CLOSE READER.
SUSPEND	SUSPEND HASP BUT MAINTAIN LINE CONNECTION.
QUIT	TERMINATE HASP.
ABORT	EMERGENCY QUIT.
DISPLAY<description>	DISPLAY STATE OF TERMINAL DEVICES.
PRINT<text-file-name>	PRINT TEXT FILES.

Figure 2: CV HASP Command Summary



THE FOLLOWING IS AN IBM JOB CREATED ON THE CV SYSTEM:

```
OTHR.KILL.&BCD.HASPDEMO1
 5-26-81   16:54:43

/ /AHK3 JOB (          MDNS),'KILLINGER',REGION=200K,TIME=(0,10),
/ /   MSGLEVEL=(0,0)
/ /*MAIN CLASS=PRIME
/ /      EXEC FORTGCLO,PARM.FORT=NODECK
/ /FORT.SYSPRINT DD DUMMY
/ /FORT.SYSIN DD *


/*
/ /LKED.SYSPRINT DD DUMMY
/ /GO.FT06001 DD SYSOUT=A
```

ACCOUNT NO.

IBM STANDARD
JOB CONTROL CARDS

ANY FORTRAN
PROGRAM

IBM STANDARD
CONTROL CARDS

Figure 3

THIS SLIDE IS BUSY

THE NEXT SLIDE REMOVES
SYSTEM RESPONSES.

OPERATOR ENTERED
COMMANDS ARE UNDERLINED.

```
6>HASP


     CV HASP WORKSTATION
     FULL VERSION, REV 1.00
        2-16-81 14:40:58

CV: PC00 ATTACHED AS PC
CV: PC01 ATTACHED AS PG
CV: DOWN-LOADING PCU
CV: DOWN-LOAD COMPLETE

CV: SESSION NOT ACTIVE

CV: READY FOR CONSOLE INPUT
@CONNECT
1338152   IAT7560 RM4A2 ON LINE96     SIGNED OFF
1338167   IAT7576 LINE HAS TERMINATED
1441033   IAT7575 RM4A2 STARTED ON LINE96    (085) WITH CONSOLE SUPPORT

CV: READY FOR CONSOLE INPUT
@OPEN R1
CV: R1 PENDING
1441511   IAT6306 JOB 9054 IS CR          , CALLED BY RM4A2
CV: R1 READY

@SEND R1: OTHR.KILL.HASPDEMO1
CV: R1 SENDING
OTHR.KILL.HASPDEMO1
CV: R1 READY

@CLOSE R1
CV: R1 CLOSED

1442246   IAT6201 JOB 9062=RM4A2-01,01 JOBS,00019 CDS,P=15
1442526   IAT6101 JOB 9069 IS AHK3      , PRTY=06, CL=PRIME
1442548   IAT7450 JOB 9062, RM4A2-01 PURGED
1444132   4=  AHK3        IEF4031 AHK3 - STARTED - TIME=14.44.13
1444506   4=  AHK3        IEF4041 AHK3 - ENDED  - TIME=14.44.50
1444587   IAT7001 JOB AHK3       (9069) IS ON WRITER RM4A2PR1 (085),
RECORDS=000070.
1444589 •IAT7030 AHK3 (9069) ON RM4A2PR1 (085), SETUP F=F017006T.
S PR1,H=N
1445163   IAT7018 AHK3      (9069) RM4A2PR1 (085) REQ STARTED
1445164 •IAT7030 AHK3 (9069) ON RM4A2PR1 (085), SETUP F=F017006T.

CV: FILE:SPOOL.161440.P1.001
    OPENED FOR STREAM P1
CV: P1 CLOSED
1446241   IAT7450 JOB 9069, AHK3      PURGED
```

TO THE IBM, CLOSING THE READER
INDICATES COMPLETION OF THE
BATCH JOB INPUT.

NAME OF CV FILE
CONTAINING OUTPUT

OUTPUT FILE COMPLETE

Figure 4

```
6>HASP ↲
CV:READY FOR CONSOLE INPUT
@ CONNECT ↲
@ OPEN R1 ↲
@ SEND R1:OTHR.KILL.HASPDEMO1 ↲
@ CLOSE R1 ↲
S PR1 ↲
CV:FILE:SPOOL.161440.P1.001
   OPENED FOR STREAM P1
CV:P1 CLOSED
```

TO THE IBM, CLOSING THE READER INDICATES COMPLETION OF THE BATCH JOB INPUT.

NAME OF CV FILE CONTAINING OUTPUT

TRANSMITTAL OF OUTPUT FILE COMPLETE

OPERATOR ENTERED COMMANDS ARE UNDERLINED.

Figure 5

```
OTHR.KILL.&BCD.CVCDCPRT1
 2-24-81   8:18:05

AHK6,T07,P6000,UPQAHK,STMFB,JCSHRT.
ACCOUNT(           )
DEST(OUTPUT,1,3K,KILL,$CVTOCDC DIRECT DEMO$)
FTN.
LGO.
5>

OTHR.KILL.CVCDC.&BCD.CDCPROG
 5-27-81   16:25:02


OTHR.HASP.&BCD.789SYMBOL
 2-19-81   12:25:21

&12345678912345678912345678972
5>
```

STANDARD CDC CONTROL CARDS

ANY FORTRAN PROGRAM

CDC END OF SECTION CARD—TERMINATES CDC CONTROL CARDS

THE ? IS A HEX 3F.

Figure 6

THE NEXT TWO SLIDES SHOW ALL MESSAGES AND RESPONSES.

THE THIRD SLIDE REMOVES SYSTEM RESPONSES.

```
6>HASP

     CV HASP WORKSTATION
     FULL VERSION, REV 1.00
      2-24-81   8:57:56

CV: PC00 ATTACHED AS PC
CV: PC01 ATTACHED AS PG
CV: DOWN-LOADING PCU
CV: DOWN-LOAD COMPLETE

CV: SESSION NOT ACTIVE

CV: READY FOR CONSOLE INPUT
@CONNECT
WPSCC/CDC INTERCOM 5.0
DATE   02/24/81
TIME   08.59.30.

         PLEASE LOGIN

         CV: READY FOR CONSOLE INPUT
         LOGIN,_____ _____ ______
         02/24/81   LOGGED IN AT 09.00.34.
                    WITH USER-ID
                    EQUIP/PORT 14/003
COMMAND—
DEFINE,CR1,029
COMMAND—
READ
COMMAND—
ON,LP1
COMMAND—
```

STANDARD LOGIN PROCEDURE FOR CDC SYSTEM.

OPERATOR ENTERED COMMANDS ARE UNDERLINED.

Figure 7

```
CV: READY FOR CONSOLE INPUT
@OPEN R1
CV: R1 PENDING
CV: R1 READY

@SEND R1:OTHR.KILL.CVCDCPRT1
CV: R1 SENDING
OTHR.KILL.CVCDCPRT1
CV: R1 READY

@SEND R1:OTHR.HASP.789SYMBOL/TEXT,NOXLATE
CV: R1 SENDING
OTHR.HASP.789SYMBOL
CV: R1 READY

@SEND R1:OTHR.KILL.CVCDC.CDCPROG
CV: R1 SENDING
OTHR.KILL.CVCDC.CDCPROG
CV: R1 READY

@CLOSE R1
09.02.47. AHK60AB.   BEG IP=00,   TID=LO
CV: R1 CLOSED
09.02.54. AHK60AB.        19 CARDS READ
CV: FILE: SPOOL.240857.P1.001 <———— CV OUTPUT FILE NAME
     OPENED FOR STREAM P1
09.03.31. AHK60AB.   BEG OP=40, TID=LO

CV: READY FOR CONSOLE INPUT

CV: READY FOR CONSOLE INPUT
CV: P1 CLOSED
```

OPERATOR ENTERED COMMANDS ARE UNDERLINED.

Figure 8

```
  6>HASP ↲
*    ʃ
  @ CONNECT
* PLEASE LOGIN
  LOGIN, USER ID, PASSWORD, ACCT. NO. ↲
*    ʃ
* COMMAND—
  DEFINE,CR1,029 ↲    ←——— DEFINE CARD READER AS 029
* COMMAND—
  READ ↲              ←——— START THE CARD READER
* COMMAND—
  ON,LP1 ↲            ←——— TURN ON THE LINE PRINTER.
* COMMAND—
  @ OPEN R1 ↲
*    ʃ
  @ SEND R1:OTHR.KILL.CVCDCPRT1 ↲
*    ʃ
  @ SEND R1:OTHR.HASP.789SYMBOL/TEXT,NOXLATE ↲
*    ʃ
  @ SEND R1:OTHR.KILL.CVCDC.CDCPROG ↲
*    ʃ
  @ CLOSE R1 ↲
*
* CV:FILE:SPOOL.240857.P1.001
*    OPENED FOR STREAM P1
*
    ʃ
* CV:P1 CLOSED
* CV OR CDC SYSTEM RESPONSES.
```

OPERATOR ENTERED COMMANDS ARE UNDERLINED.

Figure 9

IBM JOB
USES SYSTEM UTILITY
PROGRAM IEBGENER.

```
OTHR.KILL.HASP.&BCD.CVDWGTOIBM
 2-12-81   13:41:45

/ /AHK8 JOB (AP41313,MDNS),'KILLINGER',REGION = 100K,TIME = (0,10),
/ / MSGLEVEL = (0,0)
/ /*MAIN CLASS = PRIME
/ /*FORMAT PU,DDNAME = SYSUT2,DEST = MB
/ /     EXEC PGM = IEBGENER
/ /SYSIN DD DUMMY
/ /SYSPRINT DD SYSOUT = A
/ /SYSUT2 DD SYSOUT = B,HOLD = YES
/ /SYSUT1 DD DATA
5>
```

WILL ACCEPT A DATA FILE AND CREATE A PUNCH FILE AND PUT IT INTO MB'S QUE. THE PUNCH FILE WILL BE IN HOLD.

```
OTHR.HASP.&BCD.STAR
 1-27-81   16:22:23

/*
5>
```

IBM JOB
TERMINATOR.

```
6>HASP
CV:READY FOR CONSOLE INPUT
@ CONNECT
@ OPEN R1
@ SEND R1:OTHR.KILL.HASP.CVDWGTOIBM
@ SEND R1:CD3PRTS.KILL.9557D05.103.1COM/BINARY,NOXLATE
@ SEND R1:OTHR.HASP.STAR
@ CLOSE R1
```

THIS IBM JOB, WHEN EXECUTED, WILL CREATE A PUNCH FILE WHICH CONTAINS THE SUBJECT DRAWING DATA AND WILL PLACE THAT PUNCH FILE INTO THE OUTPUT QUEUE OF THE "RECEIVING" CV SYSTEM.

Figure 10

```
6>HASP
@CONNECT
I U, j=?, H=Y,N=ALL
```

THIS IS A TERMINAL COMMAND TO LIST ALL DATA SETS IN "HOLD."

```
1357205 IAT8131 J=AHK8 (1641), D=MD, T=PUN, H=Y, L=641
X,WTR,OUT=PUI,H=N,B=N
```

COMMAND TO DELETE "LACE" CARDS.

```
F U, J=1641,NH=N
CV:FILE:SPOOL.12355.U1.001
   OPENED FOR STREAM U1
CV:U1 CLOSED
@ QUIT
```

THIS IS A TERMINAL COMMAND TO RELEASE THE PUNCH FILE FROM "HOLD."

```
6>COPYFIL SPOOL.121355.U1.001,CD3PRTS.KILL.HASP.DWGIBM1.COM
```

$\underbrace{\text{KILL.HASP.DWGIBM}}_{\text{DRAWING NAME}}$

```
  SPOOL.121355.U1.001
    COPIED AS
  CD3PRTS.KILL.HASP.DWGIBM1.COM
```

THE DRAWING KILL.HASP.DWGIBM1 MAY NOW BE CALLED UP IN THE NORMAL MANNER:

```
6>CADDSCLR
 PART NAME:KILL.HASP.DWGIBM1
  ENTERING OLD PART
  REGEN NOT RESTORED
```

Figure 11

```
PRINT HASP.LOG06


HASP.&BCD.LOG06
 3-10-81    12:42:39


    CV HASP WORKSTATION
    FULL VERSION, REV 1.00
     3-10-81    12:34:04
CV: SESSION NOT ACTIVE
@CONNECT
1237198   IAT7576 LINE HAS TERMINATED
1237539   IAT7575 RM4A2 STARTED ON LINE96   (085) WITH CONSOLE SUPPORT
@OPEN R1
CV: R1 PENDING
1238559   IAT6306 JOB 6747 IS CR          , CALLED BY RM4A2
CV: R1 READY
@SEND R1:OTHR.KILL.HASPDEMO1
CV: R1 SENDING
OTHR.KILL.HASPDEMO1
CV: R1 READY
@CLOSE R1
CV: R1 CLOSED
1240358   IAT6201 JOB 6755=RM4A2-01,01 JOBS,00019 CDS,P=15
1240368   IAT6101 JOB 6756 IS AHK3          , PRTY=06, CL=PRIME
1240380   IAT7450 JOB 6755, RM4A2-01 PURGED
1240435   4= AHK3       IEF403I AHK3 - STARTED - TIME=12.40.43
1240582   4= AHK3       IEF404I AHK3 - ENDED - TIME=12.40.58
1240599   IAT7001 JOB AHK3      (6756) IS ON WRITER RM4A2PR1 (085),
RECORDS=000070.
1241000 *IAT7030 AHK3 (6756) ON RM4A2PR1 (085), SETUP F=F017006T.
S PR1
CV: FILE: SPOOL.101234.P1.001
    OPENED FOR STREAM P1
CV: P1 CLOSED
1243181   IAT 7450 JOB 6756, AHK3       PURGED
@QUIT
CV: SESSION TERMINATED
CV: HOST DROPPED THE LINE
5>
```

Figure 12

CAD/CAM DATA MANAGEMENT IN A HOST-BASED NETWORK

Raphael McBain
General Dynamics
Western Data Systems Center
P.O. Box 80847
M/S 86-5651
San Diego, CA 92138
714-692-5722

ABSTRACT

Geographically separate interactive graphics (stand-alone) systems can communicate with a mainframe host computer and thus with each other in a network. The network provides a large single-source data storage and retrieval capability with mainframe capabilities:

1. huge storage capacity
2. data access control
3. data product management with archival, backup, protection, and audit trail
4. interface to major processing functions on host computer (a two-way street)
5. message storage and routing

CAD/CAM data is digital data created on graphics systems. In many companies, the graphics systems are stand-alone turnkey systems, useful for their high productivity.

Digital data produced on stand-alone systems is subject to proliferation without mercy. Typically the data is maintained on magnetic tapes which can easily get lost or overwritten or damaged.

Even in the best of shops, redundant copies of the data are maintained on a variety of tapes, creating a data management problem. Access to the data is unrestricted except when the tapes are locked in cabinets. Meanwhile, the same data on disc is available for manipulation by virtually anyone.

1. We need a safe place to keep data.
2. We need controlled access to the data.
3. We need control over the data by management.

Solution:

Tie all stand-alone graphics systems into a network which features a main-frame host to store and control the data. Design and implement a data storage and management system which is located on the host and which has the features described below.

This retains the advantages of stand-alone systems for responsive interactive graphics processing and adds the advantages of centralized data processing for data management and integration of processing functions on other computers.

FEATURES OF A HOST-BASED NETWORK PROCESSING SYSTEM FOR DATA STORAGE AND MANAGEMENT

1. Storage of data.

After completing work on a part file, a graphics user specifies network storage for the finished part file. The data is then stored on local disc and is entered into the automated network processing system (NPS). Other useful information is stored in the catalog record, including:

date and time (from host)
userid of creator
group code (user is member of a group)
tool ID (to pin down graphics part file format)
and a status code.

The name is any valid filename on the stand-alone graphics system; the only limitation is 60 characters maximum. Multiple versions of a file can be saved; this guarantees that a new copy of a file can always be saved, even though the file already exists, without overwriting the file. The maximum number of versions allowed per file can be set by the system manager to the desired value (from 1 to 99). At GD, 10 versions are allowed.

The data is kept permanently by design of the NPS.

2. Security.

Each data storage or management function in the network is accomplished by a transaction. Every transaction requires logon to the network. This requires prior authorization of users by management.

Authorization data for each user is kept in a user control record. This data includes the userid, password, one or more group codes, and a level code associated with each group code. The level code defines how the user is allowed to interact with the data, when compared with a table of allowed status code changes. The table consists of records; each record defines the allowed change of status (from, to values) for a particular level code.

An audit trail is kept of every transaction in the system that changes any data. This audit trail is kept in an audit file which can be searched by system routines to determine all changes that have occurred to any data file. The audit trail includes installation of user control records by management.

Logon to the network gathers userid, password, and group code. Anyone having the same group code can modify data created by another member of the group. (modify = DELETE, REPLACE, or RENAME) Typically all members of a group are under the direction of one group leader. A member of any other group can only read the data; data cannot be modified by any member who logged on with a group code that does not match that stored with the file.

A user whose level code is defined as "in training" cannot modify any production data stored on the system. This user can store only temporary files and delete them, and can only read production files.

A file can be locked so that only a particular user can unlock the file. Other members of the group cannot unlock the file. A safety valve exists in that the group leader is allowed to unlock this file. This is useful for coordination of efforts when the user cannot immediately consult with other members of group or with the group leader.

3. Audit trail.

An audit trail is kept of every transaction in the system that changes any data. This includes any change to the catalog. The audit trail is kept in an audit file that can be searched by system routines to determine all changes to the data in or to the catalog record of any data file.

Every transaction that changes a catalog record creates an audit record. If desired, other transactions can be allowed to create audit trail records also. The audit trail record stores the filename, transaction data, and user data. Since the catalog record contains counters of reads and writes, any read (GET) and any write (SAVE/REPLACE) creates an audit record. A LISTing of the catalog does not change any catalog record.

4. Status of the Product.

Every shop needs control by management over the production status of its products. Data produced by a CAD/CAM group is either product data (salable) or non-product data (stays within the company; may become part of production data). NPS is designed to have the capability of controlling the status of a part file through a series of sequential processing steps.

Each step can refer to a stage in production; a status code value is assigned to each step. Typical Stages of production are:

1. engineering work (and rework)
2. analysis of "finished" data by analysis group
3. checking
   (how well does the data correlate with standards?)
4. verification by configuration management
   (how well does the data coordinate with all related data?)
5. release

The change of status from one value to another is controlled by a combination of two factors:

1. a level code assigned the user by management
   (kept in user control record by NPS).
2. a table that relates user level codes to allowed changes of status code value.

Each group code has a level code associated with it in the user control record. One can logon to NPS with any group code in one's own user control record. The level code is then taken by NPS from that user control record.

The implementation of NPS at General Dynamics uses the stages of production summarized above.

Range = 1–3:
Engineering group performs design/drafting functions.
Range = 4–6:
Analysis groups report all findings to engineering group leader. Group leader changes status when all scheduled groups approve data or require rework.
Range = 7–9:
Checking function can be done by separate group or by member of the same group (who is assigned a group code having a level code that has the "check" value).
Range = 10–12:
Config management verifies conformance to overall design.
Range = 13–15:
Only the release group can release data. No one can unrelease data. Any change to a data file that has been released requires that a copy be made under a new name. The new file can then be revised as desired and processed through the network.

A file can have rework status at any time, but the status code value remains in the range of values assigned to the group that requests rework. This means the status code will only advance, never regress, from range to range.

Note: To illustrate the use of status code values within each range, let each trio of values be represented by a,b,c in order of ascending value. When a member of the cognizant group sets the status code to "NC" the "RESULT" is achieved:
NC. . . .RESULT . . . .
a = > data enters control of this group and is "locked"
b = > a member of this group has decided the data needs to be reworked; the data is considered "unlocked" to allow a member of the engineering group to modify data.
c = > the data is approved by this group and is considered "locked" (cannot be modified while it retains this status code).

Weekly management reports are generated via a task schedule reporting system. The data records for these reports are generated by the appropriate transactions. This amounts to creating a card image record at completion of each transaction. The resulting file of text records can be used as input to a reporting system to create management reports.

Process control can be extended into manufacturing. This requires extending the table of allowed status code changes to include new values. The meanings of the new values are of course installation-dependent.

5. Sending messages.

A user or the system manager can send messages to a user, to a terminal, to all members of a group, or to all users. The message can be sent now or stored to be displayed when the recipient logs on. A user can identify all users now logged on. The message sub-system helps to control the process of changing product data. A designer can send a message late at night to the group leader to be displayed when group leader logs on in the morning.

6. Data interchange in a network.

Anyone logged on to the network has access to data stored in the network (unless the data is read-protected). One user can store a file, another can read it immediately upon being stored. This is useful in coordinating:

large projects (inter-divisional)
library (non-product) data (has inter-divisional support)
other information (charts, graphs, text, statistics)

Messages can be sent/received system-wide.

7. Data can be categorized into types.

1. Product data.

All data is considered product data in NPS unless it is flagged as library data. Product data goes through the production control process. At GD, this data must follow the standard naming convention.

Data is saved permanently (this can be 75 years).

2. Non-product data.

This includes LIBRARY data and other data.
LIBRARY data:

— goes through a control board review process
— requires a standard naming convention
(determined by project management)
— is saved permanently.
— becomes an integral part of product data, and thus once RELEASED into the production process can never be changed. Only a copy can be modified.
— The library file flag is set by a user authorized to perform this function.
— Only users authorized to modify library files can do so.

Other data might include files such as:
keyfiles and text files (parameter or PEP files).
These can be considered as library data, subject to control by a library data administrator.

These flags can be ignored if a simple file storage and retrieval system is all that is required.

8. Data maintenance.

Data backup is a function of data management performed to guarantee that a good copy of the data is always available despite accidental destruction of the primary copy. Data backup is performed periodically using well-planned and tested procedures.

All data maintenance tasks should be as automated as possible to minimize by-hand processing of the necessary functions. Maintenance work is performed on the host and does not affect users of interactive graphics systems.

GD is using a highly reliable system to guarantee data integrity. The system uses both mass storage devices and DASD (rotating disk) to store data. The storage capacity is tens or hundreds of thousands of megabytes, capable of fully-automated data management. The system is qualified for processing classified data.

9. Integration of processing functions.

Large benefits accrue to full-system users. The full-system concept accepts data early in the design process. From then on, the data is processed (changed or read) by any group in the plant assigned a role in handling this data. This will all be under control of the host-based data management system from the moment the data is first stored on the host, and will control all processing of that data through analysis functions to the end-product.

In the fully-integrated plant, all ancillary functions are controlled through the data management system, including inventory control, ordering, planning for long lead-time items, and shipping. Engineering data is no longer hand-carried by mag tape to a computer which is then directed to post-process the data.

In the full-system, data is tagged for post-processing in the system; post-processing is accomplished by transfer of data along well-defined paths to well-controlled post-processing programs within the control of the full-system. Data transfer is controlled by the system, rather than at the mercy of hand-carrying mag tapes and paper procedures. Selected post-processing functions can be set up to occur automatically when the status code arrives at particular values.

If required, the post-processed data can be stored back on the NPS. This data (derivative data) can then be linked to the originating data either by naming conventions or by fields in the data catalog that are reserved for such purposes.

In effect, NPS creates a closed-shop environment for data storage and management even when the graphics systems are operated in an open-shop fashion.

Plant & Architectural

AUTOMATED DESIGN/ANALYSIS OF PRESSURE VESSELS USING THE COMPUTERVISION SYSTEM

By
R.F. Hudale and R.K. Mysore
Westinghouse Electric
Monroeville Nuclear Center
Bay 371
Box 355
Pittsburgh, PA 15230
412-373-5569 : 373-4107

ABSTRACT

This paper describes an innovative approach to pressure vessel design and analysis developed at Westinghouse Equipment Engineering on the Computervision system. A series of P.E.P. and CV-Fortran programs integrated with the CV graphics capabilities are utilized to carry out the CADD applications described here.

Some of the applications that will be addressed are:

- Equipment Design
  - — Equipment Sizing
  - — ASME Code Calculations — Requirements for Heads, Shell Reinforcements
  - — Outline drawing generation

- Equipment Analysis
  - — Preparation of Mathematical model
    - — Pipe Model
    - — Axisymmetric Model

- Communication Networks
  - — CV-HASP

EQUIPMENT DESIGN

Equipment Sizing—One of the initial steps in pressure vessel design is to determine the overall size of the vessel in accordance with the functional, shipping, plant layout and other requirements imposed on the equipment.

Through the use of the CV system, software has been written and implemented to perform the sizing operations. By inputting the functional design requirements and other limiting parameters, the shell diameter, cylindrical length, and suitable head types are selected—all through an interactive process. The software is written using CV's Parametric Element Processor (PEP) and can handle flat, hemispherical, standard and non-standard ellipsoidal and torispherical heads.

A typical section of the head with crown and knuckle regions is shown in Figure 1. The volume of the head is the volume of the surface of revolution of knuckle and crown arcs about the vertical axis. The volume of the knuckle portion is computed from

$$V_K = \pi \{ H_1(R^2+R_2^2) - \frac{H_1^3}{3} + R \{ H_1\sqrt{R_2^2-H_1^2} + R_2^2 \text{Sin}^{-1}\frac{H_1}{R_2} \} \}$$

where

H_1 = Height of knuckle portion of head
R = X−coordinate of the locus of knuckle curve
R_2 = Radius of the knuckle curve

The volume of the crown portion is computed from

$$V_C = \frac{\pi H_2^2}{3}(3 R_3 - H_2)$$

where

H_2 = Height of crown portion of head
R_3 = Radius at the crown of head

MECHANICAL DESIGN

In order to establish the structural integrity of the pressure vessels for the internal and external pressure, temperatures, static or dynamic loads, the criteria of the ASME Code for pressure vessels (Sections III & VIII) most frequently serve as a guide.

Several questions need to be answered:

Are the nozzle openings in the vessel adequately reinforced?
Are the attachment welds between the nozzle and vessel properly sized?
Are the reinforcements adequate (in contour, etc.) to avoid areas of excessive stress concentration?

For this purpose, software has been introduced on the CV system to provide the ASME Code calculations. Included in the output are such data as required thicknesses of the vessel shell, heads, nozzles, and the reinforcement requirements for nozzle openings. Typical reinforcement details plotted by the software are shown in Figure 2.

OUTLINE DRAWING

Using the CV graphics capabilities and the data stored as a result of the previous software routines, an outline drawing of the pressure vessel is generated and displayed. The drawing provides the details of overall dimensions, nozzle attachments and the interface between equipment supports and support foundation including baseplate dimensions and anchor bolts.

In this graphical representation of the vessel, the determination of the nozzle locations on the various types of heads is of interest. Nozzles can be positioned vertically or perpendicular to the head by specifying the coordinates of the center of nozzle and the diameter of nozzle. The outline drawing used for plant layout is typified in Figure 3.

EQUIPMENT ANALYSIS

Mathematical Model—From the equipment design (outline drawing) a detailed finite element model can be created and displayed. The model, cantilever pipe or axisymmetric, can be used for dynamic and static analysis. By inputting the number of elements in the heads, support region, and shell, the PEP program automatically determines the distribution of nodes and element pattern as shown in Figures 4 and 5.

The program also translates the node and element definition to that required by a large finite element computer code, WECAN (Westinghouse Electric Computer Analysis Program). Along with the node and element description, the complete input data file is generated by responding to questions asked by the program regarding:

Shock Spectrum
Element Type
Material Properties
Coupled Nodes
Nodal Forces
Pressure Loads
Boundary Conditions

All this information is written on a data file in the appropriate WECAN-specified format.

With the addition of the required Job Control Language (JCL) the data generated can be transferred from the CV minicomputer to a large mainframe (CDC computer) for execution. This transfer will be discussed further in the next section.

COMMUNICATION NETWORKS

The link between the CV (CADD) minicomputer and mainframe that has been used successfully to transfer data is CV-HASP. CV-HASP is a synchronous communications software package that allows the CV work station to operate as a terminal to a host mainframe. This integration of computer systems is depicted in Figure 6.

The particular application that we have implemented is the transfer of the WECAN input data generated on the CV system to the CDC 7600 for execution. Through the use of the appropriate HASP commands the input file along with necessary JCL created on the CV is sent to the host mainframe. After execution the output can be viewed on the CV screen or printed.

CONCLUSIONS

The application of the CV system for pressure vessel sizing, designing, outline drawing, mathematical modeling and transmitting the data base to mainframe computer for finite element analysis has resulted in significant engineering productivity improvement. Additional benefits are gained through minimized human errors due to the use of a common data base and faster turn-around.

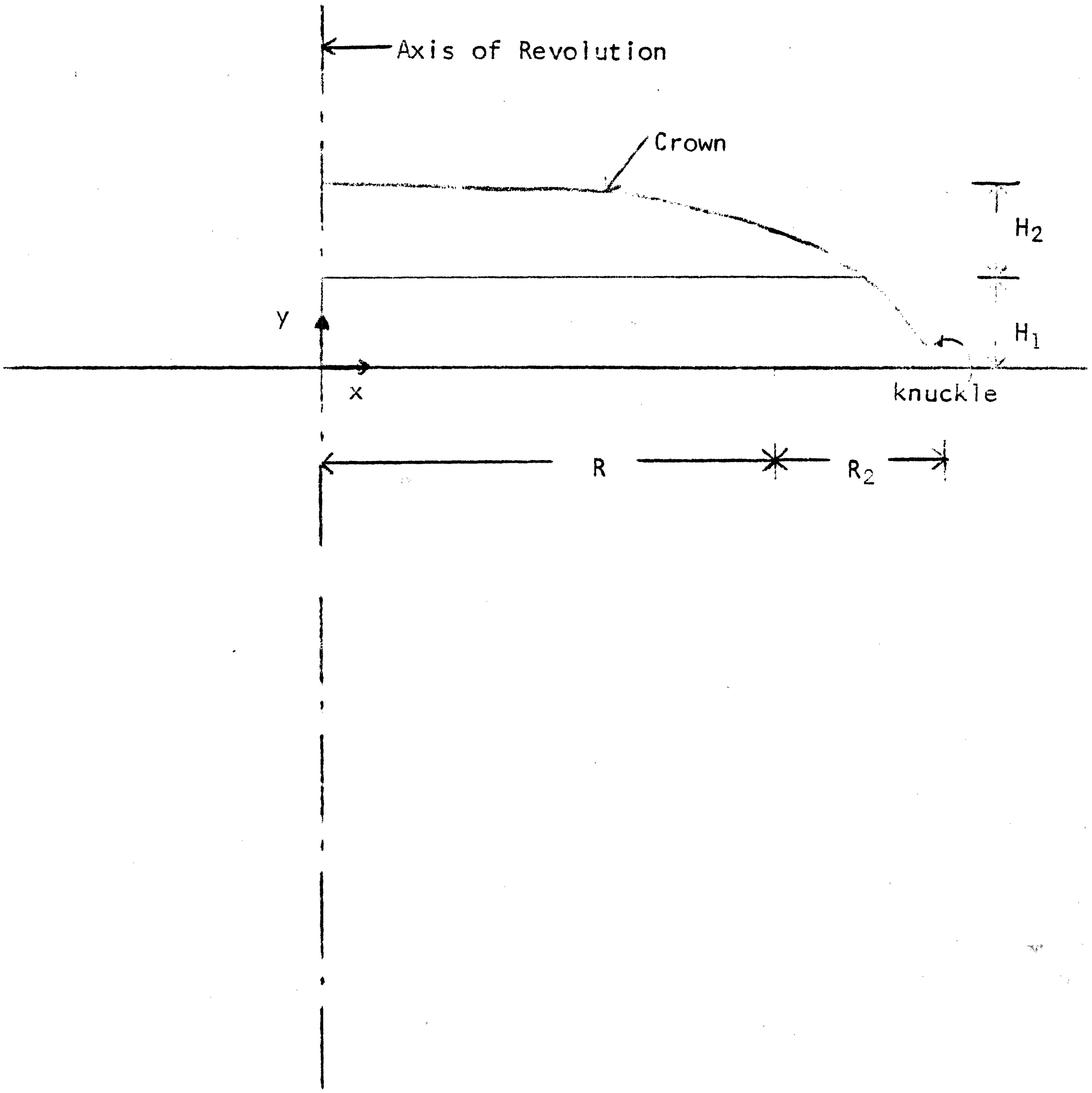


Figure 1: Parameters of a Typical Head

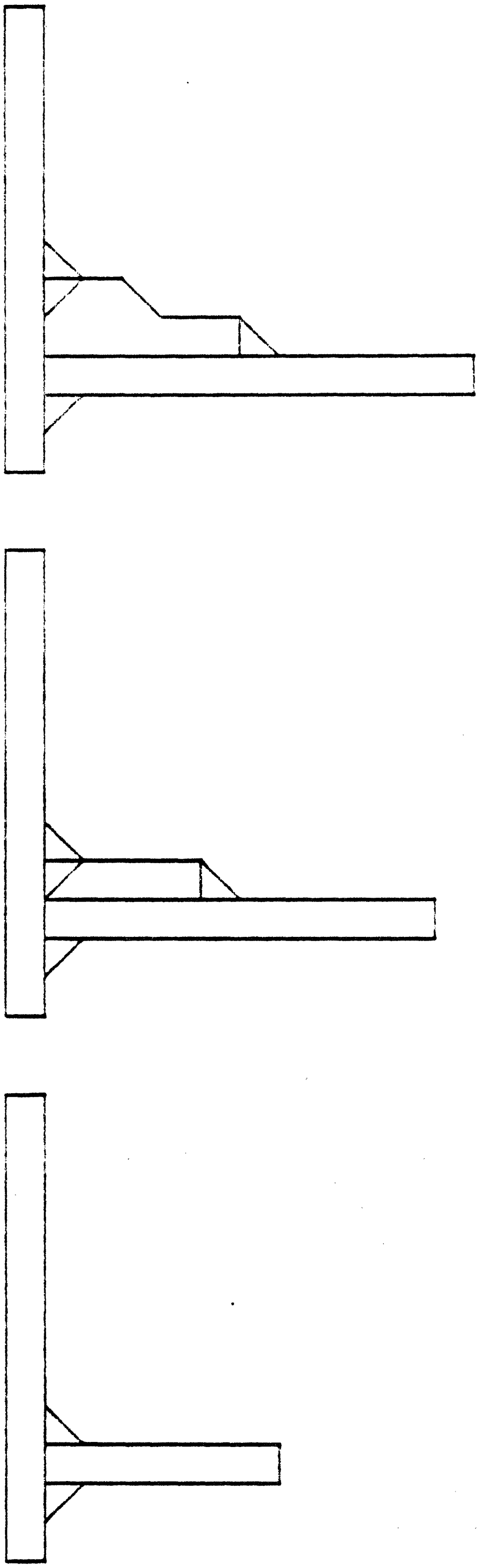


Figure 2: Details of Reinforcement for Pressure Vessel Nozzles

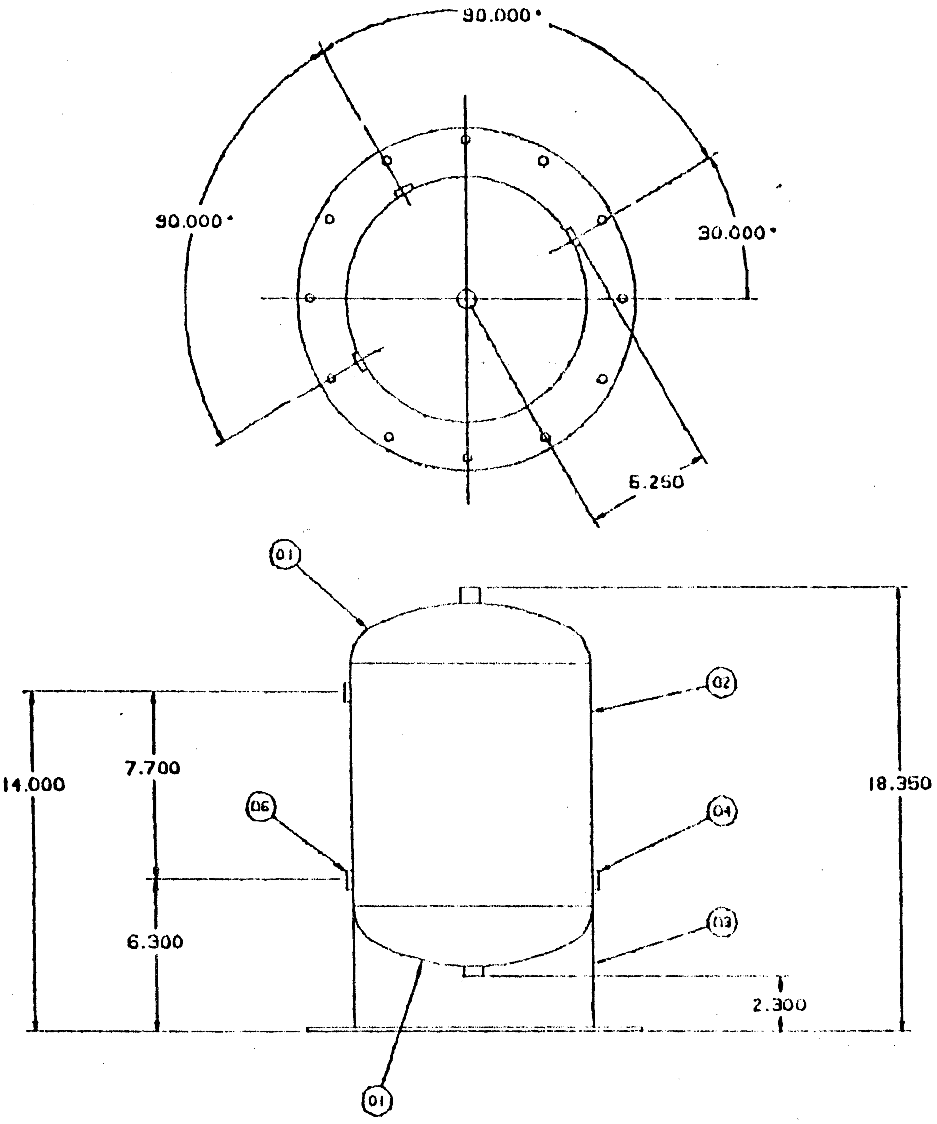


Figure 3: Pressure Vessel Outline Drawing

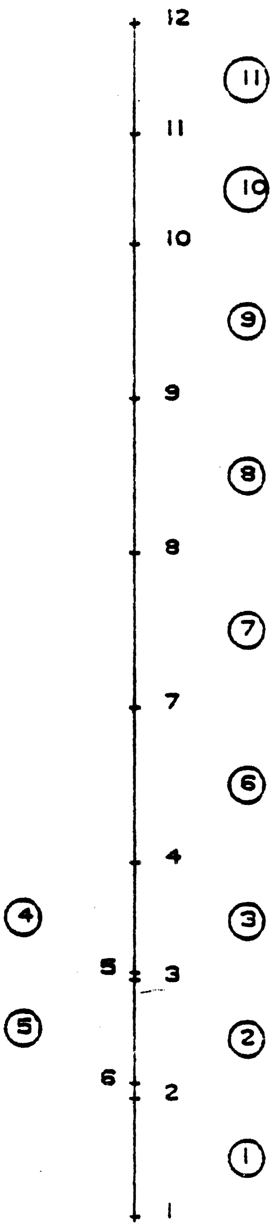


Figure 4: Simplified Finite Element Model

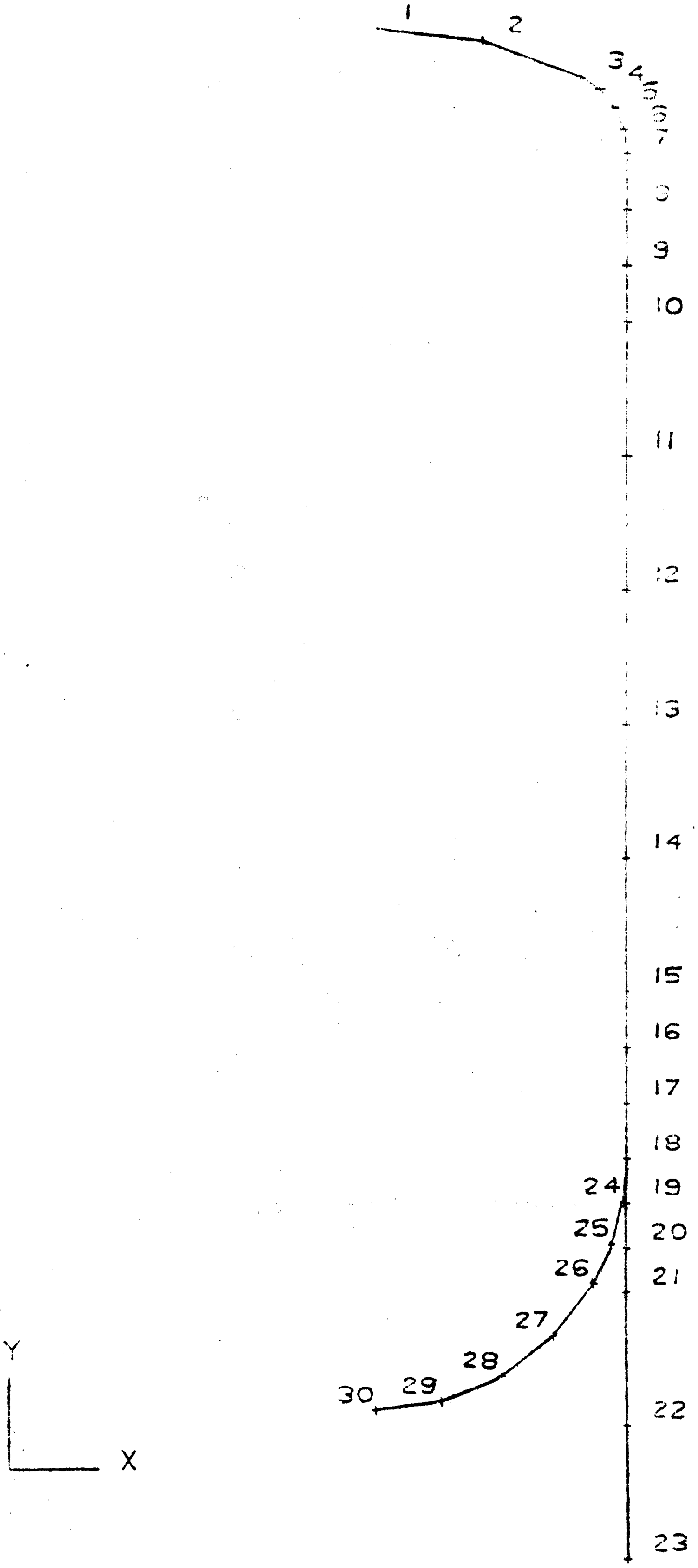


Figure 5: Axisymmetric Model

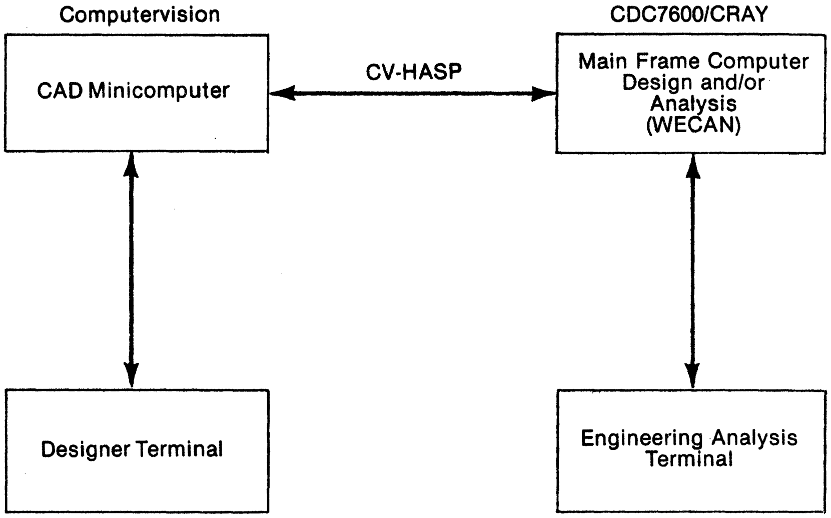

Computervision
CAD Minicomputer
CV-HASP
CDC7600/CRAY
Main Frame Computer Design and/or Analysis (WECAN)
Designer Terminal
Engineering Analysis Terminal

Figure 6: Integration of Computer Systems Via CV-HASP

PIPING DESIGN SYSTEM

Dave Wilkinson
H.A. Simons International Ltd.
425 Carroll Street
Vancouver, B.C.
Canada V6B 2J6
604-664-4315

1.0 INTRODUCTION

There are several 'Piping Design Packages' available on the Engineering Software Market today. Many of these packages involve complex design of a Piping Model including such options as pipe stress analysis, pipe sizing, and automatic pipe routing. At Simons, most of this work is done using manual procedures where the final product is a physical piping model. We undertake about 3 major projects a year, as well as several smaller projects. Major projects usually involve over 2000 piping isometrics, so a yearly throughput of over 6000 isos is not unlikely. These drawings will vary in size and complexity from large pumping systems to simple spooling diagrams.

Our objective in designing a 'Piping System' was to develop an automated procedure for converting the physical piping model into isometric drawings in a simple, fast, and economical manner. The basic development specifications for the system were the following:

1. The final output of the system must be a proper engineering document. The artwork must be acceptable to our client and be equal or superior to those drawings turned out manually by a draftsperson.
2. The system must be able to use a variety of standard component specification files. It must be able to handle both imperial and metric measurements.
3. The input of data must be oriented to the designer and draftsperson with little or no background in computers. The method of input must be logical and 'user friendly.'
4. The computing resources available are a Computervision CGP 200 graphics system and a DEC VAX 11/780 minicomputer.

2.0 SYSTEM DESCRIPTION

It was decided to use the DEC VAX system to do most of the 'bullwork' computing for this system. A set of eight programs were written to perform such tasks as data input, data file manipulation, conversion of data into graphics commands, creation of a Bill of Materials, and report generation.

PISO contains the following useful features:

1. Simplistic and logical method of data entry, for use by anyone who can read a piping model or a sketch from a designer.

2. Access to any Pipe Specification files which can be entered onto the VAX computer.
3. Creation of a Bill of Materials containing selected or all data regarding the piping system in a variety of formats using DEC's DATATRIEVE software package.
4. Automatic insertion of a drawing border containing drawing text which is entered before the user begins to 'draw' the isometric on the computer.
5. Ability to easily revise the drawing.
6. Automatic scaling of the drawing for readability.
7. Ability to easily transmit the drawing to other main frame or minicomputers which are foreign to DEC equipment.

Output from the VAX is a text file of CV graphics commands which is transferred to an execute file on the CV system. When the text file is executed, a drawing is produced which can be edited, plotted, or filed for later revision.

2.1 DATA ENTRY

The isometric data is entered into the system from the model or sketch using a program called PINPUT. A form is displayed on the screen and the user enters each component of the isometric by 'filling out' the form. Components are entered in a logical order which is determined by the user before activating the PINPUT. The user will enter the component name, size and location relative to the previous component. Drawing border text can be entered by activating a module within the input program. A component file is created by PINPUT and can be edited from within the program or by using a separate program called PEDIT.

The system allows for the user to enter the drawing in more than one sitting if the need arises. When one returns to complete the drawing, the model is automatically 'rebuilt' to the point where it was previously left off.

2.2 DRAWING CONVERSION

Two programs, DRAW and DUMPISO convert the drawing from a component file to an execute file of CV text commands.

DRAW performs two main functions. First, the coordinates of the components are converted from real space to drawing space. A de-scaling routine converts the distance between components into a more readable form, making especially long pipe lengths shorter, and vice versa. Secondly, pipelines are inserted into a model connecting the components.

DUMPISO takes the output data from DRAW and inserts it into CV text commands. Dimension insertion commands are generated by pipeline records. Commands to insert the drawing border and rotate the drawing are also generated by this progam.

2.3 REPORT PROGRAMS

Presently, all reports produced using the drawing data are generated using BASIC programs which are written only for the creation of specific reports. This method will soon be replaced by DATATRIEVE which is DEC's query language. If the user is familiar with the simple commands of this language, reports can be easily generated containing any items of data, sorted in any order.

2.4 EDITING THE DRAWING

PISO is not an intelligent drawing package, therefore there is the possibility of components being placed on top of each other, or dimensions being inserted in unrealistic positions. An operator should be available to touch up the drawing before it is filed or plotted.

When the drawing originally appears on the screen, the isometric and the border appear in 3D space. By putting the border in a separate port and flipping it to 2D, the operator can manipulate the isometric into the border. We intend to put a restriction of 20 percent on any zooming of the isometric so that there will be a fair degree of consistency in component size and dimension height among the drawings. User will take care of any 'touch up' work manually at the CV workstation.

Revisions should be done by entering changes into the edit program (PEDIT), thereby updating the component file. However, it may be easier to make small changes to the drawing at the graphics workstation and enter a group of them into the VAX at a later date.

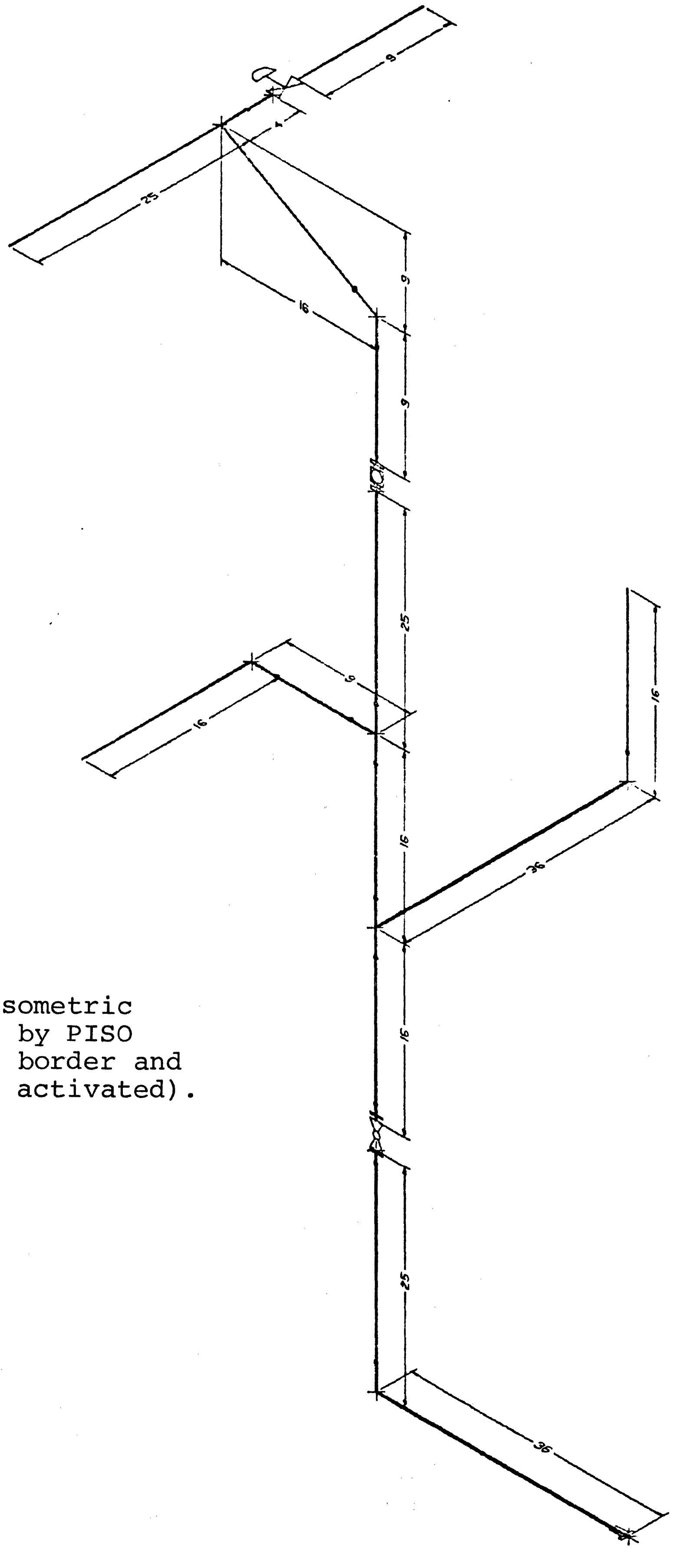
3.0 DEVELOPMENTS

PISO is still in the development stages but is now at the point where it could be used in a production mode. Development work is scheduled to continue for another six months with on-going maintenance. Two major developments will soon be implemented into the system. They are:

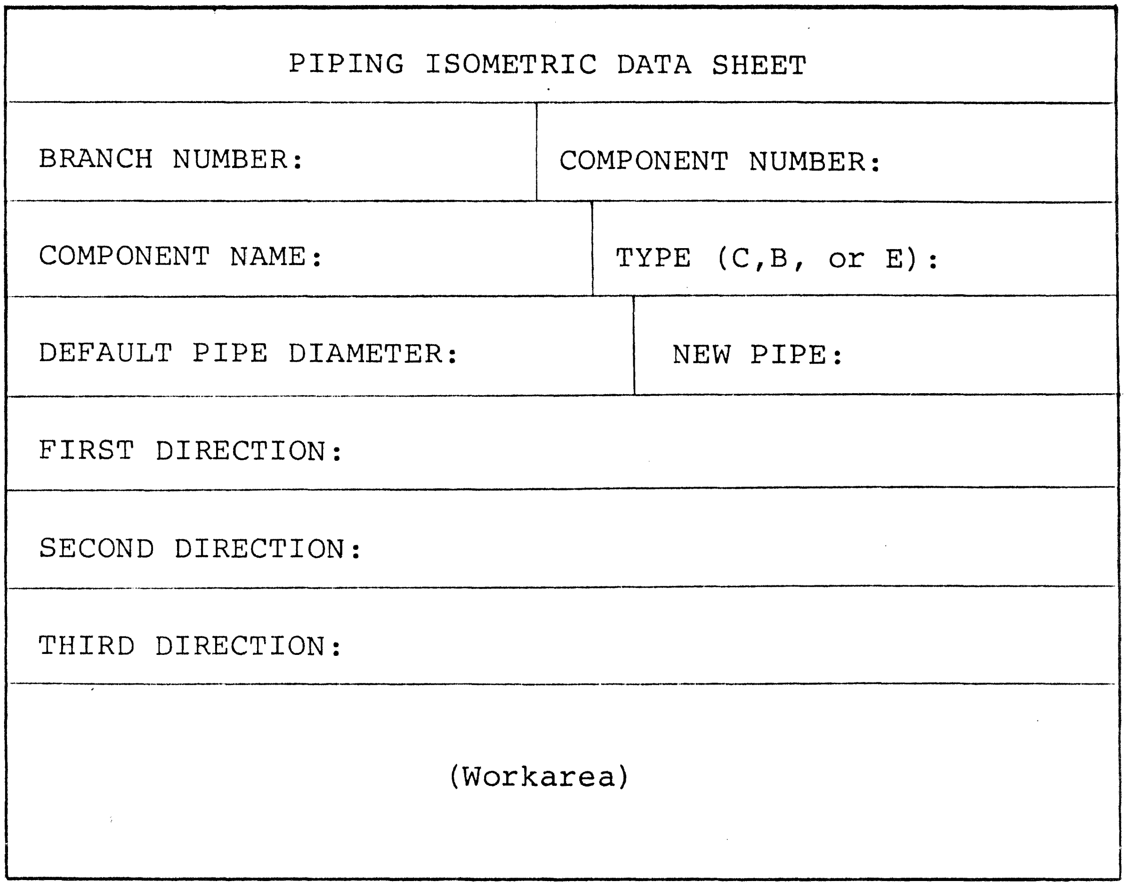
1. Automatic scaling of the drawing to fit onto the drawing border, while not de-scaling the components. This may require more than one pass through the ''DRAW'' program on the VAX to obtain the optimum scaling factor. This will provide a consistent component size throughout all the drawings.
2. A module is being added to input the data through a Tektronix CRT using a split screen. When the user enters the drawing, a 'wire diagram' representation will appear on the other half of the screen. This will eliminate the problem of 'blind drafting' and therefore reduce the amount of touch-up work to be done on the CV system.

The PISO system will be implemented on a small project in mid-1981 for final testing and proving for production. Development and improvement will continue with the addition of Manhours and Material Cost Report Modules being added to the set of programs.

PISO is seen as adding a powerful production feature to our in-house CADD ability.



Sample isometric produced by PISO (drawing border and text not activated).



PIPING ISOMETRIC DATA SHEET

BRANCH NUMBER:

COMPONENT NUMBER:

COMPONENT NAME:

TYPE (C,B, or E):

DEFAULT PIPE DIAMETER:

NEW PIPE:

FIRST DIRECTION:

SECOND DIRECTION:

THIRD DIRECTION:

(Workarea)

PISO Data Entry Screen Format

PLANT AND FACILITIES LAYOUT USING INTERACTIVE COMPUTER GRAPHICS

By
Gary L. Kahler
Harvey J. Marx
Western Electric Co.
Omaha Works Dept 735
P.O. Box 14000
Omaha, NE 6814
402-691-3808
402-691-3806

ABSTRACT

Plant and facilities layout using interactive computer graphics is a means of generating plant drawings using the speed, accuracy and power of a computer. This report describes the approach taken in its design and the structure imposed during its development to insure high quality, accurate, plant drawings.

INTRODUCTION

Plant and Facilities Layout play a major role in Interactive Computer Graphics at the Omaha Works of Western Electric Company. A well defined plan was developed to incorporate the existing Floor, Power and Service Layouts into an accurate, up-to-date graphics database. This plan was incorporated into a standard procedure at the Omaha Works and is proposed for other Western Electric locations with similar graphics equipment.

The plant, which consists of three major buildings, (two Shop and one Administration) was divided into equal parts with a suitable size for plotting and maintaining the database. Each part contains approximately 12,500 square feet of floor space. The graphic data for each part of the sub-divided plant consists of a Floor Layout, a Power Layout and a Service Layout, stored as a single file in the graphics database.

The single file concept saves time and disk space by creating single entities or inserting symbols from a parts library that may appear on any or all of three layouts. This eliminates redundancy and can be accomplished by special layering techniques.

An accurate, up-to-date database of the entire Plant is a primary goal. This will provide shop personnel, contractors and engineers with reliable, scaled drawings that can be easily modified.

Design alternatives are an important aspect of computer generated layouts. Ultimate design of plant facilities often requires several alternative approaches before a final version is approved. This design function can save considerable lead time for new, important projects.

GENERAL PROCEDURE

The main buildings of the Omaha Works are identified by number, such as Building 30. For multi-level buildings, the floor number follows the building number and is separated by a dash. Each building is divided into several rectangular subdivisions called parts which are sequentially numbered, beginning with (1) one. These parts are further divided into halves and given a suffix indicating either North, South, East or West half. An example is shown as "30.11W" which indicates Building 30, part 11 West.

Drawing forms are stored as a library part and consist of a border, title block, issue column and location key plan. General information contained in the title block is the company name and location, scale of drawing, building number, part number and the initials of the creator, checker and approvals. (See Appendix, Fig. 1). The issue column briefly describes changes to the layout. All prior revisions are retained for archival purposes. The location key plan shows the relative location of a part or half-part with reference to the building as a whole. Exact location is found by use of the column line nomenclature shown on two sides of the key plan. Crosshatching of the subject part or half-part gives a quick, visual reference to its location. (See Appendix Fig. 2, 3 & 4).

Information common to all three layouts includes a north arrow, building columns, match lines, peripheral aisle markings, dimensions pertaining to aisle and column centerlines and all restrooms. The north arrow is shown in Appendix, Fig. 5 and is located above the location key plan. When a part is rotated within the drawing form, a second north arrow is placed in the field of the drawing to indicate correct north orientation.

Building columns are drawn as three-line figures which represent the web and both flanges. They are drawn to scale of outside dimensions, grouped as one entity in the drawing and placed on a single specified layer.

FLOOR LAYOUTS

Plan view outlines of all equipment and facilities, including bench mounted capital equipment, are shown on the floor layout. Specific information about each different equipment item is specified in a callout and contains the following:

a) Name of equipment or facility.
b) Identification number.
c) Reference marks showing surfaces to be used for locating equipment at installation. (if applicable)
d) Drawing numbers of applicable installation drawings.
e) Operators normal work position.

Standard floor layout symbols are used to specify repetitive items such as partitions, channel guards, piperails, fire extinguishers, eye wash stations, water fountains, etc. These symbols are created and stored in a parts library.

POWER LAYOUTS

Only equipment and facilities requiring electrical services are shown on the power layout. The plan view outline of these items is shown with a callout which contains the following information:

a) Name of equipment or facility.
b) Identification number.
c) Drawing number of applicable installation drawings.

Electrical plug-in distribution bus ducts are shown on power layouts and are labeled as such. Power supply runs are shown from the electrical plug-in bus ducts to the point of service. This includes overhead and in-floor power runs.

These runs are also shown from the electrical circuit breaker panels to the point of use. Information contained in a callout for each power run are:

a) Square duct or conduit size
b) Number of wires
c) Gage of wires

Plan views of electrically serviced equipment and facilities show main components, such as control cabinets, main disconnect switches, motors, starters, heating elements, transformers, receptacles, etc. Wall or column mounted distribution panels show incoming service, service runs from point of distribution to point of use and a description of the distribution panel. Standard power symbols are used and stored in a parts library.

SERVICE LAYOUTS

Only the plan view of equipment and facilities requiring mechanical services are shown on the service layout. Associated with this equipment is a callout which contains the following:

a) Name of equipment or facility
b) Identification number
c) Drawing number of applicable installation drawings.

Service loops and supply piping runs from the service loop are shown, along with the service drops to the equipment. Information about these runs are shown in a callout and include:

a) Pipe sizes
b) Drop sizes
c) Elevations of horizontal runs
d) Pitch of runs, if applicable

Return and drain piping from the equipment or facility to the return loop is shown along with the same callouts. Standard symbols for piping items are created and stored in a parts library.

GRAPHIC REPRESENTATION

Graphic representation of the individual layouts within a single file are determined by layering procedures. Line weights and plots of specific layouts are also determined by layering. Floor layouts, power layouts and service layouts occupy specific layers as described below and shown in chart form in Appendix Fig. 6 & 7. The following describe and specify layout layering and plotting procedures:

Layers 10 through 19 are reserved for entities which appear on floor layouts only. Equipment and facilities that do not have electrical or mechanical service requirements are shown on Layer 12 and plotted with a medium line weight pen. Fine lines and dashed lines are on Layer 11.

Layers 9 through 49 are reserved for entities which appear on all three types of layouts. Equipment and facilities that require both mechanical and electrical services are shown on Layer 42, and plotted with medium line weight. Fine lines and dashed lines are on Layer 41.

Layers 50 through 59 are reserved for entities which appear on floor layouts and power layouts but not service layouts. Equipment and facilities requiring electrical services only are shown on Layer 52 and plotted with medium line weight. Fine lines and dashed lines are on Layer 51.

Layers 60 through 69 are reserved for entities which appear on floor layouts and service layouts, but not power layouts. Equipment and facilities requiring mechanical services only are shown on Layer 62 and plotted with medium line weight. Fine lines and dashed lines are on Layer 61.

Text associated with entities on Layers 10 through 18 is placed on Layer 19.
Text associated with entities on Layers 40 through 48 is placed on Layer 49.
Text associated with entities on Layers 50 through 58 is placed on Layer 59.
Text associated with entities on Layers 60 through 68 is placed on Layer 69.
When possible, text parameters are Font 1, Height .1 and width deferred.

PLANT LAYOUT FILE STRUCTURE

Each plant layout file on the Interactive Graphics System contains the data required to produce a Floor Layout, Power Layout and Service Layout for a single building part or half part. The plant layout file name consists of the building number, part or half part number and the issue numbers for the floor layout, power layout and service layout drawings. The files are stored in catalog form with each of the aforementioned items being treated as a subcatalog. See Appendix, Fig. 8.

Off-line storage of plant layout files are stored on magnetic tapes. These tapes are labeled with an identification number followed by a dash and then a reel number. Individual files of a plant layout are located by referring to a drawing cross reference card which identifies the magnetic tape, the reel number and the relative file location on the reel. See Appendix, Fig. 9.

RESULTS

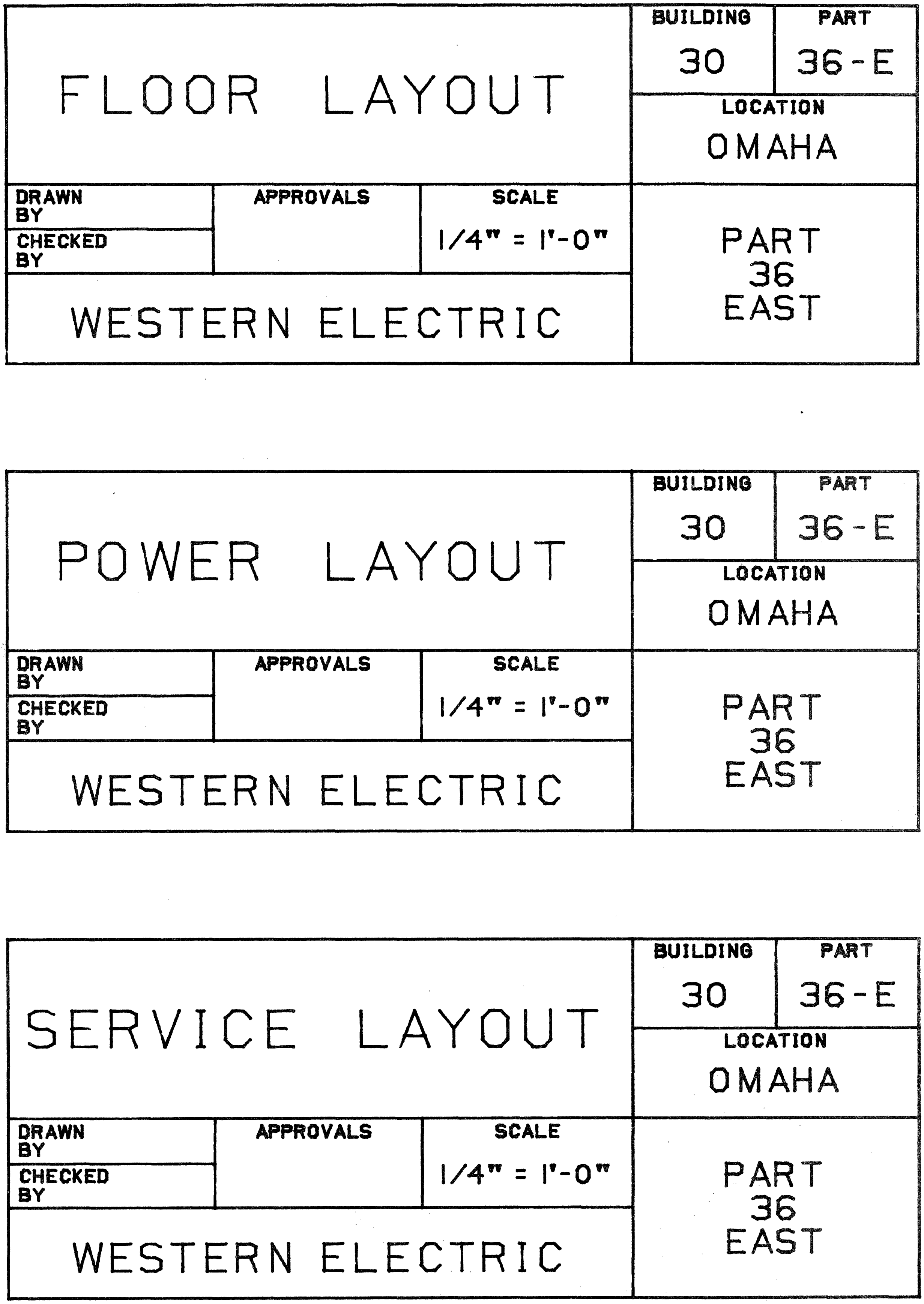
In final analysis, plant and facilities layout can be made easier, faster and more efficiently, using Interactive Computer Graphics. Many tangible and intangible benefits are the result of a well developed program. An accurate up-to-date set of plant and facility drawings save time and effort in plant rearrangement or incorporation of new products, at a time when short lead times are critical.

At Omaha, revisions, additions or complete renovation to a manufacturing or office area may be made by accessing only one database file. Changes made to any one of the three layouts in this file allow revising the remaining two layouts with minimum effort. Revisions may then be visually checked and cross-referenced from one layout to another to insure accuracy and completeness. This concept minimizes the time and effort required to generate new drawings.

Storage mediums are kept to a minimum with all data stored on, and retrieved from, magnetic tape. Fast accurate, plots and checkprints are made possible with the use of an electro-static dot matrix printer. Dot plots are used for final drawings and reproduction masters.

In summation; plant and facilities layouts, using Interactive Computer Graphics at the Omaha Works, has been a viable and justifiable process.

APPENDIX



FLOOR LAYOUT			BUILDING 30	PART 36-E
			LOCATION OMAHA	
DRAWN BY	APPROVALS	SCALE 1/4" = 1'-0"	PART 36 EAST	
CHECKED BY				
WESTERN ELECTRIC				

POWER LAYOUT			BUILDING 30	PART 36-E
			LOCATION OMAHA	
DRAWN BY	APPROVALS	SCALE 1/4" = 1'-0"	PART 36 EAST	
CHECKED BY				
WESTERN ELECTRIC				

SERVICE LAYOUT			BUILDING 30	PART 36-E
			LOCATION OMAHA	
DRAWN BY	APPROVALS	SCALE 1/4" = 1'-0"	PART 36 EAST	
CHECKED BY				
WESTERN ELECTRIC				

Figure 1

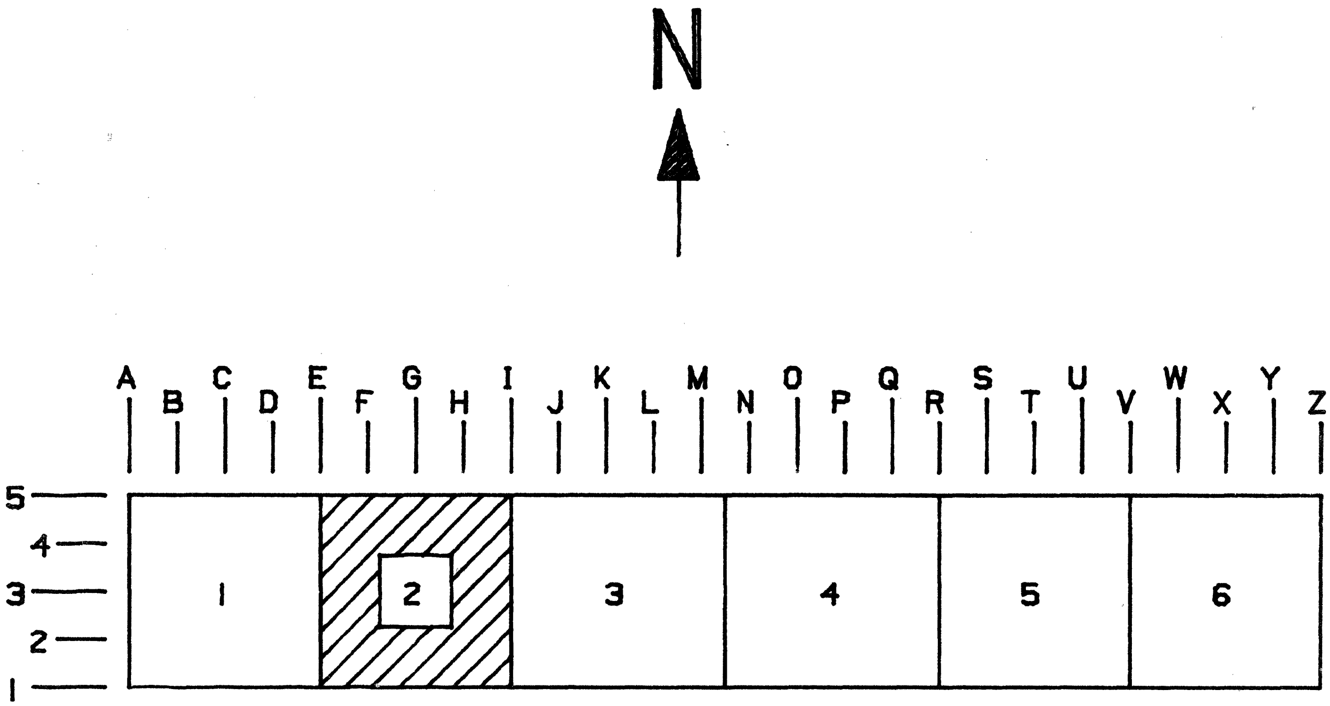


Figure 2

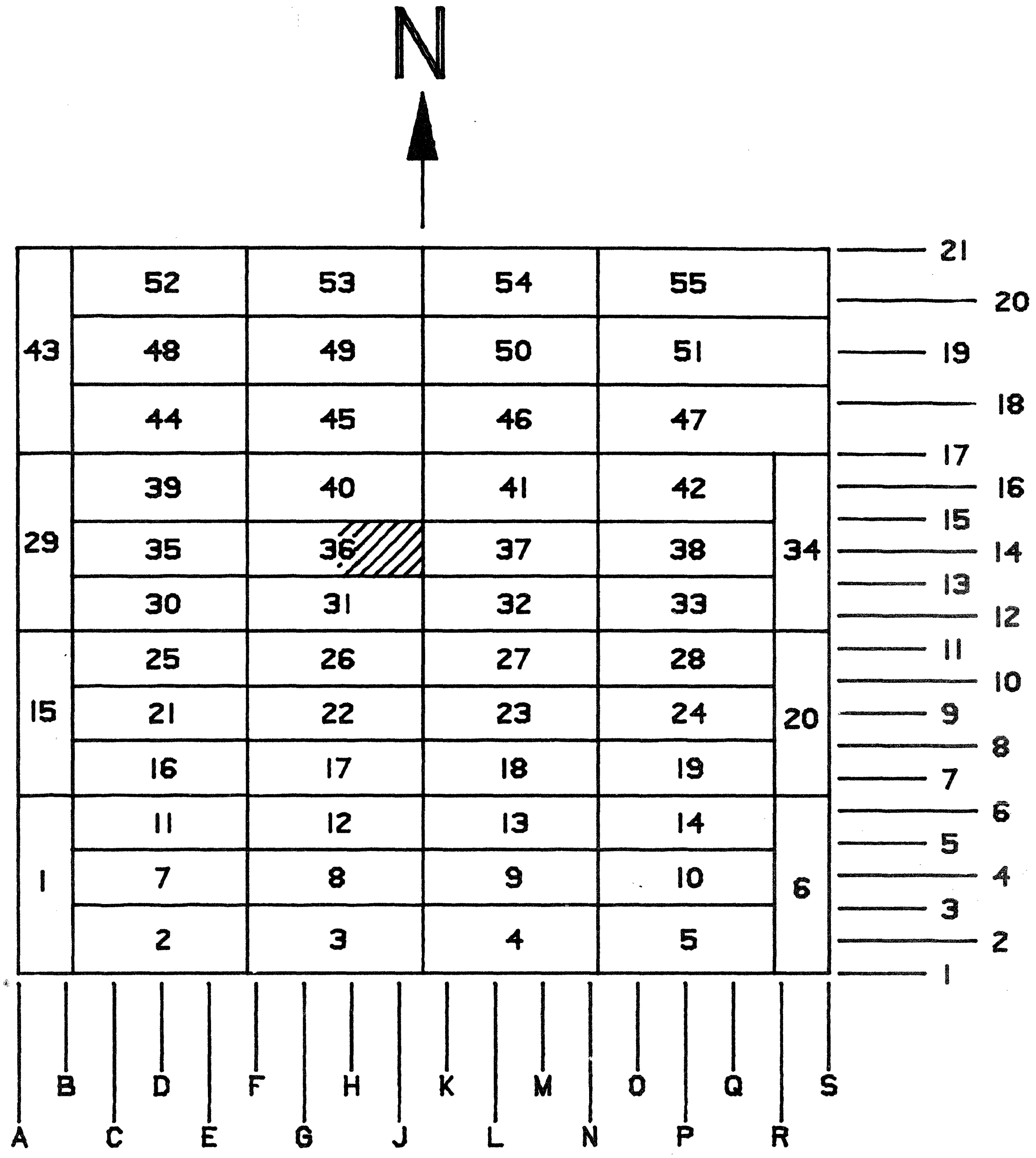

N
52
53
54
55
43
48
49
50
51
44
45
46
47
39
40
41
42
29
35
36
37
38
34
30
31
32
33
25
26
27
28
15
21
22
23
24
20
16
17
18
19
11
12
13
14
1
7
8
9
10
6
2
3
4
5
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
B
D
F
H
K
M
O
Q
S
A
C
E
G
J
L
N
P
R

Figure 3

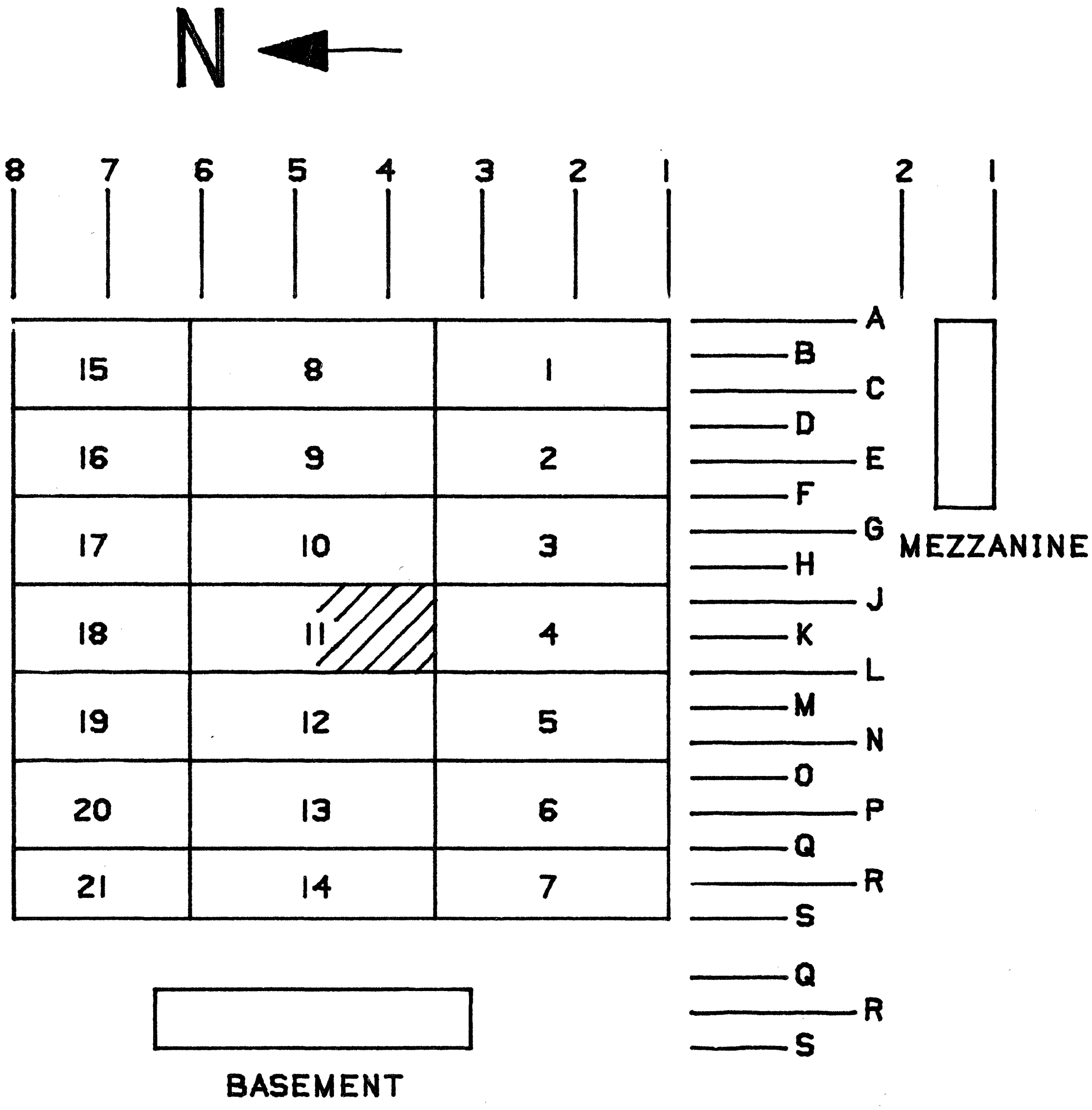


Figure 4

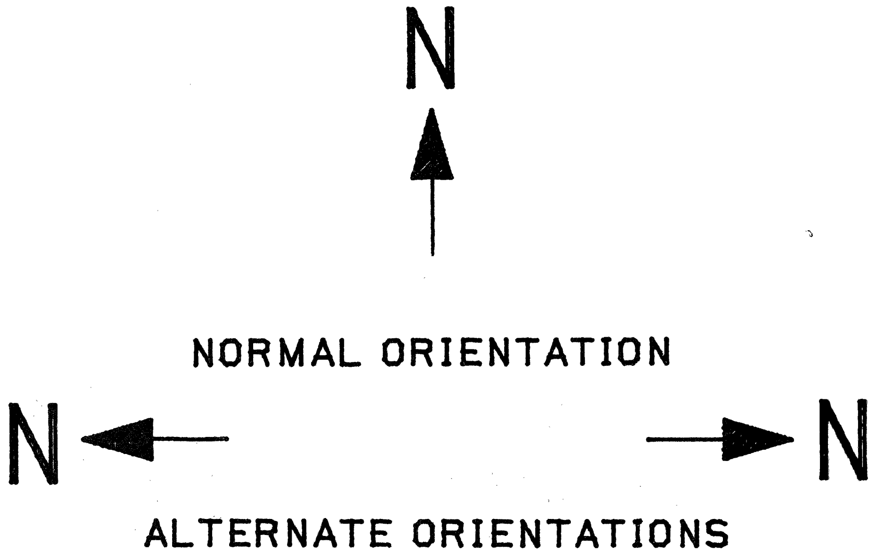


Figure 5

LAYER	CONTENT	LINE WEIGHT		
		FLO	PLO	SLO
0 - 9	FORM AND PART DESCRIPTION			
0	FORM	M	M	M
1	DIMENSIONS AND AISLE C'S	L	L	L
2	AISLE AND PART BOUNDARIES, WALLS	M	M	M
3	MATCH LINES, COLUMNS	H	H	H
4	RESTROOMS	M	L	L
5 - 9	UNASSIGNED			
10-19	FLO ONLY			
10	UNASSIGNED			
11	OPTIONAL LAYER FOR DASHED OR DETAILED	L		
12	PREFERRED LAYER FOR ALL OTHER LINEWORK	M		
13-18	UNASSIGNED			
19	TEXT ASSOCIATED WITH FLOOR LAYOUTS	L		
20-29	PLO ONLY			
20	BUS DUCTS		M	
21	OPTIONAL LAYER FOR DASHED OR DETAILED LINEWORK WHEN LIGHT LINE OUTPUT IS DESIRED		L	
22	SYMBOLS		M	
23	POWER RUNS		H	
24-28	UNASSIGNED			
29	TEXT ASSOCIATED WITH POWER LAYOUTS		L	

BUILDING LAYOUT LAYERING PROCEDURES AND
ASSOCIATED PLOTTING LINE WEIGHTS

Figure 6

(continued)

LAYER	CONTENT	LINE WEIGHT		
		FLO	PLO	SLO
30-39	SLO ONLY			
30	SERVICE LOOP PIPING			H
31	OPTIONAL LAYER FOR DASHED OR DETAILED LINEWORK WHEN LIGHT LINE OUTPUT IS DESIRED			L
32	SYMBOLS			M
33	SERVICE RUNS			H
34-38	UNASSIGNED			
39	TEXT ASSOCIATED WITH SERVICE LAYOUTS			L
40-49	FLO, PLO, AND SLO			
40	UNASSIGNED			
41	OPTIONAL LAYER FOR DASHED OR DETAILED WHEN LIGHT LINE OUTPUT IS DESIRED	L	L	L
42	PREFERRED LAYER FOR ALL OTHER LINEWORK	M	L	L
43-48	UNASSIGNED			
49	TEXT ASSOCIATED WITH FLOOR, POWER, AND SERVICE LAYOUTS	L	L	L
50-59	FLO AND PLO ONLY			
50	UNASSIGNED			
51	OPTIONAL LAYER FOR DASHED OR DETAILED LINEWORK WHEN LIGHT LINE OUTPUT IS DESIRED	L	L	
52	PREFERRED LAYER FOR ALL OTHER LINEWORK	M	L	
53-58	UNASSIGNED			
59	TEXT ASSOCIATED WITH FLOOR AND POWER LAYOUTS	L	L	

Figure 6 (continued)

LAYER	CONTENT	LINE WEIGHT FLO	LINE WEIGHT PLO	LINE WEIGHT SLO
60-69	FLO AND SLO ONLY			
60	UNASSIGNED			
61	OPTIONAL LAYER FOR DASHED OR DETAILED LINEWORK WHEN LIGHT LINE OUTPUT IS DESIRED	L		L
62	PREFERRED LAYER FOR ALL OTHER LINEWORK	M		M
63-68	UNASSIGNED			
69	TEXT ASSOCIATED WITH FLOOR AND SERVICE LAYOUTS	L		L

Figure 6 (continued)

LINE WEIGHT	PEN SIZE
L (light)	.2 mm
M (medium)	.3 mm
H (heavy)	.4 mm

LAYOUT TYPE	LINE WEIGHT	LAYERS
FLO	L	1, 11, 19, 41, 49, 51, 59, 61, 69
	M	0, 2, 12, 42, 52, 62
	H	3
PLO	L	1, 21, 29, 41, 42, 49, 51, 52, 59
	M	0, 2, 20, 22
	H	3, 23
SLO	L	1, 31, 39, 41, 42, 49, 61, 62, 69
	M	0, 2, 30, 32
	H	3, 33

BUILDING LAYOUT PLOTTING SPECIFICATIONS

Figure 7

GENERAL FORM

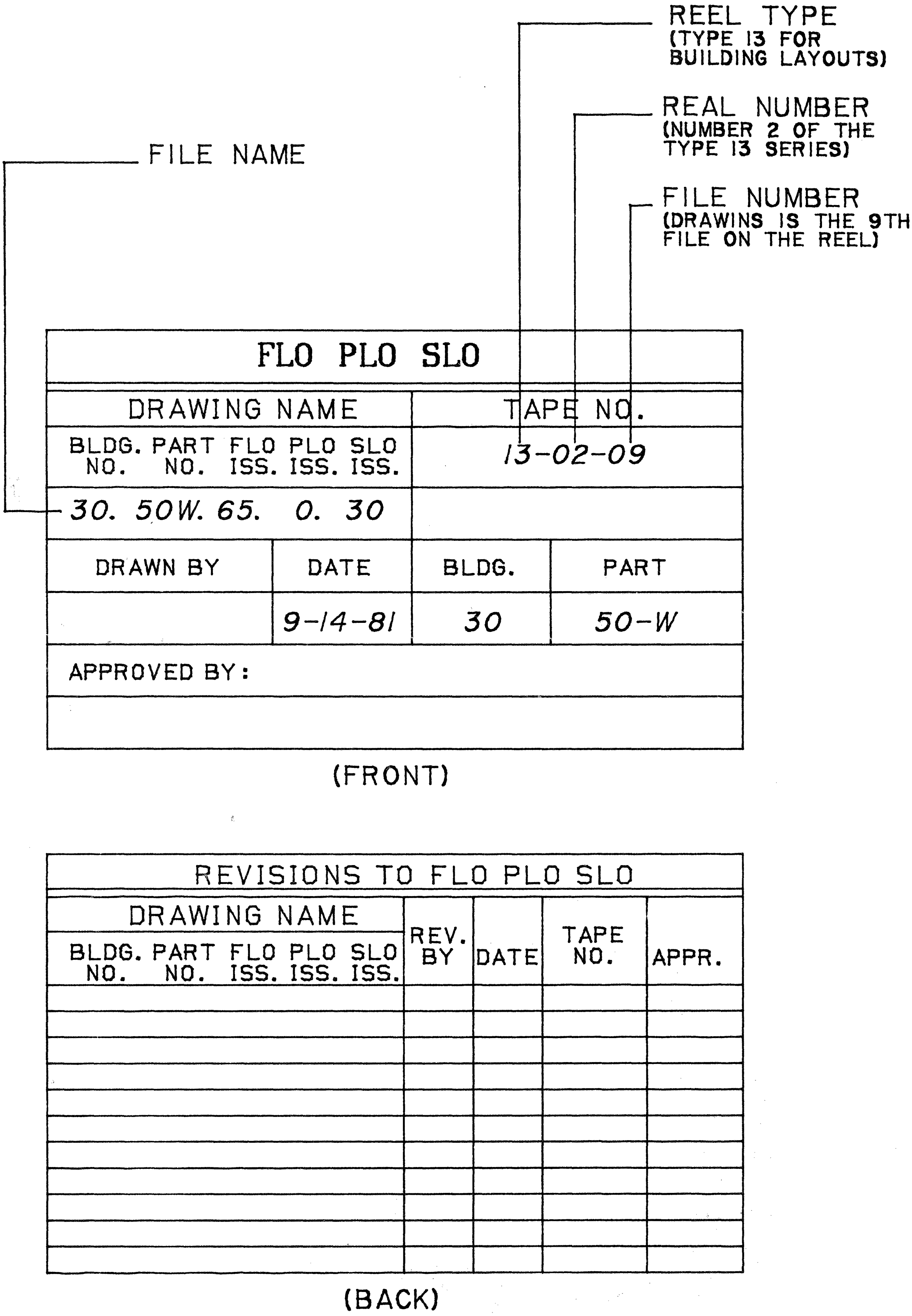
```
                  CD3PRTS.   ##.  ###.  ##.  ##.  ##
                     |        |     |     |    |    |
CADDS 3 PARTS -------+        |     |     |    |    |
                              |     |     |    |    |
BUILDING NUMBER --------------+     |     |    |    |
                                    |     |    |    |
PART NUMBER ------------------------+     |    |    |
                                          |    |    |
FLO ISSUE NUMBER -------------------------+    |    |
                                               |    |
PLO ISSUE NUMBER ------------------------------+    |
                                                    |
SLO ISSUE NUMBER -----------------------------------+
```

EXAMPLE

```
                  CD3PRTS.   30.  36E.  14.  10.  11
                     |        |     |     |    |    |
CADDS 3 PARTS -------+        |     |     |    |    |
                              |     |     |    |    |
BUILDING 30 ------------------+     |     |    |    |
                                    |     |    |    |
PART 36E ---------------------------+     |    |    |
                                          |    |    |
FLO ISSUE 14 -----------------------------+    |    |
                                               |    |
PLO ISSUE 10 ----------------------------------+    |
                                                    |
SLO ISSUE 11 ---------------------------------------+
```

BUILDING LAYOUT FILE NAME DESCRIPTION

Figure 8



REEL TYPE (TYPE 13 FOR BUILDING LAYOUTS)

REAL NUMBER (NUMBER 2 OF THE TYPE 13 SERIES)

FILE NUMBER (DRAWINS IS THE 9TH FILE ON THE REEL)

FILE NAME

FLO PLO SLO			
DRAWING NAME		TAPE NO.	
BLDG. NO. PART NO. FLO ISS. PLO ISS. SLO ISS.		*13-02-09*	
30. 50W. 65. 0. 30			
DRAWN BY	DATE	BLDG.	PART
	9-14-81	*30*	*50-W*
APPROVED BY:			

(FRONT)

REVISIONS TO FLO PLO SLO								
DRAWING NAME					REV. BY	DATE	TAPE NO.	APPR.
BLDG. NO.	PART NO.	FLO ISS.	PLO ISS.	SLO ISS.				

(BACK)

Figure 9

THE PECULIARITIES AND POSSIBILITIES OF CAD IN SMALL ARCHITECTURAL FIRMS

Peter Selby
Computervision Corporation
1040 Uxbridge Road
Hayes, Middlesex, England
1-561-2626

ABSTRACT

The impact of CAD on two modestly-sized European firms of architects and engineers points the way to a promising market for smaller CV systems, such as the two-terminal Designer M. While the architectural field may never seriously challenge the giants of PC and MD, it certainly deserves some attention to its peculiarities and possibilities.

Already a Designer V is well established at Van Kerckhove, a Belgian firm of some 30 employees specializing in factories, schools, and hospitals. In the Netherlands, a Designer IV has been installed by Haskoning, whose staff of several hundred are involved primarily in large government contracts and planning projects.

This paper reflects to some degree the ideas exchanged, and the cooperative relationship existing between Computervision and these two pilot customers. Our objective is to clarify: 1) what CAD requirements are peculiar to architects, 2) how a firm's productivity can be increased by CAD, and 3) what implications all this might have in the design of interfaces and databases.

There are, I think, three principal areas in which CAD can help the architect: production information, design evaluation, and cost control.

PRODUCTION INFORMATION

Obviously, there is a need for good communication between designer and fabricator in architecture as well as in other industries. This demands correct and comprehensive descriptions of the proposed building — traditionally in the form of working and detail drawings, and specifications of required materials and workmanship.

Initially, CAD provides the architect with automated drawing capabilities, as well as the ability to create standard libraries of components and constructional details. Included in the libraries might be standard ranges of windows, doors or sanitary fittings, or generic classes of components such as structural steel sections or wall construction types. These standard library components would be parameterized on insertion into the building model. Clearly, CAD can assist the architect in creating correct and comprehensive working and detail drawings.

A further enticing prospect is the possibility of better co-ordination of production information. Today, architects work with increasingly complex building types. Often the design process is a multi-disciplinary affair, involving teams of professionals with diverse engineering specialties. It is vitally important to coordinate the various disciplines in order to eliminate costly conflicts and ensure that any change made to one facet of the design is reflected in, and accounted for, in all other affected areas of the project. CAD's single-source database — a unique mastercopy of the building — ensures that change is consistent throughout all drawings and documentation.

DESIGN EVALUATION

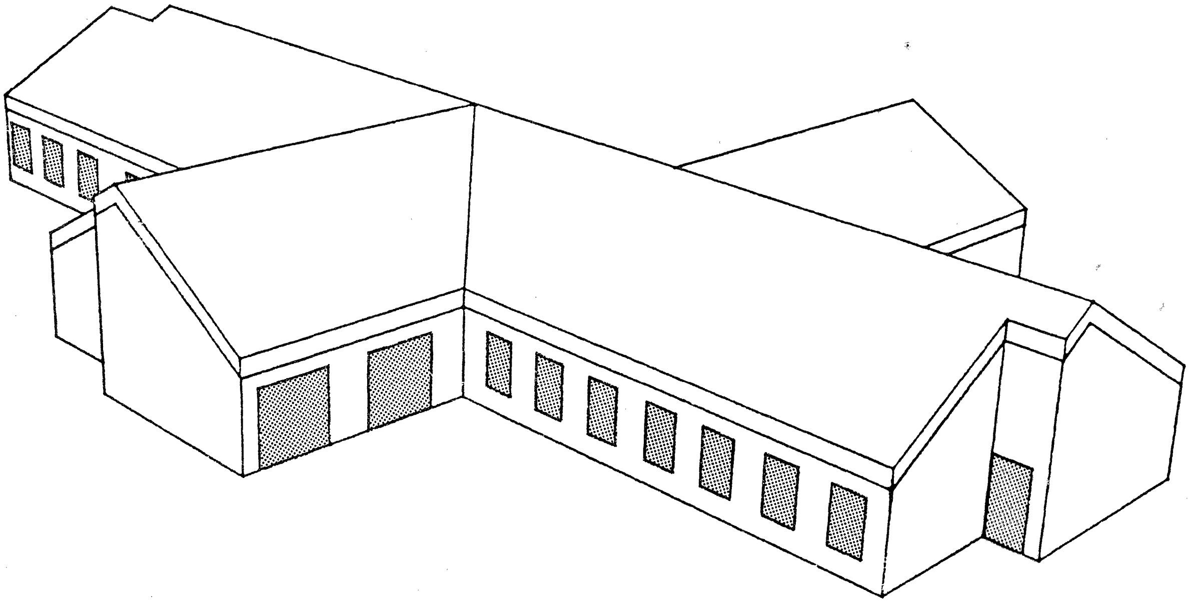
This process involves asking the CAD system to predict how a given design is likely to perform after it becomes a completed building. Design evaluation can be enhanced by CAD-generated interior or exterior perspectives, with or without hidden line removal, and environmental simulation. In real time, the system can graphically display and mathematically analyze the effect of daylighting factors, heat loss and solar gain, and sunpatch distribution, even predicting the fall of sunlight through windows onto interior surfaces.

CAD can also analyze value factors much harder to visualize and quantify — for example, the circulation of people and the building population.

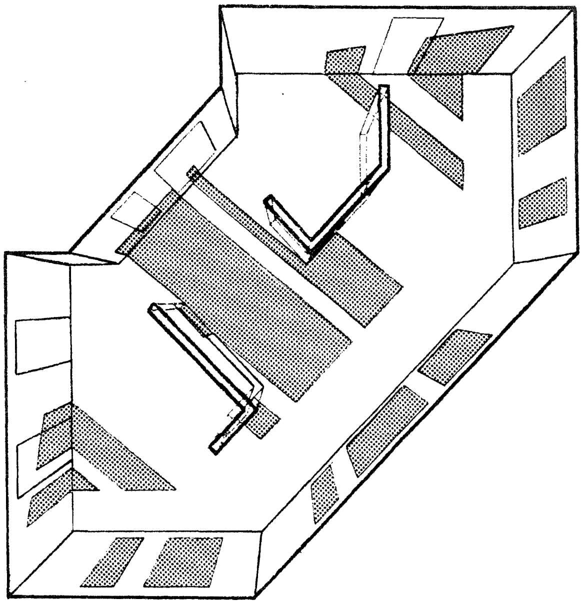
COST CONTROL

This is closely allied to production information and design evaluation, previously discussed. For one thing, cost control relates to the total building design process. It can be considred part of design evaluation since at the early design stage, CAD techniques can be used to roughly estimate the cost, as well as the ultimate value, of a commercial building — for instance, in terms of rental area.

Cost control also involves production information since bills of materials can be used as a basis for very accurate and detailed forecasting. In cost control, CAD makes its greatest contribution in the measurement of the building fabric. Done manually, even with calculators, this could be an error-prone and time-consuming process. In contrast, this trivial and repetitive procedure is ideally suited to a CAD system, which stores all the correct information required within the building model.



An exterior perspective of a health centre with hidden lines removed



A vertical perspective of a room showing the patches of sunlight from the windows at a specified time and place

OVERVIEW OF THE BUILDING PROCESS

Let us take an overview of the building process and see how it can be radically changed by CAD techniques. The task of designing a building can be broken down into four separately identifiable parts: 1) sketch design, 2) detailed design, 3) production information, and 4) on-site supervision.

Traditionally, the major portion of manpower on a project is allocated to the production information stage. Technicians and draftsmen must generate voluminous production information to meet contractual schedules. This produces a pronounced peaking of the manpower curve, as shown in Graph 1. At Van Kerckhove, the manpower ratio between the design stage and the production information stage is always at least two to one.

Yet the most important design decisions are made much earlier, during sketch design and, to a lesser extent, during detailed design. This is when strategic decisions are made about overall building form, siting, and orientation — decisions with far-reaching cost implications.

On the other hand, the production information stage involves decisions of a much more tactical, small-scale and local nature — for example, the selection of finishes or the choice of staircase balustrading. On Graph 2, notice that the decision curve and the manpower curve are way out of sync. In fact, there is almost an inverse correspondence between decision level and allocated manpower.

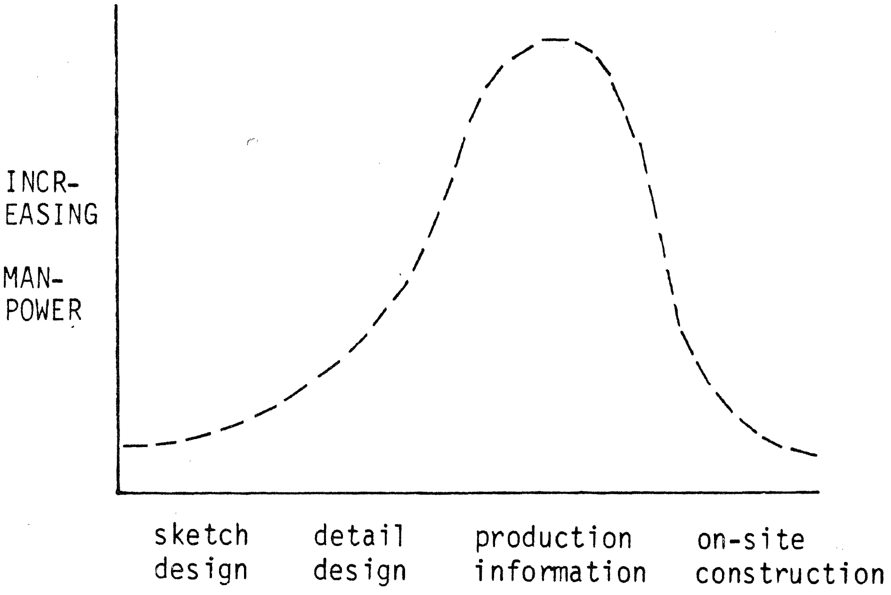
THE IMPACT OF CAD

When CAD is introduced at the production information stage, it dramatically flattens the manpower curve, as shown in Graph 3. Even if CAD is used at this stage only, the results can be significant. A company can complete projects more quickly with less manpower, and more smoothly manage the on-site stage through better co-ordinated information. The resultant productivity increases can yield substantial financial gains to the CAD user.

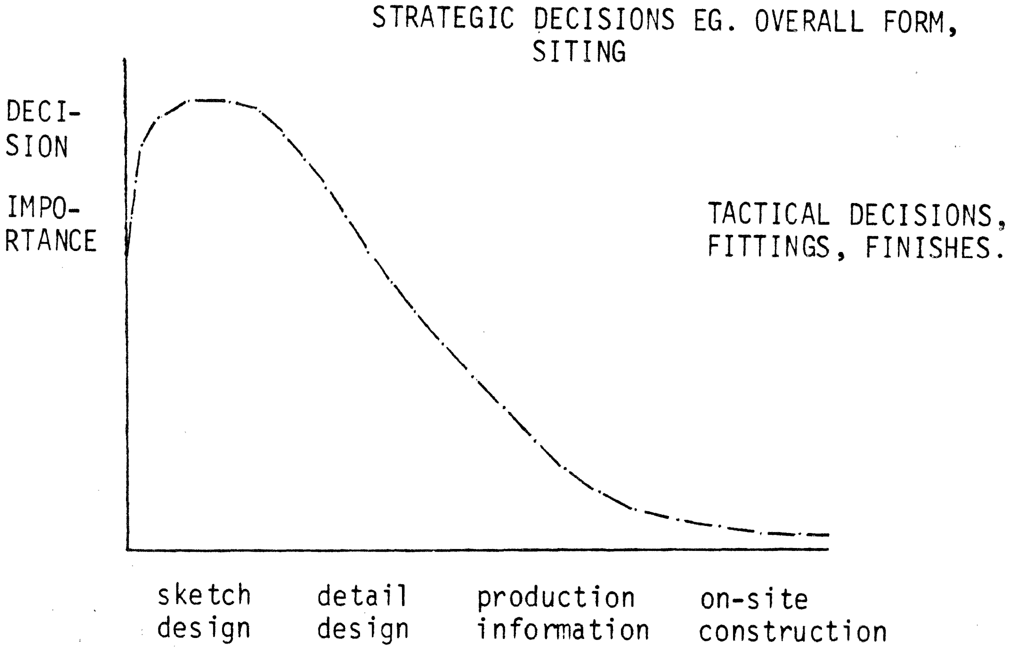
What's more, the company can now re-allocate manpower from the production stage back to the critical decision-making stages. Many more design alternatives can be researched than before, enabling architects to base strategic decisions on much improved design performance information. Thus buildings of substantially better value-for-the money can be produced.

Graph 4 illustrates the close match that now exists between the manpower resource curve and the decision curve. CAD has fundamentally shifted the resource allocation within a design project and, incidentally, has produced a much more highly motivated staff.

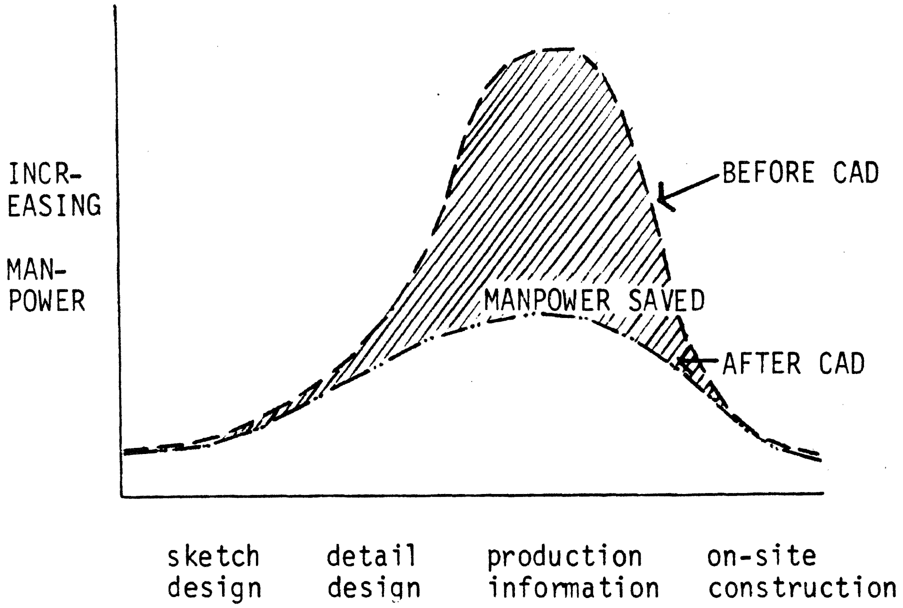
Bear in mind that all this has been achieved by simply applying CAD to the production stage. If we also use CAD for design evaluation and cost forecasting purposes at the early design stages, building quality and performance can improve significantly. It becomes possible to evaluate an even wider range of design alternatives, as well as to ask — and answer — questions about the design which previously were not even feasible to raise. The advanced computer graphics such as CADDS4 can provide ensures that clients of an architect can be much better informed about what they are getting, and be able to participate in early decision making about the building.



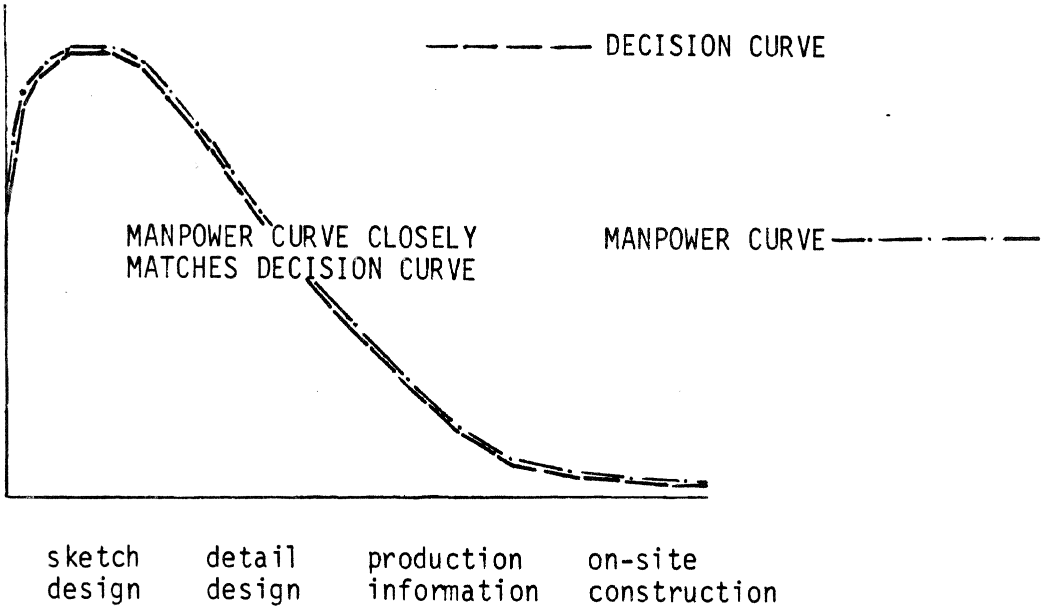
Graph 1: Manpower Resource Curve



Graph 2: Design Decision Curve



Graph 3: CAD Applied to the Production Information Stage



Graph 4: With Manpower Re-allocation

IMPLICATIONS ON DATABASE DESIGN

Finally, let us consider what implications the foregoing discussions have on the design of interfaces and databases. The special needs and problems of architects must be understood by CAD software designers. Typically, architects are more likely to think in purely graphic terms, and are less accustomed to communicating in mathematical languages than their MD counterparts. Architects have a keen awareness of 3D form, but are more apt to conceptualize new graphic images with reference to other graphic images rather than, say, coordinate systems with certain angular rotations.

To generate more than straightforward production drawings or simple graphic representations of components, an architectural database will probably need much more relational, associative data than is usually the case. This is certainly true if the architect must obtain complex area or volumetric measurements of the building fabric for cost control purposes, or for design evaluations of rooms and spaces within the building.

Analyses like these require a building model that contains intelligent descriptions of complex polyhedra — not just where entities are located in space, but also what they are logically connected to, and what their neighbors are. Quite possibly some of the work in CV's mapping and piping packages can serve as a basis for developing a more intelligent and useful architectural database.

Perhaps the most difficult problem of all will lie in the mode of connecting a suitable interface to the intelligent database. How will the required associative data be captured? How will solid models be derived? and interpreted from user input? This phase in the creation of an architectural applications package will surely require the most intense research.

CONCLUSIONS

Clearly, architectural productivity can be significantly enhanced by the application of CAD techniques. Obviously, the size of the system selected by an architectural firm will be heavily influenced by the firm's own size. Most such firms, in Europe at least, are modest in size, and likely to find a two-task system their reasonable limitation. Hence Designer M should prove a cost-effective investment in a majority of cases.

Operations Management

INITIAL GRAPHICS EXCHANGE SPECIFICATION (IGES): A PERSPECTIVE

Albert J. Gibbons
Senior Engineer
Westinghouse Electric Corporation
Pittsburgh, Pennsylvania

ABSTRACT

At Westinghouse a translation was written to convert files of Computervision (CV) CADDS3 part geometry into Initial Graphics Exchange Specification (IGES) output files. The first phase of the translation, written in early 1981, converts a subset of CADDS3 data including points, lines, circular arcs, and conic arcs. Two of the translator objectives are fast translation and easy portability to other computer systems. Implementation of the translator and service on three of the IGES committees has provided the opportunity to observe and evaluate IGES and its usefulness to Westinghouse (and other companies) on a long range basis.

INTRODUCTION

Westinghouse is diverse in activity and geographic location, as well as large in size with a multitude and variety of functional resources. Many computer aided design and drafting (CAD) systems, and an amount of equipment for computer aided manufacturing (CAM), supplied by an assortment of vendors, are used at plant sites throughout the country. These CAD systems are important both in designing new products and equipment and in modifications to existing designs. Several corporate data centers, with powerful hardware and sophisticated software, have been established to assist in the interpretation and analysis of the vast data generated by routine business and engineering activity including the structural analysis of new or modified equipment designs. Product definition data often must be transmitted from one location to another for purposes of manufacturing, subsequent design steps, analysis, or long term storage. Because of growth in the amount of accumulated information, establishment of additional Westinghouse locations, and installation of additional CAD/CAM systems, greater accessibility and portability of the available information will be the key to generation of improved products and methods at reduced costs.

For Westinghouse the three major reasons for transferring product definition data are: (1) to serve as input to the Westinghouse structural analysis system, (2) to allow exchange of information within Westinghouse (or to and from Westinghouse suppliers and customers), and (3) to provide long term recoverable storage of the data in a neutral database (independent of any specific vendor). Input to the Westinghouse structural analysis system requires neither annotation (dimensioning and notes) nor precise geometry. For example, several circular arc segments could be used to approximate an ellipse or a spline without introducing significant errors. When information is transferred to another location for use there or for long term (archival) storage, however, accuracy and completeness are essential to preserve the original design intent and to facilitate duplication of the part or creation of other adjacent parts. Because of retention requirements, which may be as long as 50 to 100 years, the use of an independent, neutral, standard form for representing the data is important.

HISTORY OF IGES

To date, there is only one neutral database specification for the representation of product definition data, the Initial Graphics Exchange Specification (IGES). The need for a standard for communication of drawing and geometry information between CAD/CAM systems surfaced in September of 1979 at a meeting of the Department of Defense (DOD) Manufacturing Technology Advisory Group (MTAG). Because of the conversions required, communication and interpretation of geometric data between turnkey systems and between suppliers and customers utilizing different graphics software and databases involved substantial cost and loss of time. Therefore, the CAD/CAM Interaction committee (of MTAG) requested immediate action by DOD. A draft document was targeted for the beginning of 1980.

The IGES report (Version 1.0) was completed and made available on January 30, 1980. By May of 1980 the National Bureau of Standards (NBS) had received over 500 requests for the IGES report. The goals for IGES as stated in the report were: "(1) to produce a format that would permit the communication of basic geometry, drafting, and structural entities, (2) to produce an open end format, to facilitate the communication of new material defined after the IGES was published, (3) to minimize, where possible, the burdens imposed on pre- and post-processors by the IGES, and (4) to gather as much input from the interested community as possible."

IGES is a project of the Air Force Integrated Computer-Aided Manufacturing (ICAM) Program. Funding comes from the Air Force, Army, Navy, and the National Aeronautics and Space Administration (NASA). The National Bureau of Standards (NBS) is the administrative agency for the IGES program. Over 42 companies are involved in the IGES effort with one or more members on various IGES committees. Employee time and travel expenses for IGES participation are provided without cost by the companies. Two major committees were formed, the working committee providing technical guidance and the steering committee for guidance on management decisions. Two subordinated committees are Extensions and Repairs (E&R), to maintain the technical intensity of the IGES document, and Test, Evaluate, and Support (TE&S), to coordinate implementations of IGES, prepare test cases for translators, and catalog recommended practices as a means for maximizing compatibility among the various translators which are produced. The February, 1981, E & R committee meeting brought 24 people from 20 companies plus 2 from NBS, while the TE & S meeting that followed had 15 people from 14 companies plus the 2 from NBS. The May meetings were similarly attended. The involvement by these companies is an indication of their commitment to IGES.

Over the last two years IGES has progressed through the various preliminary and public review stages to the final draft of Version 1. The American National Standards Institute (ANSI), through its subcommittee Y14.26 (Computer Aided Preparation of Product Definition Data), has been following the development of IGES and providing many helpful suggestions. On May 1, 1980, the subcommittee voted to issue IGES as the first three parts of a five part proposed standard. IGES meetings in February of 1981 produced revisions and clarifications in response to the ANSI review during late 1980. Further polishing of the document was the result of IGES meetings in May of 1981 with the final draft of Version 1 published by NBS in June. ANSI approved the draft of the proposed standard in July and submitted the document for printing. The completed national standard should be available before the end of 1981, possibly as early as October.

Adoption of the national standard is likely to propagate increasing contractual requirements for the use of the standard for documentation purposes. Even where not designated in a contract, the standard will be used as a convenience for transfer of information between the parties. The vendors of CAD systems, in anticipation of customer demand, have been implementing translators in both directions (pre- and post-processors) between the native databases and IGES.

WHAT IGES IS, AND IS NOT

IGES is meant to be used for the exchange of data between CAD/CAM systems and for the archiving of data from a CAD/CAM system. IGES was designed with knowledge of the technical aspects of several CAD/CAM systems. Therefore, translations between vendor systems, while not necessarily one for one, should be possible. The committees working on IGES were presented with a variety of data exchange formats, including those of the Boeing CAD/CAM Integrated Information Network and the General Electric Neutral Data Base. IGES, as a set of geometrical, drafting, structural, and other entities, has the ability to represent most of the information in CAD/CAM systems. Furthermore, IGES can be extended through the use of the mechanisms provided, including user-defined entities.

IGES is not designed to reflect the technical aspects of any one of the available CAD/CAM systems, nor does it copy exactly any of the exchange formats given to the committee. IGES is neither a perfect nor a complete specification for all data in all CAD/CAM systems. While newer, more complete versions of the specification will be developed in the future, translations to and from IGES under the current version may result in some loss of information.

HISTORY OF THE TRANSLATOR PROJECT

A PRIME computer system, located at the Westinghouse Power Systems Computer Center (PSCC), is used for mesh generation, hidden line removal, and other functions as preparation for data input to the subsequent analysis stages. To use the PRIME, however, it was necessary to provide the part geometry information. Direct (electronic) transfer of the information from the Computervision (CV) system would eliminate duplication of data entry to the two systems.

In mid-1980 it was decided that a Computervision (CV) CADDS3 to IGES translator should be produced to operate on the Control Data Corporation (CDC) system at PSCC. There has been some concern that the operation of a translator on a CAD system used for production drafting would have an undesirable impact on production. Furthermore, to centralize translations at one location seems preferable to providing a translator at each of the potential sites. The CDC system at PSCC is a good choice as host for the translator because it also is the host for the Westinghouse structural analysis software.

The Westinghouse division requesting the translator was primarily interested in the transfer of information to the structural analysis system. Westinghouse Electric Computer Analysis (WECAN) has substantial capabilities including (1) linear and nonlinear elastic analysis, (2) plasticity and creep analysis, (3) steady-state and transient heat conduction, (4) mode and frequency calculations, (5) seismic response spectrum analysis, (6) harmonic, linear, and nonlinear dynamic response, (7) modal superposition time history, (8) substructuring capabilities, (9) generalized linear constraints, and (10) wave front equation solvers.

Design effort for the translator occurred during the third and fourth quarters of 1980. Implementation (writing and testing of the software) began in January of 1981. Translator phase one (points, lines, circular arcs, and conic arcs) was demonstrated for a sample part file in early May. Possible extensions, such as annotation, are under consideration.

TRANSLATOR IMPLEMENTATION

The group within Westinghouse responsible for the translator implementation is the Engineering Systems department of the Corporate Systems Integration Division, a part of the Westinghouse Productivity Center. Engineering Systems provides expertise and consulting in computer aided engineering (CAE) hardware/software, applications, operations, and training.

The translator was written in FORTRAN with special emphasis on modularization. Portions of the software which have knowledge of the details of the CADDS3 database or of the number system used in the CDC equipment are isolated from the more general purpose routines which produce the IGES output to allow for easy modification in support of a different data input or a different translator host.

Differences in terminology between CV and IGES, sometimes only to enforce IGES neutrality, often tended to obscure the relationship between input and output information. Moreover, architectural differences between the computers (e.g. 16 bit versus 60 bit word length, one's complement versus two's complement representation of numbers, radically different floating point representation, and 8 bit ASCII characters versus 6 bit display codes) added to the challenge of deciphering the CADDS3 database. Because 60 is not evenly divisible by 8 or 16, many routines were written to compensate for the uneven data manipulation.

The mechanism for extracting information from the input part file was determined by careful study of the CADDS3 database reference manual and an octal dump of the part file of a sample part. The two parts of the output file which should be generated simultaneously (the Directory Entry section and the Parameter Data section) are written into parallel output file structures and subsequently merged as part of the final step. Information which must be supplied by the translator user is obtained interactively.

LIMITATIONS IN DATA TRANSFER

As a neutral database IGES exhibits differences from any vendor database. For example, a circular arc is represented in IGES by specifying the XYZ coordinates of the center and the XYZ coordinates of the end points. Some vendor systems specify the XYZ coordinates of the center, the length of the radius, and angles which define the end points. In IGES, an ellipse is represented by specifying the six coefficients of the general equation for a conic arc plus a number which indicates the type of the conic arc. Some CAD systems specify the lengths of the major and minor axes, the angle of rotation of the ellipse in the plane, and the type of the conic arc. Although the representations may differ, there has been much effort to ensure the accuracy of the IGES entities.

To simplify internal processing for display purposes or to support special system features, some CAD systems store more information than the minimum necessary to represent a geometric entity. Translations to and from that CAD database may not preserve the extra information unless user-defined entities are defined to contain the additional data.

Different CAD systems exhibit differences in their support of certain features, geometric entities, or other information. Data transfers between such systems could cause a reduction in the information. Data transfers between such systems could cause a reduction in the information content at the receiving end. Similarly, if the features are supported by both systems, but with significantly differing representations for the data, information could be lost in a transfer. Such losses, however, are independent of whether IGES is used as the intermediate form of representation.

IGES has not matured as yet. The entities defined in version 1.0 were intended to support mechanical geometry and annotation. Extensions in support of electrical and piping entities are the subjects of current subcommittee activity. The IGES E & R committee, in the future, will resolve any oversight in the original specification and add extensions in support of other disciplines. The TE & S committee, through publication of recommended practices, will further enhance the communication of CAD/CAM information between systems.

CONCLUSIONS AND RECOMMENDATIONS

IGES is a viable mechanism for data exchange which can be valuable to Westinghouse and other companies. Use of IGES should be encouraged so that the resulting experience can shorten the cycle time for corrections and enhancements leading to the eventual stability and universal acceptance of the standard. Companies should support the IGES effort, as well as the related ANSI activity, through distribution of information to all employees using and managing CAD/CAM systems and through participation on IGES and ANSI committees.

Translators produced by independent software vendors or by CAD/CAM system vendors and written in FORTRAN would allow Westinghouse and other companies a choice between pre- and post-processing of part files on the CAD/CAM systems or at a data center. Such non-resident translators could spread the cost of development and maintenance over a customer base if demand is sufficient.

We now have a national standard for communication of drawing and part information among CAD/CAM systems. Let us all use and support the standard and help it grow.

BIBLIOGRAPHY

Initial Graphics Exchange Specification Report, Version 1.0, January 30, 1980, National Bureau of Standards

IGES Newsletter, Volume 1, Number 1, May 1980, National Bureau of Standards

Final Draft, Sections 1, 2, 3, and 4, Proposed American National Standard, Engineering Drawing and Related Documentation Practices, Digital Representation for Communication of Product Definition Data, June 1981, IGES Y14.26M Response Committee

The System for Structural Analysis WECAN-WAPPP-FIGURES, May 1978, Advanced System Technology, Westinghouse Electric Corporation

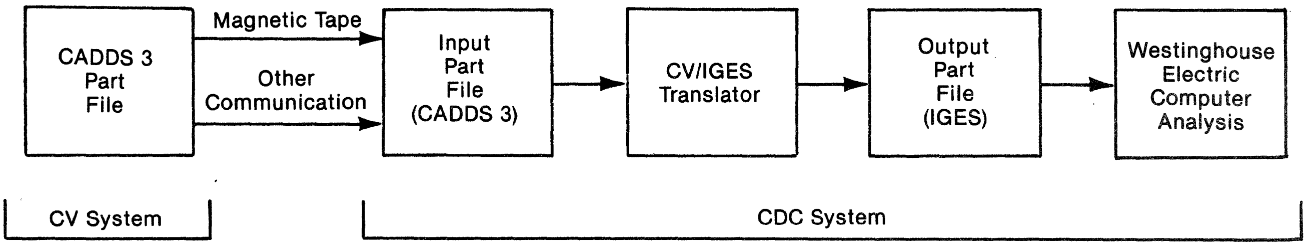


Figure 1: Translator Configuration

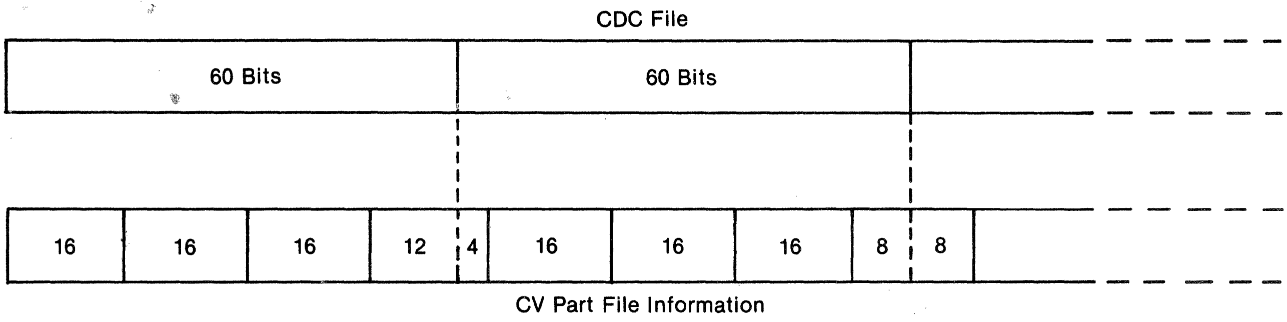


Figure 2: Packing of CV Information in CDC File

"A GUIDE TO CAD FACILITIES ORGANIZATION FOR THE FIRST TIME USER" (Or, Turnkey Systems Aren't)

Fred Sukow
General Electric Company
Advanced Reactor Systems Dept.
310 DeGuigne Drive
P.O. Box 5020 M/S 5-40
Sunnyvale, CA 94086
213-738-7445

ABSTRACT

This installation guide is directed primarily towards the neophyte system buyer, although there is information that might be valuable to the veteran buyer/user. The discussion is divided into three general categories: Preparation, Installation and Implementation.

The first step of the Preparation category involves intensive research by the prospective user which includes visiting CAD installations, gathering CAD operation documentation and studying trade publications. The next steps involve preparing people in the organization, readying the facilities and witnessing in pre-delivery testing.

The second category, Installation, is subdivided into receiving, assembly and testing, and some general guides and hints are discussed in these areas.

The last category, Implementation, is comprised of three subtopics — training, operation and security. On the subject of training, there is discussion of scheduling, in-house vs. vendor site training, and follow-on advance training. Operation covers work scheduling, work priorities, and suggestions on work tracking methods. The discussion of security touches on the physical such as door locks, isolation, offsite tape storage, and data/system security which includes utilization of the system's protection facility by use of passwords and file access authorization.

The paper concludes with a summary of the salient points to be considered when organizing your CAD facility for effective operation.

You have all seen advertisements for Computer Aided Design Systems and you'll probably agree that, in general, the copy reads like one normally associated with the "Ace Vegomatic Little Kitchen Aid". "Buy this little self-contained modular turnkey beauty and, with just a few hours of instruction, you'll be making 3-D designs of your very own space shuttle." Right!!

I may exaggerate just a little, but having bought one of those "turnkey beauties", it has become apparent that there are a few things you aren't told concerning installation and implementation. I'm sure it isn't because there is a desire to keep you in the dark; but, I think, probably it is assumed you know what you're doing and it would be presumptuous to tell you what to do.

It is my intention to cover the basics of CAD System installation and implementation based on my personal experience with our first system, and I'm sure for every incident that is mentioned, the veterans that have fought this battle can relate a half a dozen of their own.

This discussion is divided into three general categories:

1. Preparation
2. Installation
3. Implementation

PREPARATION

Prior to vendor selection, there is one relatively simple and possibly enjoyable activity you can participate in to prepare yourself mentally for this madness you are about to precipitate. I would encourage you to visit as many CAD installations as you can afford, within your own company and at other companies. Do not limit your trips to vendor demo sites. For this installation survey, it really doesn't matter if the CAD equipment is all the same manufacturer as you wish to buy or that the application is always the same as yours. As you look, ask yourself questions such as "Is this aesthetically appealing?" . . . "Would I like to work in this environment?" . . . "What do I like/dislike?" . . . "What could be improved?" . . . "What are the apparent problems?". The list can go on almost endlessly.

Another favor you can do for yourself, while touring the world of CAD users, is to beg, borrow, wheedle and cajole any and every procedure or instruction that these users have on CAD system operation. Give serious consideration to either hiring support personnel (e.g., System Specialist) or engaging the services of a CAD consultant. In a nutshell, if you can't talk CAD, find someone who does.

PEOPLE

Now comes the most important part of your preparation. This part is labeled PEOPLE. If your system is to be utilized successfully, you had better prepare *all* of your people for the blessed event and this doesn't mean just the "chosen ones" that will become your first operators. You should consider all persons including potential operators and those that may never get the opportunity. You have to try to create a positive attitude that will displace the fear that CAD systems seem to generate:

— Fear of being replaced by a machine . . .
— Fear of failure if they try to learn operation . . .
— Fear of the unknown, and on and on.

We held meetings almost monthly during the acquisition stage for all the designers and drafters to report on progress and to explain what CAD could do to assist them in their design work. For the meetings, we borrowed video tapes from the vendor and from internal sources that provided a system orientation and an introduction to Computer Aided Design. Articles from trade journals and technical publications were routed around. People were encouraged to take introductory classes offered by local junior colleges. In general, everything that could be done was done to create a positive interest in CAD.

Personnel selection is the most important and sensitive part of the PEOPLE category. You must decide whether to train your people from the top down (that is, best designers first) or only your detailers or a combination thereof. You must also decide whether or not you are going to reclassify your design people as CAD operators.

During our visits to CAD installations, we were able to observe a variety of business products and personnel situations, and from this decided that for *us,* because of our R&D business orientation, the best approach would be to train our best and most experienced designers and to *not* reclassify them at this time. Our philosophy for the time being is that the CAD System is just another tool (albeit a very sophisticated one). Moreover, we felt that the best people would be more likely to learn faster and use the equipment most effectively in the design mode.

Since our organization is a Research and Development operation and not manufacturing, we have placed emphasis on the design aspect and, as a result, our training is designer oriented. Digressing still further, it is our opinion that most CAD systems are most effective in the design mode rather than drafting, although productivity gains may be demonstrated in much of the drafting work performed on CAD. Back to the topic of discussion — PEOPLE.

Age is not a factor in the trainability of the people for CAD operation. We have people with 35 and 40 years of service with the company that have been just as eager and trainable as the younger individuals. This may be due, in part, to the earlier positive information and status meetings.

Some key individuals and your system manager must be scheduled for training classes on system operation before system delivery, or of course an even better approach would be to hire someone with the expertise in addition to scheduling system operation classes. The point is, someone in your organization must know at least the basics about the system before it is received. Otherwise, you could find yourself waiting hours or days with an unusable system and have a vendor technician to tell you all you needed was to ''flip this switch'' or ''press that button''.

Don't deceive yourself into thinking you can operate anything if you have the owner's manual. Many years ago when I worked on a wheat ranch, we hired a new caterpillar tractor driver from the crowd in front of the employment office. When asked if he could drive a D-8, he answered to the effect that he ''could drive anything''. He was taken to the tractor and told to give it a try. After circling the tractor a few times and carefully checking it over, he announced, ''I can drive anything you can *start''*. I certainly wouldn't make that statement about a CAD system. The system sales representative should offer training, but, if he doesn't, insist on it before signing any contract.

PROCEDURES

Establish some interim instructions and procedures during this early stage. You'll find it much easier now than when the world is falling apart during system installation. The procedures can cover such things as file labeling, design station scheduling, work request forms, system protection, etc. They may not be exactly what you will want formally issued later, but at least they will provide some direction and guidance for the users. It will also demonstrate to your management that you really know what you're doing. This is important for establishing the credibility you will need when you make all the seemingly extravagant requests for your facilities.

It is extremely important that you make every effort to educate upper management on the benefits and realities of CAD. They must be prepared for less than miracles the first year and to do this you must keep them informed.

FACILITIES

The facilities design challenge/problem is probably one of the most subjective areas (i.e., open for personal interpretation) and can be the most difficult to sell to management.

Our choice of facilities design was based on observations made during our tour of installations as well as what we were able to dig out of technical articles on the subject. (As an aside, a guide to system installation was published after we were done that can provide much needed information that you might not find elsewhere.)

We began with three general ideas for our installation:

1. Separate CPU, plotter and disc drives from design stations.
2. Provide a quiet, creative atmosphere with subdued, adjustable lighting for the design stations.
3. Provide separate A/C for CPU area.

Each of these ideas was examined separately to determine the most effective facilities arrangement. Starting with the first item, the goal was to provide a creative atmosphere that would attract the designers, make them want to use this new tool, and to provide an aesthetically appealing installation to show visitors (and you will have visitors) that this is an important part of your design operation. We had the walls painted in atypical colors, the floor was carpeted, special color-coordinated chairs were purchased, art work is displayed on the walls, and all of this was done at surprisingly low cost. The biggest hurdle is convincing people that a nice pastel brown or orange wall costs the same as institutional white and does a heck of a lot more for positive worker attitude.

Now for the "computer room" which is what we are calling the room housing the CPU, disc drives and plotter. First, the room was designed to be twice as large as needed for the initial installation because we knew what we were going to expand and we also knew it would be a lot easier to make it big now than to try to add on later. This also applies to the air conditioning system (A/C); size it for the heat load of your future arrangement. The computer room was made separate from the rest of the system for two main reasons . . . system environment and system security. System security will be covered in later paragraphs.

In spite of what you may have been led to believe, the CPU operates the most effectively at 68°–72°F and at 45% to 55% humidity. In our case, this certainly precludes using the building air conditioning system, so we bought a stand-alone unit with bottom discharge. This unit was installed over a standard computer room false floor such that the cooled air flows under the floor and up through cutouts and into the base of the CPU, the disc drives and the plotter. It became readily apparent in our installation that having a separate air conditioned room was justified because the noise of the A/C unit would not be long tolerated by the designers at the work stations. Another problem that was discovered, after the fact, was flooding. Like many similar installations, our raised floor was installed in an area without floor drains and, subsequently, the first time the A/C condensate pump decided not to function, about 30 gallons of condensate flooded under the wall and into some boxes of copy paper next door. In fact, we didn't know there was a problem until an irate neighbor came to visit. We since have installed a condensate overflow drain pump to prevent this from reoccurring.

To help the A/C system in maintaining the environment at a steady state, large viewing windows were installed along the wall adjacent to the corridor to provide viewing of computer equipment and activities by large groups of visitors without them actually being in the room.

This also is an aid to maintaining security. It is necessary to go through one of our offices to reach the personnel entry door to the computer room which obviously allows screening of visitors and helps assure no unauthorized entry. The office door and the door between the office and computer room have locks and they are keyed differently. Keys to the computer room lock are limited and signed out to key people (no pun intended) in order to prevent unauthorized entry. This may seem a bit paranoid, but after finding curiosity seekers wandering around in the computer room, touching this and that, when the door was inadvertently left unlocked, we became believers in this selective paranoia. I'm sure you can imagine this little scenario . . . "Gee, I wonder what this button does?" CLICK!!

You now have an aesthetically pleasing computer room with attractive walls and a steady-state environment. About the only thing left to consider is the electrical power for the CPU and disc drives and the plotter; the A/C and maintenance, etc., etc.

Our philosophy was "cleaner is better". The cleaner, the more isolated your power to the CPU and drives is, the more effectively the system will operate.

A branch circuit with its own breaker panel was installed in the computer room to supply the CPU, disc drives, plotter and utility outlets. The power goes from the panel to the individual power conditioning/isolation transformers and then to each drive and the CPU. In addition, after a couple of power failures, we installed a low voltage trip between the breaker panel and the power conditioning transformers. This device trips the CPU and disc drive off-line with approximately a 10 volt drop in voltage and they remain off until manually reset. At the time this was installed, we had a high temperature switch installed in series with the low voltage trip and set it to trip off-line at 80°F. Since the installation of these devices, we have had about six low power trips and have had no disc or CPU problems as a result. Prior to installation, we had a "near" head crash because of tandem power failure, approximately 5 seconds apart, in which the heads retracted with the first failure, but the retract circuit wasn't sufficiently charged to retract the heads a second time.

Four utility outlets were located under the floor to provide power for maintenance and these, plus the wall outlets, are on a separate circuit that branches off the main panel upstream of the power conditioning equipment. This helps prevent electrical "noise" in the power delivered to the computer.

The A/C power is on an entirely separate electrical panel and branch circuit because the potential power fluctuations and the electrical "noise" precluded having the A/C and the computer near each other electrically.

While on the subject of noise, it is probably appropriate to mention the audible output of the isolation transformer for the CPU. I believe it is reasonable to suggest that you *not* install this device near the operators, or, if you must, design a special sound-proof, air-conditioned booth to enclose it. This piece of equipment is by far the noisiest component in our computer room; its hum can be even heard above the noise of the A/C unit. During your visits to the various installations, ask about this problem so you'll better understand what needs to be done.

The noise is not loud enough to be damaging to the hearing, but it has an effect more like the ancient water torture. The first 50 or so drops don't bother you at all, but after an hour or so, you begin to get a little dingy. Because of the heat generation of this equipment, be sure you have adequate cooling air flow in your sound proof enclosure.

One other tip regarding the power conditioning equipment is to request delivery of the devices well in advance of the computer equipment so that you may have these installed and ready to power the computer during installation and check-out by the vendor field engineers.

As you make your final preparations for delivery, it is advisable to check all the aisleways, corridors, etc., to see if the equipment, crated or uncrated, will be able to pass through all the way from the loading dock to the CAD area. You might be surprised to find some of the corners too tight, some doorways too narrow or stairs that need to be negotiated. Keep in mind that the disc drives, uncrated, weigh over 500 lbs. You may want the shipping company to move the equipment rather than your own facilities people.

A final note on facilities design is to check the NPFA No. 75, the National Fire Protection Association Code covering computer installations. You should determine, in advance, the application of this code to your installation. Local interpretation may have a direct impact on the cost of your facilities.

Just prior to system delivery you have the opportunity to witness the pre-delivery testing. This can be beneficial to you in at least two ways. One is that it will be the first opportunity for you to see your system totally assembled and secondly, it will be your last chance before delivery to repair or replace malfunctioning hardware and check your software package for missing programs. Your being present at the system's "birth" possibly will cause some imprinting to take place with you becoming a surrogate parent for the system which will subsequently obey your every command.

INSTALLATION

Your presence during installation is almost mandatory. It will definitely be a learning experience and, at the very least, you may be able to provide some help like holding a screw driver or provide some muscle in moving a heavy crate. In any case, you should be there during unpacking to check off for receipt, in good condition, of the equipment. If there is anything missing from the shipment, order the replacement immediately. If you wait to order, you could find your startup schedule beginning to slip. Be there as your system is assembled. Again, anything and everything you learn about your system very likely will be of value to you sooner or later. Besides, this is like Christmas and all your birthdays combined and you should be the one to have the fun.

In our case, during installation, a grounding cable was not run between the CPU and two of the terminals. This was subsequently forgotten and, later, when there was some static electricity problems, the solution was readily apparent, but obviously the problem could have been avoided.

After the system is completely assembled, you will go through lengthy and thorough hardware and software acceptance tests. As parts of the system are checked, you will be asked to sign for acceptance. However, if there is any piece that doesn't check, you may sign the acceptance with exceptions. In our case, lacking a MODEM, we were unable to check the communications hardware/software with our mainframe computer. We added a note to the acceptance sheet that these hardware/ software features were not accepted.

It is important that your vendor has two resources: a 24-hour hot line for emergency service and a national accounts representative for your firm. A call to either of these resources will usually get prompt response. We are fortunate that, in addition, we had very talented and responsive local vendor personnel.

Whenever you have a system problem, call the vendor hot line. Although it may be a bother to call sometimes, every call is important. A running record is made of your system problems and, if a trend begins to show in the nature of these problems, the vendor will probably spot it before you and be better prepared to provide a solution to the problem.

IMPLEMENTATION

Finally, the last topic is that of implementation which I have divided into three subjects labeled training, operation, and security.

As a preface to this discussion, it should be pointed out that your approach to training, job selection and work priority will naturally be related to your function in your company's organization. This discussion is biased by the fact that we are part of an engineering services type organization.

In any case, do plan to send your system manager and/or system specialist to the System Operations class *prior* to delivery. Even if they have some experience in CAD operation, this type of class covers many important aspects of system operation, some of which are certain to be beneficial. If you plan to have the vendor provide instructors for the "Basic Operators Class" to be taught at your site, then you must schedule this well in advance. Don't wait until your system is running and then call to schedule classes; there may be a 6–8 week waiting list. You'll find yourself with a working system and no one to use it for 8 weeks. Look ahead at the next phase of the training you will need and begin to rough out schedules and lesson plans. You will be so busy in the ensuing months that it will seem extremely difficult to do this planning then.

Prior to actually beginning your CAD operation, there are some work logistics to be considered. On the surface, work scheduling would appear to be rather straightforward and trouble free and, if you are very good at preplanning, this might be true for you. However, for the rest of us, let me outline some considerations that may help you begin organizing your work scheduling.

The ability to do work on the CAD System is a dynamic, growing capability. To cope with this variable situation, we developed a method of job selection and distributed an extract to the engineering functions that will be requesting work.

JOB SELECTION

For selecting the particular work to be applied to the system, a number of specific criteria can be considered. In initial applications, the complexity of the job will be one of the most important. Since the operators (I prefer designers) will initially lack the system experience and sophistication required by most complex 3-D tasks, nearly all of the early use will probably be limited to 2-D layouts and diagrams. As the designers gain knowledge and experience, the complexity of the work will increase. Other selection criteria can include: job schedule pressure, whether the work is old or new, and if old, is there time to reconstruct the data base (drawings) on the system; and finally, there may be some tasks that will be assigned to the system in order to provide some "showcase" jobs for reports and demonstrations.

In our case, the actual job selection process has three basic steps involving Engineering and Drafting.

1. Drafting and Documentation will formulate a list of candidate jobs based on the criteria discussed above. The inital listing will be formulated shortly before first application and continuously updated.
2. Engineering Management representatives will be given the opportunity to review the list and input their definition of priority. Obviously, there may be particular jobs that are not listed that they would like on the system and other jobs listed that would better be done manually.
3. The last step of the selection process is the negotiation to resolve any differences.

We followed the lead of the CAD installations we visited by developing a special CAD Work Request form. This form parallels a standard drafting work request with the addition of a CAD time estimate as well as a manual time estimate. An interesting event occurs with work on the CAD System and it follows a pattern something like the following:

Mr. Engineer brings in a design job that he estimates and budgets will take 200 hours to do manually. You assign it to a Designer and he proceeds to flash through the job on your CAD System in 100 hours. As soon as Mr. Engineer realizes that he is going to have a lot of time/money left over, he begins to ask for iterations on a few areas he may be less than 100% satisfied with (that qualification might cover a number of areas). In this manner, the whole 200 hours is used up for the job. He then says, "Well, you did a good job, but it took you the same amount of time as it would have manually, so where is the savings?" Gotcha!

On the back of the CAD Work Request, there is a ruled area for listing iterations that were not part of the original job scope. In this way, the total time may be justified in terms of increased scope and still show a cost improvement of CAD over manual.

The subject of operation is so big that it needs separate coverage, so with these few tips, we will move onto the next topic which is security.

Security may be divided into sub-categories which I have labeled physical and data. Physical security, which was previously discussed, includes the obvious such as locks on the doors to control access to the CPU, and no one is allowed access to the computer room without authorization of the System Manager or the Operations Specialist.

As mentioned earlier, this room contains the disc drives plus it also contains locked storage cabinets for the magnetic tapes and the disc packs used for back-up purposes (i.e., if the data on a disc becomes questionable or lost because of a head crash, etc., you have a disc(s) that only needs reloading of current data).

Data security includes the use of the system protection facility commands which provide for controlled user access. An authorize file may be established that defines user access and this access control allows you as Systems Manager to prevent unauthorized access of either the system or specified files within the system.

You may also protect against changes to issued data except by special approval or even prevent filing of any data without authorization.

Included in the data security category is the protection of generated data stored on the disc by use of regular taping routines. If you transfer data from the disc to tape on a daily basis, you will never lose more than a day's design work. This applies to a single shift application and, obviously, if you are working multiple shifts, it might be best to tape at the end of each shift. We also tape on a weekly basis to provide additional backup to the daily taping and minimize the tape inventory. Duplicate tapes are stored off-site, for common mode damage protection, including copies of all the required system tapes as well as our library tapes.

We have had the computer room made a check point for the roving night guard for security protection, but also to check on the power conditions of both the air conditioning system and the CAD System and, if there are any problems, he has a list of people to call and notify.

This has been just a cursory look at system implementation and not an attempt to provide an in-depth look at this subject. There is just too much to discuss as an addition to the topic of this write-up.

SUMMARY

We have covered many aspects of the preparation for a CAD System and the installation. Hopefully, what has been discussed here will at least trigger questions to ask during your review of user sites. You should have an idea of what to look for at an installation and some leading questions to ask the users.

In planning your facility, remember to think of establishing an atmosphere with aesthetic appeal as well as one that is conducive to operation creativity. You want the operators to feel comfortable with this new tool and proud of being a part of its use and development. Therefore, not only the atmosphere must be right, but the users must be kept informed of what is happening and excited with the potential of CAD.

You must do all you can to provide a clean, secure containment home for the CPU and disc drives and assure the "cleanest" power source you can afford to ensure data integrity.

System operation classes should be taken prior to delivery and as many of the operator classes as you can foresee should be scheduled well in advance of your need. Schedule each step of the delivery and installation and be sure that it is physically possible to move the equipment from the loading dock to the CAD area. You or your representative must be present during delivery and installation and remember to promptly order missing or damaged items.

Establish preliminary operating instructions and procedures to provide initial guidance to the operating personnel during the hectic early days. System security must begin immediately and be maintained consistently throughout operation.

Implement a method of work scheduling and work tracking. You *must* have a means of work tracking in order to judge work effectiveness and, just as importantly, system justification.

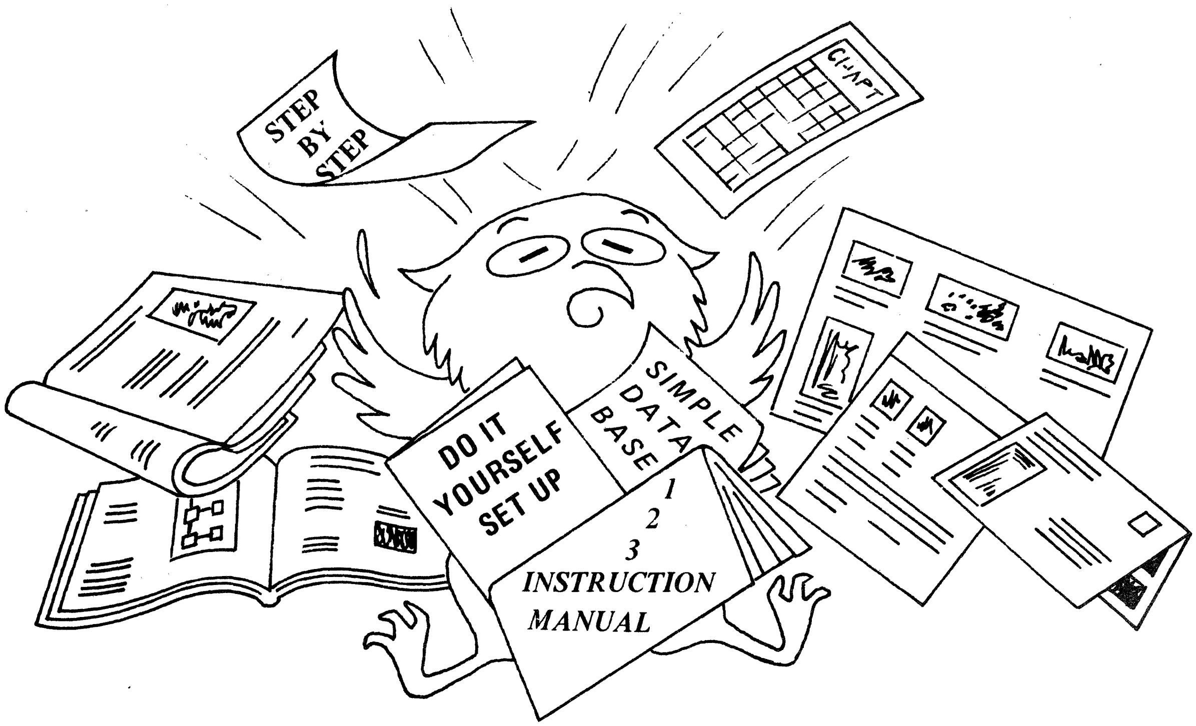
You are going to have problems and growing pains, but if you remember you have friends as close as the telephone hot line and if you promptly discuss your problems with them, most of these will be rectified very quickly.

Wasn't that easy? Now that you have your first system running like a clock, what's next? Get out your long range plans. When do you need to expand? What is the current equipment lead time? Maybe you should have placed your order today! What about networking? What software will you need? Do you have room? What about money?

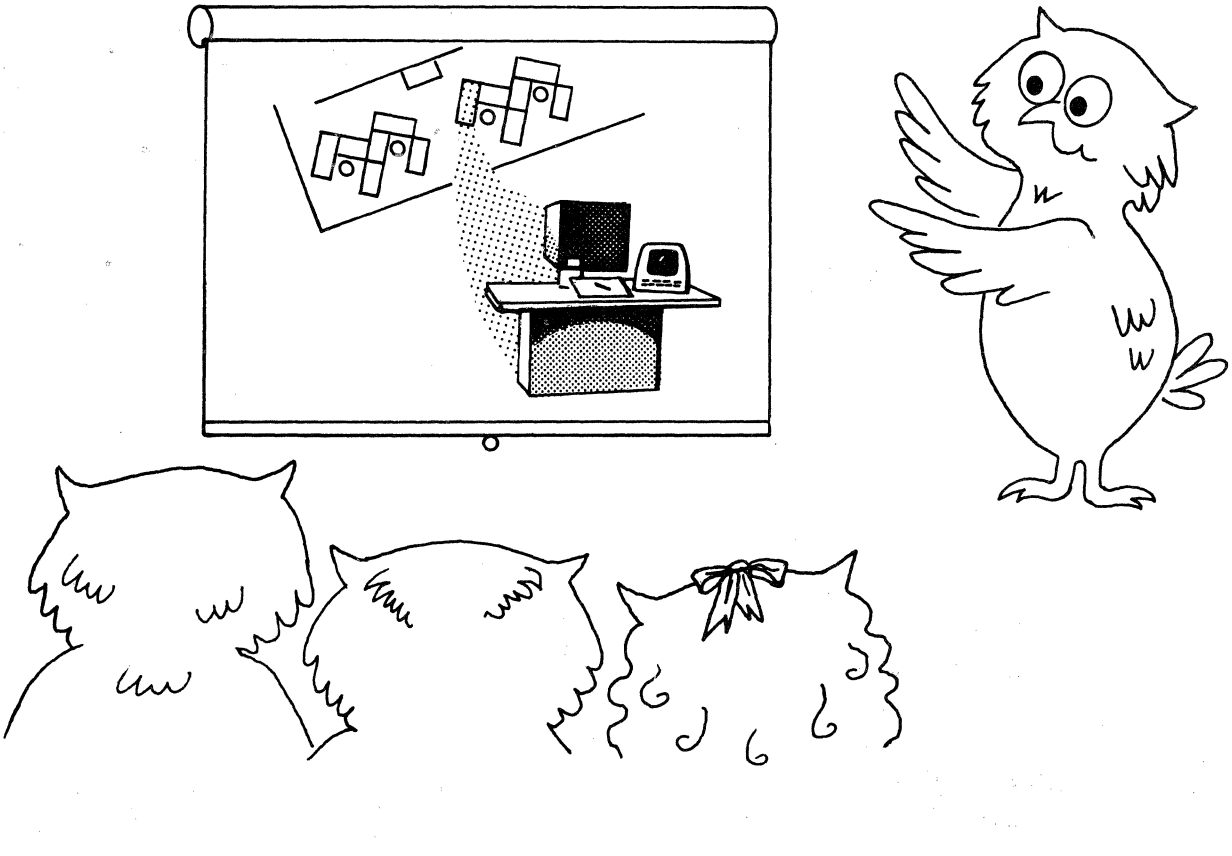
Exciting isn't it?

A GUIDE TO
CAD FACILITIES ORGANIZATION FOR THE FIRST TIME USER
(OR, TURNKEY SYSTEMS AREN'T)

Presented By
FRED SUCKOW
For
GENERAL ELECTRIC



THIS IS CAD, YOUR NEW WORK ASSISTANT



- PREPARATION

- INSTALLATION

- IMPLEMENTATION

PREPARATION

- PRELIMINARY
- PEOPLE
- FACILITIES
- PREDELIVERY

PRELIMINARY

VISIT SEVERAL INSTALLATIONS

ASK QUESTIONS

GATHER INFORMATION
AND DOCUMENTATION

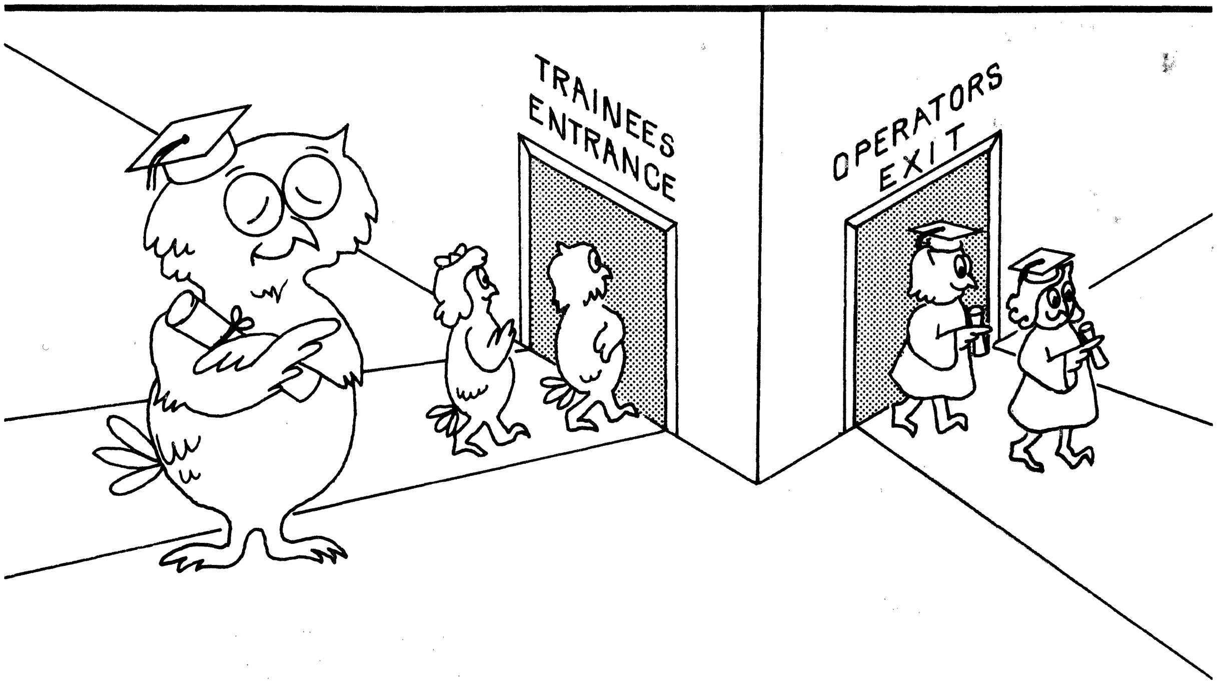
PEOPLE

- MEETINGS
- VIDEO TAPES
- PERSONNEL SELECTION
- MANAGEMENT EDUCATION
- PROCEDURES

FACILITIES

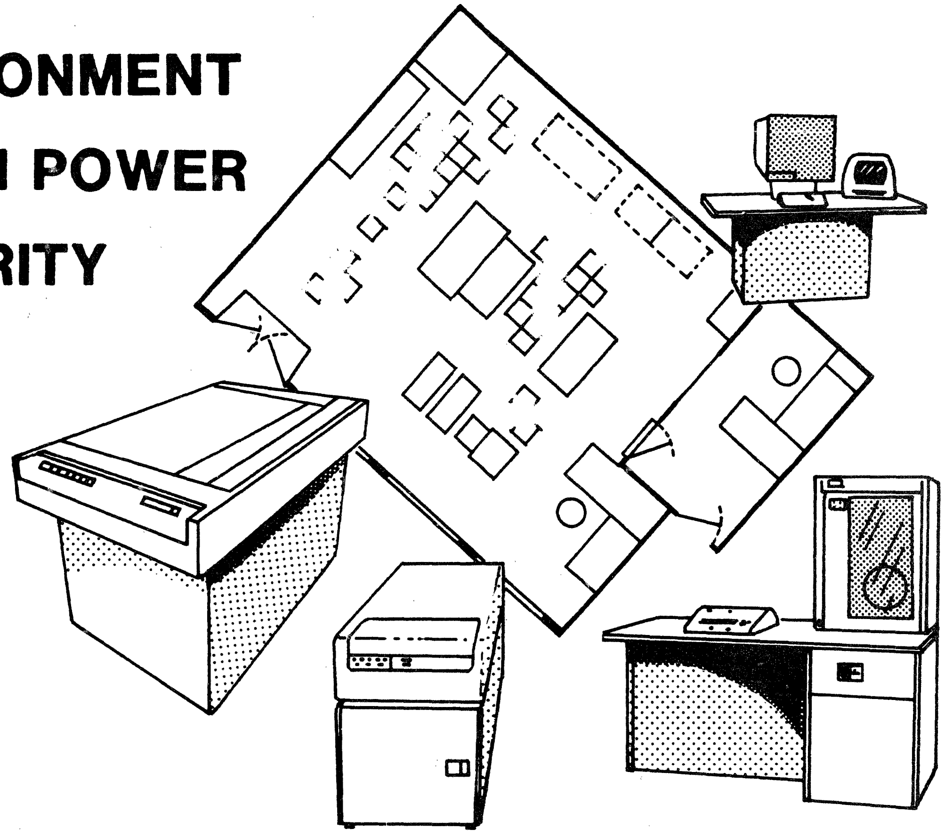
- CREATIVE ATMOSPHERE
- ESTHETICS
- ENVIRONMENT
- POWER
- LOGISTICS
- FIRE CODE COMPLIANCE

ESTABLISH TRAINING PRIORITIES AND SCHEDULES



COMFORTABLE CREATIVE ESTHETIC ATMOSPHERE

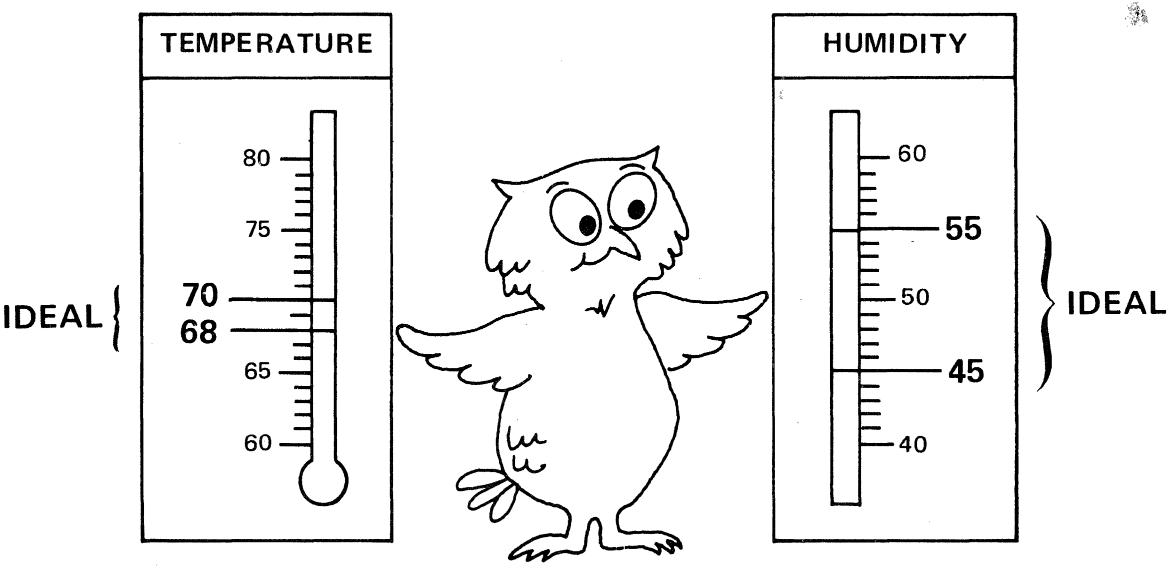
- ENVIRONMENT
- CLEAN POWER
- SECURITY



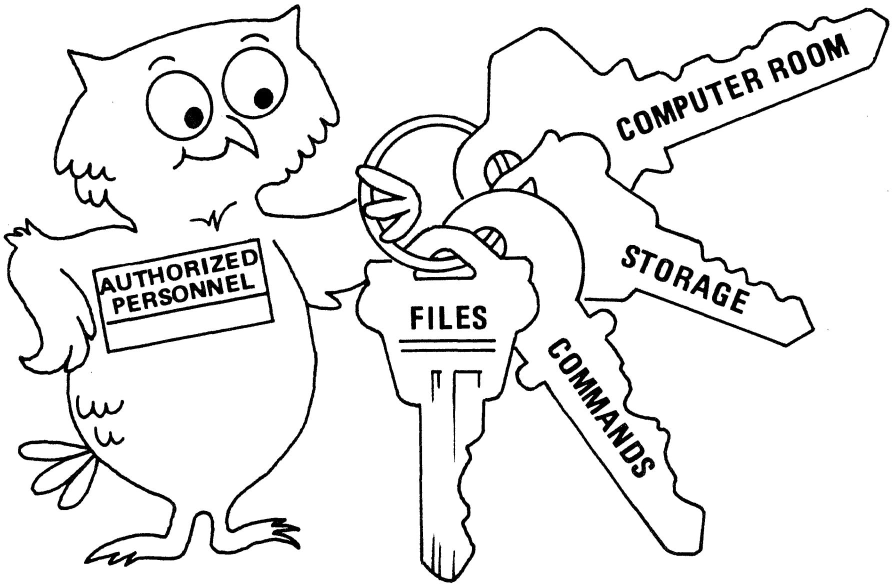
ESTABLISH PROCEDURES

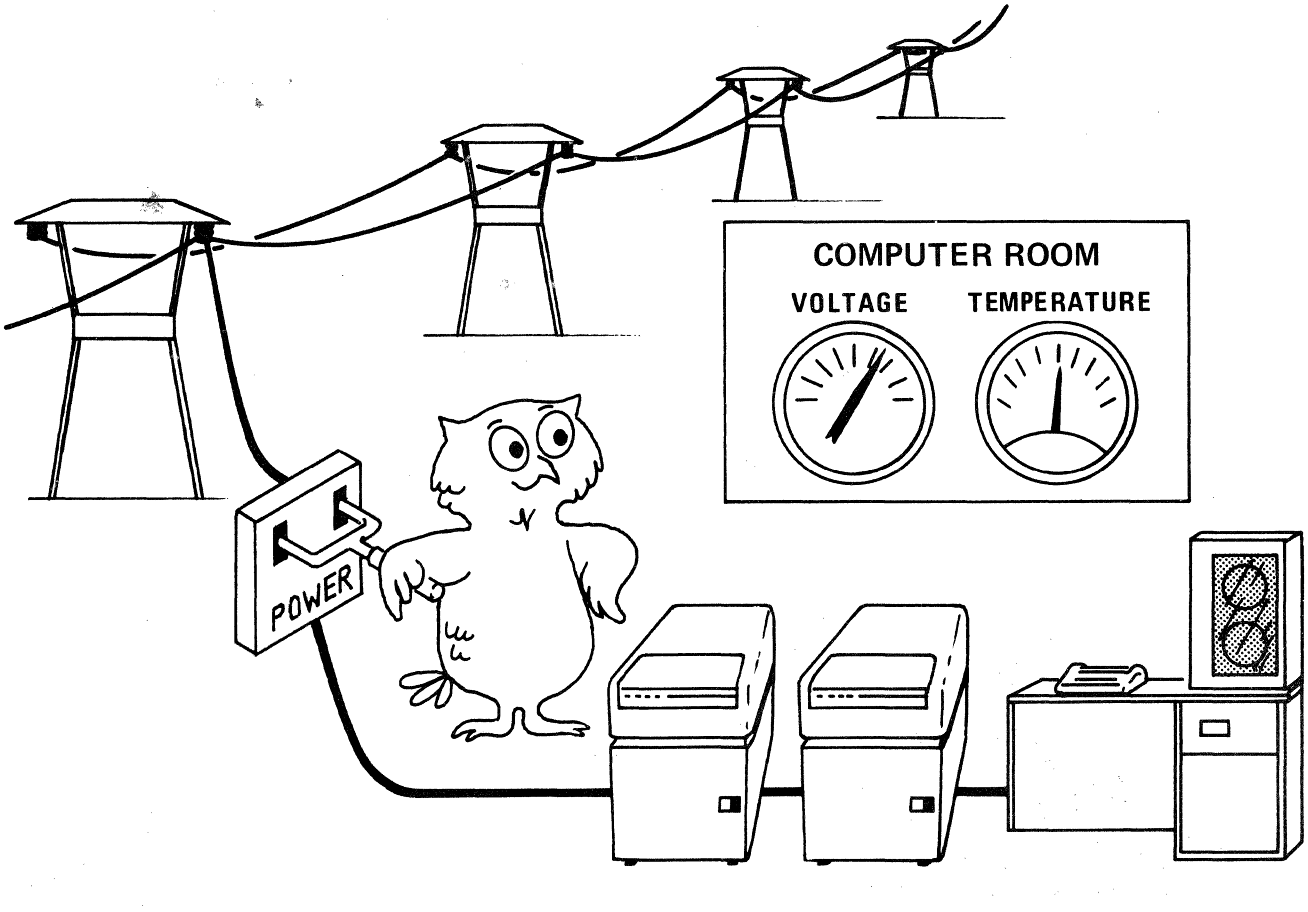
- FILES
- SCHEDULING
- WORK FORMS

MOST EFFECTIVE OPERATING ENVIRONMENT

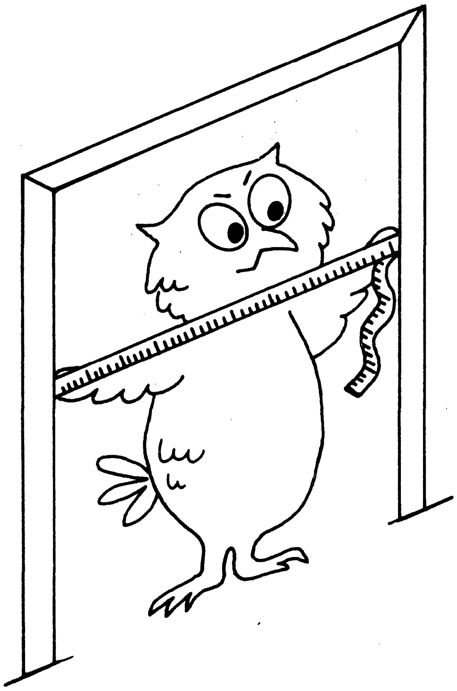


ENVIRONMENTAL MAINTENANCE AND SECURITY





PREDELIVERY CHECKLIST



✓ POWER EQUIPMENT DELIVERED AND INSTALLED

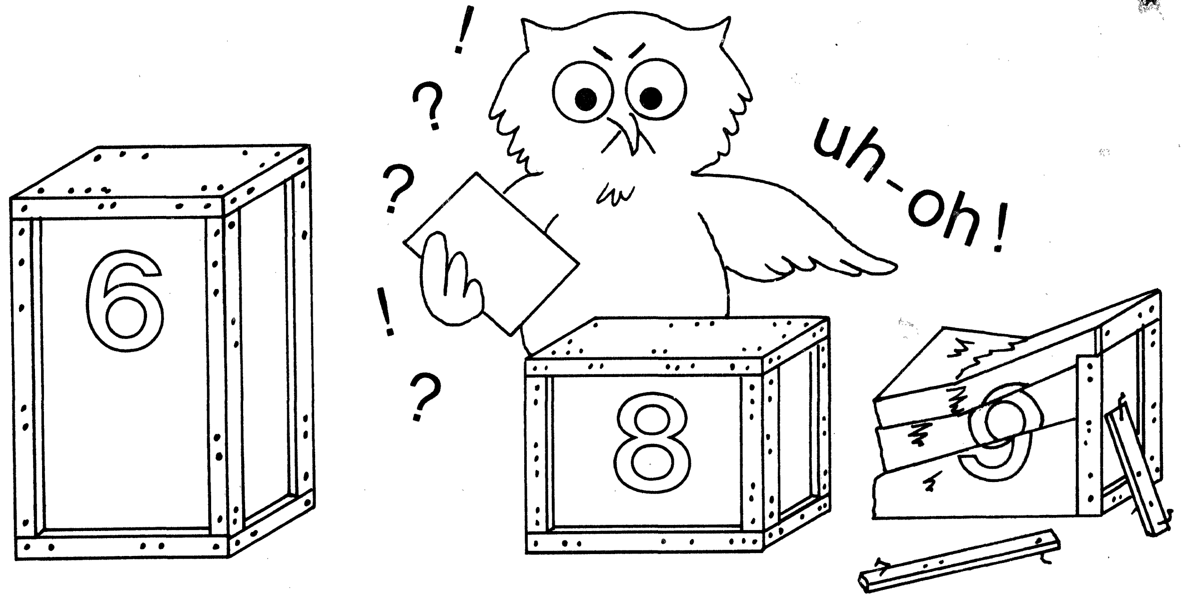
✓ A/C UNIT INSTALLED AND WORKING

✓ CHECK CLEARENCE MEASUREMENTS FROM DOCK TO COMPUTER AREA

✓ COMPLIANCE WITH NFPA No. 75

✓ ATTEND PREDELIVERY TESTING

ORDER MISSING OR BROKEN PARTS-NOW



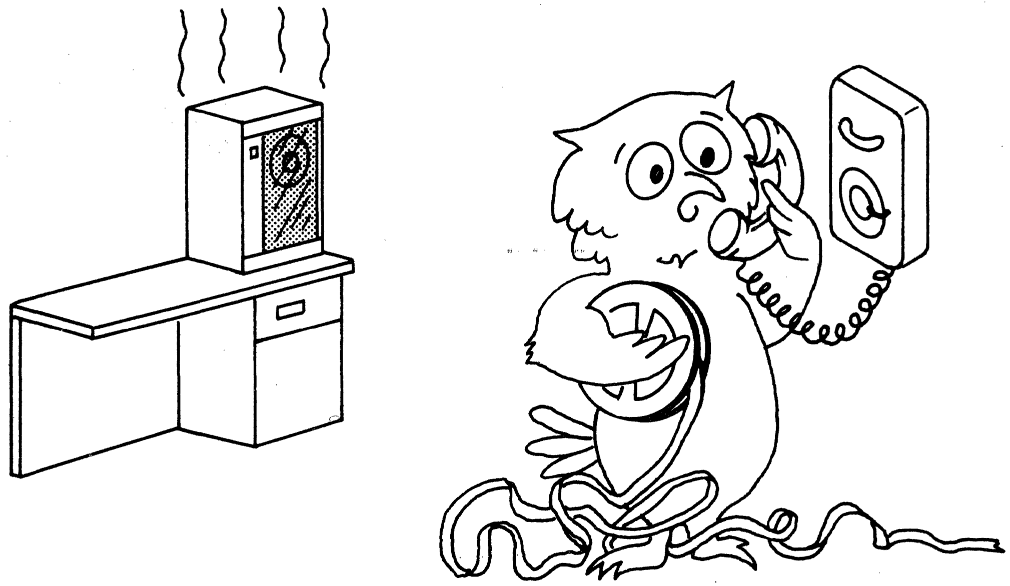
INSTALLATION

- RECEIVING (YOUR PRESENCE IS REQUIRED)
- MISSING/BROKEN PARTS
- ASSEMBLY
- TESTING
- VENDOR HOT LINE

ASSEMBLY AND TESTING

- WORK AND LEARN TOGETHER
- COMPLETE HARDWARE/ SOFTWARE TESTS
- ACCEPTANCE/NON-ACCEPTANCE
- REORDER REPLACEMENT PARTS

CONTACT VENDOR HOT LINE



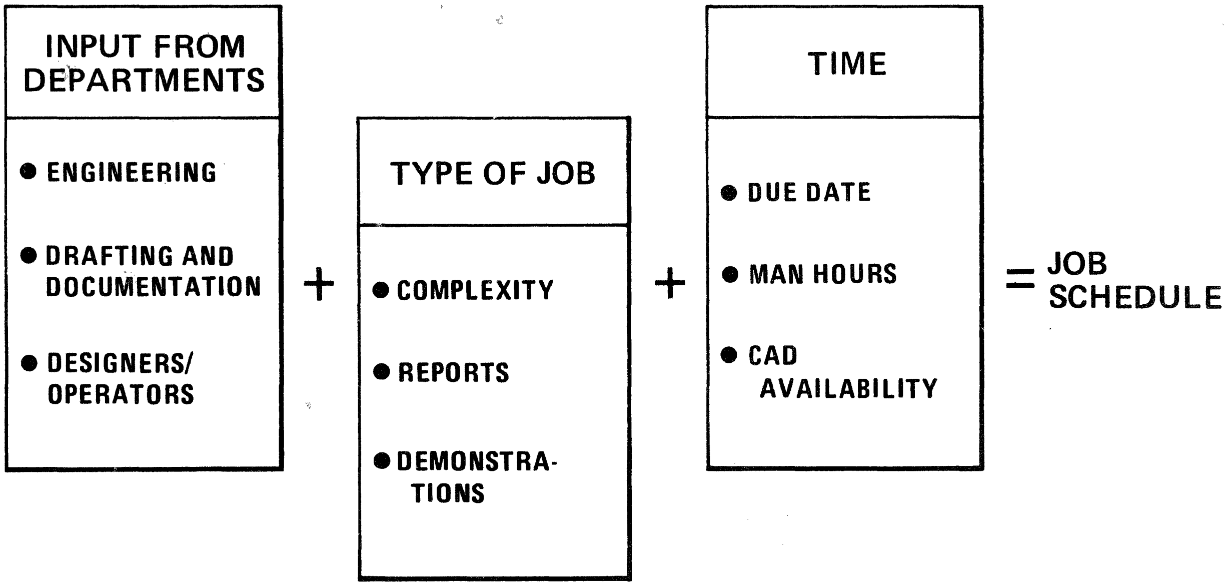
IMPLEMENTATION

- TRAINING
- OPERATION
- SECURITY
- PROBLEMS

TRAINING

- MANAGERS
- SYSTEM SPECIALISTS
- EXPERIENCED OPERATORS
- ON-SITE OPERATOR'S CLASS

JOB SELECTION AND SCHEDULING



INPUT FROM DEPARTMENTS

- ENGINEERING
- DRAFTING AND DOCUMENTATION
- DESIGNERS/ OPERATORS

+

TYPE OF JOB

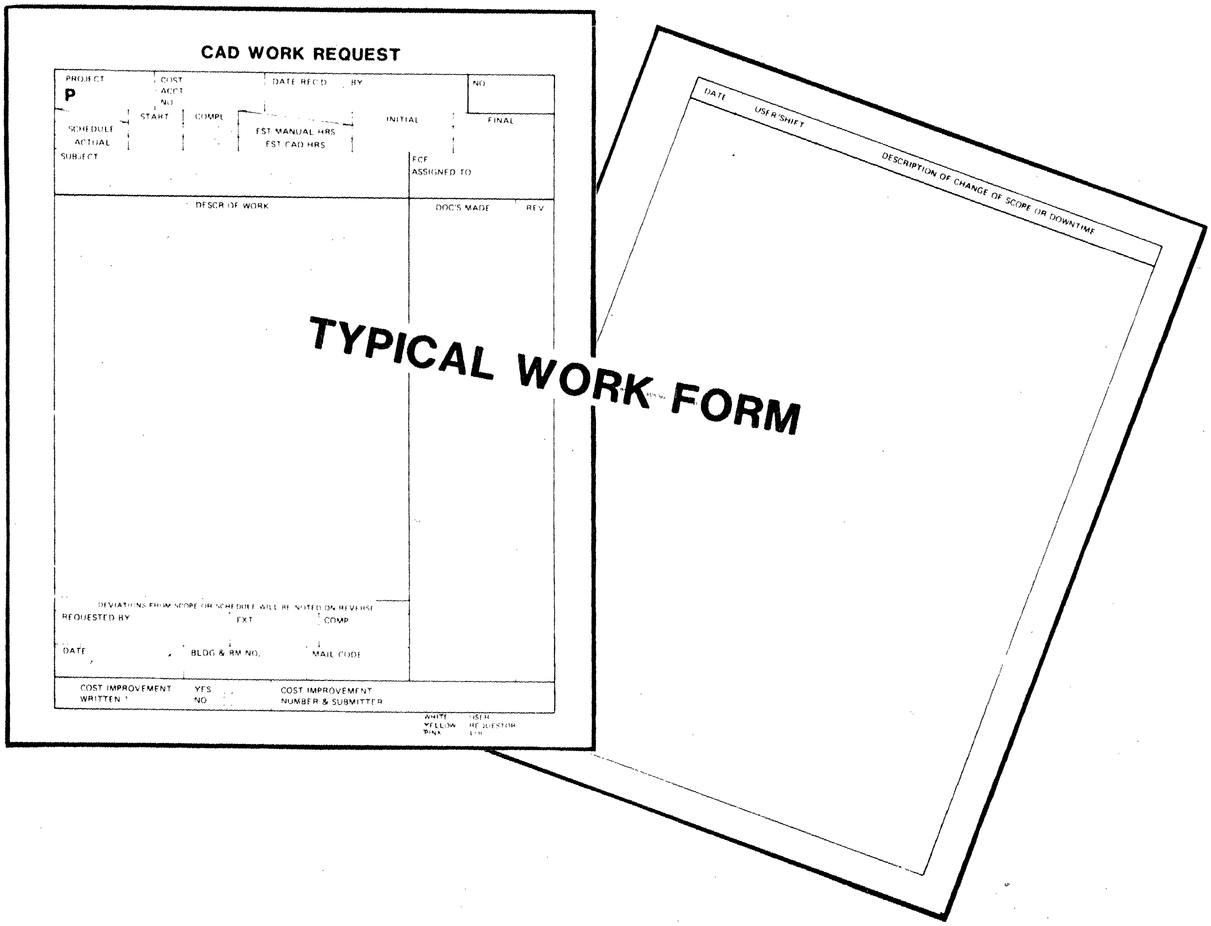
- COMPLEXITY
- REPORTS
- DEMONSTRATIONS

+

TIME

- DUE DATE
- MAN HOURS
- CAD AVAILABILITY

= JOB SCHEDULE



CAD WORK REQUEST

PROJECT P	COST ACCT NO	DATE REC'D	BY	NO
SCHEDULE ACTUAL	START	COMPL	EST MANUAL HRS EST CAD HRS	INITIAL	FINAL
SUBJECT				FCF ASSIGNED TO
DESCR OF WORK				DOC'S MADE	REV

DEVIATIONS FROM SCOPE OR SCHEDULE WILL BE NOTED ON REVERSE

REQUESTED BY	EXT	COMP
DATE	BLDG & RM NO.	MAIL CODE

COST IMPROVEMENT WRITTEN ? YES NO COST IMPROVEMENT NUMBER & SUBMITTER

WHITE	USER
YELLOW	REQUESTOR
PINK	[illegible]

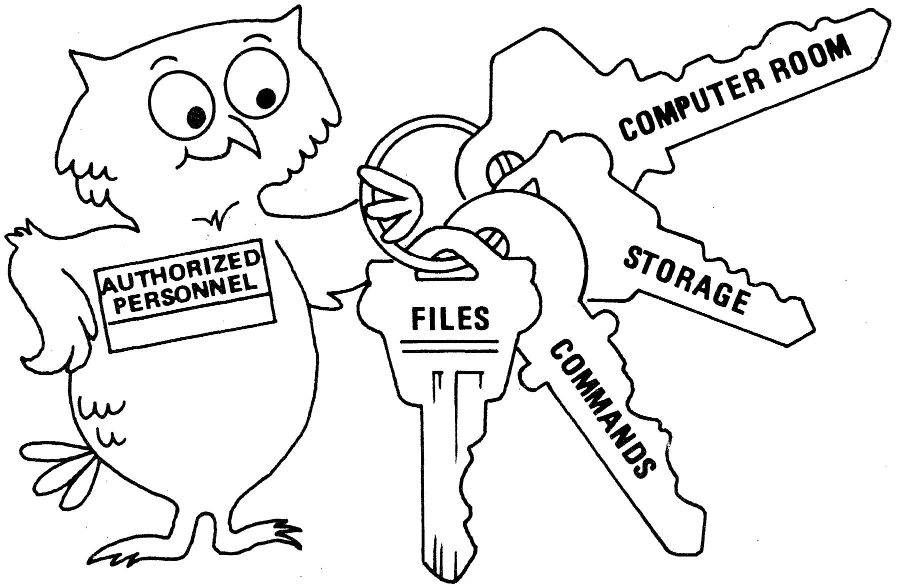
DATE	USER SHIFT	DESCRIPTION OF CHANGE OF SCOPE OR DOWNTIME

TYPICAL WORK FORM

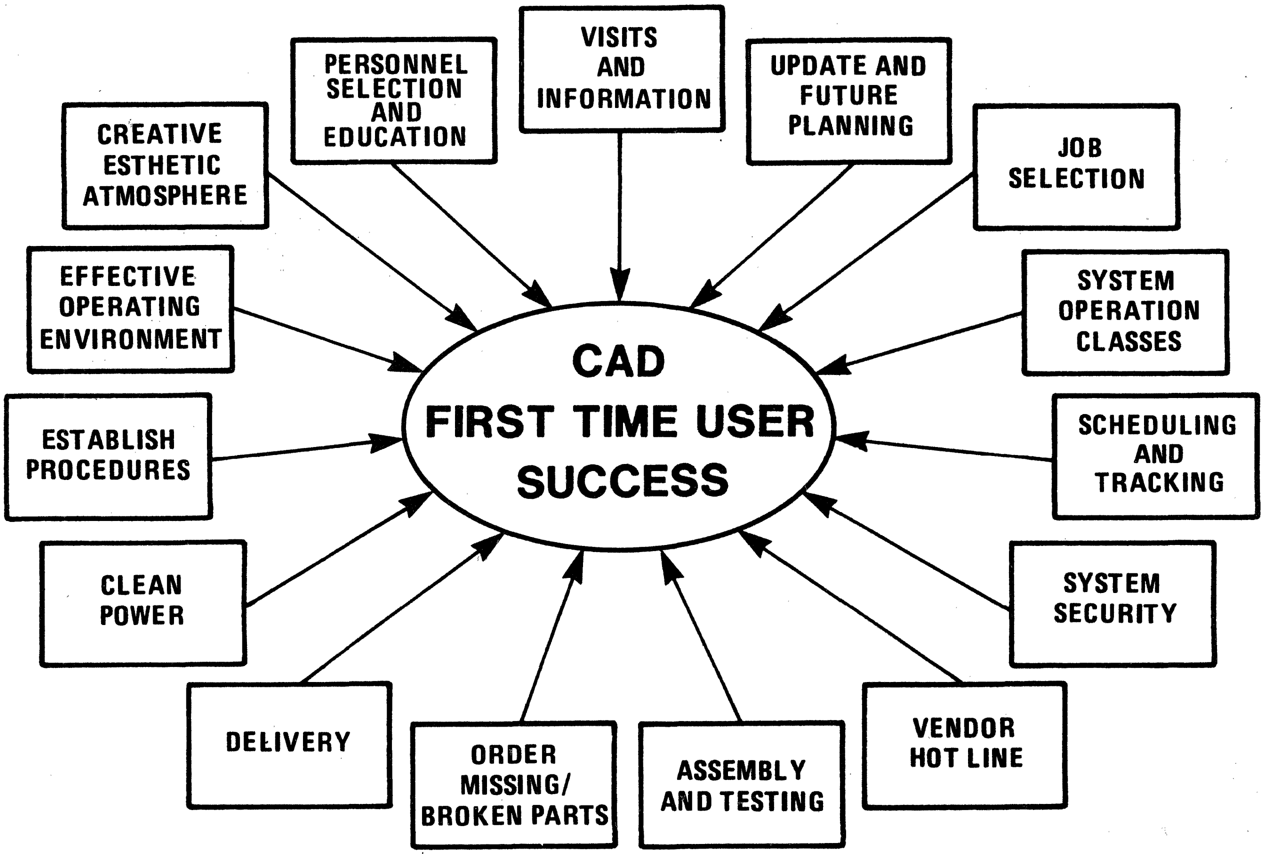
SECURITY

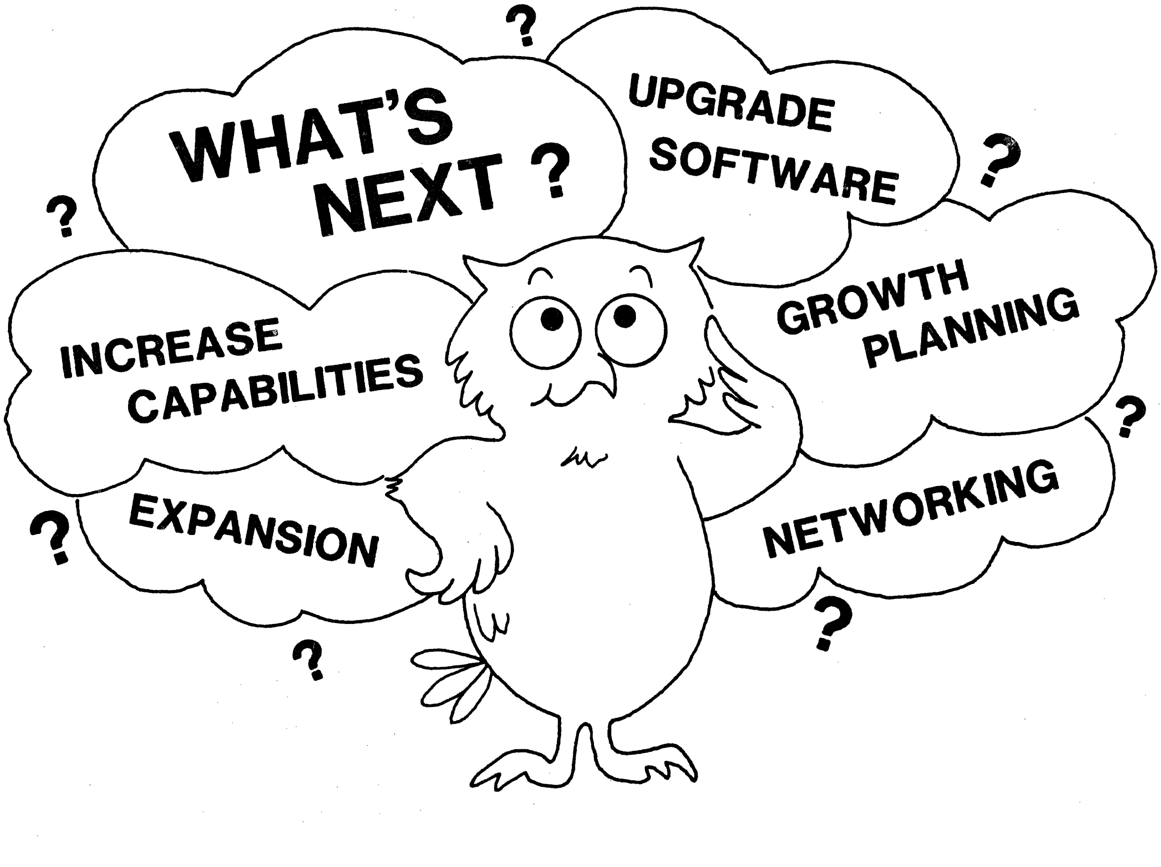
- ● PHYSICAL
- ● DATA

ENVIRONMENTAL MAINTENANCE AND SECURITY



- ✓ DOORS LOCKED
- ✓ AUTHORIZED PERSONNEL
- ✓ POWER CONDITIONS
  - ✓ AIR CONDITIONER
  - ✓ CAD SYSTEM
- ✓ EMERGENCY NUMBERS


CAD FIRST TIME USER SUCCESS
VISITS AND INFORMATION
UPDATE AND FUTURE PLANNING
JOB SELECTION
SYSTEM OPERATION CLASSES
SCHEDULING AND TRACKING
SYSTEM SECURITY
VENDOR HOT LINE
ASSEMBLY AND TESTING
ORDER MISSING/ BROKEN PARTS
DELIVERY
CLEAN POWER
ESTABLISH PROCEDURES
EFFECTIVE OPERATING ENVIRONMENT
CREATIVE ESTHETIC ATMOSPHERE
PERSONNEL SELECTION AND EDUCATION


WHAT'S NEXT ?
UPGRADE SOFTWARE
GROWTH PLANNING
INCREASE CAPABILITIES
EXPANSION
NETWORKING
?
?
?
?
?
?
?

PRODUCTIVITY AND PROFITABILITY WITH CAD/CAM

Chuck Cummings
Babcock & Wilcox
20 S. Van Buren Avenue
Barberton, OH 44203
216-753-4511

ABSTRACT

The purpose of this paper is to examine productivity and profitability with CAD/CAM. It will provide a simple framework to measure the tangible savings of CAD/CAM, determine the economic "break-even" point based on those savings and their associated costs, look at the things to which profitability is sensitive, and point out a problem with intangible benefits.

INTRODUCTION

CAD/CAM systems are almost always justified on the basis of increased drafting productivity and a series of intangible benefits. Looking through the current literature will show that the only "tangible" savings commonly attributed to the use of CAD/CAM is a savings of drafting man-hours. Furthermore in most cases, the "intangible" benefits (improved design, shorter product cycle, flexibility, etc.) are the real incentive to purchase a system. A serious confusion exists between the concepts of productivity and profitability. If the price of increased productivity is very high, perhaps we can't affort it. Typically, it is claimed that as a pure drafting tool a CAD/CAM system is a marginal investment; but when intangible benefits are considered, it becomes a very good investment. The implication is that intangible benefits make CAD/CAM systems profitable. This may or may not be true. It certainly hasn't been proven. I believe it is true, however, and the challenge is to prove it.

A FRAMEWORK FOR MEASURING TANGIBLE BENEFITS

To begin we will look at the tangible savings from using a CAD/CAM system; namely, increased drafting productivity. We will design a simple framework for measuring the increase.

For the sake of simplicity, we will assume that there is a CAD/CAM system that has six workstations and all other required hardware and software. This system was purchased by a company (fictitious) for a total of $600 thousand. The hardware cost $550 thousand and the software $50 thousand. A room was prepared for the system at a cost of $50 thousand. The company paid cash for all of this and believes it will all last for five years.

Over five years, the total system cost is:

$550 + 50 + 50 = $650 thousand

The system, both the hardware and the software, will be maintained by the vendor and to start, the company will pay 1% per month of the cost of the hardware for the maintenance. The maintenance contract will be negotiated every two or three years and the average escalation in the cost will be 2% per year. (That is a pretty good deal.) 1% of $550 thousand is $5500 per month to start. Over five years, the total cost of maintenance is:

$(5.5)\ (12)\ (1+1.02+1.02^2+1.02^3+1.02^4)$ = $343.5 thousand

Supplies (plotter paper and pens, magnetic tapes, etc.) will cost $400 the first year and will escalate 10% per year. Over five years the total cost of supplies is:

(4) (6.1051) = $24.4 thousand

Since a number of things in this example will be escalated at 10% per year, note that the total value of $1. escalated over five years is:

$(1.)\ (1.+1.1+1.1^2+1.1^3+1.1^4)$ = $6.1051

Also note that the average value of $1. escalated over five years is:

6.1051/5 = $1.22102

This is only the first system. If things work out well, there may be more. There is a person who keeps track of all of that. He is called the CAD/CAM man. He costs the company $35 thousand a year. There is another person, the system manager, who schedules the terminals, makes sure everything is running properly and much more. That person costs the company $30 thousand a year. There is a technology exploiter to make sure CAD/CAM is being used as well as possible — $25 thousand a year, and finally a clerk (plotting, file maintenance, etc.) for $20 thousand. These people all attend user group meetings, seminars, etc., so, they have fairly heavy expenses — 15% of their costs above. Taken together they are overhead personnel chargeable to CAD/CAM. This will be referred to as people costs. Over five years, the total cost of people is:

(20+25+30+35) (1.15) (6.1051) = $772.3 thousand

That is almost three quarters of a million dollars and appreciably more than the cost of the system.

Over five years, the total cost of CAD/CAM is:

System	$ 650 thousand
Maintenance	343.5
Supplies	24.4
People	772.3
TOTAL	$1790.2 thousand

The average cost per year is:

$1790200/5 = $358,040.

If one shift at a workstation is 2000 hours per year and the workstations are available for productive use 95% of the time, then the total workstation time available on the system is:

(6 workstations) (.95) (2000) = 11400 hours per year

The cost of CAD/CAM itself for a one-shift operation is:

358040/11400 = \$31.41 per workstation hour

This is based on salary and fringes plus expenses for overhead personnel.

The primary economic justification for CAD/CAM is increased drafting productivity. The change in productivity, Rp, is expressed as the ratio of the hours required to do a job manually versus the hours required using a system.

$$\text{Rp} = \frac{\text{manual hours}}{\text{hours on system}}$$

THE BREAK-EVEN PRODUCTIVITY RATIO

If productivity increases slightly, say 10% (Rp = 1.1), the increase is going to be very expensive for the company. If it increases greatly, 1000% (RP = 10), then the company will make a lot of money. There is a point at which CAD/CAM will just pay for itself, the company will simply break even. At that point, we have the break-even productivity ratio, R.

The hourly cost for doing a job on CAD/CAM is the cost of the person for an hour plus the cost of the workstation hour. The hourly cost for doing the job manually is the cost of the person for the hour. The ratio of those two is:

$$\frac{\text{Cost of workstation hour + cost of person hour}}{\text{Cost of person hour}}$$

We already know that the workstation costs \$31.41 per hour. Currently, an operator costs the company a total of \$11 per hour; but, like overhead personnel, his cost escalates at 10% per year. The average value of a dollar escalated at 10% over five years is \$1.22102, so the average cost of an operator is:

$$(11)\,(1.22102) = \$13.43 \text{ per hour}$$

The above ratio, then is:

$$\frac{\$31.41 + 13.43}{13.43} = 3.34$$

It costs the company 3.34 times as much per hour to do a job on a CAD/CAM system as it costs to do it by hand. If the operator can do 3.34 times as much work in that hour for 3.34 times the cost, then the company will break even, so the break-even productivity ratio in general is:

$$R = \frac{\text{Cost of workstation hour + cost of operator hour}}{\text{Cost of operator hour}}$$

and in this example:

$$R = \frac{31.41 + 13.43}{13.43} = 3.34$$

CAD/CAM must make it possible to do the drafting work 3.34 times as fast for the company to break even.

If the system were used two full shifts, then:

$$R = \frac{15.70 + 13.43}{13.43} = 2.17$$

SENSITIVITY ANALYSIS

If we call the cost of a workstation hour, W, and the cost of an operator hour, O, then:

$$R = \frac{W + O}{O}$$

The cost of a workstation hour is the total cost of CAD/CAM divided by the total number of workstation hours available. Call the total cost of hardware, software, site, maintenance and supplies, S. Call the cost of people, P and the number of workstation hours, H. Then:

$$R = \frac{(S+P)/H + O}{O} = \frac{S+P}{OH} + 1$$

R is the productivity ratio required for the company to break even. If R is small, the increase in productivity required to break even is small and likewise, the productivity increase required to be profitable is small. Profitability is inversely proportional to R. As R gets smaller, profitability gets larger. Therefore, to increase profitability, we want to decrease R. How do we do that? What is R sensitive to?

$$R = \frac{S + P}{OH} + 1$$

Any change in S, P, O or H will cause a change in R. As the cost of hardware, software, site maintenance and supplies increases, R increases and profitability decreases. As the cost of overhead personnel increases, profitability decreases. On the other hand, as the cost of operators increases or the number of hours of use increases, profitability increases. Based on this analysis, then, to increase profitability, do the following:

1) Decrease S. Buy the cheapest system that will do the job. Negotiate the least expensive maintenance contract possible. Spend as little as possible on the site. Use supplies as sparingly as possible.
2) Decrease P. Reduce the number of overhead personnel.
3) Increase O. Save man-hours for the most expensive operators possible. All this means is that the more expensive the operator, the more profitable CAD/CAM is.
4) Increase H. Use the system as many hours as possible. Minimize downtime.

It should be noted that as the number of overhead people increases, the number of workstation hours available for productive use decreases because overhead people tend to use workstations. Both of these things depress profitability, so that increasing the number of overhead people has a doubly bad effect. Overhead personnel should never be added without justification! (See Appendix)

Some of these actions are incompatible. For example, it probably isn't possible to minimize both the cost of maintenance and downtime. Some common sense is helpful, too. It isn't reasonable to give all drafters a big increase in salary to save more expensive operator hours.

A PROBLEM WITH INTANGIBLES

In the one-shift operation in the earlier example, the break even productivity ratio was 3.34. If the company is doing mechanical drafting and using the system as delivered, that may be very difficult to achieve. It may be impossible to achieve in which case the company will continue to lose money on drafting. However, the drawings are cleaner and neater and more consistent. They are also more accurate. They have fewer errors and they are completed more quickly. All of those things have an appreciable amount of value, but it is intangible — it is difficult to judge the dollar savings represented. Nevertheless, when added to the productivity increases in drafting, it clearly makes CAD/CAM very profitable. Or does it?

If we look at the equation implied, we see that something is missing. Let tangible savings be S_t and tangible costs be C_t and intangible benefits be S_i. Then:

$$\text{Profit} = S_t - C_t + S_i$$

A little trickery was used to illustrate a point. S was used for both tangible savings and intangible benefits as though savings and benefits were the same thing. They are not. When we speak of tangibles we understand that benefit means net savings or total savings minus total costs. When we speak of intangibles we imply that the net value is positive — total savings are greater than total costs, or more likely, we completely fail to recognize that there are any intangible costs. Just as there are tangible costs associated with tangible savings (cost of CAD/CAM associated with the savings in drafting hours), so there are intangible costs associated with intangible savings. The assumption that the net intangible savings — the intangible benefit — is positive, can easily be challenged. It needs to be investigated. A more accurate equation is:

$$\text{Profit} = S_t - C_t + S_i - C_i$$

A couple of examples will clarify what sort of things are intangible costs.

The management of the company in the earlier example recognized that the attitudes of the operators would have an effect on their productivity. If they felt intimidated and unhappy, they wouldn't be very productive; but if they felt confident and happy, they would be very productive. Since this was the first system it was important to make a good showing. Therefore, in order to instill confidence for the 12 drafters chosen from among the 110 available, they were given new job titles. To make sure they were happy, they were given an extra two dollars an hour. Surely enough, they became confident and happy and very productive. Although the effect of their attitudes on productivity is intangible, it is real, and the system has an overall productivity ratio of 3.5, which makes it not only productive but also profitable. The best that could have been expected with poorer operator attitudes would have been 3, which would make the system unprofitable. Alas, when the other 98 drafters heard about the job titles and the extra two dollars an hour, they became very unhappy and productivity at the drafting board fell sharply. In fact, overall drafting productivity is way down. The intangible savings — increased productivity because of high morale among the twelve operators — has been more than offset by the intangible cost of low morale among the other 98 people. The intangible benefit is definitely negative.

Another company, a small one, has even worse problems. When they installed CAD/CAM, they recognized that they would cut the engineering time by 35%. They went out and told all of their customers. Their customers were very happy. The company even got a lot of new contracts. Now however, the drawings from the system are piling up in the mail room because the shops aren't ready for them. They have searched in vain for qualified welders. Without them they have a manufacturing bottleneck. They have had to build an expensive new facility for the inventory of raw materials that is accumulating. The customers are now angry and manufacturing is really feeling the pressure. Morale there is very low, which is delaying shipments even more. The intangible saving was shorter engineering time. The intangible costs are extreme.

The point is not that intangible benefits are always negative, but rather that they must be investigated. Intangible savings are difficult to measure. Intangible costs are much worse, but if everyone is aware of them, we will certainly learn more about them. As intangibles become clear, they become tangible. When that happens, we can prove how profitable CAD/CAM really is.

CONCLUSION

The normal measure of CAD/CAM system productivity is the ratio of the hours to do a given task on the system, to the hours required to do the job by hand. We may require, however, that the job be done not only faster but also cheaper. Profitability is another measure of the "goodness" of CAD/CAM and one that gets a lot less attention. To a great extent, profitability is presented in terms of intangible benefits. But, there is a problem with intangible benefits; namely, intangible costs which are almost always ignored. In order to understand the true benefits of CAD/CAM, we must look more closely at intangibles. As we do we will learn more about them and some intangibles will become tangible. The technology and all of us involved with it will be better off for the effort.

APPENDIX

Clearly, there is a conflict between the short term goal of making a CAD/CAM system as productive and profitable as soon as possible and the long term goal of making it as productive and profitable as possible; namely, overhead people. In the short term, the cost of overhead needs to be minimal to achieve the best immediate results. However, overhead people do things that can greatly enhance the benefits of the technology. The company in the example has a CAD/CAM man. They are planning to expand the technology as widely as it will be beneficial. Without the CAD/CAM man, that would be undirected and possibly disastrous. The Technology Exploiter has already opened up lucrative new channels by providing links to the ''main-frame'' finite element programs and to the numerical control operations. In the long run these benefits will far outweigh the cost of the people involved. All decisions concerning overhead personnel should be considered very carefully and include both short term and long term ramifications.

CADDS 4 REV 2 —
A USER'S REVIEW EVALUATION

William G. Beazley, Ph.D.
Marine Division
Brown & Root, Inc.
Houston, TX 77001

1. INTRODUCTION

Computervision Corp. plans to release their new CADDS 4 REV 2 software sometime in July, 1981. The Marketing buildup and heightened user's expectation of the performance of CADDS 4, particularly REV 2, is based in part on the widely recognized limitations of CADDS 3. The author has had an opportunity to work both with CADDS 3 and CADDS 4 REV 2, and a comparison of these two software packages can guide others in selecting the one appropriate for their application.

The present review will begin with the limitations of CADDS 3 addressed by CADDS 4. The second revision of CADDS 4, meant to meet or exceed the performance of CADDS 3 REV 11, will then be discussed in terms of its practical utility. Finally, the remaining areas of improvement for CADDS 4 will be reviewed, with suggestions for future work. This paper will exclude issues such as bugs or minor command modifiers, and concentrate on the conceptual issues of the software packages themselves.

2. CADDS 3 — CADDS 4 COMPARISON

CADDS 3 is a popular CADDS software package developed principally to run on CV's CGOS 100/10 Operating System and a CGP100 CPU (developed from CV's experience with the Data General NOVA minicomputers). It has limited support for a variety of 3D data entry and manipulations, and excellent support for the more common 2D computer-aided drafting applications. Prominent among these are the schematic drawings using small symbol libraries and drawing extracted reports, i.e. PC/ES, P&ID, etc. The support for MD capabilities are typical of the wire-frame geometric approach; lines, points, strings, with no particular user use of their connective associations in any general way (except in associative dimensioning and chaining digitizes). Speciality applications, such as, Finite Element Modelling (FEM) are added to the existing entity support as a kind of forced retrofit, but with acceptable results for most users. The main mechanisms for a particular CADDS 3 drawing to "borrow" data from another drawing, were via drawing copying and editing, INS PART FIG and NFIG commands, layering conventions (one drawing per layer set), and selected data extraction procedures, i.e. MERGE NET. The most popular mechanism by which engineering data was partitioned into separate drawings was, simply, to have separate CADDS parts for each one. This approach required endless waves of drawing updates with each design change.

For these and other limitations CV took a fresh approach, CADDS 4. A summary of the limitations of CADDS 3 which CADDS 4 appears to address is given in TABLE 2.1. The introduction of CADDS 4 involved the use of a new processor (GPU) and a new type of terminal (Instaview), both discussed here as CADDS 4 features. The limitations listed in TABLE 2.1 will now be briefly discussed.

2.1 MODEL/DRAWING DICHOTOMY

The single feature of CADDS 4 which separates it from CADDS 3 is the introductions of the Model/Drawing partitioning on database graphics entities. It is a clear recognition that a difference exists between the information necessary to describe a physical object and the numerous drawings and documents necessary to record this data on paper. The model and model entities are data describing this object and visually displaying it in approximate form to the user. The drawing entities are the appearance the model and associated data has in the drawing. This means that the model must be made available to all draftsmen drawing up the design while they create the drawing itself. Only one user may be in the model mode, where the model itself can be changed. All others must remain in the drawing mode, where only drawing entities can be added.

Table 2.1:
Apparent Impact of CADDS 4 on CADDS 3 Problems

CADDS 3 Problem	CADDS 4 Solution
1. Model/Drawing Dichotomy	1. Model/Drawing Modes
2. New Orientations Difficult, Slow	2. ICU (Image Control Unit) Temporary, Permanent Orientations
3. No Independent Coordinates	3. Construction Plane Concept
4. Limitations of Repaint Technology	4. GPU/VGU Raster Refresh Technology
5. Wire Frame or Symbolic Graphics	5. Solids Modelling, Generate DETAIL Command
6. Inconsistent Punctuation	6. Consistent Punctuation
7. Few System Commands While in CADDS	7. Resume System, Resume CADDS
8. FORTRAN, PEP, VARPRO, PAREX Limitations	8. VARPRO2
9. Response Time	9. Faster Commands, Distributed Processing

2.2 GENERATION OF NEW ORIENTATIONS

The two main uses for obtaining a new orientation of something being drawn are to draw a view more clearly showing important detail and better interactive identification of entities on a cluttered display. Rotating to a new orientation in CADDS 3 invoked a REG REG via the SEL VIEW command. This step is eliminated after the first view is activated by direct operations of the Temporary View File (TVF) using the Image Control Unit (ICU) at the display. Temporary view manipulations of the views may be made permanent.

2.3 INDEPENDENT COORDINATES FOR DATA ENTRY

CADDS 3 lacks the ability to enter spatial data in terms of an independently defined coordinate system. (The DEF VIEW command sets no origin for data entry.) CADDS 4 introduces a Construction Plane Concept which solves this problem. The author used this CADDS feature to construct a CADDS FEM part for the tubular structural joint shown in Figure 2.1.

2.4 STORAGE TUBE TECHNOLOGY LIMITATIONS

CADDS 3 uses a regen file approach developed to support a storage tube hardware configuration requiring frequent repaints. CADDS 4, when used with the refresh-type raster screen Instaview terminal, uses a special processor GPU and direct video transmission from VGU Boards located at the computer to change screen appearance much faster. The limitations of raster displays themselves are well known and will not be discussed here.

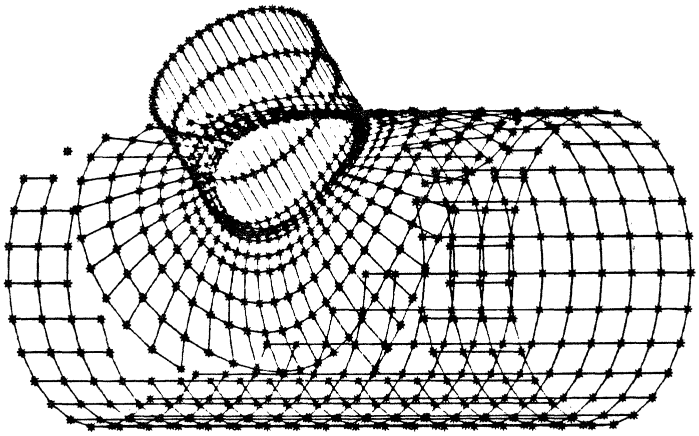


Figure 2.1: Finite Element Model of Structural Tubular Joint Generated Using Two User-Defined Construction Planes.

2.5 WIRE FRAME GRAPHICS

Wire frame graphics, quite acceptable for most Mechanical and symbolic applications, require extensive editing for certain 3D applications requiring pictorially correct views, i.e. Plant Design and Architecture. CADDS 4 has introduced GENERATE APPEARANCE and GENERATE DETAIL commands which, consistent with their notion of drawings as representations of data models, examines selected model entities and generates their true appearance. This solids modelling is a new departure from the limited surface modelling and fonting capabilities of CADDS 3.

2.6 INCONSISTENT PUNCTUATION

CADDS 3 is famous for its liberal use of contractions, abbreviations and punctuations conventions. CADDS 4 has made this more consistent, although changing verbs and nouns liberally and adding hierarchical modifiers to many commands. The author views the productivity gain here as a probable draw.

2.7 SYSTEM COMMAND ACCESS

CADDS 3 suffered from endless bouncing back and forth between system and CADDS level to use features in both. CADDS 4 may now be suspended to allow full use of system commands. Unfortunately the menu is suspended also, but this is a minor limitation.

2.8 FORTRAN, PEP, VARPRO, PAREX LIMITATIONS

The main vehicles for drawing automation for CV are FORTRAN, PEP, VARPRO and PAREX. A complete discussion of the limitations of these routines is outside the scope of this paper. The principal tradeoff presented to the applications programmer was ease of programming vs. full use of CADDS commands. VARPRO2, with text file reads and writes, vector operations, FORTRAN routine calls, and other features appears to address these shortcomings.

2.9 RESPONSE TIME

The worst criticism of CADDS 3 is (at least for the author) response time. This is an outgrowth of several of the factors discussed above. Depending on the command, CADDS 4 is much faster than CADDS 3. This has been accomplished mainly by two improvements. First, much of the computational overhead has been distributed to the GPU which has its own memory and more direct access to the terminal displays. Tests by the author on a 5,000 line drawing executing a zoom all (ZOOM VIEW ALL) command showed CADDS 4 REV 1.01 response time to be about 25% of CADDS 3. Secondly, many of the commands themselves have been rewritten for speed, though more preparation is required in library generation (PREP NFIG, etc.). Response time has increased, however, for some commands from CADDS 4 REV 1 to REV 2. CV must continue to regard response time as one of CADDS 4's principal merits.

3. CADDS 4 REV 2

A preliminary list of CADDS 4 REV 2 enhancements may be found in TABLE 3.1. Brown & Root has experimented with preliminary versions of most of these enhancements, with the exception of the numerical control applications. The overall objective of REV 2 was to incorporate the features generally present in CADDS 3 REV 11. Selected enhancements will now be reviewed.

3.1 FINITE ELEMENT MODELLING

FEM software has now become a much more generalized package with user-definable elements, checking aids (element shrinking) and several methods of displaying analytical results (i.e. distorted shape, mode shape, time response display, etc.). Grid points are tagged at insertion, but numbering cannot support NASTRN standard 8 digits. These capabilities are not well integrated into the database. Separate commands must be used for moving and copying. You cannot dimension to grid points and you apparently have no control over grid point symbol size (currently too large). This will be an exceptionally strong product if these and other deficiencies are corrected.

3.2 MD DIMENSIONING

Dimensioning is definitely improved with modifiers permitting alignment of dimensions across the page (good practice), and conformance to JIS and IOS dimensioning standards. Still, conformance to good practice in the United States is necessary on items such as solid isometric dimensioning, selective editing of witness lines and editing to dimension text and symbols (i.e. follows the dimension in some shops).

3.3 VARPRO2

A major change in philosophy of applications programming by users is VARPRO2, already discussed. This is a much more powerful VARPRO, following the variable execute file generation approach. The use of online variables in CADDS 4 makes this a strong feature.

3.4 PIPING

A major feature of the REV 2 release of CADDS 4 will be the Piping and Plant Design software features. These commands are simpler than CADDS 3 REV 11 though they cannot be simplified beyond the complexity of plant design itself. Piping has evolved in CADDS 4 into a family of commands obeying a CV set of property conventions on entities. To use the full range of capabilities, one cannot deviate from these conventions, hence, the users problem is now one of converting CV format results into in-house formats. It would be almost unthinkable to attempt to reconstruct the fittings library, for example.

New features have been added, primarily, automatic hidden line removal using a solids modelling approach (see Figure 3.1) and a pipeline extraction command for isometric and spool drawings. But with these come new questions. How large a plant can be successfully modelled on CV? What is the impact of the additional drawing entities generated for each drawing, also stored in the same file as model entities? How are the large but distinct plant areas, certain to be designed on separate CV systems, to be combined into a single database?

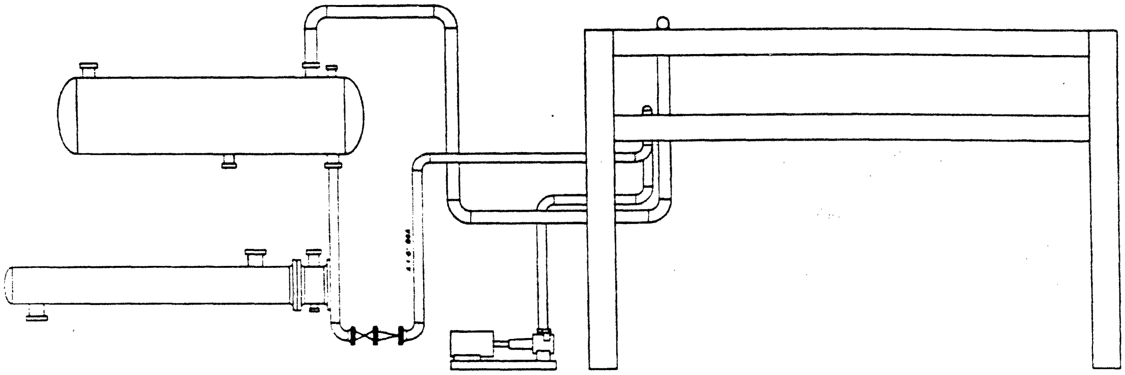


Figure 3.1: Hidden-Line Detail Generated from Solid Geometric Component Descriptions.

3.5 OTHER CHANGES

The fast refresh capabilities of Instaview raster technology make animation possible. The author has experimented with this for both FEM mode shapes and for two time history simulations. The FEM capability is too slow but is better controlled than the experimental interface coreloads used for the time histories. A faster approach could be along the lines of the tool path simulator where entire figures are manipulated rather than individual entities.

There may still be another change to file manipulation syntax on the operating system. FUTIL will change in the allowed delimiter character, although the old FUTIL syntax will be supported. Defaults need to be selected as the ones which create the most useful output for the least input, the most commonly expected request, the most upwardly compatible. It would be unwise to have the system command LIST ©CR□ to list the entire SYSCATLG, or CADDS 4 command LIST PART ©CR□ to begin searching the entire SYSCATLG for files ending in &PD.

The Graphics Nodal Architecture (GNA) is a new feature allowing CV systems to exchange files with a central CPU. It is expected to permit the satellite systems to download their drawing and library file storage requirements to a dedicated CPU. This is another step towards the distributed processing environment of the future. More attention will need to be given to file management of big multi-system jobs, where users may overwrite files accidentally at the central node.

4. SUMMARY AND CONCLUSIONS

The new or residual problems of CADDS 4 REV 2 are not simple to categorize. They arise, in part, because other problems are solved. The general areas of concern are now discussed.

4.1 PARTITIONING OF DATA ON LARGE JOBS

CV seems serious about doing business in Plant Design, Architecture and other disciplines where the Engineering database will be large and needs of the contributors will be larger than one system. Some means of properly partitioning these databases among several systems must be found.

4.2 GENERALIZATION OF WHAT CONSTITUTES A MODEL

The Engineer constructs models quite unlike the lines and circles currently used in Mechanical Design. These might be Finite Element Models, Kinematic Chains, etc. For example, Figure 4.1 shows a FEM model that could be utilized in structural drafting steps following Engineering. These models should be capable of generating the drawings detail necessary for fabrication.

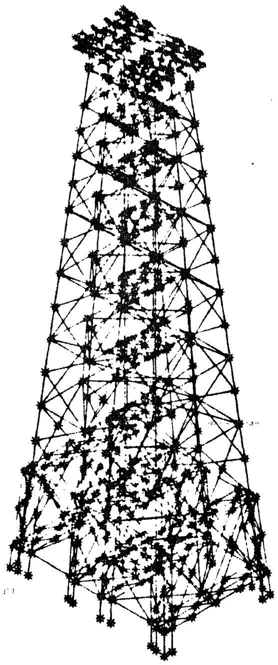


Figure 4.1: Finite Element Engineering Model Typically Underlying Offshore Structural Design.

4.3 BETTER 3D MANIPULATION SUPPORT

Better support of the user in both direct view manipulation and man-machine efficiency sense is needed. Commands should be added permitting rotation of views and scrolling using both model and drawing entities digitizes and maintaining locational relationship between them if desired. Orientation feedback aids such as the construction plane symbol cannot disappear when dynamic view is entered, as well as grid. A joystick might do better for rotations than buttons.

4.4 CONCLUSIONS

CADDS 4 and in particular REV 2 will probably transform the way Engineering is done by the very power of the model/drawing dichotomy. The tendency will be for Engineering designers to construct the model, while drafters detail it in the drawing mode. It truly represents the next level of attainment in a drawing-as-database approach to Computer Aided Design. It will be adequate for smaller model designs common in Mechanical Design, schematic drawings and similar applications.

The extensions to the next level is a concern, however. Serious questions should be raised about the extensibility of the CADDS 4 concept into the larger design disciplines it is aimed. The author expects the software to change at a rate approximately appropriate with its acceptance in these disciplines. The users, however, must play a large part in guiding CV toward this goal.

5. ACKNOWLEDGEMENTS

The examination of CADDS 4 was made possible by several individuals both at CV and Brown & Root. Particularly at CV, I would like to thank John T. Davis, Allan Vaughn, Carl Rabickow, Jr., Mike Leesley, Ernie Heimberl, Steve Handel, Randy Roy, Tom Nickerson, and many others. Here at Brown & Root, I would like to thank Paul R. Corder, Dave L. Carleton, Marie McCarthy and a number of others who helped evaluate CADDS 4 REV 2. Figure 3.1 was contributed by Dan Zubrick of Computervision Corporation. This paper was typed by Myra Fitts.

Table 3.1:
Preliminary List of CADDS 4 REV 2 Enhancements to CADDS 4 REV 1

Items Modified

Item No.	Functional Area
2.1.1.1	MD — Finite Element Modelling — User-Defined Elements
2.1.1.2	MD — Finite Element Modelling — Enhanced Mesh Generation
2.1.1.3	MD — Finite Element Modelling — Internal + Minor Updates
2.1.1.4	MD — Finite Element Modelling — External Load Definitions
2.1.1.5	MD — Finite Element Modelling — Distortion Plotting
2.1.1.6	MD — Finite Element Modelling — Shrink Elements
2.1.2.1	MD — B-Splines — Improve and Expand Math. Definitions
2.1.2.2	MD — B-Splines — Smoothing
2.1.3.1	MD — Surfaces — Fillet Surfaces
2.1.3.3	MD — Offset Surfaces
2.1.3.5	MD — Surfaces — Improve and Expand Math. Definitions
2.1.4.1	MD — Dimensioning — Japanese Industrial Standards
2.1.4.4	MD — Dimensioning — Angular Dimensions
2.1.4.5	MD — Dimensioning — Enhanced Cross-Hatching
2.1.4.6	MD — Dimensioning — Piping Features
2.1.4.7	MD — Dimensioning — Expanded Change Dimension Capabilities
2.1.4.8	MD — Dimensioning — Automatic Text Centering and Alignment
2.2.1	NC — Check Surface — 3-Axis
2.2.2	NC — MABS/NCPEP Interface
2.2.3	NC — Tool Library
2.2.5	NC — SCALLOP Ht. Improvements
2.2.6	NC — User Interface Improvements
2.2.7	NC — 5-Axis Check SURFACE MSING/MABS
2.2.9	NC — Pocket with Islands
2.2.10	NC — Surface Intersection with Tapered Cutter
2.2.11	NC — Thick offset option
2.2.14	NC — User generated subfigure in tool display process
2.2.21	NC — Lathe Macro Improvements
2.2.23	NC — Fanning
2.2.24	NC — 3-Surface Startup in MSINT/MABS
2.3.4	ED — Title-Erase changed to Delete Network (only change)
2.3.6	ED — Deferred Node Identification — Insertion of Interconnect Paths
2.3.7	ED — Title-Optional Annotation of Symbols at Insertion Time changed to Symbol Annotation; also text changes
2.3.8	ED — Merge Net/Compare Net Improvements
2.3.9	ED — Schematic Design Rules Check
2.4.1.2	System Management — Implementation of Rectangle Entities
2.4.1.3	System Management — Implementation of Shape Entities
2.4.1.4	System Management — Establishing Tags (not Properties) at Entity Insertion Time — text changed also
2.4.1.6	System Management — Only changed title from Copy NFIG to Copy NFIG/SFIG/PFIG
2.4.2.1	System Management — Report Generation — Data Extract/Data Merge Enhancements
2.4.3.5	System Management — Command Functionality — Extended NFIGs
2.5.2	Autoplacement/Autoroute Features — Autoroute
2.6	System Management — VARPRO2
2.7.1	Piping — GETDATA Improvements for PIPING
2.7.2	Piping — CHECK CLEARANCE Command
2.7.3	Piping — ROUTE PIPE Command for 3-D Pipeline Routing
2.7.4	Piping — Independent X, Y, and Z Scaling of Figures
2.7.5	Piping — INSERT FITTING Command for 3-D Piping Modelling
2.7.6	Piping — View-Dependent Drawing Graphics
2.7.7	Piping — Hidden Line Removal
2.7.8	Piping — Dimension Options
2.7.9	Piping — User-Defined Cross Hatch
2.7.10	Piping — Selection Files from Piping Specifications
2.7.11	Piping — Report Generation and Manipulation Features
2.7.12	Piping — Creation of Plant Sections
2.7.13	Piping — Isometric and Spool Drawing Creation
2.7.14	Piping — CUT PANEL and JOIN PANEL for Piping
2.9	Graphics Network Architecture — GNA (was Local Networking)
2.9.1	GNA — Star Configuration
2.9.3	GNA — Enhanced Batch Processing
2.9.4	GNA — File Archiving
2.9.5	GNA — Sub-Fig. Libraries (only changed Networking to GNA)
2.9.6	GNA — New System Administrator Functions (“ ” “)

Item No.	Functional Area
2.10.1	System Software — Support of New Peripheral Devices
2.10.3	Other System Software Features — FUTIL
2.11.1	Relationship to CADDS 3 — CADDS 3 Concurrency
2.11.2	Relationship to CADDS 3 — CADDS 4 to 3 Database Conversion
2.12.4	Mapping — On-Line Calibration
2.12.10	Mapping — Text Placement Relative to Existing Entities
2.12.11	Mapping — Property-to-Text Generation
3.1	Hardware Configuration Rules — Required Hardware
3.2	Hardware Configuration Rules — CADDS 4 and Concurrent CADDS 3/4 Memory Requirements

APPLICATIONS FOR CADDS IN THE DESIGN PROCESS

Jorge L. Perez
Vought Corporation
P.O. Box 225907
Dallas, TX 75265
214-266-2671

ABSTRACT

Computer graphics systems are being used more extensively than ever before. The introduction of these systems into the design process has been difficult for many companies. At Vought, this process is being accomplished in a timely and efficient manner.

Vought's computer graphics system utilizes the Computervision CGOS 200 using CADDS III software. To incorporate this system into the design area, Vought began by organizing a CADD System Development group. This organization is responsible for the training of operators, the ordering and distribution of hardware and software, and the writing of CAD/CAM interim operating procedures. The writing of procedures is simultaneous with the development of some designs which use CADDS. Some of the initial designs that utilized CADDS demonstrated the full capabilities of the system and were useful in establishing procedures. In addition, close coordination with manufacturing allowed maximum use of the generated database.

INTRODUCTION

In order to organize the development of Vought's CAD/CAM capabilities, a group was formed called the CADD System Development group. Job descriptions for positions in this group were formulated and an interim facility was provided. CAD/CAM departmental project managers were identified and program management office positions were filled.

This area was responsible for establishing the objectives of Vought's CAD/CAM development. Another responsibility was the development of additional CAD/CAM capabilities. Progress in the development of capabilities was made on several fronts. The Computervision graphics system in Manufacturing Engineering was upgraded to insure better utilization of generated databases. A Cyber 175 scientific computer was obtained for use in Research and Engineering. The CADDS IV graphics software package was obtained for training, development and evaluation. Electronic computer-aided design was developed. Remote CRT terminals and printers were installed at various plants. For use on common programs, a data phone link was established between Vought and Boeing's Computervision systems.

As the CAD/CAM organization began to expand, an applications laboratory was established. A program was started with Texas A & M University to develop a direct numerical control system. CAD/CAM system requirement specifications were released and data interface matrices were identified. Interface and application engineering program tasks were completed.

Training of users was another responsibility of the CAD System Development group. Basic and advanced training courses were established. A modernization of Vought's CAD/CAM technology was accomplished. A Dallas-Ft. Worth Computervision Users Group was established to provide areas users a forum for the exchange of data.

CAD/CAM FACILITIES

Vought's Computergraphics system proposes to use a Computervision Graphics Control Processing Unit (CPU) with an 80 mega-byte disk to link together the development lab, quality, logistics, materials, and facilities group. (See Figure 1)

The current system used by manufacturing uses the IBM 3033 computer with one Computervision Graphics CPU and six designer stations. (See Figure 2.) This system, located in the manufacturing area, will be expanded to use two Computervision Graphics CPUs with eleven designer stations (See Figure 3).

The current system used by engineering uses the CDC 175 computer with three Computervision Graphics CPUs and sixteen designer stations (See Figure 4). Five of these stations are located remotely, in individual design areas, for better utilization. The computer graphics system will be expanded to use a mainframe computer as a control computer host. Five Computervision Graphics CPUs will operate a total of thirty-one designer stations (See Figure 5). To obtain full utilization of Vought's CAD/CAM capabilities, groups such as structures, electronics, systems design, and the computer department would be linked together thru a mainframe computer to provide complete analysis and design support capabilities (See Figure 6).

CAD/CAM INTERIM PROCEDURES

To coordinate the work of different users areas, the CAD System Development group developed CAD/CAM interim operating procedures governing different areas of computer graphics work. These areas include computer operations, facilities provisioning, software support, software development and applications. Each of these areas has a number of procedures that apply to various aspects of the work. For instance, in the applications area, procedures have been written for the following:

- Data organization for ICG part database
- Standard item libraries for ICG system
- Technical data library
- Systems records accounting
- User training
- Handling of computer generated classified data
- NCDM drawing generated from ICG data
- CAD data preparation
- CAD data release
- CAD data revision
- Data configuration and control

Each procedure is written to define specifically the responsibility of various groups or personnel. A collation of released procedures is provided for each work area.

AIRCRAFT CONTOUR LINES

In January 1981, Vought began development of a composite access panel that would be a replaceable part for an existing metal panel. Being a replaceable part, existing contour lines had to be maintained. However, a new tool was required that would mold the composite to this contour. This new tool was to be machined on a five axis milling machine that operated by numerical control. To utilize computer graphics, the existing lines were obtained from the master drawing to create a three dimensional model. The existing lines were actually portions of conics with varying rho values, shoulder points, and end points. At each station, the rho value was computed and the shoulder point and end point coordinates were determined. This data was inserted in a 3-D format by assigning each station a certain depth. By viewing the generated 3-D model on end, a body stack was then visible (See Figure 7). Plots made of this body stack verified a perfect match to the existing master lines. Section cuts at various angles along the body stack were used to verify that the model surface matched the existing lines.

Once the body stack and section cuts were properly labeled, a 2-D master drawing was generated on a Gerber plotter using ink on mylar. The 3-D model was transferred on magnetic tape to the manufacturing area where it is being used to prepare the numerical control tapes that will machine the new tool. Further utilization of CADDS on this program was not attempted because of the small scope of the program.

BULKHEAD FITTING

In aircraft design, complex fittings that could transfer loads from one portion of the airframe to another had to be designed to join large components together. One example was this bulkhead fitting (See Figure 8). This fitting attached the vertical tail to the aft fuselage. Manufacturing was to machine this fitting from stock aluminum on a multi-axis milling machine run by numerical control. The design was done on CADDS to allow manufacturing to use the database in creating the numerical control tapes. The fitting was constructed in 3-D and was designed while maintaining close coordination with the designers working on the attaching bulkhead. At the time, this fitting was designed at a work station located in a control area near the CPUs. The designer working on the bulkhead was located in another building. While close coordination was maintained, at times it was awkward and inefficient. This eventually led to locating some terminals remote from the CPUs and in the actual design areas.

Once the 3-D model was completed, two views were dimensioned and labeled to produce a drawing. The 3-D model database was transferred to manufacturing where numerical control tapes were developed. The fitting was then machined and installed on the aft fuselage.

AIRFOIL TEST FIXTURE

CADDS was used in a program to design and build a reusable airfoil holder as part of a test fixture. The airfoil holder was made of composite materials and was composed of three main pieces. These pieces were a left hand block, a right hand block, and a collar. The three dimensional model that was created shows how these pieces are assembled (See Figure 9). The left hand and right hand block fit together over the airfoil blade to hold it in place. A small gap between the blocks assures positive clamping force on the blade. To complete the assembly a collar is pressed over the circular bottom portions of the blocks. This collar is held in place by tapered pins driven in between the collar and the blocks.

Once the 3-D model was completed, different views of the model were used to create a 2-D dimensioned drawing (See Figure 10). The collar, pins, and each block were detail dimensioned to allow machining from a solid block of composite material.

CONCLUSIONS

At Vought, the task of introducing a computer graphics system into the design process is being accomplished in a timely and efficient manner. The CAD System Development group is responsible for the development of CAD/CAM capabilities. Standard procedures are being written to organize and control data generation, storage, revision, and distribution.

In order to apply CADDS to best advantage, 3-D models should be used to produce 2-D drawings where 3-D databases can be utilized by manufacturing to produce numerical control tapes. Both dimensioned and undimensioned drawings can be produced from 3-D models. Commitments by Manufacturing to utilize the generated database in the fabrication of a part helps to substantiate the utilization of CADDS.

Proposed Development System
Facilities - Quality - Materials - Logistics

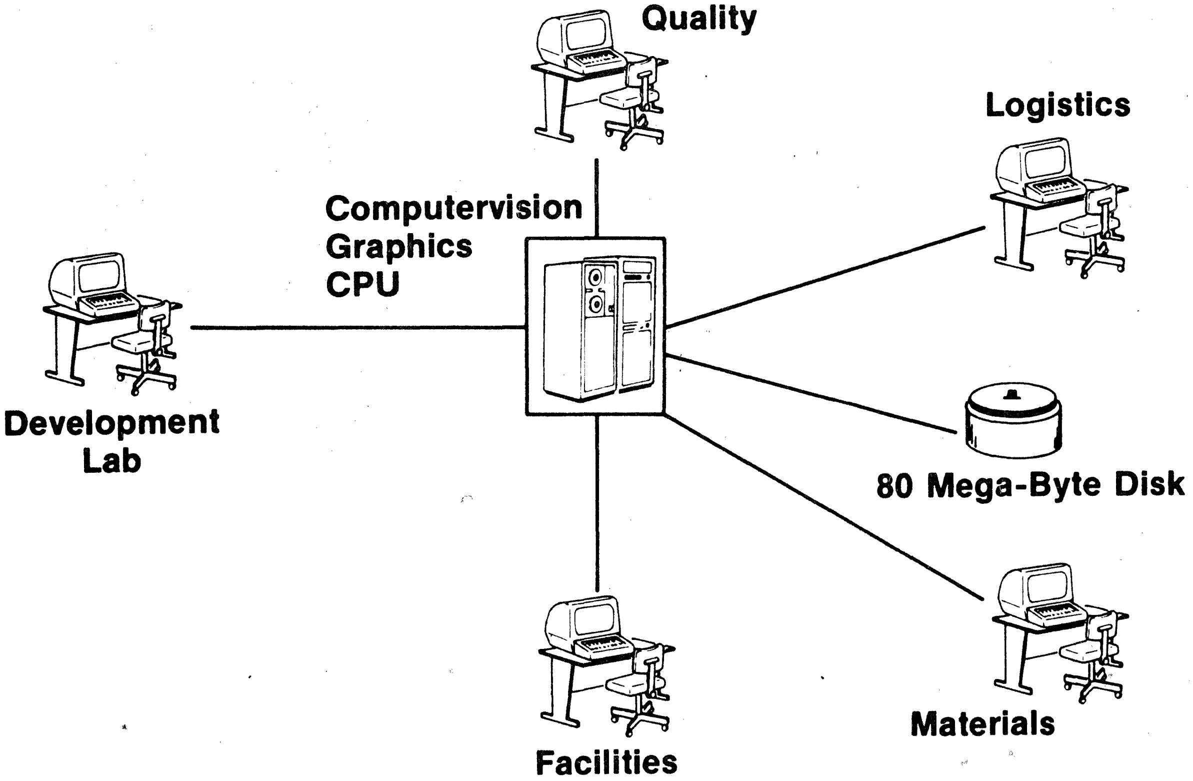


Figure 1

Manufacturing
Current System Architecture

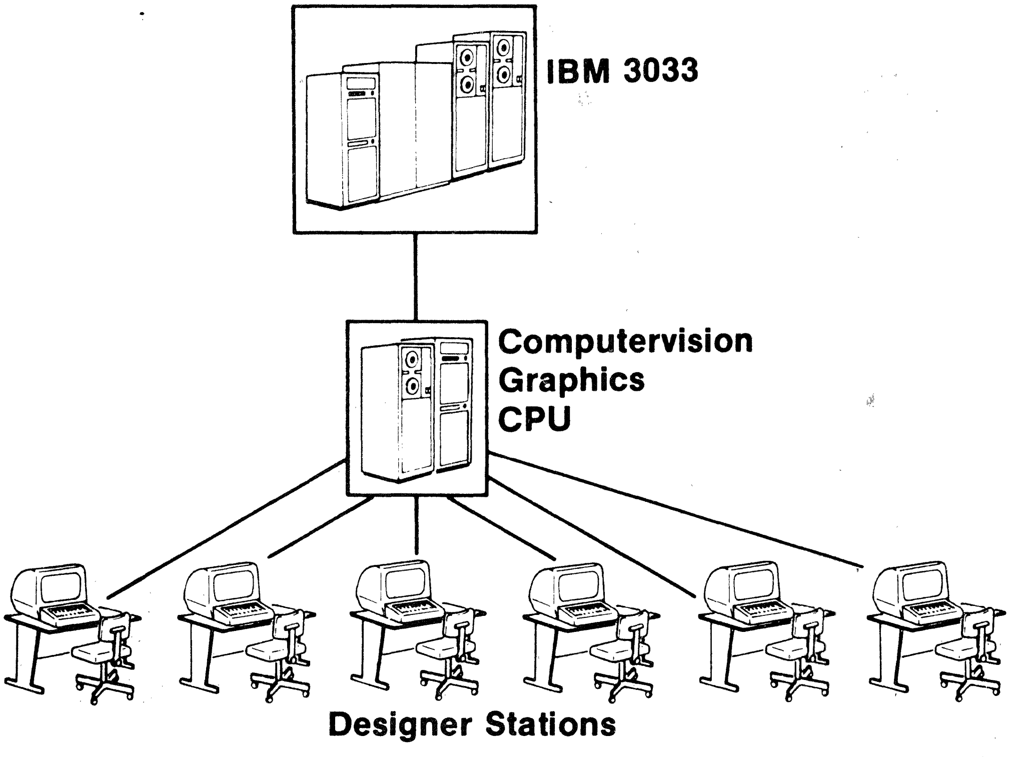


Figure 2

Manufacturing
Proposed System Architecture

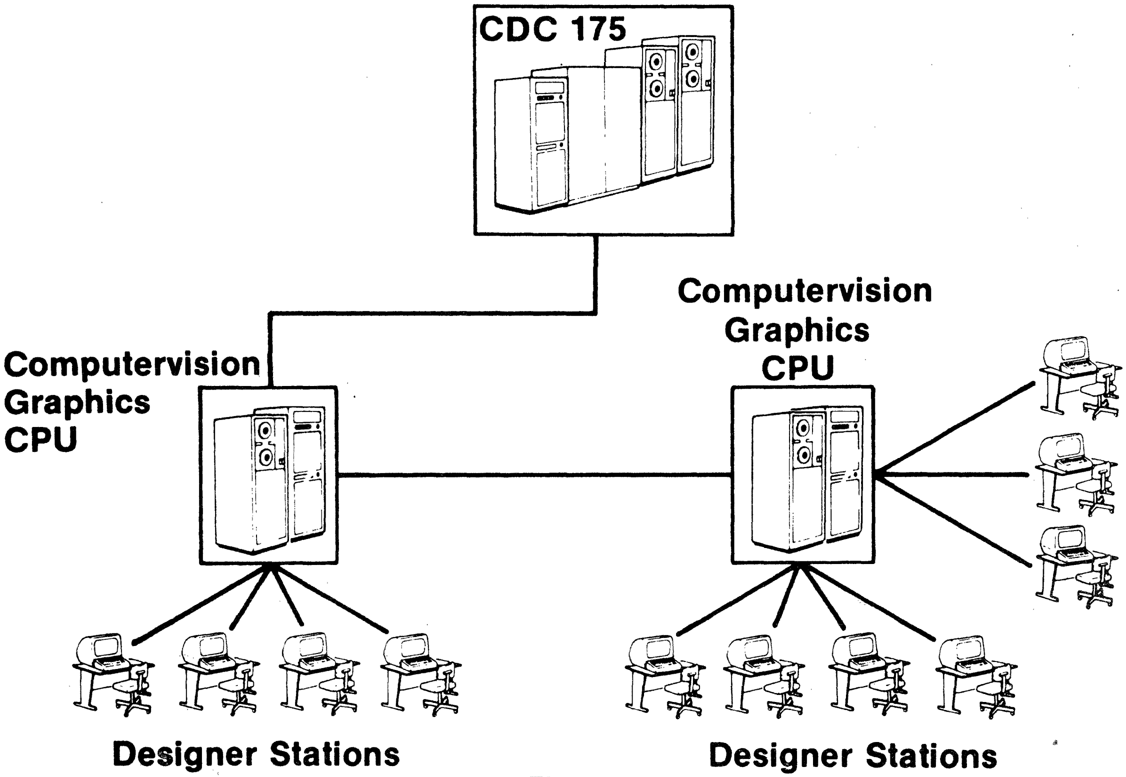


Figure 3

Engineering
Current System Architecture

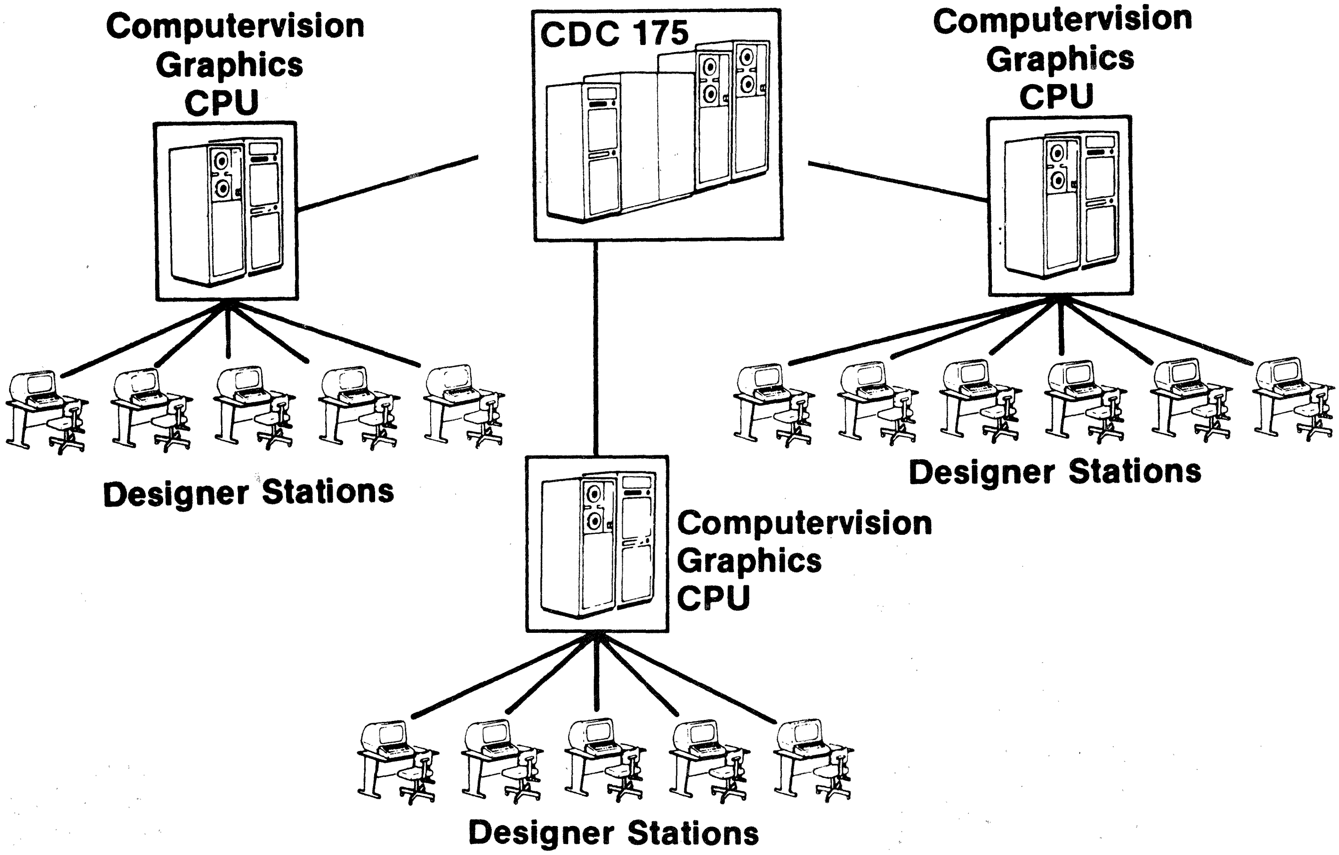

Computervision Graphics CPU
CDC 175
Computervision Graphics CPU
Designer Stations
Designer Stations
Computervision Graphics CPU
Designer Stations

Figure 4

Engineering Proposed System Architecture

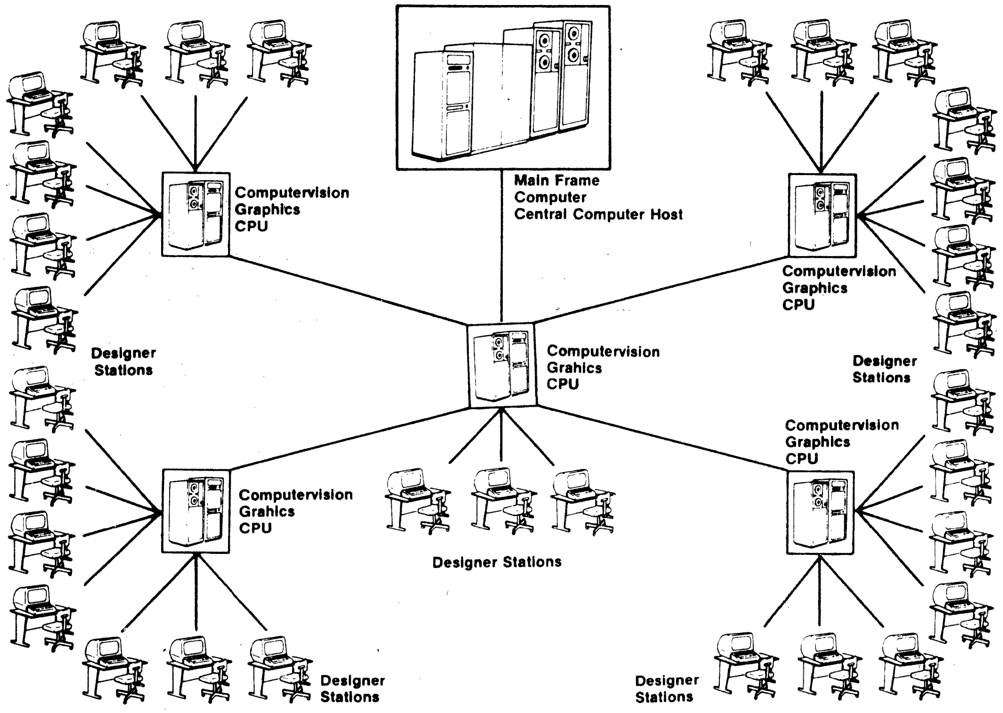


Figure 5

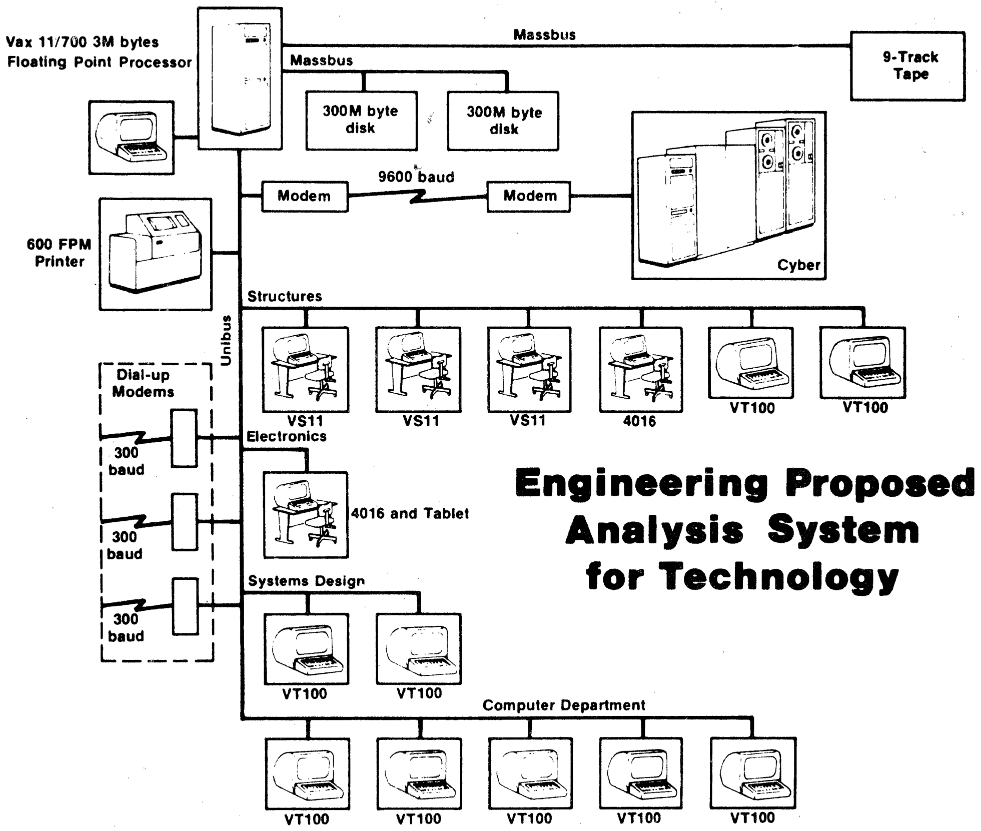


Figure 6

Aircraft Contour Lines

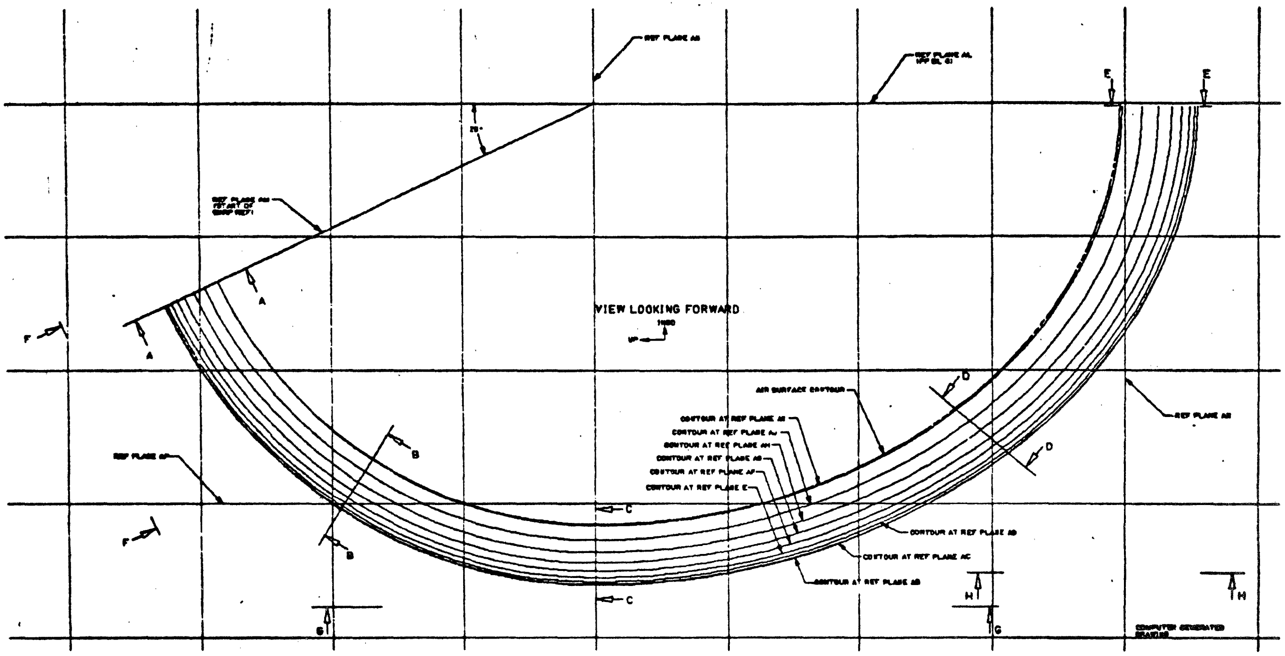


Figure 7

Bulkhead Fitting

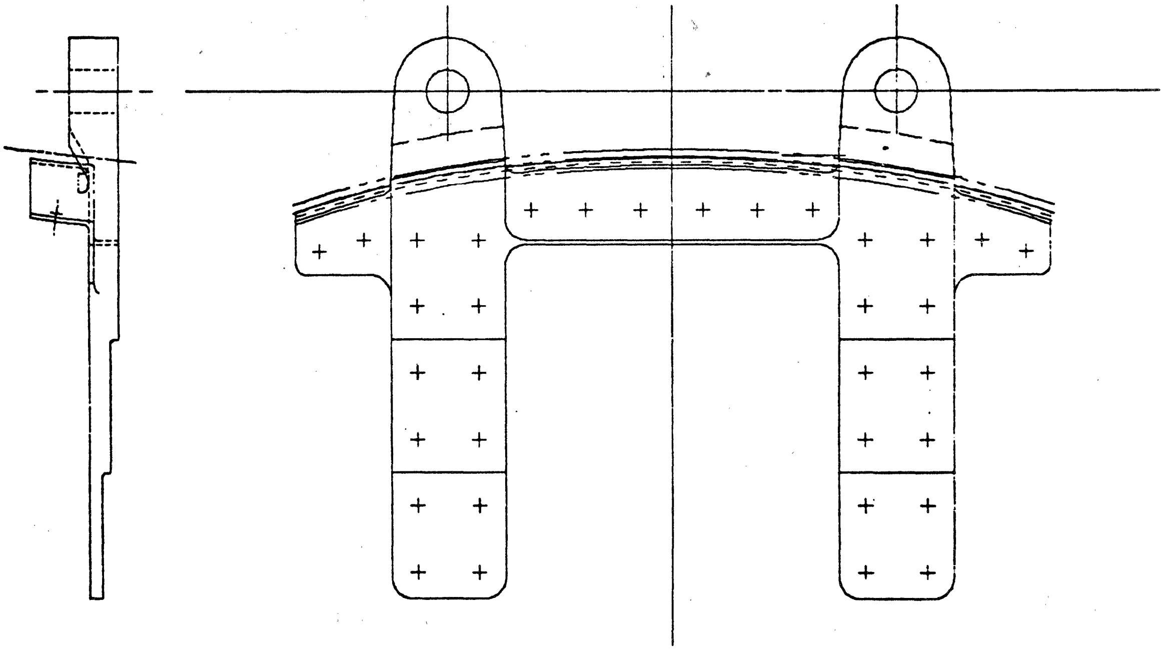


Figure 8

Airfoil Test Fixture
3-D Model

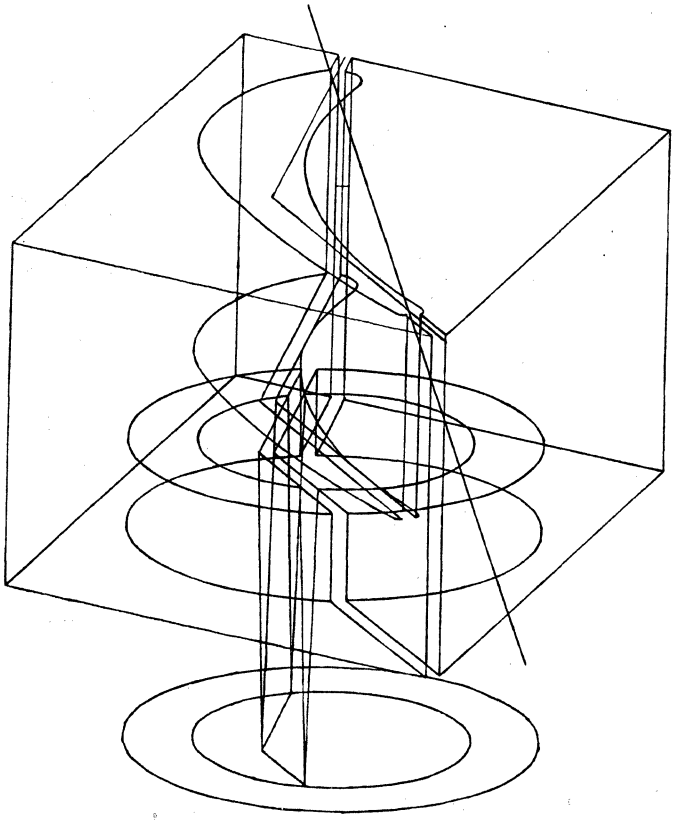


Figure 9

Airfoil Test Fixture 2-D Drawing

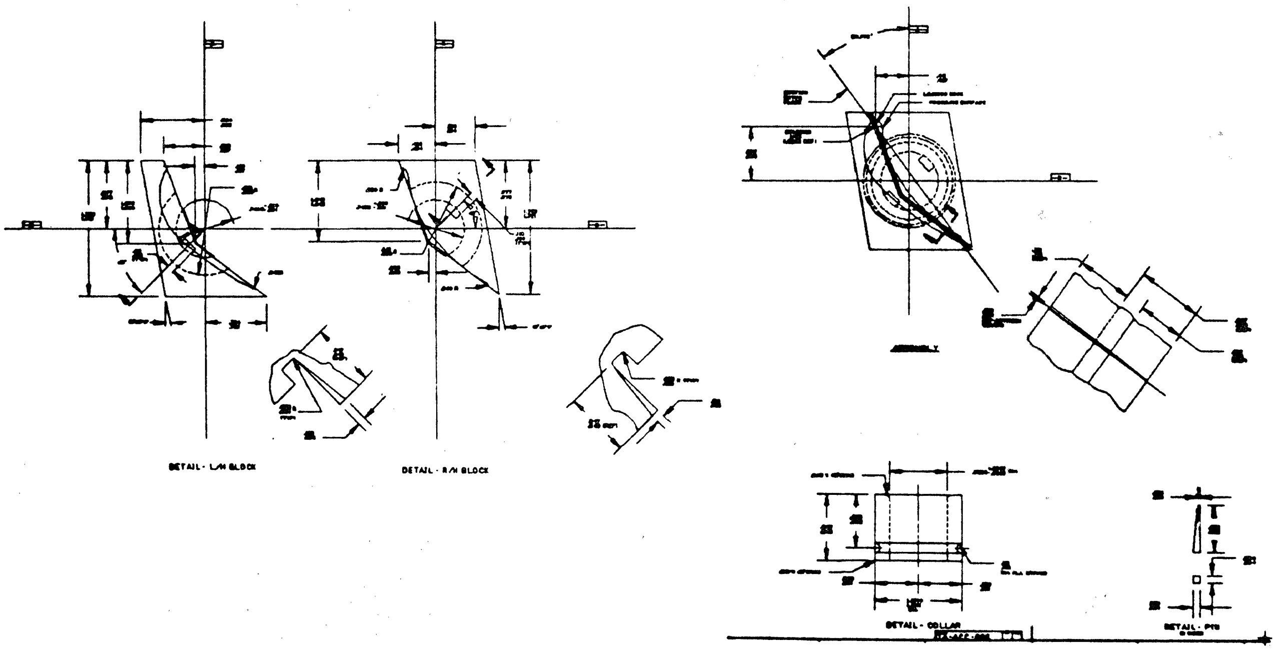


Figure 10

PRODUCTIVITY RISK ANALYSIS AND RETURN ON INVESTMENT FOR INTEGRATED CAD/CAM PROJECTS—A METHOD AND EXAMPLE

Henry Jacobson
Bruce Ziegler
Boeing Aerospace Company
P.O. Box 3999
M/S 44-08
Seattle, WA 98124
206-655-7040

ABSTRACT

A quantitative method to predict Integrated CAD/CAM Systems "performance measures" in terms of Productivity Ratios (PR) and Return on Investment (ROI) with associated risk levels is presented. Risk levels for estimated Productivity Ratios (PR) are generated using a "Risk Assessment" computer program which "fits" probability distributions through a set of input parameters. The Productivity Ratio distributions are mathematically related to savings and the computer program is used again to generate distributions for annual dollar savings from the CAD/CAM System. Finally, expected annual savings over the depreciable life of the equipment are used to compute the Internal Rate of Return (IRR) of the capital investment required to implement the System.

Key considerations for using the method to evaluate proposed Integrated CAD/CAM projects are identified.

INTRODUCTION

Prediction of "performance" is of primary importance for future success. "Performance measures" for CAD/CAM projects are needed so that the required capital investments may be compared to alternative uses of the capital.

Like other capital investments, the evaluation and justification of Integrated CAD/CAM projects is subject to two sets of interrelated "performance measures." The first set comprises technical "performance measures" which involve the analysis of the improvements to the performance of certain functions that Integrated CAD/CAM can provide. The second set comprises financial "performance measures" which involve an analysis to determine if the capital investment that is required to emplace the technological improvements can provide an adequate rate of return.

Investments in Integrated CAD/CAM projects differ in several respects from more typical capital investments such as buying N/C machine tools, process control equipment, material handling equipment, etc. These differences involve the following factors:

- Significant uncertainty in estimating "a priori" the magnitude of the technical improvements.
- Long learning curve "effects" to achieve system usage proficiency.
- Time span required for the organization to absorb the modifications and adaptations that must be made, because the effects of Integrated CAD/CAM cross functional and organizational boundaries.
- The economic benefits of Integrated CAD/CAM Systems increase substantially over time, as experience is gained.
- CAD/CAM Systems are typically an aggregate of ICG and other systems which makes finite measurements of any one specific area difficult.

The method for dealing with the "performance measures" and determining a return on investment (ROI) over a range of risk levels is described below. It is an attempt to deal explicitly with those unique factors associated with Integrated CAD/CAM projects.

The numerical data cited in this paper are intended for illustrative purposes only and are not necessarily representative of a specific Boeing project.

PROBLEM SCOPE

The decision to invest or not to invest requires judgment on the part of Management. Judgment is typically based on predictions of "performance measures" provided by experts. Most predictions are obtained either by single-valued analysis; or, alternatively, a range of performance predictions is obtained by "best case," "worst case," and "average case" analysis. Either method is subject to strong criticism. The possibility that the elements in complex systems like integrated CAD/CAM will combine to attain "best case," "worst case" or "average case" performance is remote. These methods reveal little about the potential variability of system performance. What is needed is a method that allows Management to base its decisions on predictions for System performance levels consistent with acceptable risk levels.

The method presented in this paper meets the criteria of relating a range of performance levels, expressed in terms of *Productivity Ratios* and *Return On Investment*, to risk levels, expressed in terms of *Probability of Achievement*.

The example that follows traces the ROI analysis, at different risk levels, for the capital investment required to install a number of Computervision ICG Systems. The ICG Systems installed in Engineering are coupled by telecommunications links to ICG Systems installed in Manufacturing to form a "full-thread" integrated CAD/CAM System. The Systems are used to support Engineering Design/Analysis and Manufacturing Engineering activities.

The case illustrated requires a capital outlay of $2,686,000, which includes the ICG Systems costs and associated facilities.

DEFINITIONS

In addition to the productivity gains realized through use of the systems, the *outputs* from CAD/CAM systems provide technical improvements to other activities; for example, the quality of N/C medias generated on Interactive Graphics can markedly reduce the "tape tryout" time on N/C equipment in the shop.

Benefits are defined as the technical improvements to a set of activities that are possible by the emplacement of CAD/CAM Systems. For clarity, two categories of benefits are defined in the analysis that follows:

1. *DIRECT BENEFITS* result from using CAD/CAM to improve the performance of specific tasks. Examples of activities that provide Direct Benefits are:

    - Automated Drafting
    - Cutter path generation for N/C
    - "Fit" checking for Assemblies
    - Engineering Design/Analysis applications (e.g., Finite Element Modelling, Tubend Analysis, Computer Nesting, etc.)

2. *INDIRECT BENEFITS* result from using the *outputs* from CAD/CAM Systems to improve the performance of other activities. Examples of activities that provide Indirect Benefits are:

- N/C Tape Tryout Time reductions
- Re-creation of Engineering Data in Manufacturing Engineering
- Fewer unplanned Manufacturing revision requests
- Computer Output Microfilm (COM)
- Quality Assurance N/C medias
- Mechanized parts lists generation
- Simpler and faster interfaces between Engineering Design and Engineering Analysis.

EQUATIONS

The Productivity Ratio estimates, combined with estimates for hourly man-hour rates, terminal work station hourly rates, and savings from using *outputs* of CAD/CAM Systems are used to compute annual dollar savings versus probability of achievement for the 7 year depreciable life of the system. The discount rate at which the annual dollar savings stream equals the required capital investment outlay to implement the system (Internal Rate of Return) is then computed as the measure for Return On Investment (ROI). Computer programs are used to perform the computations and plot results.

1. Total Benefits = Direct Benefits + Indirect Benefits
   = (Manual Costs – CAD/CAM Costs) + Indirect Benefits

   Total Benefits may be in terms of \$ Savings:

$$\text{2. Savings} = \left[\underbrace{\overbrace{H_m.R_m}^{\text{Manual costs}} - \overbrace{H_c.(R_t + R_m)}^{\text{CAD/CAM costs}}}_{\text{Direct Benefits}}\right] + \text{\$ Indirect Benefits}$$

Where:

H_m = # of hours to perform activity manually
R_m = manhour rate (\$/hr.)
H_c = # of hours to perform activity with CAD/CAM
R_t = CAD/CAM terminal rate (\$/hr.)

3. Productivity Ratio (PR) is defined as:

$$PR = \frac{H_m}{H_c};\ H_c = \frac{H_m}{PR}$$

Substitute H_c in (2.) with H_c value under (3.);

$$\text{4. \$ savings} = H_m.R_m - (R_t + R_m).\frac{Hm}{PR} + \text{\$ Indirect Benefits}$$

By using the equations above, the risk analysis described in the next Section is made. The elimination of H_c from the equations, permits the \$ savings and risk analysis to be performed using estimates for Productivity Ratios (PR), manual hours (H_m), manhour rates (H_m) and terminal hourly rate (R_t). Estimates are obtained from personnel who have both broad functional experience and familiarity with CAD/CAM.

COMPUTER RISK ANALYSIS

A computer program is used to perform a two stage risk analysis of the benefits. In the first stage, the estimated Productivity Ratios (PR) are subjected to a risk analysis. In the second stage, the $ savings as defined in equation 4. of the previous section are analyzed for risk.

The PR risk analysis begins by obtaining yearly estimates for the four parameters illustrated in Figure 1. The computer program treats the PR as random variables and generates a BETA probability distribution "curve fit" for the PR's, using the four parameters as program input. Output from the computer program includes the "probability density" distributions and the "cumulative probability" distributions. The cumulative distributions are illustrated in Figure 2. As an example, points on Figure 2 are read as follows: The probability that the PR in 1981 will be *at least* 2.0:1 is 7%; the probability that the PR in 1982 will be *at least* 2.0:1 is 56%; the probability that the PR in 1983 will be *at least* 2.0:1 is 95%. Other points on the curves are read in a similar manner. It is assumed that the PR's for the CAD/CAM applications under consideration here reach "maturity" in the years after 1983.

In the second stage of the risk analysis, the computer program uses the calculated PR's from stage one, and equation 4 to calculate $ savings at each "risk level" (probability of achievement). The cumulative distributions for $ savings for each year of the analysis are illustrated in Figures 3 & 4. Figure 3 are DIRECT Savings only (see DEFINITIONS Section). Figure 4 are cumulative DIRECT and INDIRECT Savings. As an example, points on Figure 4 are read as follows: The probability that the $ savings will be *at least* $1.4 MM in each of the years 1981–1987 is:

0.0 in 1981 (no chance)
0.48 in 1982 (48%)
0.49 in 1983 (49%)
0.999 in 1984–1987 (almost certain)

BREAKEVEN ANALYSIS

Breakeven means the point at which the *manual costs equal the CAD/CAM costs.* It is important to know the breakeven Productivity Ratio (PR), in order to determine what the required PR is for the CAD/CAM System to be cost effective. Equation 4 in the previous section is solved for the breakeven PR. Set $ Savings = 0; Set $ Indirect Benefits = 0; then solve equation 4 for PR.

$$\text{PR (Breakeven)} = \frac{Rt + Rm}{Rm}$$

Figures 5 and 6 present breakeven PR's for the years 1981–1987 at estimated Engineering man-hour rates (Rm) and terminal hour rates (Rt). The likelihood of achieving *at least* the breakeven PR may be determined from Figure 2. The likelihood of achieving PR of 1.8 in 1981 is approximately 16%. The likelihood of achieving the required breakeven PR in 1982 is 83% and in the succeeding years (1983–1987) is virtually 100%. The breakeven points are the 0.0 $ Savings on the curves in Figures 3 and 4.

RETURN ON INVESTMENT CALCULATIONS

The Internal Rate of Return is defined as the discount rate (r) at which the Net Present Value equals zero (0) in the formula below; it is the rate at which the net dollar inflows (taking into account the time value of money) exactly equal the initial investment outlay.

$$0 = \text{NET PRESENT VALUE} = \sum_{i=1}^{n} \frac{\$\ \text{Savings}(i)}{(1+r)^i} - \text{Capital Outlay}_{(0)}$$

A computer program is used to calculate the Internal Rate of Return for selected points from Figures 3 and 4. The annual $ savings flow and the calculated rates are shown in Figures 7 and 8.

CAD/CAM PRODUCTIVITY RATIO ESTIMATES

	1981	1982	1983	1984	1985	1986	1987
PR LOWEST	.9:1	1.2:1	1.6:1	→			
PR HIGHEST	3:1	3:1	4:1	→			
PR MOST LIKELY	1.3:1	2:1	2.5:1	→			
MOST PROBABLE PR RANGE THAT EXPRESSES 80% CONFIDENCE IN MOST LIKELY	1:1 to 1.8:1	1.5:1 to 2:3	1.8:1 to 2.8:1	→			

Figure 1

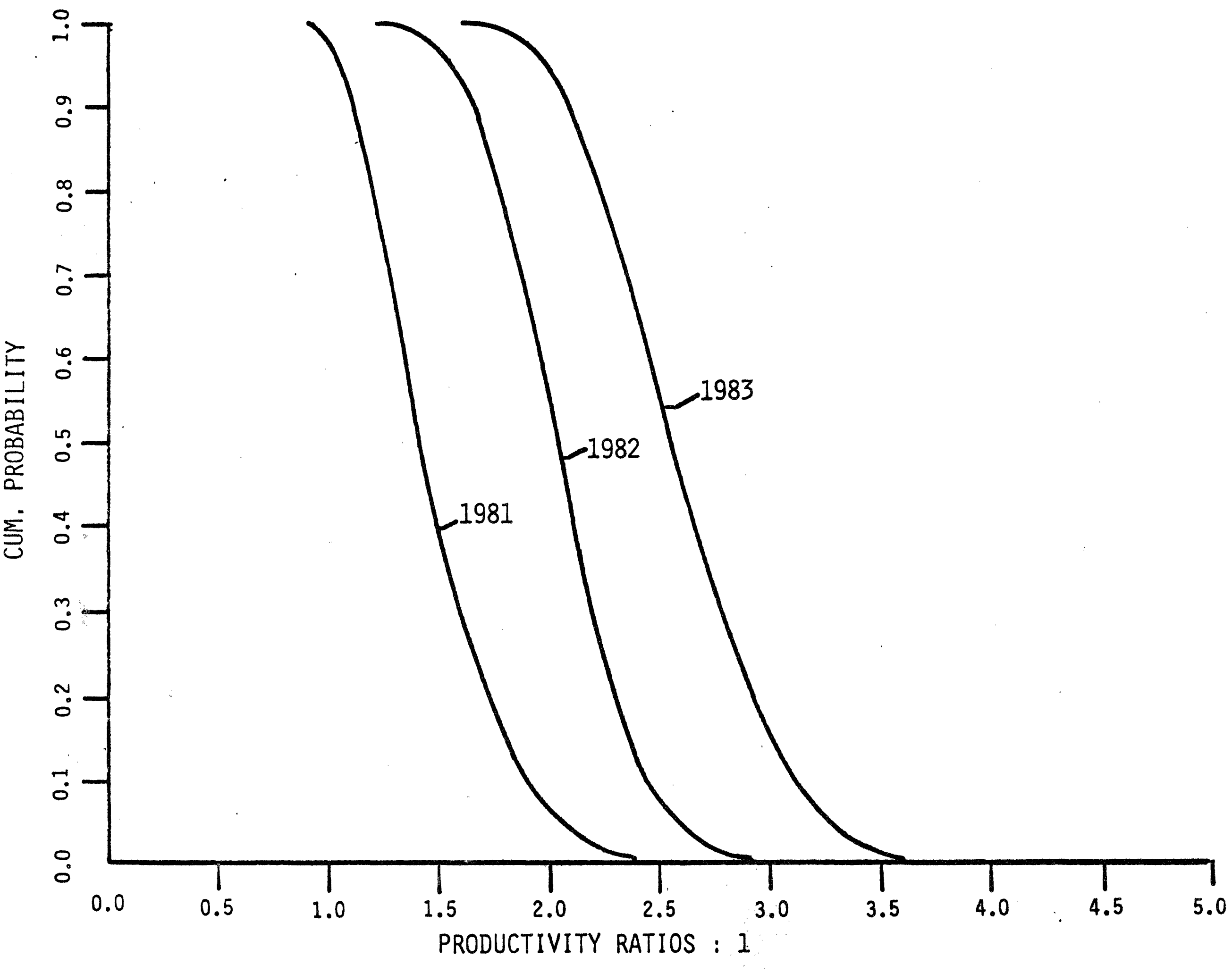


Figure 2

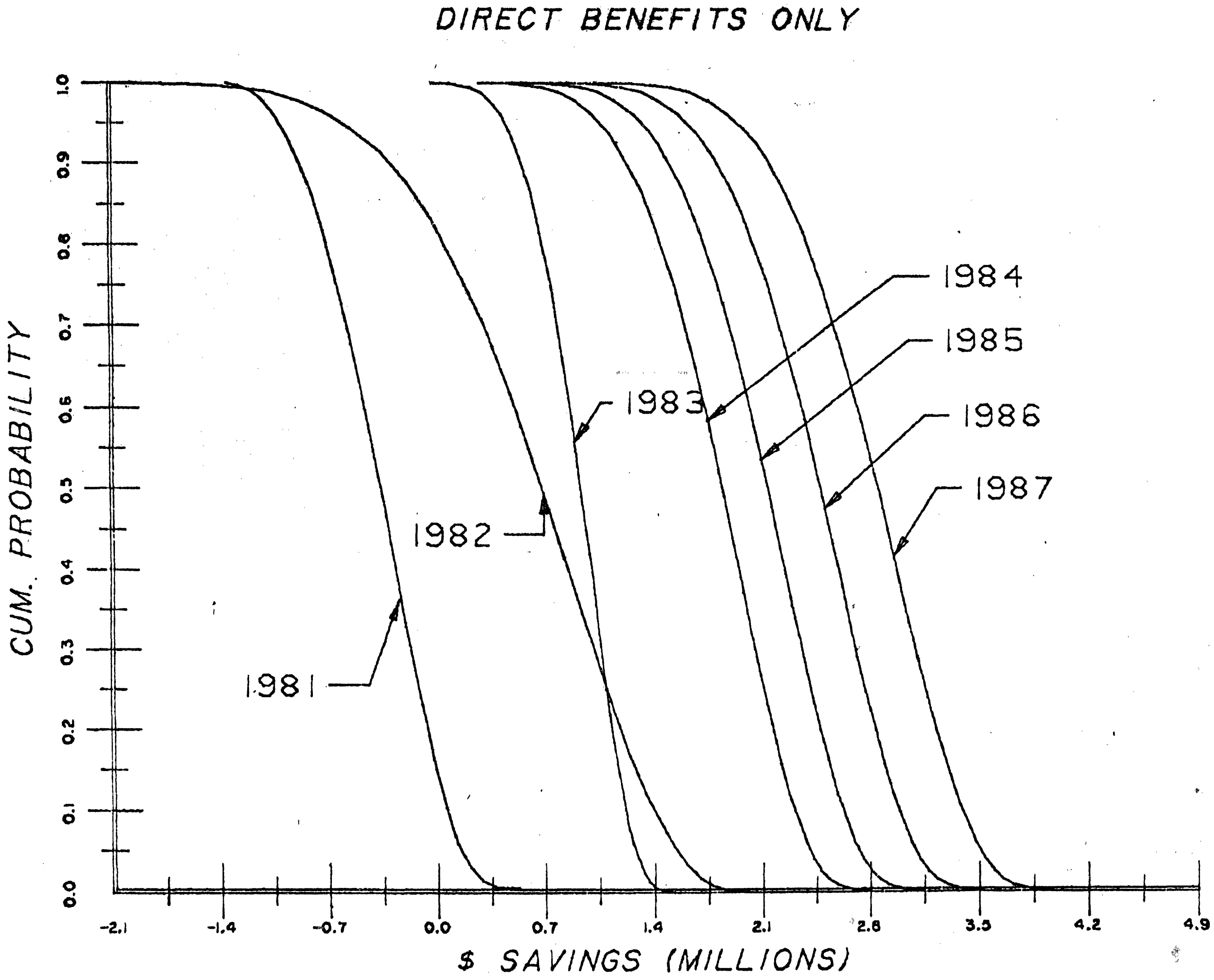

DIRECT BENEFITS ONLY
CUM. PROBABILITY
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-2.1
-1.4
-0.7
0.0
0.7
1.4
2.1
2.8
3.5
4.2
4.9
$ SAVINGS (MILLIONS)
1981
1982
1983
1984
1985
1986
1987

Figure 3

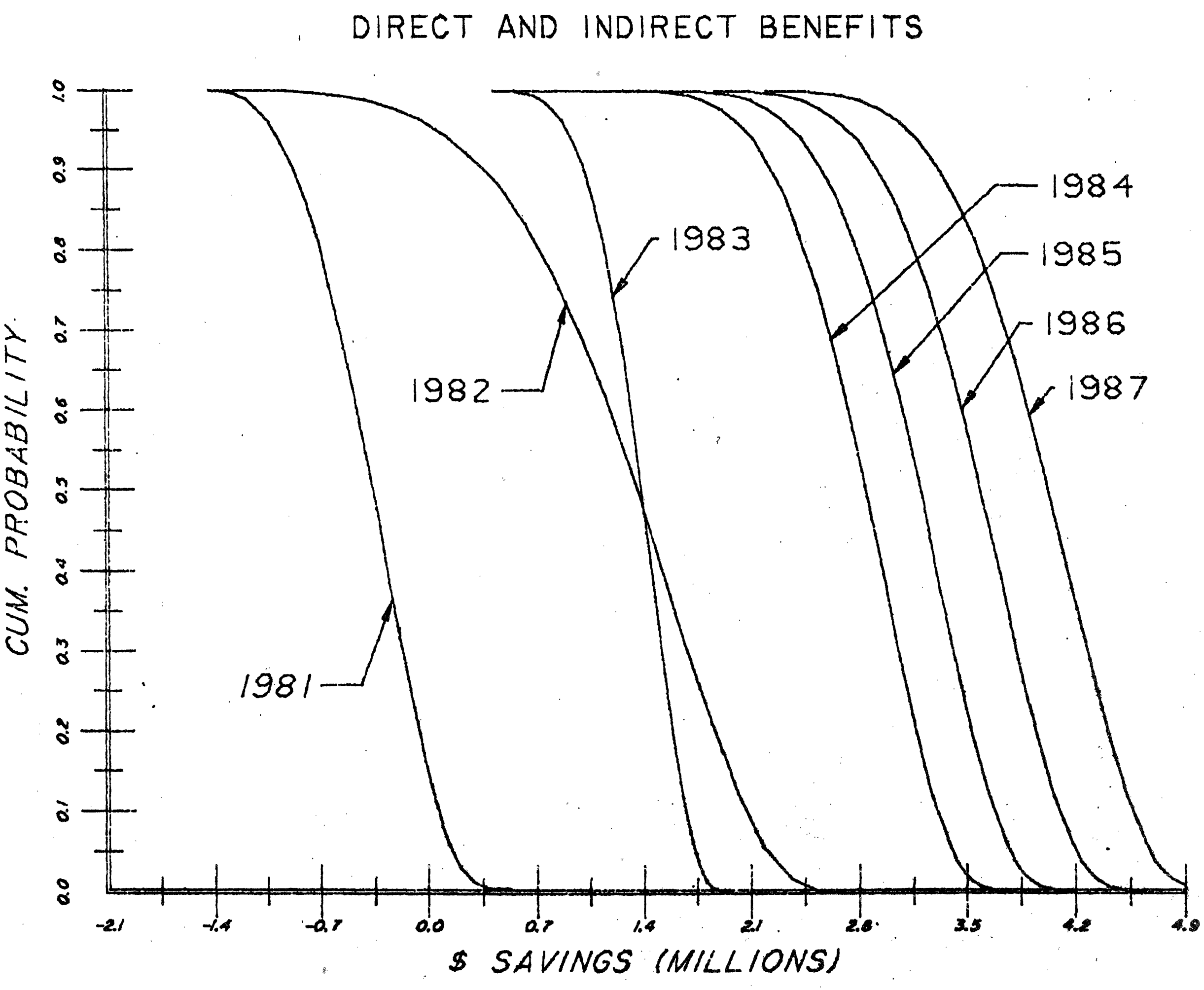


Figure 4

DIRECT BENEFITS ANALYSIS

MX FSED BREAKEVEN PRODUCTIVITY RATIOS (PR)

YEAR	1981	1982	1983
ENG. MAN-HOUR RATE (R_m)	53.00	57.00	65.00
TERMINAL HOUR RATE (R_t)	42.00	41.00	39.00
BREAKEVEN PRODUCTIVITY RATIO (P_r)	1.8	1.7	1.6

Figure 5

BREAKEVEN PRODUCTIVITY RATIOS 1984-1987

YEAR	1984	1985	1986	1987
ENG. MAN-HOUR RATE (R_m)	75.00	82.00	90.00	99.00
TERMINAL HOUR RATE (R_t)	37.00	35.00	33.00	31.00
BREAKEVEN PRODUCTIVITY RATIO (P_r)	1.5	1.4	1.4	1.3

Figure 6

DIRECT BENEFITS ONLY

	GROSS $ SAVINGS							
% confidence	1981	1982	1983	1984	1985	1986	1987	% Internal Rate of Return (ROI)
10	72,000.	1,417,000.	1,245,000.	2,310,000.	2,650,000.	3,020,000.	3,420,000.	44.5
50	-436,000.	711,000.	972,000.	1,860,000.	2,170,000.	2,510,000.	2,900,000.	30.6
90	-866,000.	-272,000.	592,000.	1,270,000.	1,540,000.	1,840,000.	2,180,000.	14.0

CAPITAL OUTLAY REQUIRED IS $2,686,000.

Figure 7

DIRECT AND INDIRECT BENEFITS

	GROSS $ SAVINGS							
% confidence	1981	1982	1983	1984	1985	1986	1987	% Internal Rate of Return (ROI)
10	72,000.	2,081,000.	1,697,000.	3,320,000.	3,710,000.	4,130,000.	4,590,000.	57.8
50	-436,000.	1,378,000.	1,413,000.	2,870,000.	3,220,000.	3,620,000.	4,060,000.	44.7
90	-866,000.	376,000.	1,045,000.	2,270,000.	2,590,000.	2,950,000.	3,380,000.	29.9

CAPITAL OUTLAY REQUIRED IS $2,686,000.

Figure 8

CONCLUSIONS

Productivity Ratio (manual hours/CAD/CAM hours) to perform a set of given tasks is typically used in the industry as the technical "performance measure" for CAD/CAM Systems. Except for a few specific applications such as automated drafting, the achievable productivity ratios are subject to significant uncertainty.

Integrated CAD/CAM System projects are strategic long term investments. Simple payback calculations will typically not justify the required capital investments, because long "learning curve" effects limit benefits that can be achieved in the short run. The relatively long "learning curve" effects occur from two causes:

- The time required to develop system usage proficiency.
- The time required to incorporate the organizational adaptations that must be made to make integrated CAD/CAM applications operational.

System throughput capacities often limit the magnitude of the DIRECT benefits that can be achieved. Therefore, it is essential to identify and measure both DIRECT and INDIRECT benefits that result from integrated CAD/CAM Systems.

Return on Investment (ROI) evaluations must take into account all relevant cash flows over the depreciable life of the System. The ROI calculation method should include the "time value of money," as described in the previous ROI section of this paper.

SAVINGS DEVELOPMENT AND JUSTIFICATION FOR THE PURCHASE OF A COMPUTERVISION CADDS 3 SYSTEM FOR A CAD/CAM APPLICATION

By George D. Cook
Manager, Tool Engineering and Services
(716) 422-4452
Xerox Corporation
North American Manufacturing Division
800 Phillips Rd. Bldg 207A
Webster, New York 14580
(716) 422-4452

ABSTRACT

This paper shows a method that can be followed to develop the cost justification for the purchase of a Computervision system for a CAD/CAM application. It includes the development of the return on investment, net present value, and pay-back years for the investment. Detail calculations are included in the appendix that show the source of the figures used in the summary data of the report.

INTRODUCTION

We are faced with rapidly growing and highly competitive industrial products that are emanating from Japan, which are already challenging and surpassing the American product quality and cost. If we are to meet this challenge, we must become more productive through the maximization of capital equipment. This leads to a need for a clear and concise manner of presenting information to top management for the purchase of capital equipment to provide the increased productivity capability. The purpose of this report is to provide a systematic approach for presenting the problem to management with alternatives and a recommended solution.

DISCUSSION

A format was developed to cover the key factors for requesting capital dollars in support of a new or expanded project. These factors are as follows:

- Problem
- Alternative
- Solution
- Benefits
  - Economic
  - Noneconomic
- Risks and assumptions

PROBLEM

The statement of the problem should be clear and concise, with enough supportive information to answer the most likely questions the capital approval committee might ask. It is very similar to a resume in that the written information must sell the readers on accepting your problem and want to help to correct it. Keep in mind that the capital approval committee is receiving many requests for capital expenditures from various areas in the company and must rank these requests from most important to least important in order to operate within the annual available capital funds. Therefore, it is important to you to show the relationship of your problem to the major objectives of your division or company that will help to sell your project.

Page 1 of the capital approval request (CAR) shows a typical format used to present the problem, which is covered in six specific statements. Note the first statement relates the problem with one of the division's objectives, which is to reduce the people required to perform the same amount of work and make them available to fill new openings in the division.

The second statement provides an overall view of Computer Graphics as it relates to your business and industry.

The third statement defines the initial Computer Graphics system and the areas of application to show the base operation from which the growth will take place.

The fourth statement describes the system expansion for obtaining maximum utilization in the drive for more productivity.

The fifth statement describes the test approach for determining speed degradation with the variable job mix used on this system to develop proper system size for meeting a minimum 40 percent return on investment.

The sixth statement describes the test results in terms of the recommended system changes that meet the financial buy criteria and the equipment required to solve the problem.

Page 2 of the CAR shows the seventh through ninth statements, which tell the capital review committee how the incremental savings will be obtained.

ALTERNATIVES

The approach here, as shown on Page 3 of the CAR, is to state the dollar loss caused by the system operating at a slower speed that lowers the expected return on investment below minimum requirement. The only alternative you have, in this case, is to operate with less savings, since the equipment is presently owned, or change the system to meet the financial requirement.

SOLUTION

The alternatives are examined objectively, analyzing the advantages and disadvantages as appropriate to the problem. Then, based on the overall review of the alternatives, a solution is presented that best meets the requirements for solving the problem. If you can relate your solution to not only solving your problem, but also to enhancing your ability to help improve other areas of the company, your chances of obtaining approval will be greater.

BENEFITS

The key factors to present for the economic benefits are the return on investment, net present value, and pay-back years. These are computed using standard financial formulas on a business computer, with the detail run shown in the appendix. A clarifying statement was used here to explain what was included in the cost figures since this was expanding a prior purchased system.

The noneconomic benefits cover those areas which are extremely difficult to quantify financially, but still represent improvement in company operations. In this particular case, we will be able to communicate with product engineering electronically for transmitting engineering drawings.

We have included the savings summary, which includes actual savings from the existing system, plus the projected savings from expanding the system for the total period over which the equipment will be depreciated. This provides the capital review committee with an easy way to see how the savings are obtained.

RISKS AND ASSUMPTIONS

All possible risks to the investment are reviewed in arriving at a decision. In this case, the system performance can be improved by software updates and the computer power obtained by add-on modules. It is assumed that the business will grow over the next five years providing the savings opportunity.

FINANCIAL DEVELOPMENT

The savings justification was developed from five types of work activities performed on the graphics system, which are represented by four categories — tool design, N/C savings, plant layout, and tool build.

Tool design includes machining fixtures, product assembly tools, and the design of computer based test equipment for the product assembly line.

The savings claimed for these activities are based on the increased productivity of the designers working on two shifts versus the drafting board. The detail calculations are shown in the appendix, where the productivity factors for electrical and mechanical design are given for the five years over which the equipment will be depreciated. The dollar savings are based on the reduction in contract designers by the increased output from the full-time employed tool designers. The man-year hours are first adjusted down 15 percent for holidays, break period, and engineering interface. Then, the hours are adjusted down 15 percent for development; downtime utilized to create better terminal usage. The total saved hours for each year are then adjusted down using the "five station degradation rate," which takes into account the speed loss of the system due to five terminals operating instead of four terminals. This corrective factor is presented in the paper titled "Speed Degradation Testing and Evaluation of a Computervision CGOS 100/10 Multiterminal System," by B. Burnett, during this same international meeting. We have used a conservative approach on the savings, which helps to sell the project to the financial review group.

The N/C savings are developed from the hours saved by utilizing Computer Graphics instead of the manual method. The reliability in tape accuracy from graphics has further reduced tool tryout on the factory floor. The estimated N/C workload, based on future new products, was used to develop the savings in this activity.

The plant layout activity savings were based on the reduction of one layout engineer due to the speed and versatility of interactive graphics in this area.

The tool build savings were provided by the reduction in cost to build wire wrap circuit boards by utilizing our post-processor run on the CV system to create the N/C tapes for driving a semi automatic wire wrap machine. In a like manner, we create N/C tapes to drive the wire EDM machine in the tool room for making punches and dies, as well as repair parts. Now, 85 percent of our chart gages are produced utilizing graphics, as well as special CAM profiles.

In the appendix, there are four pages that show the calculations for developing the return on investment, net present value, and pay-back years for the three investments that will provide the final system. This system will consist of two computer driving five design terminals each versus eight terminals operating from a single computer. The dollar savings summary shown on Page 5 of the CAR has lower values than the number shown to compute the return on investment. Since this is a combined CAR covering two prior purchases, plus the new investment, you must add in the dollars saved in 1980 to the five-year total and allow 10 percent growth in each year's savings to account for inflation. The result of the return on investment calculations provide an acceptable 41.3 percent, thus meeting management's requirement to purchase the system.

CAPITAL ASSET REQUEST

	DATE 01/05/81
PROGRAM NAME Computer Graphics--Tool Engineering and Services	PROGRAM NUMBER

PROVIDE AND REFERENCE EXHIBITS (CHARTS OR GRAPHS) TO ILLUSTRATE ANY ASPECT OF REQUEST WHICH WILL SUPPORT THE REASONING PRESENTED.

TYPE OF JUSTIFICATION:

☒ ECONOMICS
□ REPLACEMENT (Mandatory)
□ QUALITY
□ SAFETY
□ VOLUME (Expansion)
□ PROPRIETARY
□ PROGRAM (Unique – Critical)
X NEW TECHNOLOGY

CHECK ONE:

☒ BUDGETED
□ NON-BUDGETED

A. PROBLEM: Explain fully why this request is necessary. Present conditions to support this need.

Manufacturing in Xerox is being driven as never before to increase quality and productivity in a cost-effective manner. An enabling strategy to increase productivity and attain manpower reductions equal to 1:1 ratio objectives is increased utilization of Computer Graphics.

Computer Graphics is an increasingly sophisticated system integrating technical manpower with the power of a computer to improve the design and manufacturing processes. New applications for the process called CAD/CAM are being discovered daily, and it is enjoying unprecedented growth and acceptance in industry.

Tool Engineering, CMO, had the foresight of this coming revolution and entered the world of CAD/CAM in 1979. The initial configuration of the system included four designer terminals incorporating the disciplines of electrical tool design, mechanical tool design, sheet metal fabrication (N/C), and Plant layout.

Based on system configuration data of other leading Computer Graphics users and an objective to maximize productivity, a plan was made to expand the Tool Engineering installation. During 1980, a fifth terminal was added for Division Plant Engineering and three additional terminals were added to increase volumes of tool design.

Since no statistical data was available to determine the system response time impact of eight stations, it was necessary to "load up" our first system and do comprehensive testing to measure speed or response time degradation. Using projected productivity savings, Tool Engineering was able to determine return on investment impact of speed degradation and measure the optimum number of terminals per mini-computer.

Based on the data collected, a decision was made to plan two 5 station systems with two mini-computers and achieve maximum utilization while maintaining an attractive return on investment. In view of the productivity experience to date and statistical data collected, Tool Engineering recommends the purchase of one mini-computer and two designer terminals. The incremental savings will be realized through the following applications.

A. Problem (continued)

Electrical Test Equipment Design

Interactive graphics can be used to both design and document printed circuit boards and wire wrap boards. It can also be used to generate schematic diagrams and design the mechanical aspects of the test equipment. Thus, speed and efficiency of the computer enables us to realize a significant savings. A good designer should reach a productivity level of 3:1.

Mechanical Tool Design

Interactive graphics can be used to both design and document mechanical machine tools and fixtures. "Visual aid" of the tool on the system prior to actual build allows the designer more creativity while reducing design time. A good designer should reach a productivity level of 2:1.

N/C Tape Creation

The menial and time-consuming task involved in part layout and cutter tool path calculation are eliminated with interactive graphics. The engineer is able to display the cutter in motion, ensuring no debug time wasted on the floor. The engineer will improve his productivity by 25 hours per part based on actual experience to date.

B. ALTERNATIVES: Explain other courses of action available (i.e. buy vs. make, lease vs. purchase, add shift vs. overtime).

1. Continue to operate with existing eight station system, with known speed degradation problems documented through extensive testing. This would result in the expected ROI of 41.3 percent to drop to 24.3 percent. Thus, the available savings of $1,456.0K will be lost over the ensuing five-year period.

C. SOLUTION: Discuss each alternative presented objectively, advantages and disadvantages of each. Why is proposed course of action most beneficial.

1. This alternative is not recommended based on a very detailed speed degradation study, which illustrates productivity loss and represents a negative impact on previous investments.

The purchase of an additional CGP100 mini-computer and two design stations will enable Tool Engineering to reconfigure Computer Graphics into two 5 station systems and attain productivity levels which justify the additional investment. The additional purchase will allow us to expand our N/C application in Fabrication and continue to realize productivity improvements in Manufacturing's first CAD/CAM endeavor;

Additional new applications will be explored to improve productivity in Manufacturing and Tool Design.

D. BENEFITS: Explain economic and non-economic impact of request. Graphs, charts, and exhibits should be supplied to illustrate rationale.

Economic: Effect on cash flow – Investment return – Cash payback.

Non-Economic: Maintaining quality stds. – protection of proprietary rights – safety – preventing production delays – labor – management relations, etc.

Economic

Return on investment percent. 41.3%

Net present worth at 20 percent. $411.8

Pay back. 3.4 Years

This estimate is based on an additional outlay of capital for a CGP100 mini-computer, two additional design stations, and supporting equipment. The return on investment is based on expenditure for the new equipment, combined with the capital costs of the original CAR and the costs for the CAR for three additional stations. Previously stated expense costs and new additional expense costs are also included in the calculation of return on investment. Combined capital and expense costs are compared with total savings generated from the new ten station system configuration.

BENEFITS (Con't)

Noneconomic

The Computer Graphics installation has proved the value of Versatec Plotters used with Computer Graphics equipment providing a net potential for sales increase for Versatec Corporation. Components included with this CAR will provide the capability to communicate across the site with the RTG MDGS network system for electronic transmission of part designs to manufacturing. This will help to further enable productivity gains in our operations. Computer Graphics has the interface potential with computer-aided process creation.

Savings Summary

	1980	1981	1982	1983	1984	1985	Total
Tool Design	$19,200	$324,203	$423,883	$454,535	$510,643	$510,643	$2,243,107
N/C Savings	22,100	7,600	21,200	27,600	72,400	15,600	166,500
Plant Layout	10,800	40,000	40,000	40,000	40,000	40,000	210,800
Tool Build	25,900	118,600	118,600	118,600	118,600	118,600	618,900
Total Savings							$3,239,307

E. RISKS AND ASSUMPTIONS: Indicate the possible risks which might occur if the solution is adapted (i.e. Design change could effect obsolescence; Technical changes could reduce asset life; Proposed method might be radical increasing failure possibility.

Equipment obsolescence is minimized since the vendor purchase price includes software upgrade.

Hardware upgrades in the mini-computer can be accomplished with additional electronic boards for specific application and addition or subtraction of memory to suit system configuration.

It is assumed that new product/accessories will continue to provide about the same, or increased, volume of business over the next five years.

F. APPROVAL SIGNATURES AND DATES:

CAR ORIGINATOR	APPROVAL	ADDITIONAL APPROVAL
ADDITIONAL APPROVAL	PLANT/SECTION CONTROLLER	PLANT/SECTION MANAGER

APPENDIX

Computer Graphics
Tool Engineering
Tool Design Productivity

Based on 14 Terminal Shifts
Computed at Five Station Degradation Rate of .869

	First Year	Second Year	Third Year	Fourth Year	Fifth Year
Electrical	2.0/2.5 2.25	2.7	2.9	3.0	3.0
Mechanical	1.5/1.8 1.65	1.8	1.8	2.0	2.0
Design Hours Saved	17,716	23,163	24,838	27,904	27,904
Heads Saved	10.4	13.6	14.6	16.4	16
Dollar Rate	18.30	18.30	18.30	18.30	18
Dollars Saved	$324,203	$423,883	$454,535	$510,643	$510,643

Total Five-Year Savings = $2,223,907

Computer Graphics
Tool Engineering
Tool Design

Productivity Calculation:	250 Days × 8 Hours = 2,000 × .85 (Productive) = 1,700 Hours 1,700 × .85 (Development-Downtime) = 1,445
Assumptions:	1 Man-Year = 1,445 Hours
(Two Shifts)	Dollar Rate = $18.30 Per Hour = 1981 Contract Design Rate

7 terminals = 2 shifts = 14 shifts (6 shifts electrical – 8 shifts mechanical)

First Year Productivity Rate
Electrical = 1,445 × 2.25 = 3,251 – 1,445 = 1,806 hours saved
1,806 hours per terminal × 6 terminal shifts = 10,836 hours saved
Mechanical = 1,446 × 1.65 = 2,384 – 1,445 = 939 hours saved
939 hours per terminal × 8 terminal shifts = 7,512 hours saved

Electrical = 10,836 Five Station
Mechanical = 7,512 Degradation Rate
18,348 × .869 = 15,944 hours saved

15,944 ÷ 90 = 17,716 hours paid
17,716 × 18.30 hour = **$324,203**
17,716 hours ÷ 1,700 = 10.4 heads

Second Year Productivity Rate

Electrical = 1,445 × 2.7 = 3,902 − 1,445 = 2,457 hours saved
2,457 hours per terminal × 6 terminal shifts = 14,742 hours saved

Mechanical = 1,445 × 1.8 = 2,601 − 1,445 = 1,156 hours saved
1,156 hours per terminal × 8 terminal shifts = 9,248 hours saved

Electrical = 14,742 Five Station
Mechanical = 9,248 Degradation Rate
23,990 × .869 = 20,847 hours saved

20,847 ÷ .90 = 23,163 hours paid
23,163 × 18.30 per hour = **$423,883**
23,163 ÷ 1,700 = 13.6 heads

Third Year Productivity Rate

Electrical = 1,445 × 2.9 = 4,191 − 1,445 = 2,746 hours saved
2,746 hours per terminal × 6 terminal shifts = 16,476 hours saved

Mechanical = 1,445 × 1.8 = 2,601 − 1,445 = 1,156 hours saved
1,156 hours per terminal × 8 terminal shifts = 9,248 hours saved

Electrical = 16,476 Five Station
Mechanical = 9,248 Degradation Rate
25,724 × .869 = 22,354 hours saved

22,354 ÷ 90 = 24,838 hours paid
24,838 × 18.30 per hour = **$454,535**
24,838 ÷ 1,700 = 14.6 heads

Fourth Year Productivity Rate

Electrical = 1,445 × 3.0 = 4,335 − 1,445 = 2,890 hours saved
2,890 hours per terminal × 6 terminal shifts = 17,340 hours saved

Mechanical = 1,445 × 2.0 = 2,890 − 1,445 = 1,445 hours saved
1,445 hours per terminal × 8 terminal shifts = 11,560 hours saved

Electrical = 17,340 Five Station
Mechanical = 11,560 Degradation Rate
28,900 × .869 = 25,114 hours saved

25,114 ÷ 90 = 27,904 hours paid
27,904 × 18.30 per hour = **$510,643**
27,904 ÷ 1,700 = 16.4 heads

Fifth Year Productivity Rate

Electrical = 1,445 × 3.0 = 4,335 − 1,445 = 2,890 hours saved
2,890 hours per terminal × 6 terminal shifts = 17,340 hours saved

Mechanical = 1,445 × 2.0 = 2,890 − 1,445 = 1,445 hours saved
1,445 hours per terminal × 8 terminal shifts = 11,560 hours saved

Electrical = 17,340 Five Station
Mechanical = 11,560 Degradation Rate
28,900 × .869 = 25,114 hours saved

25,114 ÷ 90 = 27,904 hours paid
27,904 × 18.30 per hour = **$510,643**
27,904 ÷ 1,700 = 16.4 heads

NUMERICAL CONTROL SAVINGS

Based on CMO program summary, projected NC parts over the next five years were calculated using the following assumptions. Fifty percent of the parts will be NC sheet metal or machining.

Based on actual experience of Computer Graphics NC tape creation, 13 hours per part will be saved in the tape creation cycle, and seven hours per part in tool tryout. Assuming a change rate of two per NC part, an additional five hours per part will be saved. Thus, an average savings of 25 hours per part is used to compute savings based on a man-year of 1,920 hours. Savings are dollarized using a salary plus benefits of $40K per year.

Total Five-Year Savings **$144.4K**

PLANT LAYOUT SAVINGS

The application of Computer Graphics to Plant Layout provides a productivity improvement to the extent that one layout engineer can be reduced. This yields a salary plus benefits savings amounting to $40K per year.

Total Five-Year Savings **$210,800**

NC-CG POTENTIAL SAVINGS

(BASED ON ESTIMATED CMO PROGRAM SCHEDULE)

	1981	1982	1983	1984	1985
PRODUCT 1	11 FAB MAKES (EST) 5 NC PARTS				
PRODUCT 2	15 FAB MAKES (EST) 7 NC PARTS				
PRODUCT 3		68 FAB MAKES (1) 34 NC PARTS			
PRODUCT 4			88 FAB MAKES (1) 44 NC PARTS		
PRODUCT 5				152 FAB MAKES (1) 76 NC PARTS	
PRODUCT 6				80 FAB MAKES (EST) 40 NC PARTS	
PRODUCT 7					50 FAB MAKES (EST) 25 NC PARTS
(2) 20 HOURS SAVED PER NC PART	12 x 20 = 240 HOURS	34 x 20 = 680 HOURS	44 x 20 = 880 HOURS	116 x 20 = 2,320 HOURS	25 x 20 = 500 HOURS
(3) 5 HOURS SAVED PER PART CHANGE RATE OF 2.0	24 x 5 = 120 HOURS 360 HOURS (4) .19 MAN/YEAR (5) $7.[illegible]	68 x 5 = 340 HOURS 1,020 HOURS .53 MAN/YEAR $21.2K	88 x 5 = 440 HOURS 1,320 HOURS .69 MAN/YEAR $27.6K	232 x 5 = 1,160 HOURS 3,480 HOURS 1.8 MAN/YEAR $72.4K	50 x 5 = 250 HOURS 750 HOURS .39 MAN/YEAR $15.6K

TOTAL = $144.4K

(1) SOURCE = CMO PROGRAM SUMMARY

(2) TWENTY HOURS SAVED CONSISTS OF 13 HOURS AVERAGE REDUCTION IN TAPE CREATION AND SEVEN HOURS AVERAGE REDUCTION IN TRYOUT

(3) ESTIMATED CHANGE RATE TO LINE BALANCE x NUMBER NC PARTS - HOURS SAVED CONSISTS OF REDUCED TAPE CREATION AND TRYOUT

(4) MAN-YEAR BASED ON 1,920 HOURS PER YEAR

(5) SAVINGS = MAN-YEARS x $40K PER MAN-YEAR

TOOL BUILD SAVINGS

Fourth quarter savings in 1980 were documented for 25.9K. Of this savings, approximately 50 percent was in wire wrap build and 50 percent was in the Computer Graphics wire EDM programming. To summarize yearly savings, the 25.9K per quarter was expanded to 103.6K per year. The 103.6K per year is considered conservative since all personnel have been in a learning curve and in a procedures modification mode of operation. The other build savings reference is in chart gages. Of the approximate 100 chart gages completed annually, 85 will be done in Computer Graphics, yielding a build savings of $175 per job. Thus, a savings of $15K per year in chart gages is estimated.

Total Five-Year Savings Chart Gage	$ 75K
Total Five-Year Savings WW/EDM	518K
Grand Total	$593K

COMPUTER GRAPHICS TOOL ENGINEERING INVESTMENT

	Capital	Expense
Original (Actual)	$483.3K	
- Associated		$ 65.0K
- Ongoing		196.2K
Second Three Station CAR (Actual)	147.7K	
- Associated		15.0K
- Ongoing		147.4K
New Outlay	305.3K	
- Associated		5.0K
- Ongoing		85.5K
Overhead Included in Return on Investment		192.0K
	$936.3K	$706.1K

```
COMPUTER GRAPHICS - 3 C.A.R. COMBINED
RETURN ON INVESTMENT (ROI)  =        41.3%

NET PRESENT VALUE           =$     411.8

PAYBACK PERIOD              =        3.4 YEARS


ASSUMPTIONS SUMMARY

TAX RATES: 52.0% ALL YEARS

DISCOUNTING IS DISCRETE
DISCOUNT RATES: 20.0% ALL YEARS

                        ---- INVESTMENTS ----

                          TYPE            DEPRECIATION
                                         METHOD       START   LIFE
  CPU & 4 ST.           DISCRETE     STRAIGHT LINE    1980   5 YRS
  3 ST.                 CONTINUOUS   STRAIGHT LINE    1980   5 YRS
  CPU & 2 ST.           CONTINUOUS   STRAIGHT LINE    1981   5 YRS


                          TOTAL AMOUNT      SALVAGE          INVESTMENT
                                          YEAR   VALUE       TAX CREDIT
  CPU & 4 ST.                   483.3     1984      7.0  6.67%
  3 ST.                         147.7     1984      2.2  6.67%
  CPU & 2 ST.                   305.4     1985      5.0  6.67%

                            INFLATION RATES
  CPU & 4 ST.           ;1980-1985    10.0%
  3 ST.                 ;1980-1985    10.0%
  CPU & 2 ST.           ;1980-1985    10.0%


                   ---- SAVINGS AND REVENUE ----

                         TYPE         TOTAL AMOUNT
 TOOL DESIGN SVGS        CONTINUOUS      3063.6
 N/C SVGS                CONTINUOUS       224.0
 LAYOUT MNPWR            CONTINUOUS       279.4
 TOOL BUILD SVGS         CONTINUOUS       822.4

                           INFLATION RATES
 TOOL DESIGN SVGS       ;1980-1985    10.0%
 N/C SVGS               ;1980-1985    10.0%
 LAYOUT MNPWR           ;1980-1985    10.0%
 TOOL BUILD SVGS        ;1980-1985    10.0%
```

```
             ---- ASSOCIATED COST AND EXPENSE ----

                                 TYPE        TOTAL AMOUNT
4 ST. ASSOC.                     DISCRETE          65.0
4 ST. MAINT.                     CONTINUOUS       254.9
3 ST. ASSOC.                     CONTINUOUS        15.0
3 ST. MAINT.                     CONTINUOUS       191.5
2 ST. ASSOC.                     CONTINUOUS         5.5
2 ST. MAINT.                     CONTINUOUS       114.8
OVERHEAD                         CONTINUOUS       246.9

                             INFLATION RATES
4 ST. ASSOC.           ;1980-1985   10.0%
4 ST. MAINT.           ;1980-1985   10.0%
3 ST. ASSOC.           ;1980-1985   10.0%
3 ST. MAINT.           ;1980-1985   10.0%
2 ST. ASSOC.           ;1980-1985   10.0%
2 ST. MAINT.           ;1980-1985   10.0%
OVERHEAD               ;1980-1985   10.0%
```

COMPUTER GRAPHICS - 3 C.A.R. COMBINED

CASH FLOW SUMMARY	TOTAL	1980	1981	1982	1983	1984	1985
SAVINGS AND REVENUE							
TOOL DESIGN SVGS	3063.6	19.2	356.6	512.9	604.9	747.6	822.3
N/C SVGS	224.0	22.1	8.4	25.7	36.7	106.0	25.1
LAYOUT MNPWR	279.4	10.8	44.0	48.4	53.2	58.6	64.4
TOOL BUILD SVGS	822.4	25.9	130.5	143.5	157.9	173.6	191.0
TOTAL SAV. & REV.	4389.3	78.0	539.4	730.5	852.8	1085.8	1102.9
DEPRECIATION							
CPU & 4 ST.	476.3	95.3	95.3	95.3	95.3	95.3	.0
3 ST.	145.5	14.6	29.1	29.1	29.1	29.1	14.6
CPU & 2 ST.	300.4	.0	30.0	60.1	60.1	60.1	90.1
TOTAL DEPRECIATION	922.1	109.8	154.4	184.4	184.4	184.4	104.7
COSTS AND EXPENSES							
4 ST. ASSOC.	65.0	65.0	.0	.0	.0	.0	.0
4 ST. MAINT.	254.9	25.2	37.6	41.4	45.5	50.1	55.1
3 ST. ASSOC.	15.0	15.0	.0	.0	.0	.0	.0
3 ST. MAINT.	191.5	18.9	28.3	31.1	34.2	37.6	41.4
2 ST. ASSOC.	5.5	.0	5.5	.0	.0	.0	.0
2 ST. MAINT.	114.8	.0	18.8	20.7	22.8	25.0	27.5
OVERHEAD	246.9	32.0	35.2	38.7	42.6	46.9	51.5
TOTAL COST & EXPENSE	893.6	156.1	125.4	131.9	145.1	159.6	175.5
NET INCOME BEF TAXES	2573.6	-187.9	259.6	414.2	523.3	741.8	822.7

```
INCOME TAX                 1338.3    -97.7    135.0    215.4    272.1    385.7    427.8

INVESTMNT TAX CREDIT
  CPU & 4 ST.                32.2     32.2       .0       .0       .0       .0       .0
  3 ST.                       9.9      9.9       .0       .0       .0       .0       .0
  CPU & 2 ST.                20.4       .0     20.4       .0       .0       .0       .0
TOTAL TAX CREDITS            62.5     42.1     20.4       .0       .0       .0       .0

NET TAX PAYMENT            1275.8   -139.8    114.6    215.4    272.1    385.7    427.8

NET INCOME AFTER TAX       1297.8    -48.1    145.0    198.8    251.2    356.0    394.9

SALVAGE VALUE
  CPU & 4 ST.                 7.0       .0       .0       .0       .0      7.0       .0
  3 ST.                       2.2       .0       .0       .0       .0      2.2       .0
  CPU & 2 ST.                 5.0       .0       .0       .0       .0       .0      5.0
TOTAL SALVAGE VALUE          14.2       .0       .0       .0       .0      9.2      5.0

TOTAL DEPRECIATION          922.1    109.8    154.4    184.4    184.4    184.4    104.7

NET CASH INFLOWS           2234.1     61.7    299.4    383.2    435.6    549.7    504.5

CAPITAL INVESTMENTS
  CPU & 4 ST.               483.3    483.3       .0       .0       .0       .0       .0
  3 ST.                     147.7    147.7       .0       .0       .0       .0       .0
  CPU & 2 ST.               305.4       .0    305.4       .0       .0       .0       .0
TOTAL INVESTMENTS           936.4    631.0    305.4       .0       .0       .0       .0

NET CASH FLOW              1297.8   -569.3     -6.0    383.2    435.6    549.7    504.5

P.V. OF N.C.F.              411.8   -569.3     -5.0    266.1    252.1    265.1    202.8
```

‘THE CASE FOR A GREATER DEFINITION OF CAD’

Dr. A. J. Medland
Department of Engineering and Management Systems, Brunel University
Uxbridge
Middlesex, England
895-37188 x 221

INTRODUCTION

The development of Computer based Design techniques has been, up until now, on a piecemeal basis following the specific needs of the developing group or supporting company. This process, of devising solutions for particular needs should be recognized as being the natural course of development in all engineering situations (as opposed to a science-based approach where general principles would be extracted). The ‘target’ nature of engineering activities will thus not follow systematically through the design process by converting each stage at a time into a computer based procedure. Development will commence on the areas of greatest need or areas of easiest justification. A patchwork of techniques have thus been developed, having no common base, but all existing under the general umbrella of ‘CAD’.

THE DESIGN PROCESS

In order to be able to categorize the existing techniques and to specify new areas of need, it is necessary to investigate in some depth the design process itself. Again, the ‘target’ nature of engineering must be recognized in that the activities included within the process will vary in relative weight and order depending on the objectives and targets set to control the company and/or its product. The classical description of the design process is that it is the method whereby the conceptual solution of an engineering problem is turned into reality.

The process is thus assumed to start with the acceptance of a design specification. No design can, however, be deemed to be successful unless the objectives have been clearly stated at the outset. This definition seems to suggest that the designer takes no part in (or assumes no responsibility for) the original specification. The design team (and hence the design process) must however include some participation in the original formulation of the problem definition and its constraints.

Once the need has been established, various approaches can be tested against the proposed target. This is usually termed the concept stage in which the ideas are gradually refined, to give a single workable solution. As the proposed solutions gradually ‘solidify’ into a defined hardware proposal, the design problems move out of the concept stage and into the analysis phase. This transition may not be obvious as the detailed consideration of a number of alternative solutions may be considered as a conceptual investigation whilst the calculation of design parameters for the final design would naturally fall into the analysis section.

Once the proposed design has become sufficiently firm, the details of the manufacturing processes and constraints can be chosen, to turn the proposed solution into reality. This activity is usually termed the 'detailing' phase.

Whilst the design process can be considered theoretically as a simple linear system containing three elements (shown in Fig. 1) in practice it takes on a far more complex format. Rarely is one phase completed before another started. Usually differing aspects of the design develop at differing speeds. Some areas of the design are hardened down immediately (perhaps there is a desire to use an existing component or format) whilst others are allowed to float until a very late stage in the complete development of the product ('Last Minute' decisions). The non-uniform advance across all the elements can also create confusion in the process as pictured due to the interdependence of many of these elements. Some aspects even after final detailing may be forced to return to the analysis (or even the concept) phase in order that they can be made compatible or function with all other elements. The process being iterative in nature is thus seen to take on the characteristics of both oscillating and cycling systems rather than that of a simple linear one. It then becomes confusing to talk simply about the 'concept', 'analysis' and 'detailing' phases as these terms tend to suggest the single once-only operation rather than the possible large number of iterations. It seems incongruous to talk about a concept or analysis activity for a particular element when the major activities have reached the detailing stage. It then becomes more appropriate to call them the primary, secondary and tertiary phases of design, (as shown in Figure 2).

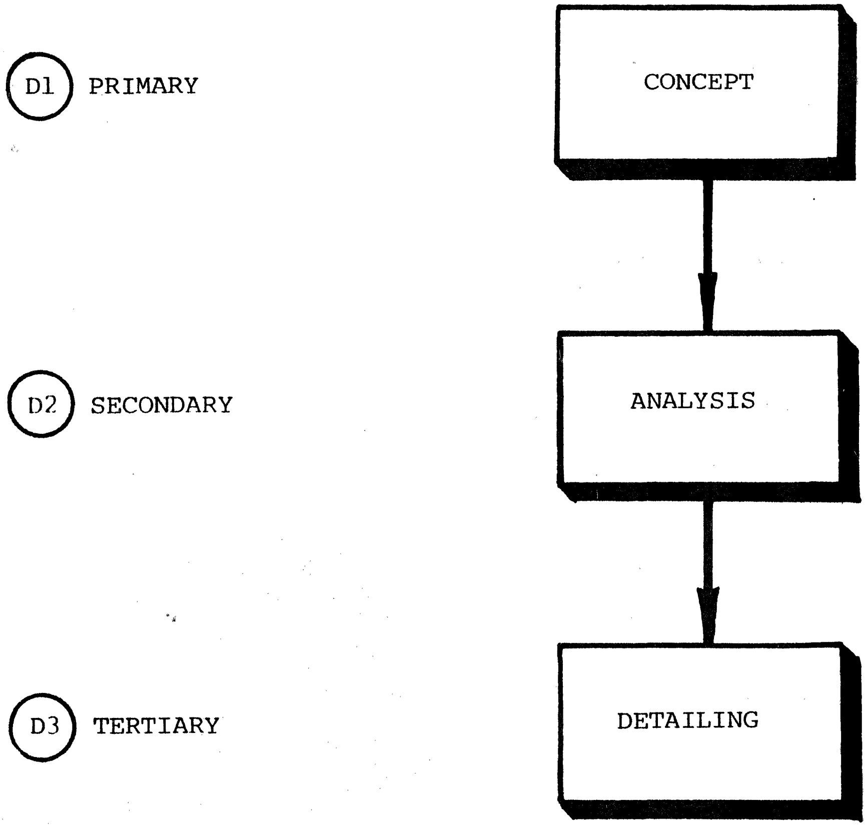


Figure 1: Theoretical Design Process — Major Classification

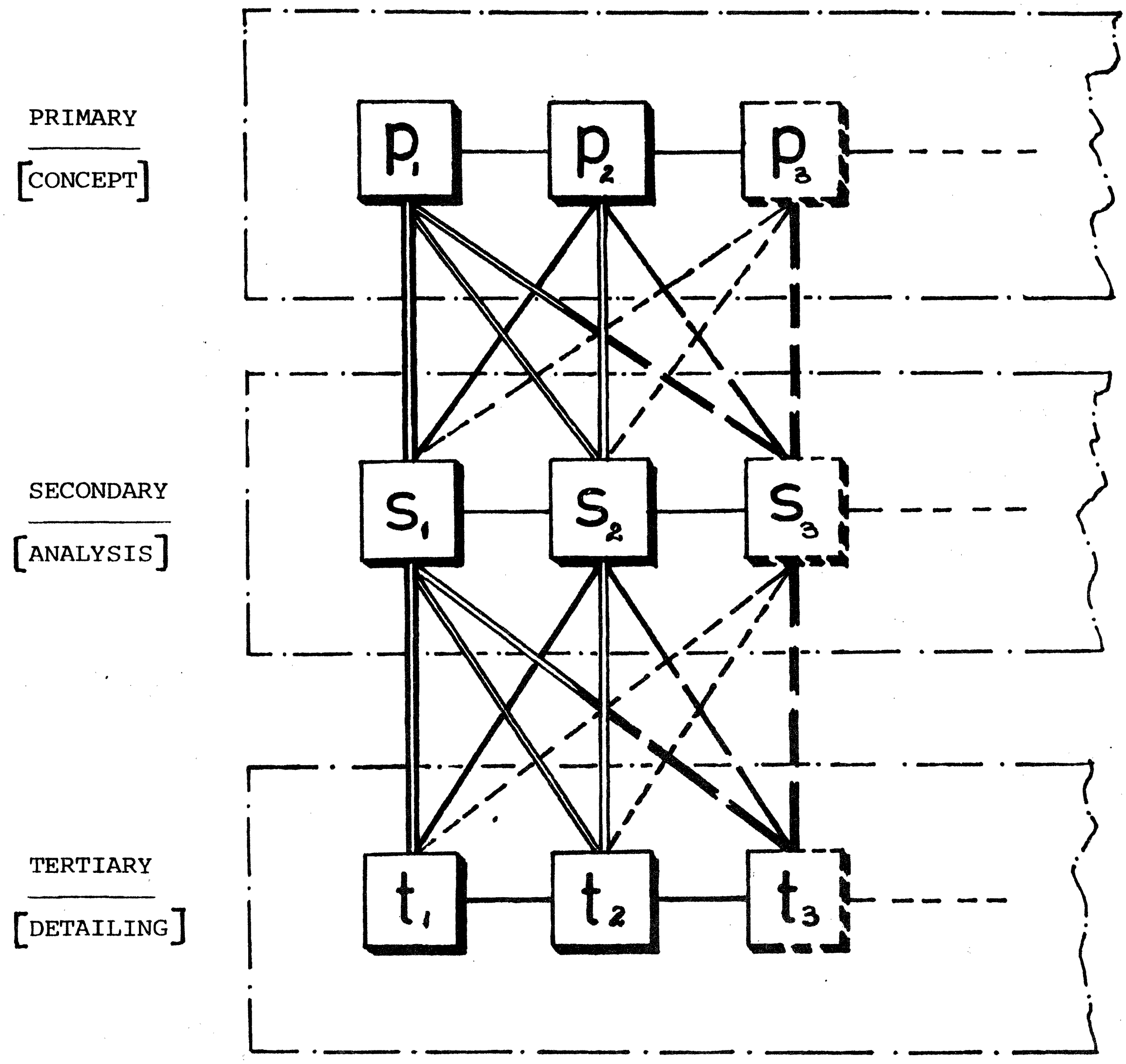


Figure 2: Iterative Design Process

CATEGORISING EXISTING TECHNIQUES

If then the existing techniques (or available programs) are categorised under these three headings their role within the overall design process becomes apparent. It also helps to clarify those areas of design which have been successfully computerised and those areas which have been neglected.

This grouping of techniques within 'CAD/CAM' allow the potential user to choose or assemble a system, from all that is available, which will best fit his needs. No longer is it necessary for us to all use the same words to mean completely different things.

The table in Figure 3 gives the broad outline of the classification for electrical, mechanical engineering and architecture.

CLASS TOPIC	PRIMARY DESIGN CAD 1 (CONCEPTUAL)	SECONDARY DESIGN CAD 2 (ANALYSIS)	TERTIARY DESIGN CAD 3 (MANUFACTURING INFORMATION)
ELECTRICAL ENGINEERING		CIRCUIT ANALYSIS	LSI & PLB LAYOUT
		COMPONENT SELECTION	
MECHANICAL ENGINEERING		STRESS (F.E.M.)	DRAFTING
		KINEMATIC/ MECHANISM	
		VIBRATIONS/ DYNAMICS	NC. TAPE PRODUCTION
		PIPE RUNS	
ARCHITECTURE		STRUCTURAL ANALYSIS	CARTOGRAPHY
		PURPOSE/ LAYOUT ANALYSIS	PLANS

Figure 3: Categorisation of CAD Programs

TERTIARY DESIGN

When such a classification is employed it becomes obvious that the majority of work has been directed towards the tertiary activities. The main aim of the normal turnkey CAD/CAM system is to operate in this region.

This approach is not surprising as whilst the inputs (in terms of component requirements) are varied and numerous the output format is very well defined. These fall into two well documented categories of either man-readable instructions or machine-readable instructions.

Two types of man-readable data are prepared; drawings or schedules. The drawings contain both pictorial and symbolic representations. Many levels of pictorial description are available from the simple two-dimensional representation of the drawing sheet through to the full three-dimensional solid body of complex shape. The correct level of representation can then be chosen to fit the needs of the particular work in hand. The type of drawing (or 'language' used) again depends upon the particular application. The rules of presentation and representation change depending whether the output is an electrical diagram, engineering drawing or site plan. The techniques employed for the geometric representation of three-dimensional bodies varies as does the methods storing and retrieving the data. These again must be chosen to match the needs. (The construction of solids for interference analysis imposes quite different geometric constraints to those for the construction of isometric (or perspective) pictorials).

Whilst the production of numerically controlled tapes for production is usually separated off as 'CAM' it is really the machine readable part of the tertiary design phase. The decision to instruct a machine to automatically produce an item rather than produce a pictorial representation takes the designer down the 'CAM' route.

SECONDARY DESIGN

The word 'design' when used in common English tends to imply that some creative activity is taking place. Our expectation of what is deemed to be 'creative' is equally vague. Anything that is simple and obvious is not creative and hence not design. There is thus a tendency for some engineers to say that drafting (and hence tertiary design) is not really design. The mundane activities necessary to produce an artefact are not really design; solely production engineering. Design really takes place only from the conception of a idea up until it can be seen to be a fully developed and workable solution. Thus when many people talk (or think) about CAD they are referring to the secondary design activity wherein the seed of the idea is fully investigated and developed.

Within this secondary category are grouped all the engineering analysis activities normally performed using computers. Many such programs have been developed over the years to exploit the computers ability to retrieve and/or manipulate large quantities of data. Complex matrix and iterative processes can be employed to resolve or optimize a large number of variables in a complicated interactive model. It is thus not surprising that to these people design means the application of high level analysis programs such as finite element structural analysis and automatic circuit analysis programs. Whilst these are perhaps the most complex pieces of analysis that need to be performed they are not at present really interactive design programs; simply the analysis of a specified configuration. However much analysis and design work can be performed on the existing turnkey computer system by making use of its large graphics capabilities with a limited amount of numerical manipulation. Much design at present is performed against codes of practice and standards. Programs can be provided or written which interpret the component design in accordance with these rules. The program output can be in the form of specified data or given as nomograms which can be interpreted by the designer. Many companies have developed a range of these programs to satisfy their own needs and requirements. They thus tend to be very company dependent and remain unpublished as their content is only of interest to their direct competitors.

It is thus in the areas of secondary design that the software houses are most active. Packages are developed in accordance with customer requirements or by recognising that a common need exists within a particular field.

PRIMARY DESIGN

Little or no work has been published on this conceptual activity. Whilst the desired output and constraints can be specified in both the secondary and tertiary activities it is extremely difficult at times to specify the form of the output required in the primary stage and more difficult to attempt to quantify it.

When asked, many designers are either unaware that they are performing any creative activity or are unable to rationally describe it. ('It just comes' or 'It's obvious'). Without an understanding of the logic, weighting and constraints of their design situation it is impossible to provide the supportive design aids. This is the area where the greatest benefit could occur but the effort necessary to develop the techniques may be enormous.

It is strangely in this area that the greatest fears against 'CAD' exist. There is the almost paranoic feeling that the machines will take over. Artefacts will be designed and produced without human control. This is caused possibly by the lack of clear understanding of the designer's role in the primary phase. It is thus difficult to assert that the computer will not be able to take it over. It then, becomes seen as a threat rather than a design aid. The clearer division of activities within the overall design process, it is hoped, will help to alleviate this problem.

CONCLUSIONS

The subdivision of 'CAD' into its Primary, Secondary and Tertiary phases helps to clarify the activities that, at present, can be performed by the existing turnkey packages and specialised programs. Much work is still required to be performed in the initial conceptual phase in order to provide the designer with tools necessary to hold and manipulate a larger number of alternative schemes and constraints.

Many analysis programs and geometric packages now exist. Effort is required in establishing which techniques are best suited to handle particular range of engineering problems. Companies must be encouraged and helped to investigate the optimum form of the design process within their industrial constraints. Once this has been done the computer based design packages can be tailored to provide a truly integrated computer aided design facility.

NOTE: Specific companies and their needs will be used as illustrations during this paper

SPEED DEGRADATION TESTING AND EVALUATION OF A COMPUTERVISION CGOS100/10 MULTITERMINAL SYSTEM

Blaine E. Burnett
CAD/CAM Systems Engineer
Xerox Corporation
North American Manufacturing Division
800 Phillips Road, Building 207A
Webster, New York 14580
716-422-9459

TABLE OF CONTENTS

Abstract

This paper explains the results of extensive speed degradation testing done on a Computervision CGOS100/10 multiterminal system. The intent of the paper is to aid the manager of a CAD/CAM system in predicting speed degradation as terminals are added to a system. Using this data, the optimum number of terminals per CPU on a return-on-investment basis can be determined.

Introduction

At the time that this test was conducted, June, 1980, there seemed to be little information available concerning speed degradation of multiterminal Computervision systems. The system users knew that the system got slower as more terminals were added, but no one could say just how much slower. It seemed that the only way to know the optimum number of terminals per system was to fully load one CPU and perform time testing. Using the test results, the terminals could then be split up between CPU's for maximum productivity per system.

Test Validity

It should be noted that the large number of variables makes it impossible to exactly duplicate these tests. This variability also prevents an exact prediction of expected system productivities. (Remember: Compare these estimates to the estimated degradation from other tests. You may get "different" degradation depending on how you use your system, the environmental conditions, length of usage, and the condition of your system. Actual overall degradation will probably be less than the overall estimate.) (Sound familiar?)

The following are some of the parameters affecting test results:

Hardware Configuration
- Number of stations and location
- Number of disk drives and location
- Number and type of additional peripherals
- Placement of boards within CPU
- Etc.

Software Configuration
- Number of tasks
- Revision level
- Task priorities
- Location of software
- Location of data
- Number of sectors per cluster
- Amount of data on disc(s)
- Etc.

Physical Environment
- Temperature
- Humidity
- Power isolation
- Air flow

Activity
- Number of tasks in use
- Type of activity
- Part sizes involved
- Time of day (believe it or not)

System Description

The system used in this test consisted of the following hardware:

- One CGP100 with tape drive
- Eight design stations; one located over 1,800 feet away
- One Decwriter
- One CVD digitizer with Lear-Siegler
- Two 80 MB disc drives
- Two Versatec electrostatic plotters (24 inch and 42 inch)

The software consisted of the following:
- CGOS100/10 O/S
- Revision 10.021

- Applications software for MD, PC/ES, N/C, sheet metal development, wire-wrap development, plant layout, engineering, and full programming capability
- Four sectors per cluster
- Ten tasks (as used)
- Default reformat procedure
- Two drives treated as one large volume
- Status of approximately 18,000 (hex) sectors available and 14,000 (hex) sectors contiguous

Note: This was probably the heaviest loaded CPU in the Computervision world. Consequently, it was also probably the slowest. However, it should be noted that the system performed as well or better than it did when there were only four stations on it. The Computervision system seems to be a real "workaholic."

Test Procedures

The first portion of the test was run on June 17, 1980, on third shift. The four application execute files were run on the Decwriter separately for determination of time test length.

The second portion of the test was run on June 19 and June 20, 1980, on third shift. The mechanical execute file was timed on the remote station (37) for comparison with local stations. Then, the system was loaded with four stations using the four application execute files and the other stations using the mechanical execute file. These were simultaneously tested starting with eight stations and ending with four stations—eight stations while plotting; eight stations while taping; eight stations; seven stations; six stations; five stations; four stations; four stations while plotting; four stations while taping; finally, the four application execute files were run on Station 21 for comparison of time test length. In all of the second portion, the unused stations were logged out.

The third portion of the test was run on June 23, 1980, on third shift. The test was run to compare time differences between unused stations being logged on or off. In this test, all unused stations (including the Decwriter) were logged on and put into cards. Then, the four-station test of portion two was duplicated. It was assumed that if there were any differences caused by the state of unused stations, those differences would be greatest when the test was run with just four stations.

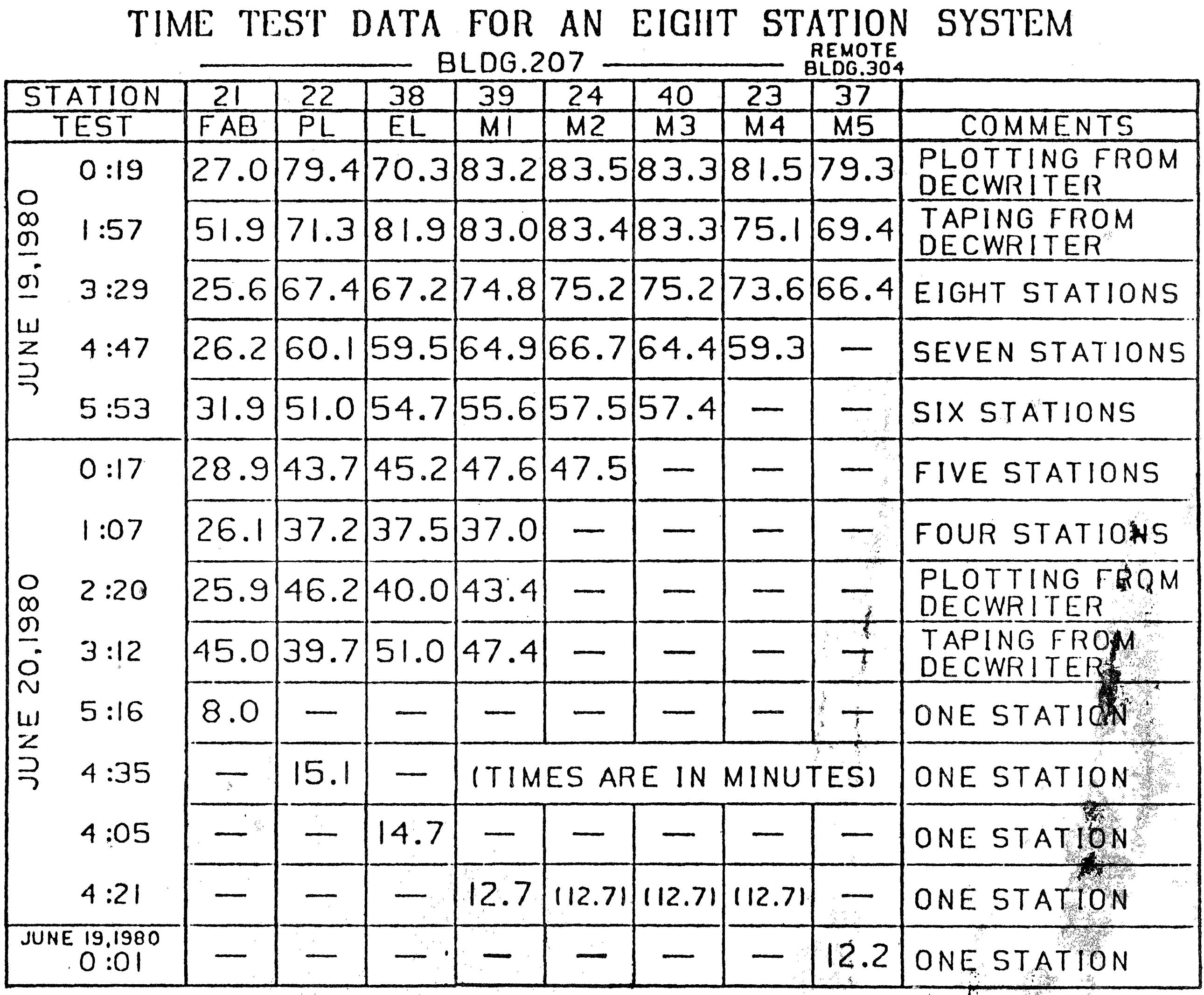
The last portion of the test was run on June 26, 1980, on first shift. The test was run to compare time differences between hardware configurations. In this test, the system hardware and software configuration was changed to four stations. Then, the four-station test of portion two was duplicated. It was assumed that if there were any differences caused by configuration, those differences would be greatest when the configuration was reduced to four stations.

Test Conclusions

Part 1 (when compared with Part 2) showed that there was little difference between test times on the Decwriter and those at the design stations. The exception to this was the fabrication test, which was considerably longer because of all the carriage returns between each of the outputs.

Part 2 showed that there was little difference in test time between the remote and local stations. The rest of the results of Part 2 have been charted and graphed (see Figures 1 through 7). Figure 1 shows the test times in minutes. Figure 2 shows the ratio of degradation based on a one-station operation, and Figure 3 shows the ratio of degradation based on a four-station operation. Figures 4 through 7 are plots of the degradation for each of the applications tests. Figure 4 shows the fabrication application runs about the same speed regardless of the number of stations being used. The fabrication test seemed to share time with the other tests when five and six stations were being used. But, as the seventh and eighth stations were added, the fabrication test went back to the speed it ran at when there were four stations running. This is probably due to the priority of operations involved. Figures 5 through 7 showed that the other applications have approximately the same amount of linear degradation as the number of stations in use increases. The mechanical test showed a slightly higher degradation than the others, but this could be because the same mechanical test was used for five of the stations. If the electrical test was used for the extra terminals, it would probably have shown more degradation than the others. The plotting and taping tests showed the following percentages of degradation.

	Plotting With		Taping With	
	Four Stations	Eight Stations	Four Stations	Eight Stations
Electrical Tool Design	6%	4%	36%	22%
Mechanical Tool Design	17%	11%	28%	10%
Fabrication	−1%	6%	72%	103%
Plant Layout	24%	18%	7%	6%



TIME TEST DATA FOR AN EIGHT STATION SYSTEM

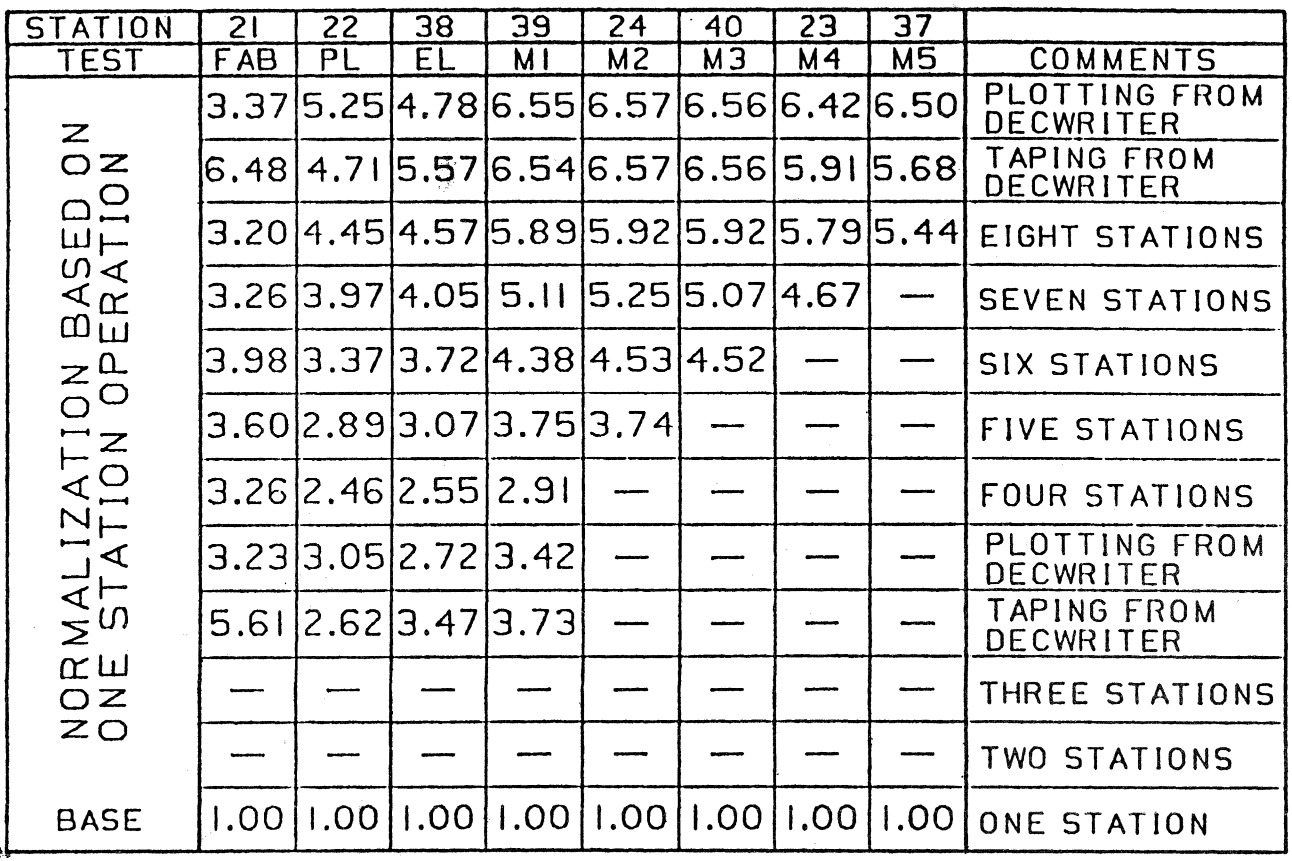
BLDG.207 — REMOTE BLDG.304

	STATION	21	22	38	39	24	40	23	37	
	TEST	FAB	PL	EL	M1	M2	M3	M4	M5	COMMENTS
JUNE 19,1980	0:19	27.0	79.4	70.3	83.2	83.5	83.3	81.5	79.3	PLOTTING FROM DECWRITER
	1:57	51.9	71.3	81.9	83.0	83.4	83.3	75.1	69.4	TAPING FROM DECWRITER
	3:29	25.6	67.4	67.2	74.8	75.2	75.2	73.6	66.4	EIGHT STATIONS
	4:47	26.2	60.1	59.5	64.9	66.7	64.4	59.3	—	SEVEN STATIONS
	5:53	31.9	51.0	54.7	55.6	57.5	57.4	—	—	SIX STATIONS
JUNE 20,1980	0:17	28.9	43.7	45.2	47.6	47.5	—	—	—	FIVE STATIONS
	1:07	26.1	37.2	37.5	37.0	—	—	—	—	FOUR STATIONS
	2:20	25.9	46.2	40.0	43.4	—	—	—	—	PLOTTING FROM DECWRITER
	3:12	45.0	39.7	51.0	47.4	—	—	—	—	TAPING FROM DECWRITER
	5:16	8.0	—	—	—	—	—	—	—	ONE STATION
	4:35	—	15.1	—	(TIMES ARE IN MINUTES)					ONE STATION
	4:05	—	—	14.7	—	—	—	—	—	ONE STATION
	4:21	—	—	—	12.7	(12.7)	(12.7)	(12.7)	—	ONE STATION
JUNE 19,1980	0:01	—	—	—	—	—	—	—	12.2	ONE STATION

Figure 1

TIME TEST DATA FOR AN EIGHT STATION SYSTEM

——— BLDG.207 ——— REMOTE BLDG.304

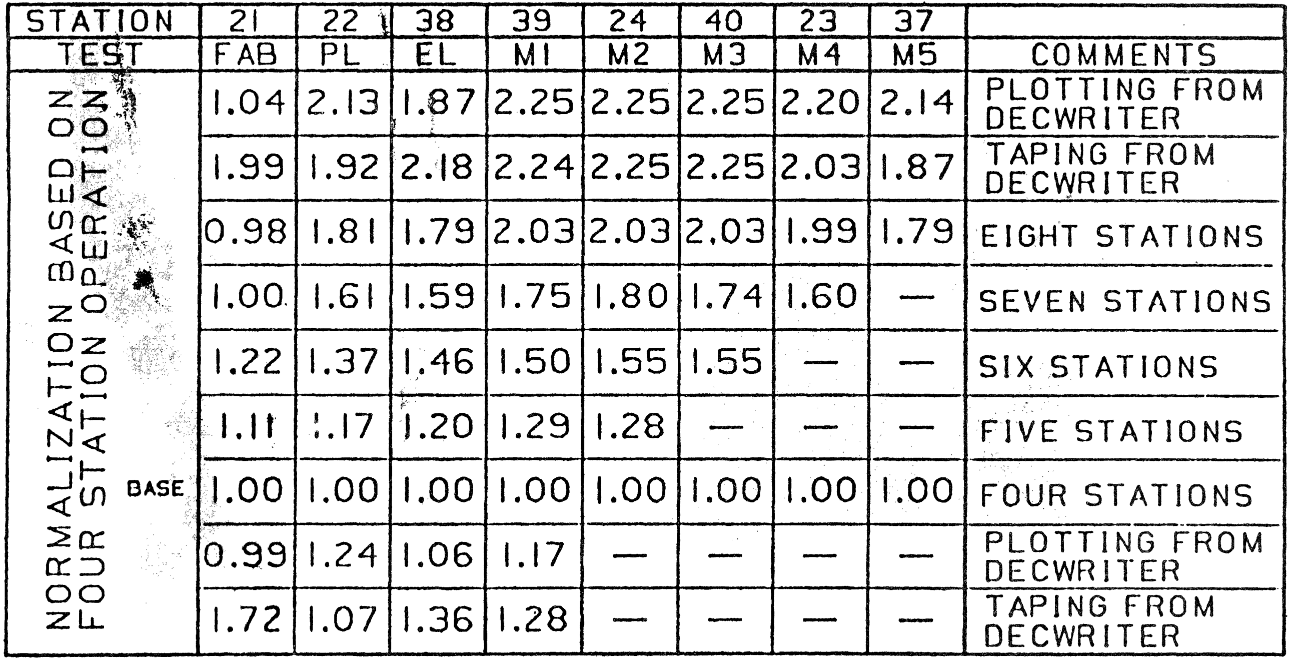


STATION	21	22	38	39	24	40	23	37	
TEST	FAB	PL	EL	MI	M2	M3	M4	M5	COMMENTS
NORMALIZATION BASED ON ONE STATION OPERATION	3.37	5.25	4.78	6.55	6.57	6.56	6.42	6.50	PLOTTING FROM DECWRITER
	6.48	4.71	5.57	6.54	6.57	6.56	5.91	5.68	TAPING FROM DECWRITER
	3.20	4.45	4.57	5.89	5.92	5.92	5.79	5.44	EIGHT STATIONS
	3.26	3.97	4.05	5.11	5.25	5.07	4.67	—	SEVEN STATIONS
	3.98	3.37	3.72	4.38	4.53	4.52	—	—	SIX STATIONS
	3.60	2.89	3.07	3.75	3.74	—	—	—	FIVE STATIONS
	3.26	2.46	2.55	2.91	—	—	—	—	FOUR STATIONS
	3.23	3.05	2.72	3.42	—	—	—	—	PLOTTING FROM DECWRITER
	5.61	2.62	3.47	3.73	—	—	—	—	TAPING FROM DECWRITER
	—	—	—	—	—	—	—	—	THREE STATIONS
	—	—	—	—	—	—	—	—	TWO STATIONS
BASE	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	ONE STATION

Figure 2

TIME TEST DATA FOR AN EIGHT STATION SYSTEM

——— BLDG.207 ——— REMOTE BLDG.304



STATION	21	22	38	39	24	40	23	37	
TEST	FAB	PL	EL	MI	M2	M3	M4	M5	COMMENTS
NORMALIZATION BASED ON FOUR STATION OPERATION	1.04	2.13	1.87	2.25	2.25	2.25	2.20	2.14	PLOTTING FROM DECWRITER
	1.99	1.92	2.18	2.24	2.25	2.25	2.03	1.87	TAPING FROM DECWRITER
	0.98	1.81	1.79	2.03	2.03	2.03	1.99	1.79	EIGHT STATIONS
	1.00	1.61	1.59	1.75	1.80	1.74	1.60	—	SEVEN STATIONS
	1.22	1.37	1.46	1.50	1.55	1.55	—	—	SIX STATIONS
	1.11	1.17	1.20	1.29	1.28	—	—	—	FIVE STATIONS
BASE	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	FOUR STATIONS
	0.99	1.24	1.06	1.17	—	—	—	—	PLOTTING FROM DECWRITER
	1.72	1.07	1.36	1.28	—	—	—	—	TAPING FROM DECWRITER

Figure 3

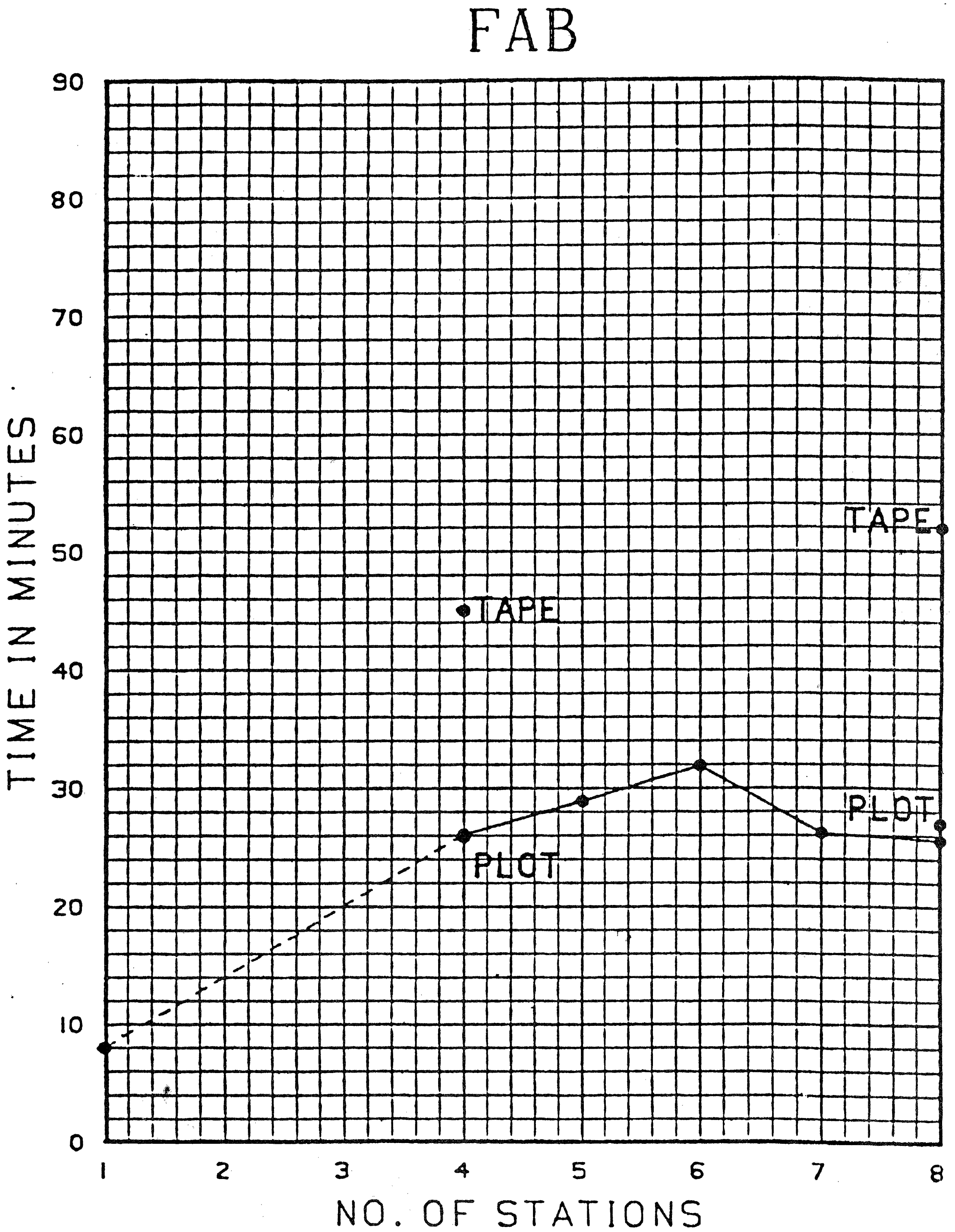

FAB
TIME IN MINUTES
0
10
20
30
40
50
60
70
80
90
NO. OF STATIONS
1
2
3
4
5
6
7
8
TAPE
TAPE
PLOT
PLOT

Figure 4

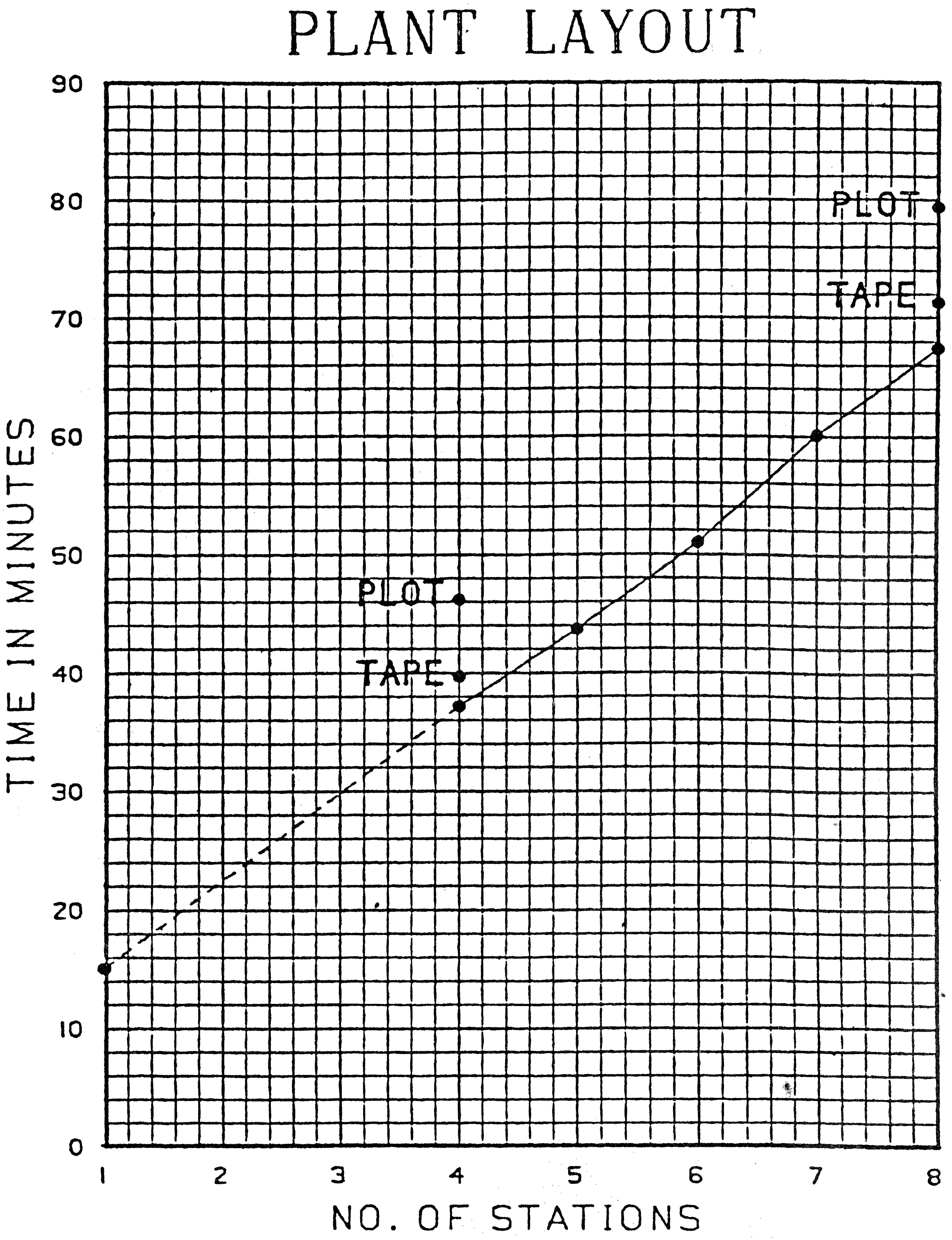

PLANT LAYOUT
TIME IN MINUTES
0
10
20
30
40
50
60
70
80
90
NO. OF STATIONS
1
2
3
4
5
6
7
8
PLOT
TAPE
PLOT
TAPE

Figure 5

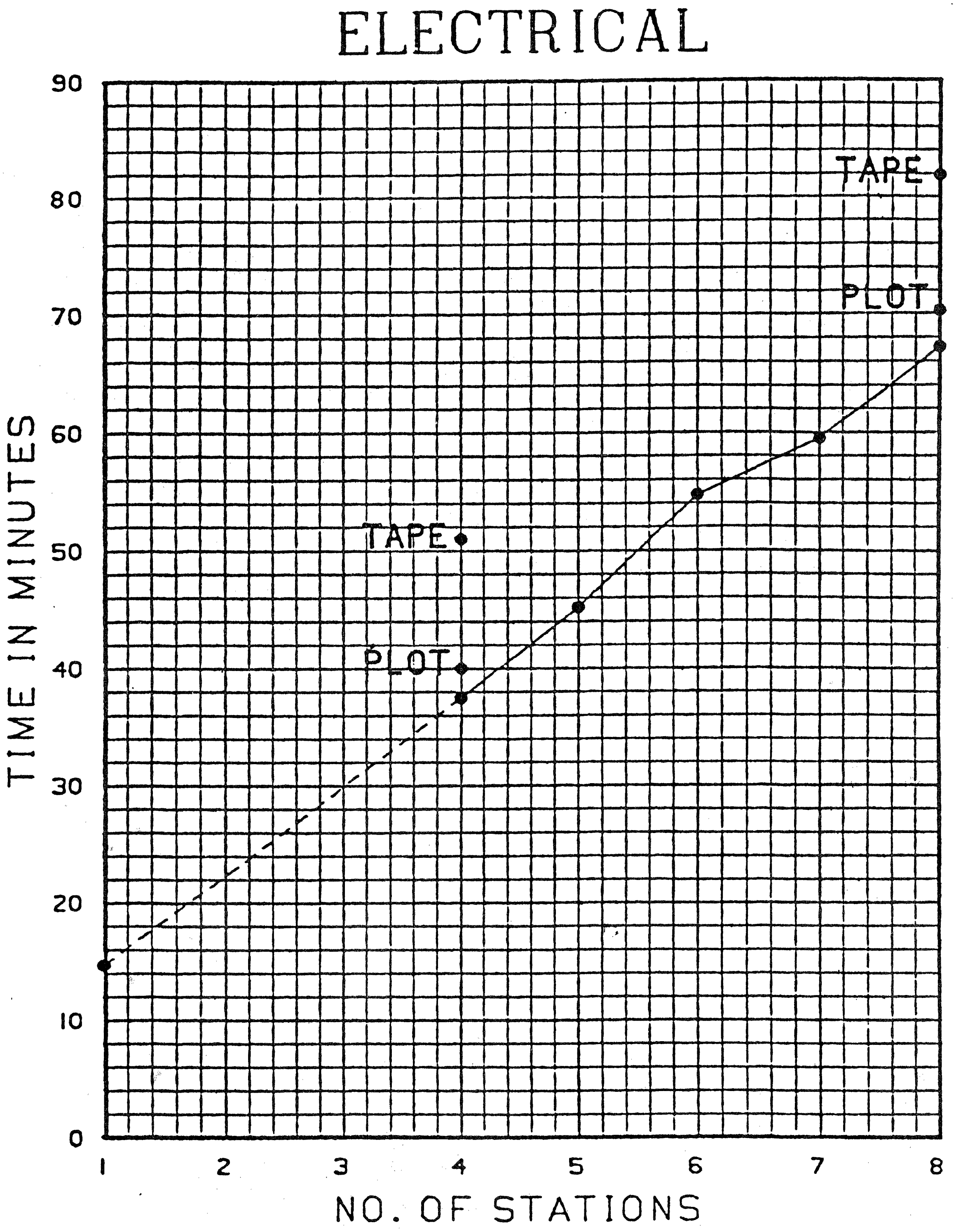

ELECTRICAL
TIME IN MINUTES
0
10
20
30
40
50
60
70
80
90
1
2
3
4
5
6
7
8
NO. OF STATIONS
TAPE
PLOT
TAPE
PLOT

Figure 6

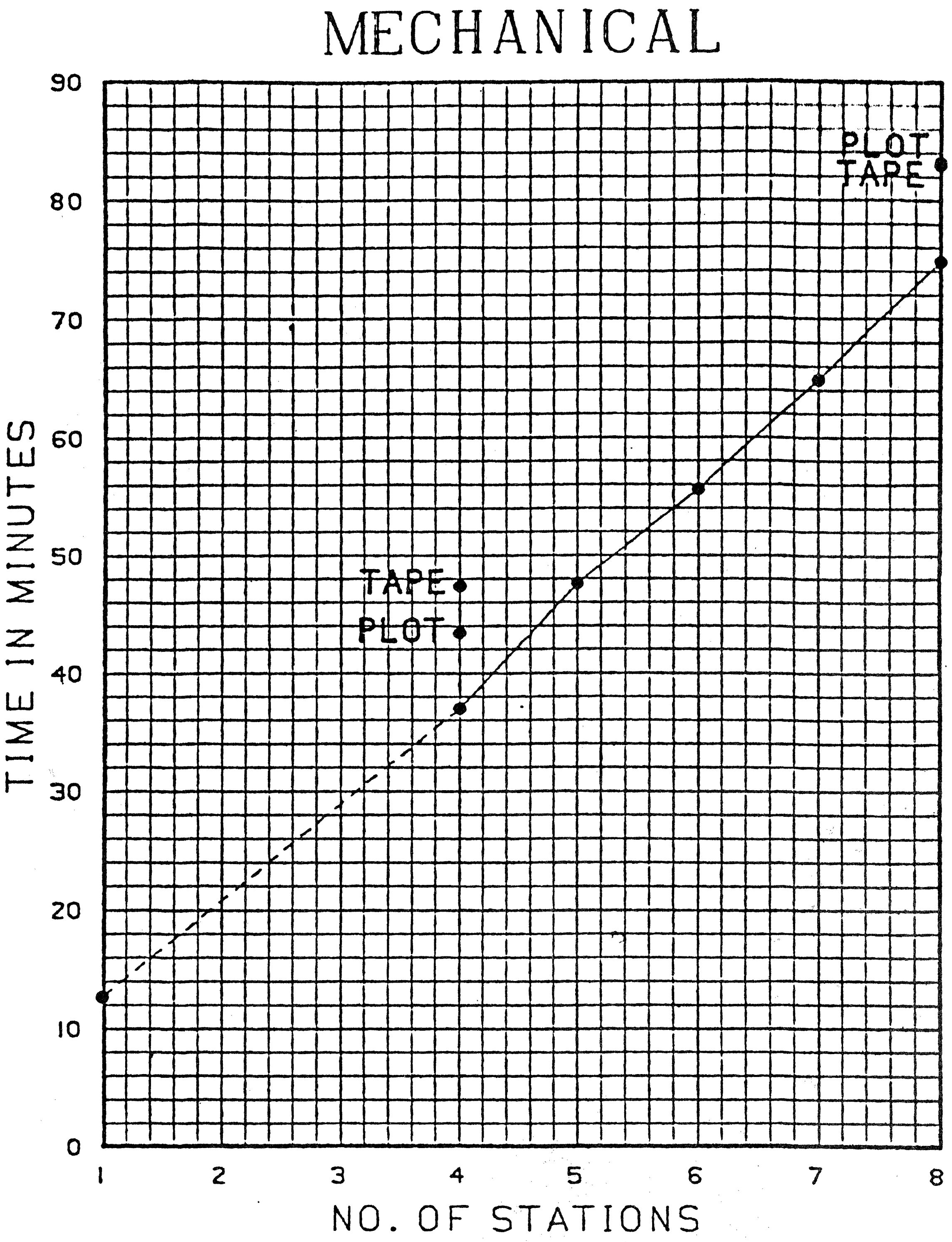

MECHANICAL
TIME IN MINUTES
0
10
20
30
40
50
60
70
80
90
NO. OF STATIONS
1
2
3
4
5
6
7
8
TAPE
PLOT
PLOT
TAPE

Figure 7

Part 3 showed that it makes no difference whether the unused stations are logged on or off.

Part 4 showed that there is little change in degradation when the hardware and software configurations are changed. While all the tests ran faster, the change was only an average of 3 percent.

Productivity Analysis and Evaluation

The main purpose for the tests is to determine the optimum configuration per CPU on a Computervision computer graphics system. With all the variables involved, it is very difficult to predict exactly what to expect in any given situation. The tests showed some trends that, until more data is available, we will have to base our decisions upon.

Using Figure 3 on Page 403, we can determine what will happen to productivity when terminals are added to the basic four-station system. The base times of the four application execute files were normalized to one to develop a ratio of speed degradation. The fabrication data shows no apparent change as terminals are added. This implies that the fabrication application takes first priority over all else, regardless of the system load. If the fabrication test had been run against itself, then the fabrication data would show a speed degradation similar to the other applications. We will average the degradation of the plant layout, electrical, and mechanical tests.

Average degradation of the execute file tests:

Five-station system	23.5%
Six-station system	48.6%
Seven-station system	68.2%
Eight-station system	92.4%

In determining productivity decrease, there are two factors that must be considered along with the test data itself. They are the percentage of execute file speed at which the user works and the percentage of design time actually using the system. Because the designer does not keep the system continually busy like the test execute files do, we will factor in a percentage of efficiency. Since we do not have any exact figures, we will compute for 70 percent, 80 percent, and 90 percent efficiency. In addition to efficiency, a designer does not spend all of his time using the system. Again, not having any exact figures, we will compute for 70 percent, 80 percent, and 90 percent system usage. These two factors will reduce the percentage of degradation and give us a more accurate picture of the productivity decrease.

First, we will use the following formula to calculate the increased time needed to complete a job due to degradation.

Percent Degradation × User Efficiency = Increased Time Needed to Complete Job

Number of Design Stations	Percent Degradation	Increased Time Needed to Complete Job at Efficiency of		
		70%	80%	90%
5	23.5%	16.45%	18.80%	21.15%
6	48.6%	34.02%	38.88%	43.74%
7	68.2%	47.74%	54.56%	61.38%
8	92.4%	64.68%	73.92%	83.16%

Then, we plug the increased time value into this next formula to determine the productivity ratio adjustment factor.

$$\frac{\begin{array}{c}\text{Design Time}\\ \text{Utilizing}\\ \text{the System}\end{array} + \begin{array}{c}\text{Design Time}\\ \text{Not Utilizing}\\ \text{the System}\end{array}}{\begin{array}{c}\text{Design Time}\\ \text{Utilizing}\\ \text{the System}\end{array} + \left[\begin{array}{c}\text{Increased Time}\\ \text{Needed to}\\ \text{Complete Job}\end{array} \times \begin{array}{c}\text{Design Time}\\ \text{Utilizing}\\ \text{the System}\end{array}\right] + \begin{array}{c}\text{Design Time}\\ \text{Not Utilizing}\\ \text{the System}\end{array}} = \begin{array}{c}\text{Productivity}\\ \text{Ratio}\\ \text{Adjustment}\\ \text{Factor}\\ \text{(PRAF)}\end{array}$$

Example using 100 hour job with 80 percent efficiency and 80 percent usage on five stations:

$$\frac{80\text{ Hours} + 20\text{ Hours}}{80\text{ Hours} + (18.8\% \times 80\text{ Hours}) + 20\text{ Hours}} = .869$$

Assuming that a user spends 70 percent of the design time using the system:

Number of Design Stations	The PRAF for a 70 Percent Efficient Designer Will Be	The PRAF for an 80 Percent Efficient Designer Will Be	The PRAF for a 90 Percent Efficient Designer Will Be
5	.897	.884	.871
6	.808	.786	.766
7	.750	.724	.699
8	.688	.659	.632

Assuming that a user spends 80 percent of the design time using the system:

Number of Design Stations	The PRAF for a 70 Percent Efficient Designer Will Be	The PRAF for an 80 Percent Efficient Designer Will Be	The PRAF for a 90 Percent Efficient Designer Will Be
5	.884	.869	.855
6	.786	.763	.741
7	.724	.696	.671
8	.659	.628	.600

Assuming that a user spends 90 percent of the design time using the system:

Number of Design Stations	The PRAF for a 70 Percent Efficient Designer Will Be	The PRAF for an 80 Percent Efficient Designer Will Be	The PRAF for a 90 Percent Efficient Designer Will Be
5	.871	.855	.840
6	.766	.741	.718
7	.699	.671	.644
8	.632	.600	.572

Now, we will use these PRAF's to determine what will happen to our total system output when compared with the original CAR prediction. The original prediction was that we would achieve the following increases in productivity as a result of using computer graphics:

Electrical Tool Design	2.0 to 1
Mechanical Tool Design	1.8 to 1
Fabrication Applications	3.5 to 1
Plant Layout Applications	1.5 to 1

To obtain an estimate of the total system output, we will simulate our usage of a four-station system. During the first shift, the fabrication and plant layout applications would each use one station. The other two stations would be used by tool designers — electrical or mechanical depending on the week. During the second shift, all four stations would be used by tool design. Two stations would be used by mechanical and the other two stations would be used by either mechanical or electrical tool design, again, depending on the week. Thus, the estimated hours saved would be affected as follows in the data below.

Application	Number of Eight-Hour Users	Manual Methods	Four-Station System	Productivity Ratio
Electrical Tool Design	2	16.0 Hrs.	32.0 Hrs.	2.00 to 1
Mechanical Tool Design	4	32.0 Hrs.	57.6 Hrs.	1.80 to 1
Fabrication	1	8.0 Hrs.	28.0 Hrs.	3.50 to 1
Plant Layout	1	8.0 Hrs.	12.0 Hrs.	1.50 to 1
Totals	8	64.0 Hrs.	129.6 Hrs.	2.03 to 1

Assuming that a user spends 80 of the design time using the system and that a user works at 80 percent of the execute file speed:

With the addition of a fifth station being used for plant layout applications on the first shift and electrical tool design on the second shift, the following amounts of work would be accomplished:

Application	Number of Eight-Hour Users	Manual Methods	Five-Station System	Productivity Ratio
Electrical Tool Design	3	24.0 Hrs.	41.8 Hrs.	1.74 to 1
Mechanical Tool Design	4	32.0 Hrs.	49.9 Hrs.	1.56 to 1
Fabrication	1	8.0 Hrs.	24.3 Hrs.	3.04 to 1
Plant Layout	2	16.0 Hrs.	20.8 Hrs.	1.30 to 1
Totals	10	80.0 Hrs.	136.8 Hrs.	1.71 to 1

With the addition of a sixth station being used for mechanical and electrical tool design on both shifts, the following amounts of work would be accomplished.

Application	Number of Eight-Hour Users	Manual Methods	Six-Station System	Productivity Ratio
Electrical Tool Design	4	32.0 Hrs.	49.0 Hrs.	1.53 to 1
Mechanical Tool Design	5	40.0 Hrs.	54.8 Hrs.	1.37 to 1
Fabrication	1	8.0 Hrs.	21.4 Hrs.	2.67 to 1
Plant Layout	2	16.0 Hrs.	18.2 Hrs.	1.14 to 1
Totals	12	96.0 Hrs.	143.4 Hrs.	1.49 to 1

With the addition of a seventh station being used for mechanical and electrical tool design on both shifts, the following amounts of work would be accomplished:

Application	Number of Eight-Hour Users	Manual Methods	Seven-Station System	Productivity Ratio
Electrical Tool Design	5	40.0 Hrs.	55.6 Hrs.	1.39 to 1
Mechanical Tool Design	6	48.0 Hrs.	60.0 Hrs.	1.25 to 1
Fabrication	1	8.0 Hrs.	19.5 Hrs.	2.44 to 1
Plant Layout	2	16.0 Hrs.	16.6 Hrs.	1.04 to 1
Totals	14	112.0 Hrs.	151.7 Hrs.	1.35 to 1

With the addition of an eighth station being used for fabrication applications on the first shift and electrical tool design on the second shift, the following work would be accomplished:

Application	Number of Eight-Hour Users	Manual Methods	Eight-Station System	Productivity Ratio
Electrical Tool Design	6	48.0 Hrs.	60.5 Hrs.	1.26 to 1
Mechanical Tool Design	6	48.0 Hrs.	54.2 Hrs.	1.13 to 1
Fabrication	2	16.0 Hrs.	35.2 Hrs.	2.20 to 1
Plant Layout	2	16.0 Hrs.	15.0 Hrs.	0.94 to 1
Totals	16	128.0 Hrs.	164.9 Hrs.	1.29 to 1

All of these test results are dependent upon the test parameters as outlined in Pages 2, 3, and 4. The large number of parameters involved shows the high degree of variability that is possible in actual system use. However, when attempting to maximize the capital investment, the manner in which the system is used has a much greater impact on total system output. It is, therefore, essential that systems be examined on an individual basis before making an investment decision.

While the data may be inconclusive because of the high degree of variability involved, the information gained is invaluable. We were unable to find any real world data on speed degradation anywhere. In summary, the following statements can be made based on the test results (when conditions are similar to the test conditions):

- Time sharing between design stations is equivalent.
- Speed of operation is not affected by distance from the CPU.
- Leaving unused terminals logged off does not affect the speed differently than leaving them logged in.
- Reconfiguring the hardware and software to eliminate unused terminals can increase system speed by as much as 3 percent.
- Plotting does not degrade the system speed any more than normal design operation would (except for plant layout).
- Taping slows down operations by as much as one third (plant layout hardly affected, and fabrication affected by as much as 100 percent).
- Speed degradation is linear with the addition of the fifth, sixth, seventh, and eighth terminals.
- N/C applications take a high priority in comparison with other applications.
- Generally, system users should see the following productivity losses:

Five Stations	13%	Seven Stations	30%
Six Stations	24%	Eight Stations	37%

Additional Comments

During the time that our system was loaded so heavily, we were able to do a great deal of training of designers. As trainees, the degradation was not as great of a problem. By purchasing our second CPU and splitting up the terminals after the bulk of the training was completed, we were able to maximize the return on investment.

Also, the reduced speed of the system forced us to off-load work, where possible, to the third shift, and use various means to do productive work without requiring the designer's presence at the terminal. We might not have tried so hard to off-load work had the degradation been less. The techniques learned during that time when degradation was bad are being applied to further increase our productivity.

A 59 page report of all the details of this test is available by contacting:

Blaine E. Burnett
Xerox Corporation
800 Phillips Road
Building 207A
Webster, New York 14580
(716) 422-9459

STANDARDIZED RESPONSE TIME MODULUS DETERMINATION ROUTINE

Doug Schuler
Boeing
P.O. Box 3707
M/S 6A-26
Seattle, WA 98124
206-251-2487

ABSTRACT

The purpose of this paper is to demonstrate a simple routine to determine the systems response time.

This paper describes the use of a modifiable routine to find the response time modulus through a series of repeated commands and calculations based on the elapsed time. The routine has several forms for time checks in various modes. Standardization execute files running on other tasks with test parts are used to establish relative modulus readings. These routines can be effectively used in determining task, budget, shift, or manpower allocations. It could even be used to determine which types of routines are best left to a second or third shift due to their extreme predilections towards slowing down the system. The routine can also be utilized in benchmark and acceptance testing.

This paper is to demonstrate a simple method for determining the response time of the system. The routine uses the DATE command 100 times and divides the elapsed time by 100 to determine the average time per response or response modulus.

The routine as it now exists (before any major revisions or enhancements for more demanding or specific applications are made) consists of two portions; an execute file and a parex file. The initial and main portion is the execute file which first deletes the files that were produced last time the program was run. In a more complex incarnation of the program the new information might be simply appended onto a preexisting file for a running account. The file then makes a file of the conditions at the onset by nesting a LSTTSK in HARDFILE NN.CONDITIONS. The command DATE is then given 100 times within a HARDFILE. NN.DATE. 100 is an arbitrary number here and any number will do. I picked 100 because, I felt, it would not give undue attention to a transient surge or lag, nor give rise to errors in rounding. Using DATE for the command to be repeated is a reasonable choice for two reasons. 1) It gives the time directly, thus keeping the number of commands used to a minimum, and 2) It is a low priority command.

The execute file then edits the brand new file of 100 dates and times. It deletes lines 2-99, leaving the residue of the first and last DATE command. After some copies of lines are made and some characters are deleted, the line "RUN PAREX NN.MODULUS" is inserted as the first line of the file. The file now looks like this:

```
RUN PAREX NN.MODULUS
HO  (hours at onset)
MO  (minutes at onset)
SO  (seconds at onset)
HE  (hours at end)
ME  (minutes at end)
SE  (seconds at end)
```

The edited NN.DATE is filed and main execute program proceeds to enter a dummy part and executes NN.DATE. The first line of NN.DATE says to run parex NN.MODULUS. The time variables are dropped into the parex file as type 1 parameters. Type 4 calculations within the parex file convert the hours, minutes, and seconds at the onset and the hours, minutes, and seconds at the end into a number of seconds. The difference is then found between these two derived numbers and is divided by 100 to find an average time per command or "response modulus". The last two lines of the parex file are:

```
*  CONDITIONS AT ONSET OF TEST CAN BE FOUND IN NN.CONDITIONS
*                                  RESPONSE MODULUS = &MOD
```

(where &MOD is the obtained ratio)

After the parex file has run its course the original execute file resumes. At this point it merely exits from the part, does not file, and returns to system level.

This routine utilizes no graphics and can, therefore, be run with a Lear-Siegler alone. The program as outlined can only be run on one task at a time. Running the program on more than one task may result in erroneous data or hung tasks as the programs would be deleting, creating, and editing the same files in an incorrect and arbitrary order. If it is necessary to run the program concurrently on several tasks, all files used or created must be renamed, probably with the task number as part of the file name.

The program as it exists is skeletal. It will yield a response-modulus which is significant over a specific amount of time (typically 40-200 seconds). It would require some enhancement for more ambitious testing. Although the proliferation of files might be considered a drawback, a program that created files NN.CONDITIONS.1, NN.CONDITIONS.2 and so on, along with a file that consisted of the corresponding moduli, would be advantageous for rigorous investigation. Running the enhanced program concurrently with test execute files on other tasks with RUN MARK utilized could constitute a powerful testing approach with both response-moduli and real time values documented in the created files. By conducting preliminary tests using the program and by isolating variables such as number of tasks in operation, percentage of disk filled, types of commands utilized, peripherals in use, size of parts and the like, rational allocation policies can be determined.

16 MENUS IN 1

Ken Mosher
Senzei Link
Kirkwood Industrial Park
Dept 466
Binghamton, NY 13902
607-772-3340

BRIEF

Here is a method of easily expanding one's menu capacity without having to change the menu graphics sheet. Several different versions of this method can be applied to suit the individual's needs. One such method will be presented here.

HISTORY

Our initial development of menu #1 was to make each square a complete command, ready for data input. This worked fine until we became smarter and more knowledgeable of the CADDS language. Because it was so automatic, it didn't allow for the addition of extra modifiers before data input. So, it was modified to remove some of the ":" to allow for the addition of extra modifiers. Still a lot of typing had to be done to type in the modifier names. Menu #2 was developed by revising menu #1. Unused commands were eliminated to allow room for more commonly used commands. Some were broken up into separate squares. Although this meant digitizing more squares to complete a command, additional squares were gained for new words. This menu worked fine for most design applications, but wasn't too efficient when a lot of detail drawings were being done on the system. A lot of typing still had to be done. Also there weren't enough blank squares to edit key for special functions.

Menu #3 took a completely different approach and philosophy. Each square was a different letter, number, noun or verb. Also 16 squares were left blank for Edit Key. Although several squares were required to complete a command, very little typing had to be done. When the 16 blank squares reserved for Edit Key became filled up, a copy of Menu #3 was done to make Menu #4 with 16 blank squares.

This led to Menu #5, now with 2 small 16 square paper overlays to keep track of what was on #3, #4, and #5. These paper overlays led to the creation of 16 Menus in 1.

PRESENT

Now with the original 220 square Menu #1, plus the additional 240 squares (15 × 16 = 240) from Menu #2 thru Menu #16, the flexibility, completeness, and automation exceeded our current needs.

Our operators were very receptive to this new menu, because the basic original menu was left intact. They didn't have to learn a complete new menu. They only had to become familiar with the new mini menus. Also their efficiency has definitely improved. Most of the typing errors have been eliminated. Required modifiers aren't left out. Library part names don't have to be looked up. Many of the squares now run programs; like EXEC, PEP, PAREX, and VARPRO, further increasing their efficiency.

FUTURE

We are currently structuring specialized menus for Structures-Frames, Control Panels, PC-ES, NC, etc.

HOW TO CREATE

Take an existing menu and Key File or create a new one, leaving a 4 × 4 section of squares blank, preferably in the lower 110 squares, either on the left or right side. Copy this Key File 15 times, naming each successive file Menu #2, Menu #3, . . . , Menu #16. Edit each menu's 16 blank square sections to your desired needs.

Two additional squares, preferably adjoining the blank section, are required. One is to tell the user which menu # one is currently active on. The other is a variable command to change from menu to menu.

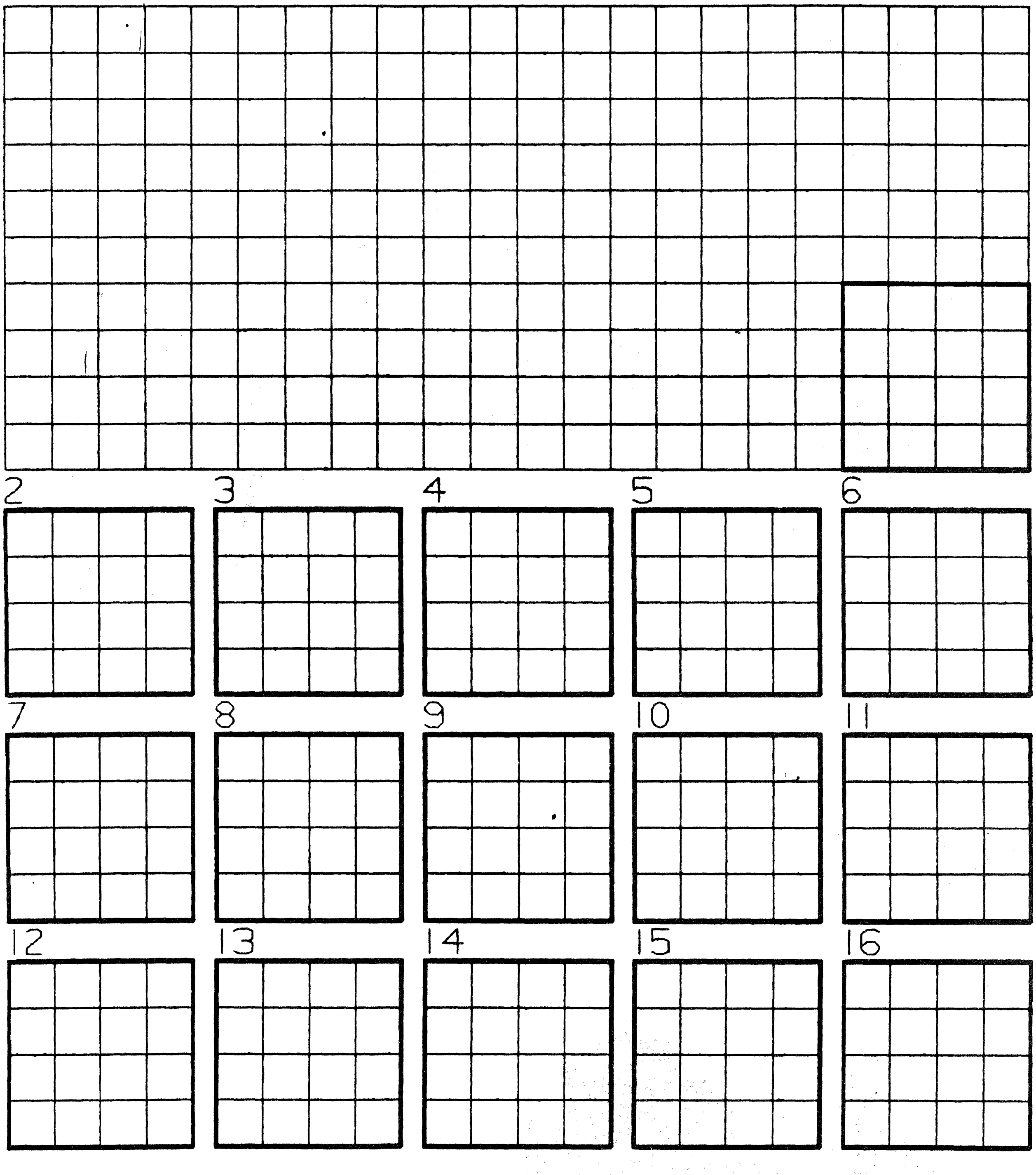
For the menu graphics add the 4 × 4 section of squares, 15 times below the original 220 squares. You now have one menu graphics sheet with 460 squares on it.

HOW TO USE

Menu #1 is used in the normal manner by digitizing the square with the desired command. Menus #2 thru #16 are used by restoring the desired menu # for the command(s) needed and digitizing the corresponding blank square of Menu #1 to activate the command.

Menu #1 blank square area can be used for special commands by doing an Edit Key for it.

MDET.MENU1



KEYFILE & MENU CONSTRUCTION

COMPUTERVISION CADDS 3 INTERACTIVE GRAPHICS SYSTEM MECHANICAL DESIGN

Joe H. Harrison
Vought Corporation
P.O. Box 225907
Dallas, TX 75265

ADVANCED KEYFILES

I. Elements in the Basic Contruction of Keyfile
   A. Layout
   B. Function Keys
   C. Color
   D. Programs
II. Other Considerations
   A. Special Characters
   B. Carriage Returns
   C. Grouping
   D. Chaining
   E. Program Calling Keys
III. Example Keyfile Problem
IV. How It Works
   A. Get Keys
   B. Hidden Keys
V. Color
   A. Grouping
   B. Color By Type

CONTENTS

LIST OF ILLUSTRATIONS

FIGURE

ABSTRACT

The purpose of this paper is to demonstrate the construction and organization of Mechanical Design Menu.

This paper describes the use of 36 extra keys described in the CADDS 3 Graphics Operator Manual. These keys are utilized in conjunction with the 220 keys found in regular keyfiles and their operation is similar to a function key on a hand-held calculator. Additionally, this paper describes the use of operator callable programs with keyfiles, the layout and design of keyfiles and the use of color in keyfiles.

INTRODUCTION

The use of keyfiles, or menus, is one of the designer's main tools in utilizing CADDS 3 Mechanical Designs Package, or any other package for that matter.

Menus allow the chaining of commands, calling of programs and the storage of information utilized within programs.

Designers utilizing such menus may out-perform any others who do not. This is due to the fact that typing errors and duplicity in "key-ins" are removed plus the greater advantage in program handling and calling from the keyboard. The principle of operator programmable keyfiles if one of Computervision's most desirable and most used programming features.

KEYFILE CONSTRUCTION

Keyfile construction may be divided into the four following areas:

1. Layout and Grouping of Keys
2. Special Function Keys
3. Color Separation
4. Programming the Keys

This discussion shall be limited to CADDS 3 Graphics System.

Other considerations in keyfile construction are the use of special characters, control characters, the use of carriage return, the grouping of commands, chaining keys, using keys to call programs and other keyfiles.

Special characters such as control "X" and control "E" may be easily included into any keyfile, and should be in sequence to allow one hand free to type. Lower case delimeter should be utilized whenever possible for the same reason.

Carriage returns should be inserted before many commands in order to omit a second digitize when one command is finished. Also, spaces, colons and question marks can be utilized on some commands.

Chaining commands are strings of commands which have the entire commands programmed into one key such as " CR INS LIN HOR :" thus replacing 2 keys and 5-key digitizes.

Program calling keys can be placed into a menu anywhere; however, most special program calling keys should be placed in overlays which will be described later.

LAYOUT AND GROUPING OF KEYS

Before layout begins, consideration must be given to the nature of operations to be accomplished by the keyfile. In this example the keyfile to be designed will be general purpose 2-D and 3-D Mechanical Design.

The Groups for consideration will be as follows:

A. All construction commands and modifiers (INS - LIN, CIR, BSPL, ARC, FIL, SPL, CONIC, ETC)
B. All 3-D construction commands and modifiers (INS TCYL, CUT PLANE ETC)
C. Views and Ports and Modifiers
D. Special GET DATA Modifers
E. Zoning, Paging and Displays
F. Text, Dimensions and Labels
G. Plotting, Printing, Gerbering
H. Numerics and Alpha Characters
I. Control Characters
J. Listing (views, ports, parameters, layer, mode, status, etc.)

Grouping should begin in the center of the keyfile for the most utilized commands, working toward the outside and bottom for the lesser used commands. Similar commands should be grouped in an overlapping fashion according to the "A" through "J" listing above. Don't waste keyfile space with too many single commands or things like the alphabet. However, do include numerical 0–9 using the same layout as on a calculator. Don't use parex for calculate expression. Add several parenthesis in calculate expression (left parenthesis) so that multiple calculations may be made. Likewise, in the right parenthesis.

SPECIAL FUNCTION KEYS

Special function keys are keys which may have any number of functions. Select 10 to 30 keys in one area (right side, or left side, top or bottom) and make a hinged overlay (many overlays may be used) which can be flipped over (like pages in a book) exposing the keys to be utilized. This permits the use of several hundred keys, 20 keys or less will work the fastest. To make the key active, build a keyfile (in the Editor) utilizing the same key number corresponding to the key in that portion of the menu selected. For example, suppose you wish to have the following commands on your menu:

1. INS TEXT
2. INS LAB
3. INS LDIM

You want all of their modifiers as well, but if you do this, you must leave out other commands which you use as much, if not more. To solve this, label three keys (in your overlay area): INS TEXT, INS LABLE, and INS DIMENSIONS. When you digitize any of these keys, flip the overlay marked TXT, LAB, DIM and all modifiers are active for INS TXT, LAB and DIM.

The key you digitize to get this overlay has the command "GET KEY CR KEYFILE.XXX" where KEYFILE.XXX is the keyfile you made in the editor. To get back, each overlay has a flip key (at the same location on each overlay) which has a copy of the original 20 keys with a different keyfile name.

HOW IT WORKS

GET KEYS only replaces the keys it lists, it does not alter any other keys. By using SELECT NO ECHO, the flip action becomes invisible to the operator, in fact, by the time the overlay is in place, you can complete your command string (if you use 20 keys or less). After GET KEYS the command is started.

The same thing can be accomplished by utilizing the 36 hidden keys. Key numbers:

T#111 THRU T#128
AND
L#47 THRU L#64

However, these hidden keys can be utilized in a more effective way, many programs require input which can be accessed via the keys. This information may be needed more than once, or chained to other commands.

EXAMPLE:

Suppose you had a program which inserted a group of standard parts and you had a part which looked like this:

and you wished to insert this part at any angle and any distance between the date and the alpha variable, with the text horizontal. The date is inserted via the flip key which activates a program that obtains the date and places it in a keyfile (then GET KEYS) the date is now held for future use.

A continuation of the flip program runs a parex which asks for two digitizes and a variable (the date is first and is called by key number invisible to the operator), the parex then inserts two circles at the digitizes and a line between them utilizing the modifiers TX, &REF DIGA, IXRar ETC. Text is then inserted on the &DIGA &DIBB. This is a simple example, which serves only as an explanation. The same logic, however, can be utilized to insert section arrows, the section number and the section location at any distance apart, any angle, any scale, as well as, auto insertion of section name and location with 3 digitizes only—the program exists.

USE OF COLOR

Color may be used to highlight keyfiles, the simplest is to make a sharp full size plot and have it copied onto colored paper. If more colors are desired, use cross-hatching on different layers, color reproductions may be plotted on the gerber, one file per color. Four colors or less is optimum, any more will defeat the purpose of colors. The purpose of colors is to highlight the most used areas (yellow does this well), define grouped commands and to point out commands where caution is advised (red). Use photo reproduction to obtain multiple copies (make only one with the gerber).

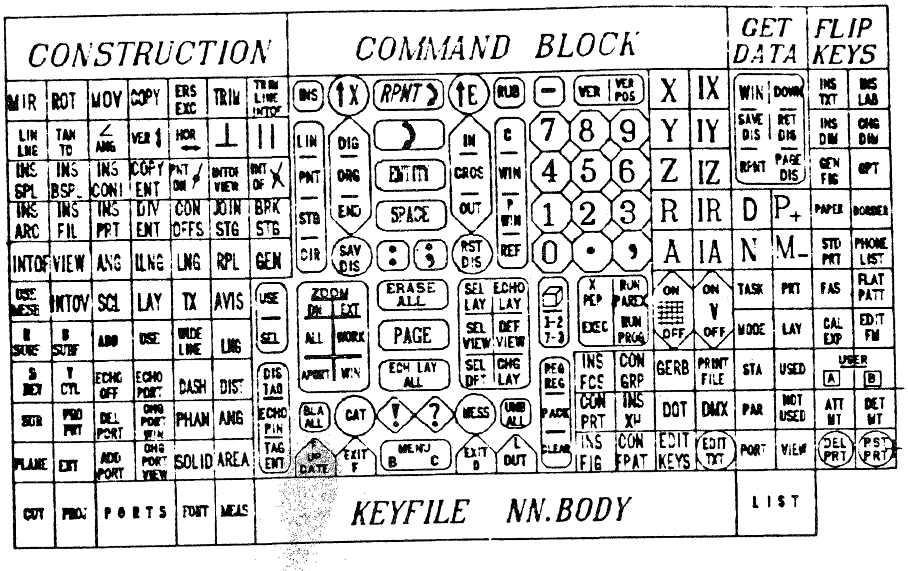
PROGRAMMING THE KEYS

Programming the keys is done at systems level. Begin by copying an existing keyfile (COPYBCD—etc.). Your new keyfile must be named keyfile.________keep the name simple. If you do not have any existing keyfiles, copy in the key number in the illustration in this paper. By observing the sample keyfile program you will notice the methods used to:

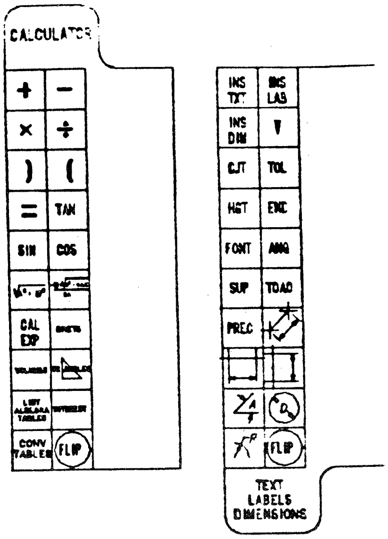
1. Add Comments
2. Define an Isometric View
3. Call Programs
4. Call Hidden Keys
5. Add Extra Keys
6. Call Personal Keys
7. Have One Key Call Another

These are just a few examples to look for, many others exist in the example shown. Note also the beginning statements and ending statements of the keyfile.

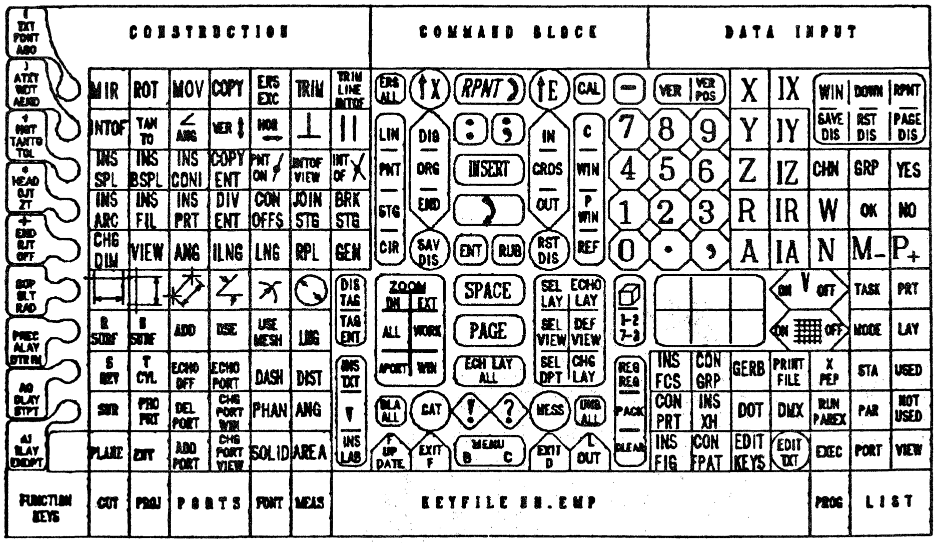
After the keyfile (the text file) is complete, enter into a dummy part, key in the command "GET KEYS," then key in your keyfile name. This will activate your new keyfile. To keep it active for the next time you need it, key in the command "SAV KEYS" and your keyfile name. Remember, if you GET KEYS on *one key only* it will replace that existing key with something new, and not alter any other key. "RST KEY" will remove old keys and replace them with the new keyfile.



SAMPLE GET KEY MENU



SAMPLE FLIP CARDS



SAMPLE FLIP KEY MENU

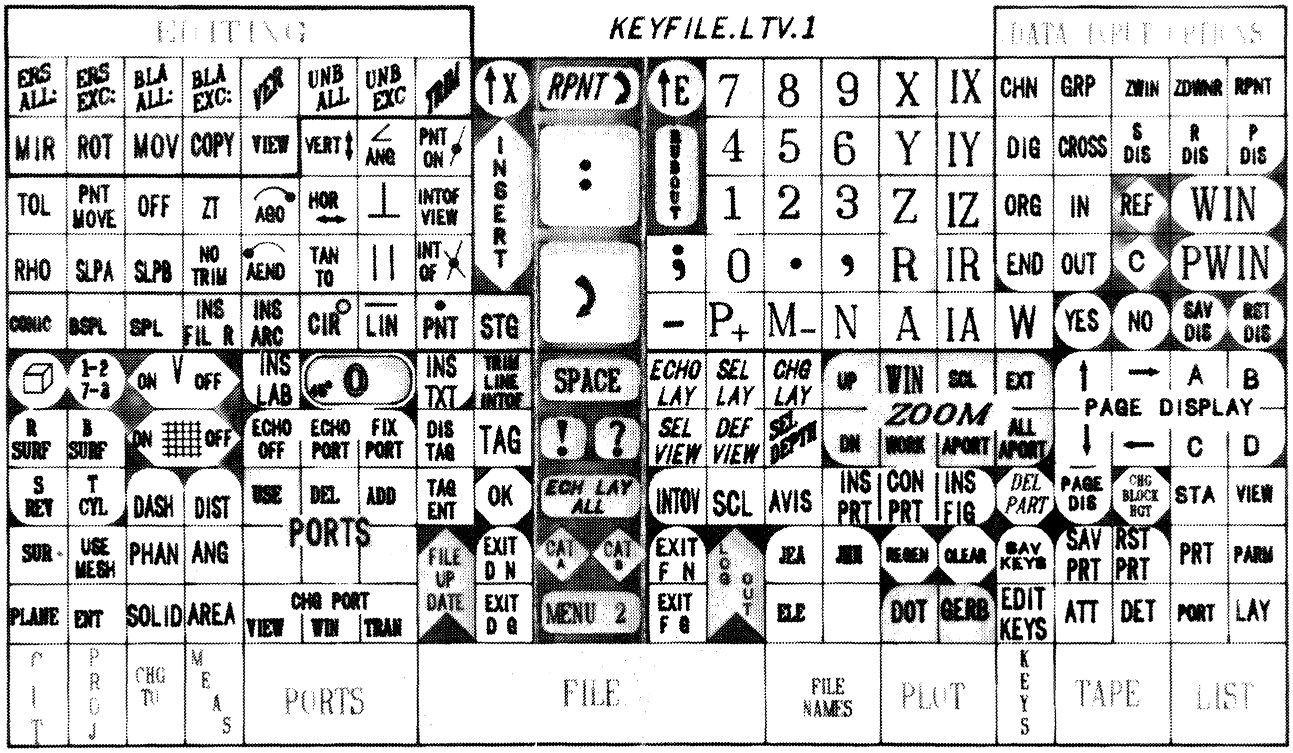
```
EDIT KEYS
C#02:<CR>INS LDIM HOR :
C#03:<CR>INS LDIM VERT :
C#04:<CR>INS LDIM PPNT :
C#05:<CR>INS ADIM :
C#06:<CR>INS RDIM :
C#07:<CR>INS DDIM :
C#08:<CR>DIS TAG<CR>:
C#09:<CR>ZOM DWN R:
C#10:<CR>ZOM EXT<CR>:
C#11: :
C#12: :
C#13:<CR>SEL LAY :
C#14:<CR>ECH LAY :
C#15:<CR>SEL VIEW 1<CR>DEF VIEW<COLN>XO YO ZO, Y-1 Y1 Z1<CR>:
C#19:<CR>SEL VCON<CR>**.....WARNING.....ALL NEW ENTITIES WILL APP$
EAR IN<CR>**                             WORKING VIEW ONLY<CR>:
C#20:<CR>SEL VALL<CR>**.....WARNING.....ALL NEW ENTITIES WILL APP$
EAR IN<CR>**                                ALL VIEWS<CR>:
C#21:<CR>RUN LSTTSK<CR>:
C#22:<CR>LST PRT<CR>:
C#24:<CR>INS RSURF<COLN>:
C#25:<CR>INS BSUR :
C#26:<CR>ADD PORT:
C#27:<CR>USE PORT:
C#28:<CR>USE MESH PATI<VAR>PATD<VAR><COLN>:
C#29:<CR>MEAS LNG STPT ENDPT<COLN>:
C#30:<CR>TAG ENT:
C#31:<CR>ZOM ALL APORT P5<CR>:
C#32:<CR>ZOM WORK<CR>:
C#33:<CR>PAG DIS <COLN>:
C#34:<CR>PAG DIS <COLN>:
C#35:<CR>SEL VIEW :
C#36:<CR>DEF VIEW :
C#37:<CR>SEL VIEW 1 2 7 3<CR>:
C#41:<CR>SEL GRD ON G<VAR><CR>ECH GRD ON<CR>DIS GRD<CR>:
C#42:<CR>SEL GRD OFF<CR>ECH GRD OFF<CR>RPNT<CR>:
C#43:<CR>LST MODE<CR>:
C#44:<CR>LST LAY<CR>:
C#46:<CR>INS SREV HIANG<VAR><COLN>:
C#47:<CR>INS TABCYL HIBND<VAR><COLN>:
C#48:<CR>ECHO PORT OFF<CR>RPNT<CR>:
C#49:<CR>ECHO PORT ALL<CR>:
C#50:<CR>CHG FONT DASH<COLN>:
C#51:<CR>MEAS DIST<COLN>:
C#52:<CR>INS TXT ':
C#53:<CR>ZOM APORT<CR>:
C#54:<CR>ZOM WIN<COLN>:
C#55:<CR>ECH LAY ALL<CR>:
C#56:<CR>ECH LAY ALL<CR>:
C#57:<CR>SEL DEP :
C#58:<CR>CHG LAY :
C#59:<CR>REG REG -ECH<CR>RPNT<CR>:
C#60:<CR>INS FCS :
C#61:<CR>CON GRP :
C#62:<CR>EXEC CAD.GERBER.PLOT<CR>:
C#63:<CR>PRINT FILE :
```

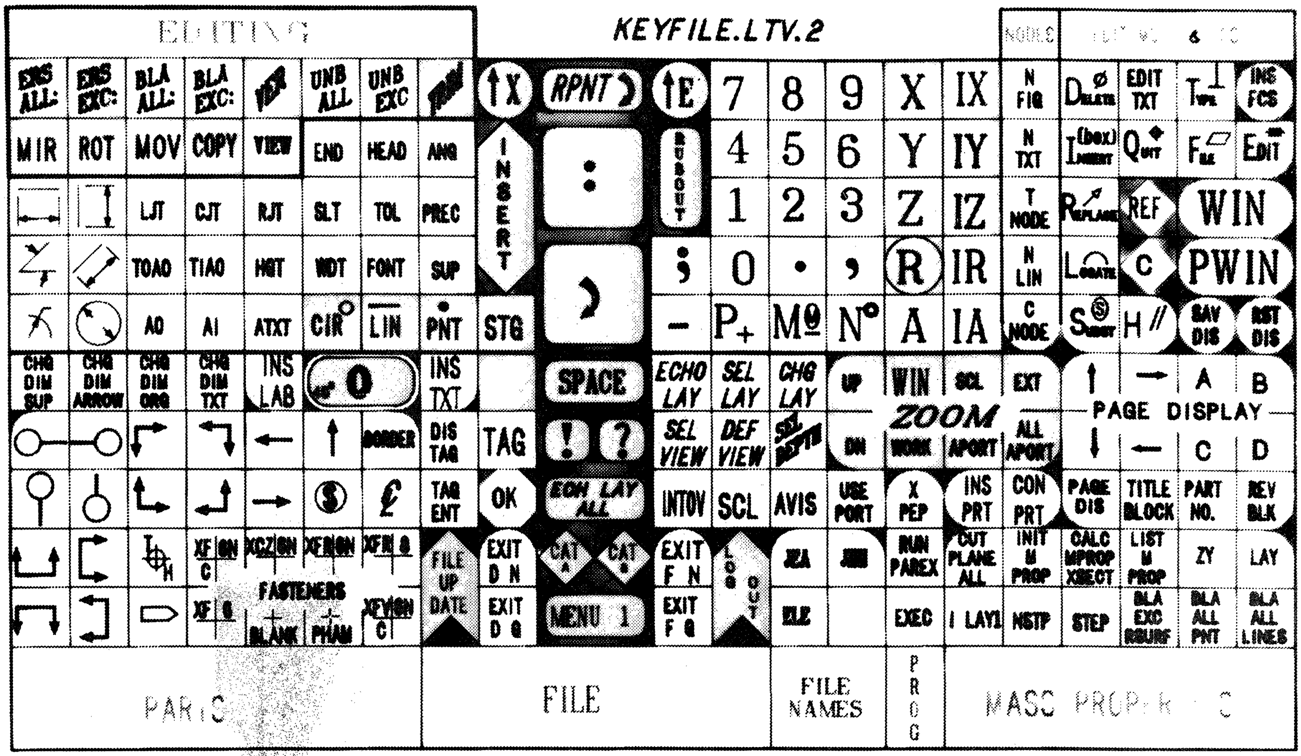
```
C#64:<CR>XPEP :
 L#01:<CR>LST STAT<CR>:
 L#02:<CR>LST LAY USED<CR>:
 L#04:<CR>CUT SURF LAY<VAR><COLN>:
 L#05:<CR>PROJ PRT :
 L#06:<CR>DEL PORT :
 L#07:<CR>CHG PORT WIN<COLN>:
 L#08:<CR>CHG FONT 2-DASH<COLN>:
 L#09:<CR>MEAS ANG<COLN>:
 L#10:':
 L#11:<CR>BLA ALL :
 L#13:!:
 L#14:<QM>:
 L#15:*             AFTER READING MESSAGES, TYPE CONTROL X <CR>$
 <VAR>PRINT FILE NN.EMP.MANAGEMENT LINE 1-20<CR><VAR>PRINT FILE NN$
 .EMP.MANAGEMENT LINE 21-40<CR><VAR>PRINT FILE NN.EMP.MANAGEMENT L$
 INE 41-60<CR><VAR>PRINT FILE NN.EMP.MANAGEMENT LINE 61-80<CR> $
 <VAR>PRINT FILE NN.EMP.MANAGEMENT LINE 81-100<CR>:
 L#16:<CR>UNB ALL<CR>:
 L#17:<CR>CLEAR<CR>DBCHK<CR>DBPACK ALL<CR>REG REG -ECH<CR>RPNT<CR>:
 L#18:<CR>CON PRT :
 L#19:<CR>INS XH <VAR><COLN>:
 L#20:<CR>PLOT DOT SCL<VAR>:
 L#21:<CR>PLOT DOT DMX1 SCL<VAR>:
 L#22:<CR>RUN PAREX :
 L#23:<CR>LIST PARAMETERS<CR>:
 L#24:<CR>LST LAY UNUSED<CR>:
 L#25:EXEC NN.STANDARD.PART.1<CR>:
 L#26:<CR>CUT PLAN LAY<VAR><COLN>:
 L#27:<CR>PROJ ENT D:
 L#28:<CR>ADD PORT VIEW :
 L#29:<CR>CHG PORT VIEW :
 L#30:<CR>CHG FONT 1-DASH<COLN>:
 L#31:<CR>MEAS AREA<COLN>:
 L#32:<CR>EDIT KEY<CR>T01<COLN><T#112><CR>T23<COLN><T#116><CR>T45$
 <COLN><T#119><CR>T67<COLN><T#124><CR>T89<COLN><T#128><CR>C01$
 <COLN><L#50><CR>C23<COLN><L#54><CR>C45<COLN><L#53><CR>L03<COLN>$
 <L#57><CR>Q*<^Z><CR>INS LAB ':
 L#33:<CR>FILE<CR><CR>OK<CR>OK<CR>:
 L#34:<CR>*                    TO ABORT, TYPE CONTROL E<CR>EXIT<CR>F $
 <CR><CR>OK<CR>OK<CR>:
 L#35:<CR>RST KEY<CR>B<CR>:
 L#36:<CR>RST KEY<CR>C<CR>:
 L#37:<CR>*  THIS EXITS WITHOUT FILING.  TO ABORT, TYPE CONTROL E$
 <CR> *  TO CONTINUE TYPE CONTROL X<VAR><CR>EXIT<CR>DG:
 L#38:<CR>*   THIS FILES AND LOGS YOU OUT.  CONTROL E WILL ABORT.$
 <CR>*     CONTROL X TO CONTINUE<CR><VAR>EXEC NN.PROP.EXIT<CR>:
 L#39:CLEAR<CR>:
 L#40:<CR>INS FIG <VAR><COLN>:
 L#41:<CR>CON FPAT <VAR><COLN>:
 L#42:<CR>EDIT KEY<CR>:
 L#43:<CR>EDIT TXT<COLN>:
 L#44:<CR>EXEC :
 L#45:<CR>LST PORT<CR>:
 L#46:<CR>LST VIEW<CR>:
 L#47:/:
 L#48:OFF :
```

```
L#49:SLT :
L#50:SUP :
L#51:RAD :
L#52:TIAI :
L#53:AO :
L#54:PREC :
L#55:TIAO :
L#56:ALAY :
L#57:AI :
L#58:TOAO :
L#59:STPT :
L#60:CLAY :
L#61:LAY :
L#62:DTRIM :
L#63:ENDPT :
L#64:ILAY :
T#02:<CR>MIR :
T#03:<CR>ROT :
T#04:<CR>MOV :
T#05:COPY :
T#06:<CR>ERS EXC<COLN>:
T#07:<CR>TRIM :
T#08:<CR>TRIM LIN INTOF<COLN>:
T#09:<CR>ERS ALL<COLN>:
T#10:<CONT>:
T#11:<CR>RPNT<CR>:
T#12:<CR>RPNT<CR>:
T#13:<^E>:
T#14:<CR>EDIT KEY<CR>T01<COLN><T#113><CR>T23<COLN><T#117><CR>T45$
<COLN><T#120><CR>T67<COLN><T#125><CR>T89<COLN><L#47><CR>0<CR>*$
<^Z><CR>CALC EXP :
T#15:-:
T#16:<CR>VER BOTH<COLN>:
T#17:VER POS<COLN><VAR>,<^E>:
T#18:X:
T#19:IX:
T#20:ZWIN:
T#21:ZDWNR:
T#22:RPNT :
T#24:INTOF :
T#25:<CR>INS LIN TANTO<COLN>:
T#26:<CR>INS LIN ANG :
T#27:<CR>INS LIN VERT :
T#28:<CR>INS LIN HOR :
T#29:<CR>INS LIN PRP<COLN>:
T#30:<CR>INS LIN PRL<VAR><COLN>:
T#31:LIN :
T#32:DIG :
T#33:<COLN>:
T#34:<SEMI>:
T#35:IN :
T#36:C :
T#37:7:
T#38:8:
T#39:9:
T#40:Y:
T#41:IY:
```

```
T#42:SDIS:
T#43:RDIS:
T#44:PDIS:
T#46:<CR>INS SPL :
T#47:<CR>INS BSPL :
T#48:<CR>INS CONIC :
T#49:<CR>COPY ENT :
T#50:<CR>INS PNT ON<COLN>:
T#51:<CR>INS PNT INTOF VIEW<COLN>:
T#52:<CR>INS PNT INTOF<COLN>:
T#53:PNT :
T#54:ORG :
T#55:<CR>INS :
T#56:<CR>INS :
T#57:CROSS :
T#58:WIN :
T#59:4:
T#60:5:
T#61:6:
T#62:Z:
T#63:IZ:
T#64:CHN :
T#65:GRP :
T#66:YES:
T#68:<CR>INS ARC :
T#69:<CR>INS FIL :
T#70:<CR>INS PRT :
T#71:<CR>DIV ENT<COLN>:
T#72:<CR>CON OFFS :
T#73:<CR>JOIN STG<COLN>:
T#74:<CR>BRK STG<COLN>:
T#75:STG :
T#76:END :
T#77:<CR>:
T#78:<CR>:
T#79:OUT :
T#80:PWIN :
T#81:1:
T#82:2:
T#83:3:
T#84:R:
T#85:IR:
T#86:D:
T#87:OK:
T#88:NO:
T#90:<CR>CHG DIM :
T#91:VIEW :
T#92:ANG :
T#93:ILNG :
T#94:LNG :
T#95:RPL :
T#96:GEN :
T#97:CIR :
T#98:<CR>SAV DIS :
T#99:ENT :
T#100:<RO>,:
T#101:<CR>RST DIS :
```

```
T#102:REF :
T#103:O:
T#104:.:
T#105:,:
T#106:A:
T#107:IA:
T#108:N:
T#109:M:
T#110:P:
T#111:FONT :
T#112:TXT :
T#113:(:
T#114:AGO :
T#115:WDT :
T#116:ATXT ':
T#117:):
T#118:AEND :
T#119:HGT :
T#120:+:
T#121:TANTO :
T#122:TOL :
T#123:CJT :
T#124:HEAD :
T#125:*:
T#126:ZT :
T#127:RJT :
T#128:END :
0**  SET KEYS COMPLETE  ***
```


EDITING
KEYFILE.LTV.1
DATA INPUT OPTIONS
ERS ALL: ERS EXC: BLA ALL: BLA EXC: VER UNB ALL UNB EXC TRIM ↑X RPNT ↑E 7 8 9 X IX CHN GRP ZWIN ZDWNR RPNT
MIR ROT MOV COPY VIEW VERT ∠ ANG PNT ON INSERT : RUBOUT 4 5 6 Y IY DIG CROSS S DIS R DIS P DIS
TOL PNT MOVE OFF ZT AGO HOR ⊥ INTOF VIEW 1 2 3 Z IZ ORG IN REF WIN
RHO SLPA SLPB NO TRIM AEND TAN TO ‖ INT OF ; 0 • , R IR END OUT C PWIN
CONIC BSPL SPL INS FIL R INS ARC CIR LIN PNT STG) – P+ M– N A IA W YES NO SAV DIS RST DIS
1-2 7-8 ON V OFF INS LAB 0 INS TXT TRIM LINE INTOF SPACE ECHO LAY SEL LAY CHG LAY UP WIN SCL EXT ↑ → A B
R SURF B SURF ON OFF ECHO OFF ECHO PORT FIX PORT DIS TAG TAG ! ? SEL VIEW DEF VIEW SEL DEPTH DN WORK APORT ALL APORT ZOOM PAGE DISPLAY ↓ ← C D
S REV T CYL DASH DIST USE DEL ADD TAG ENT OK ECH LAY ALL INTOV SCL AVIS INS PRT CON PRT INS FIG DEL PART PAGE DIS CHG BLOCK HGT STA VIEW
PORTS
SUR USE MESH PHAN ANG FILE UP DATE EXIT D N CAT A CAT B EXIT F N LOG OUT JEA JEN REGEN CLEAR SAV KEYS SAV PRT RST PRT PRT PARM
PLANE ENT SOLID AREA CHG PORT VIEW WIN TRAN EXIT D G MENU 2 EXIT F G ELE DOT GERB EDIT KEYS ATT DET PORT LAY
CIT PROJ CHG TO MEAS PORTS FILE FILE NAMES PLOT KEYS TAPE LIST


EDITING
KEYFILE.LTV.2
NODE
6
ERS ALL:
ERS EXC:
BLA ALL:
BLA EXC:
UNB ALL
UNB EXC
MIR
ROT
MOV
COPY
VIEW
END
HEAD
ANG
LJT
CJT
RJT
SLT
TOL
PREC
TOAO
TIAO
HGT
WDT
FONT
SUP
AO
AI
ATXT
CIR
LIN
PNT
STG
CHG DIM SUP
CHG DIM ARROW
CHG DIM ORG
CHG DIM TXT
INS LAB
INS TXT
BORDER
DIS TAG
TAG
TAG ENT
OK
XF ON C
XCZ ON
XF ON
XF 0
FASTENERS
BLANK
PHAM
XF ON C
FILE UP DATE
EXIT D N
EXIT D Q
↑X
RPNT
↑E
INSERT
RUBOUT
:
;
SPACE
!
?
ECH LAY ALL
CAT A
CAT B
MENU 1
ECHO LAY
SEL LAY
CHG LAY
SEL VIEW
DEF VIEW
SEL DEPTH
INTOV
SCL
AVIS
USE PORT
EXIT F N
EXIT F Q
LOG OUT
JEA
ELE
7
8
9
X
IX
4
5
6
Y
IY
1
2
3
Z
IZ
0
•
,
R
IR
P+
Mº
Nº
A
IA
UP
WIN
SCL
ZOOM
DN
WORK
APORT
ALL APORT
X PEP
INS PRT
CON PRT
RUN PAREX
CUT PLANE ALL
EXEC
I LAYI
N FIG
N TXT
T NODE
N LIN
C NODE
EXT
INIT M PROP
NSTP
DELETE
EDIT TXT
TYPE
INS FCS
INSERT
QUIT
FILE
EDIT
REPLACE
REF
WIN
LOCATE
C
PWIN
SHIFT
H
SAV DIS
RST DIS
PAGE DISPLAY
A
B
C
D
PAGE DIS
TITLE BLOCK
PART NO.
REV BLK
CALC MPROP XSECT
LIST M PROP
ZY
LAY
STEP
BLA EXC RSURF
BLA ALL PNT
BLA ALL LINES
PARTS
FILE
FILE NAMES
PROG
MASS PROPERTIES

CAD RIGHT NOW!

William Herron
Singer Sewing Products Group
321 First Street
Elizabeth, NJ 07207
201-527-6000

Back in early 1978 I was informed that I had been appointed, along with about five other guys, to a committee to investigate C. A. D. for use in the design of sewing machines.

Now I've always been a mechanical type guy and anything to do with computers just went in one ear and out the other — or in both eyes and out the back of my head — or something like that. It wasn't that I didn't want to know what made them tick but it just didn't stick.

So here I was, running around the country looking at the CV's and the Calmas, the Auto Trols and the Gerbers and hearing about the CDC's and the DEC's, the Xynetics and Calcomps — the CADDS 3's and the PEP's — and on and on. It's enough to drive a good mechanical man to drink. The main reason I was selected for this mission is that I knew our existing design process, and thereby would be of help in selecting the system most compatible with our needs.

Well, this selection process was going along just fine — I was getting more confused with every system I saw. Every new one was better than the ones before — and I seemed to be getting involved deeper and deeper in the investigation.

The other main member of the committee was one of these natural born computer guys — Masters in Com-Sci and all that good stuff. So, while I was looking at the convenience of the system related to how our design process operated and what I though it ought to do — he could look at the "ARCHITECTURE" and decide which had the most potential for future expansion. (I always thought architecture related to designing buildings.)

Well, along about September/October, 1978 we thought we had enough general background to write up what we called our "Wish List". It consisted of everything we could think of that we might want a CAD system to do, short of just being able to say "Design me a new sewing machine", and have one fall out the tube. Finally, in November, 1978, we got this "Wish List" in shape and we sent it out to about ten CAD companies with a request that they comment on their ability to grant each of our wishes, and give an approximate cost for doing it. Well, it seems that we weren't very imaginative in our wishing because several companies replied that they could grant our every desire and that their standard equipment would do most of it.

Well, then the fun really started. We had salesmen from all the companies beating our doors down and, naturally, when each salesman left, we were convinced he had the best system. The only one different was C.V. — we had to chase them till they finally caught us.

In December, at the direction of our Chairman in the Corporate Engineering Office, we selected the two most eligible systems and began to prepare an RFQ letter.

Now this R.F.Q. letter was to "completely specify" the system and also set up a timetable for implementation. (I was to find out shortly what "completely specify" could mean.) We were in such a hurry to get the system installed that I was to get the letters out within two weeks, both companies were to let us know immediately if they intended to bid, and then get the firm quotes back to us within 30 days. Then we were allowed 15 days to evaluate the bids and decide which system we wanted, and another 15 days to get the AR approved and get a P.O. out. This all got us out to March 1st, 1979, by which time someone would have a P.O. in their hot little hands and we expected a system to be *delivered* by July 1st, 1979. So much for good intentions.

The schedule was issued and we were on our way. Up to the 1st draft of the R.F.Q. letter we were still on schedule. We made up lists of equipment for both contenders and were very careful to insure that we were really comparing apple to apples in specifying both systems (and I didn't mean Apple Computers). In addition to these equipment lists and our Wish List, I wrote up a 1-1/2 page letter which I felt covered everything we were looking for or needed to pin down in a quote. It covered guarantees, training, installation and all the other things I could think of. Then I sent a draft of it to our Chairman for what I expected to be a quick approval, and I'd get it out on time. Then the fun started. He said it looks great but can't you "Flesh it out" a little. It seems as though a 1/2 million dollar piece of equipment should have more of a specification. He was obviously right, but there went our schedule.

We spent the next 3 months re-studying and discussing the systems and making sure we'd covered everything that could be covered in a specification. My 1-1/2 pages had grown to 8 and we'd even considered, but hadn't included, trying to specify exactly how the system should work. We finally agreed "if you buy a Ford you get a Ford transmission." Ford isn't going to change the "What you may call it" gear from this side to that side just because we think it would be better that way. We're buying an off-the-shelf system. We decide which one we want and take it the way it comes.

At that point we agreed we'd better get the R.F.Q. out because we'd be better off with a partially specified something than a fully specified nothing.

We then ran it by the Corporate Director of Engineering to get his blessing before it was mailed. After all, 1/2 million dollars is slightly above my approval level. Since he had been 100% behind the project to begin with, we expected a very quick O.K. (Remember this was still just an R.F.Q.) But it was delayed!

What I didn't realize was that all this time we were in the process of hiring a new Vice President of Engineering and they wanted to give him a chance to have his say about anything as important as this.

So, finally, at the end of June, 1979, we mailed the R.F.Q. We were still in a hurry though, and demanded immediate replies and complete quotes within 30 days. Early delivery was still stressed as being of *prime* importance and would weigh heavily in our decision.

When the quotes came in — on time — (within their 30 days) I was on vacation, but my computer friend handled them and got things started on the evaluation, and on an AR. By January 31, 1980, we had the Appropriation Request signed and a PO in C.V.'s hands. From that point on we had some good ones and some bad ones.

A good one was that our system was delivered in June rather than September, as originally scheduled. The bad one was that we got CADDS 3 instead of CADDS 4 because CADDS 4 wasn't available yet.

A good one was that we had "State of the Art" equipment which will be enhanced and upgraded for a long time to come. The bad one was that we had to live with "State of the Art" bugs. We had every problem that anyone using C.V. equipment had ever heard of and many that they hope no one else ever does hear of!

A good one was that the pressure of work forced us to get "up" on the system and start doing productive work very quickly. The bad one was that we couldn't take the time to learn the extra tricks that will speed things up later on.

In a committee session at the East Coast Users Meeting in the Spring, I mentioned a few of the things I thought we had done right in getting our system off the ground. At the expense of being repetitious to a few of you here, I'll mention them again, in the hope that it may help someone else get started right. If someone has a better way, please come and tell *me* about it later.

The first thing we did right was, in setting up our organization for the system. We were fortunate in having this very strong computer guy already in the organization so we split responsibility between system and operation. It was a good move. It allowed us to pick a guy to operate the design end who was very strong in our product and method of development. He didn't have to know anything about computers. Fortunately, he's learned a heck of a lot about the system since then and is doing a real good job.

The next thing we did right was selection of the operators. I've heard that some companies have had a lot of problems in this area. Some designers think the system is going to take their jobs — some resent the multi-shift operation — some who don't get on it are jealous, etc. We've been lucky!

Close to a year before we got a system we had all our designers together (about 60 people) and told them what we were planning and where we stood. At that stage we asked for volunteers and said we would try, as far as possible, to only use people who really wanted to go on.

After we'd decided on a C.V. system, had the A.R. approved and got the order in, we got together again with all the people who had volunteered initially. This time we made it sound as bad as possible, told them they'd have to do some kind of shift work. We hadn't made any plans yet as to just how we would work it or when we'd start a second shift, but wanted to be sure that everyone knew what they were in for and not have any surprises.

Well, at that point, with the prospect of shift work, a few of the original volunteers dropped out, but we still had twice as many people on the list as we could handle in the 1st training session. Fortunately, most of the top guys, — the ones I wanted to see on the system, — were still in the running. One of the principles of selection was that we'd use guys who were already top product designers on the board. We felt it would be much easier to train a good designer to use the system than to train a computer type to be a designer. In fact, many computer people might never make good designers. There again we were right. It's worked out very well.

The third thing we did right was to start off in 3D. Even C.V. didn't recommend this, but I've seen too many companies buy a system, start with 2D and never get around to changing. Anyone who intends to do real mechanical design should start with 3D and take advantage of all the "Freebies" inherent in that mode. It's harder to get started, but the longer you put it off the harder it is to change over.

The last thing I want to mention is a little trick we used to bridge the gap between training and work. After we'd had the system a few weeks, the guys were still doing canned practice programs and inserting existing parts into the system. It was time to break out of the rut and start getting some work done. They seemed to be afraid to start any real layout work or new design. At that point, we made sure that each one had a real design project to work on and then said they should spend 4 hours on the board and 4 hours on the system doing the same job. That seemed to be the key that unlocked everything. In only a few days they found that they could do most things faster and better on the system, and from then on we've had more trouble getting them off than keeping them on.

F006 ERROR RECOVERY

Patrick D. Rice
Boeing Computer Services Company
Renton, WA 98055
M/S 6C-07
(206) 251-2595

1. ABSTRACT

This paper deals with disc I/O type F006 errors. It traces their cause as a result of corrupt file manager data. It present a basic outline of CV's file manager and catalogue structure, and presents a procedure to recover data from corrupted discs by using the PACHDISC command to restore corrupt data and addresses to their proper value.

The outlined procedure has been very successful in disc data recovery of F006 error occurrences.

INTRODUCTION

Have you ever seen this message while doing an FMCLEAR or writing a file to tape?

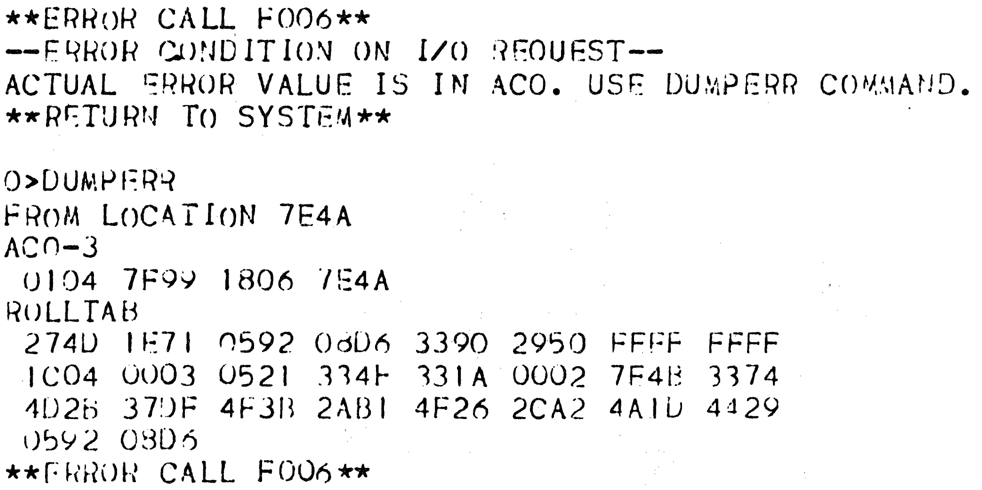
```
**ERROR CALL F006**
--ERROR CONDITION ON I/O REQUEST--
ACTUAL ERROR VALUE IS IN AC0. USE DUMPERR COMMAND.
**RETURN TO SYSTEM**
```

What is your usual course of action when this message occurs? Do you try to save whatever files you can and then rebuild the pack? This paper outlines a method of recovery that will save the entire disc in many cases and most of the disc nearly all the time.

The procedure requires an understanding of the internal workings of the CGP-100/10, specifically the file manager and the PACHDISC utility, and the ability to interpret what the system is telling you when it says F006.

WHAT IS A F006?

A F006 is a generalized system I/O error. It occurs when the system encounters a problem in a read/write operation. The problem can be as simple as the tape drive being powered down or as complex as a complete corruption of the file management area. More specific information about what the problem is can be obtained using the DUMPERR command. The resulting printout, figure 1, gives a coded message about which device is at fault and a somewhat vague description of the problem. The DUMPERR command should be issued immediately after the occurrence of a F006 or the contents of the accumulators will be overwritten. This code is found in the contents of accumulator 0, (AC0), the underlined word in figure 1.



```
**ERROR CALL F006**
--ERROR CONDITION ON I/O REQUEST--
ACTUAL ERROR VALUE IS IN ACO. USE DUMPERR COMMAND.
**RETURN TO SYSTEM**

O>DUMPERR
FROM LOCATION 7E4A
ACO-3
 0104 7F99 1806 7E4A
ROLLTAB
 274D 1E71 0592 08D6 3390 2950 FFFF FFFF
 1C04 0003 0521 334F 331A 0002 7F4B 3374
 4D2B 37DF 4F3B 2AB1 4F26 2CA2 4A1D 4429
 0592 08D6
**ERROR CALL F006**
```

Figure 1: DUMPERR Printout

The left two characters tell which device can't be communicated with and the right two characters give a generalized problem code. Interpretations of these codes can be found in section 4 of the CGOS100/10 O/S manual. They are summarized in figure 2.

Left byte	Unit	Right byte	Problem
00	TAPE	01	End of tape violation
		03	Writing exceeds MTRECMAX
		04	Hardware error in tape I/O
		05	Non-existent command
		06	I/O attempt, tape not ready
01	DISC	00	Hardware error in disc I/O
		01	Bad location on pack
		02	Bad unit argument in I/O call
		03	Bad argument in I/O call
		04	Invalid drive number argument
		05	Disc error not recoverable in retry
		06	Disc error on recalibration attempt

Figure 2: Decoding of ACØ

Figure 2 is just a partial list and it is applicable only to CGOS100, i.e., CADDS3 rev 10 and before. CGOS200, CADDS3 rev. 11 and CADDS 4, have a similar structure but a different coding.

Some F006's present no real problem, being as simple to resolve as putting the line printer on-line. The ones covered here are more difficult to resolve. Their AC0 code is 0104 and is usually preceded by the message "VOLUME NOT ON ASSIGNED DRIVE."

Most often this message appears when a bad disc address has been passed to the disc read utilities from the file manager. To understand how this can happen, we need to look into the file manager catalogue structure.

FILE MANAGER AND CATALOGUE STRUCTURE

As you know, CGOS supports a hierarchal filing system in which each file — whether it be a graphics part, textile, a keyfile, object code, etc. — is stored under a subcatalogue which is, in turn, sorted under a higher subcatalogue and so on up to the highest catalogue level, which is SYSCATLG. All files and subcatalogues can be found somewhere under this main directory.

For the system to move around in the file management area efficiently, there must be pointers, both forward and backward, letting the system know where it has been, where it is, and where it is going. These pointers can be represented graphically as in figure 3.

These pointers, as well as other pertinent data about each subcatalogue and file, are generated and stored when the file is initially created, and are updated whenever a file is accessed.

Each subcatalogue is allocated an amount of disc space equal to the cluster size established in the FMINIT command issued when the pack was built. The cluster size is an integral power of 2 and determines the smallest number of sectors that can be allocated for any data storage. This paper will assume a cluster size of 4 sectors. This means that each level of a part name delimited by periods would take 4 sectors of disc space. For a part name with the full 35 levels allowed by the file manager, a minimum of 140 sectors would be allocated just to store the name.

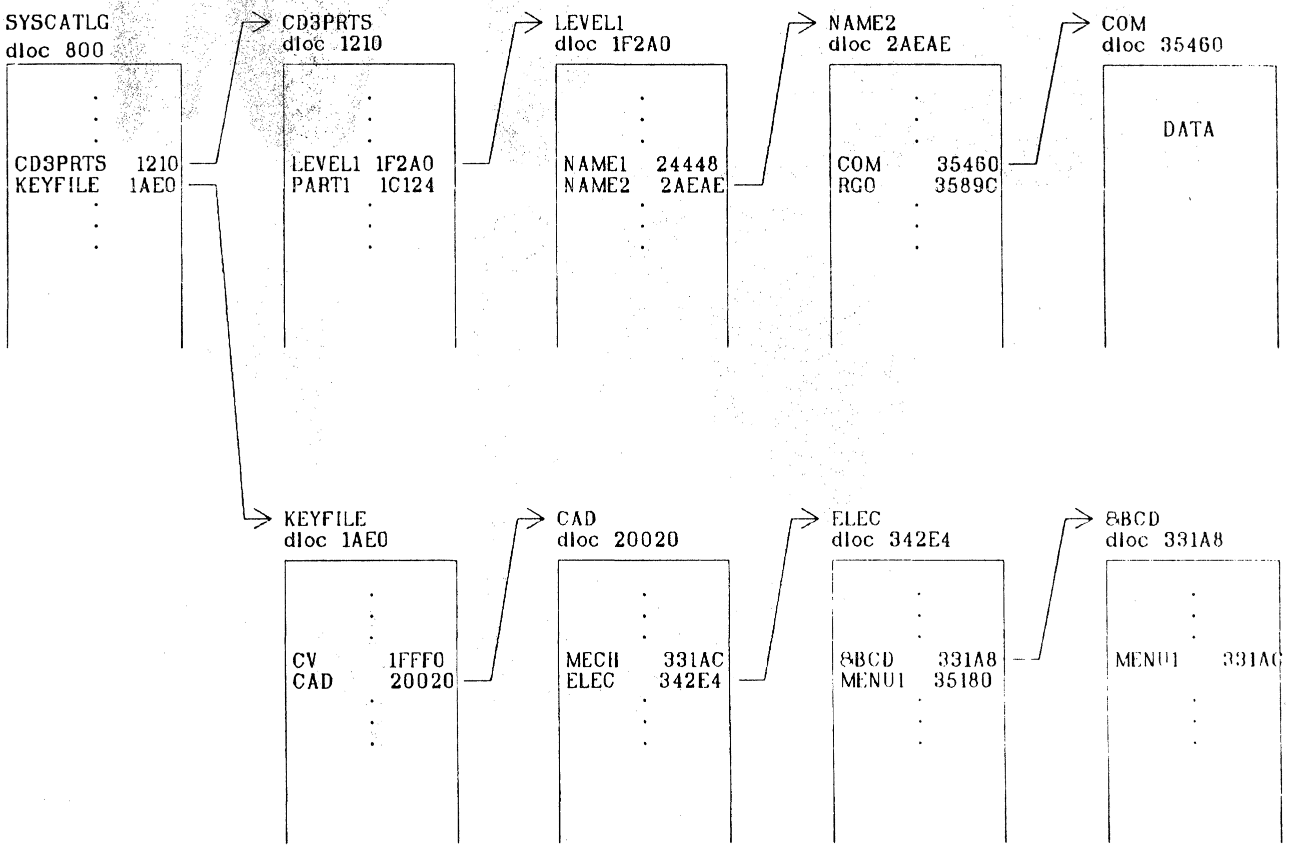


Figure 3

BASES, NUMBERS AND TEXT

Through this paper, decimal, hexadecimal and binary numbers are used. When it isn't clear which base a number is in, it will be specifically stated. As a brush up, table 1 gives hex to decimal conversions.

Table 1:
Hex to Decimal Conversion

HEX	DECIMAL
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Each sector contains 256 words. That's 100 hex. Each word has 2 bytes, each byte has 8 bits or 2 nibbles. A nibble has 4 bits.

Each nibble of 4 bits can be represented by a single hexidecimal number. So a 16 bit word — 4 nibbles — can be represented by a 4 digit hex number.

This is a good place to mention ASCII coded alpha-numeric characters. CV used an 8 bit ASCII code. Each alpha-numeric and special character on a keyboard can be represented by a unique 8 bit combination. A 16 bit word can hold 2 ASCII codes. CV's ASCII appears in figure 4.

80	NUL	A0	SP	C0	@	E0	`
81	SOH	A1	!	C1	A	E1	a
82	STX	A2	"	C2	B	E2	b
83	ETX	A3	#	C3	C	E3	c
84	EOT	A4	$	C4	D	E4	d
85	ENQ	A5	%	C5	E	E5	e
86	ACK	A6	&	C6	F	E6	f
87	BEEP	A7	'	C7	G	E7	g
88	BS	A8	(	C8	H	E8	h
89	HT	A9	)	C9	I	E9	i
8A	LF	AA	*	CA	J	EA	j
8B	VT	AB	+	CB	K	EB	k
8C	FF	AC	,	CC	L	EC	l
8D	CR	AD	-	CD	M	ED	m
8E	SO	AE	.	CE	N	EE	n
8F	SI	AF	/	CF	O	EF	o
90	DLE	B0	0	D0	P	F0	p
91	DC1	B1	1	D1	Q	F1	g

Figure 4: CV 8 BIT ASCII

(continued)

92	DC2	B2	2	D2	R	F2	r
93	DC3	B3	3	D3	S	F3	s
94	DC4	B4	4	D4	T	F4	t
95	NAK	B5	5	D5	U	F5	u
96	SYN	B6	6	D6	V	F6	v
97	ETB	B7	7	D7	W	F7	w
98	CAN	B8	8	D8	X	F8	x
99	EM	B9	9	D9	Y	F9	y
9A	SUB	BA	:	DA	Z	FA	z
9B	ECS	BB	;	DB	[	FB	{
9C	FS	BC	•	DC	\	FC	¦
9D	GS	BD	=	DD	]	FD	}
9E	RS	BE		DE	^	FE	~
9F	US	BF	?	DF	_	FF	DEL

Figure 4: (continued)

To be consistent with CV convention all database words will be presented in a format of 4 hex digits.

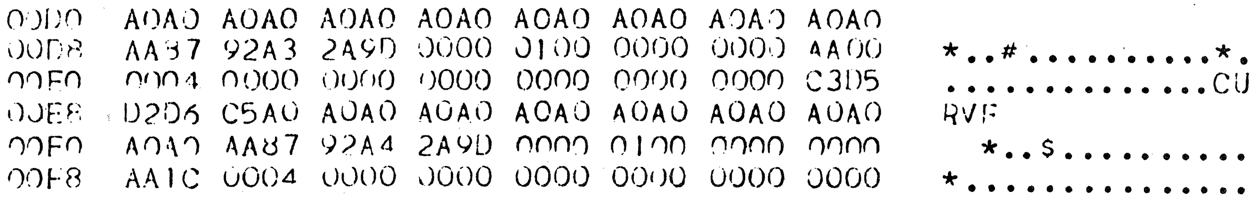
FILE MANAGER DATA STRUCTURE

With notation established we can now look at the actual structure of a subcatalogue. As mentioned above, each subcatalogue is allocated one cluster, 4 sectors in this example, of disc space. Each sector within the cluster has the same format. Figure 5 shows how a dump of a subcatalogue sector would appear on the comdev.

```
0000   0000 A721 FFFF 0000 0801 0002 C6CF CFD4    ..'.........FONT
0008   C4C5 C6D3 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    DEFS
0010   2933 57FD AA9D 1000 8400 0001 0001 6080    ...]*...........0
0018   0014 0000 0000 0000 0000 0000 0000 C2C9    ..............BI
0020   D3D9 CEC3 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    SYNC
0028   A0A0 AA87 9278 2A9D 0000 0100 0000 0000      *............
0030   A733 0004 0000 0000 0000 0000 0000 0000    '...............
0038   A6C2 C3C4 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    &BCD
0040   A0A0 A0A0 AA87 9278 2A9D 0000 0100 0000        *..........
0048   0000 A74C 0004 0000 0000 0000 0000 0000    ..'.............
0050   0000 C1D0 D4D7 CFD2 C4D3 A0A0 A0A0 A0A0    ..APTWORDS
0058   A0A0 A0A0 A0A0 AA87 9284 2A9D 0000 0100              *.........
0060   0000 0000 A81C 0004 0000 0000 0000 0000    ....(..........
0068   0000 0000 C2C3 C4C4 C5C6 A0A0 A0A0 A0A0    ....BCDDEF
0070   A0A0 A0A0 A0A0 A0A0 AA87 9285 2A9D 0000               *.......
0078   0100 0000 0000 A82C 0004 0000 0000 0000    ......(.........
0080   0000 0000 0000 C2CF CDA0 A0A0 A0A0 A0A0    ......ROM
0088   A0A0 A0A0 A0A0 A0A0 A0A0 AA87 9299 2A9D                  *.....
0090   0000 0100 0000 0000 A95C 0004 0000 0000    .........).......
0098   0000 0000 0000 0000 C3C1 A0A0 A0A0 A0A0    ........CA
00A0   A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 AA87 929C                     *...
00A8   2A9D 0000 0100 0000 0000 A980 0004 0000    ...........)0....
00B0   0000 0000 0000 0000 0000 C3C8 C1D2 C1C3    ...........CHARAC
00B8   D4C5 D2AD C3CF C4C5 A0A0 A0A0 A0A0 AA87    TER-CODE          *.
00C0   929F 2A9D 0000 0100 0000 0000 A9D4 0004    .............)T..
00C8   0000 0000 0000 0000 0000 0000 C3CF CDCD    ..............COMM
```

Figure 5: Subcatalogue Dump

(continued)



```
00D0   A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0
00D8   AA37 92A3 2A9D 0000 0100 0000 0000 AA00    *..#..........*.
00E0   0004 0000 0000 0000 0000 0000 0000 C3D5    ..............CU
00E8   D2D6 C5A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    RVF
00F0   A0A0 AA87 92A4 2A9D 0000 0100 0000 0000      *..$..........
00F8   AA1C 0004 0000 0000 0000 0000 0000 0000    *...............
```

Figure 5: (continued)

The numbers in the first column are the numbers, in hex, of the first word of each line. The next 8 columns are the hex representation, by word, of the binary data in the sector. To the right of this the system prints any ASCII coded byte that might appear in the data. For any byte that isn't an ASCII code, the system prints a period.

The first 6 words in each sector form the sector header. The first two, words 0 and 1 give the address of the next sector. All file manager addresses are sector addresses. Word 2 is the empty sector flag. This word contains FFFF if there is data in the sector, 0000 if there is none. Words 3 and 4 contain the sector address of the next higher level subcatalogue. Word 5 is an entry pointer into that sector. The function of this pointer will be clarified shortly.

Following the header are the entries for each subcatalogue to be found under this parent subcatalogue. Each of these entries contain 19 hex, 25 decimal, words. Each file management sector can hold up to 10 subcatalogue entries. The entry pointer mentioned as word 5 in the header is this position number, pointing back to a subcatalogue's position in its parent catalogue.

The format of each subcatalogue entry is shown in figure 6.

Word	Contents
1-10	20 character catname in ASCII
11	creation data
12	creation time (1 second intervals)
13	access data
14	protection group
15	status bit map
16	use count
17-18	dloc of entries
19	sector count of catalogue
20-21	dloc of RIB; else 0000
22-23	user defined attributes
24	checksum
25	file type

Figure 6: Format of Subcatalogue entry

Words 1-10 hold the subcatalogue name in ASCII. These 10 words can hold 20 ASCII characters. This is why there is a 20 character limit to subcatalogue names. Word 11 is the coded creation data, word 12 is the creation time and word 13 the access data. Word 14 is the protection group. Word 15 is a status flag bit map which will be covered below. Word 16 gives the number of users currently accessing the catalogue. Words 17 and 18 contain the address of the sector containing the subcatalogue entries for this subcatalogue and word 19 is the sector count for that subcatalogue. Words 20 and 21 contain the sector address of a RIB or retrieval information block which, simplified, holds addresses of clusters used to continue subcatalogue entries if the initial cluster becomes full.

Words 22 and 23 contain user defined attributes which can only be set or modified under system programmer control. Word 24 is the check sum for type 2, 3, 4, 5 and 8 files. Word 25 is the file type of the entry. If the entry is a subcatalogue, this word is 0000. CV file types are shown in figure 7. The status word, word 15, has a format as shown in figure 8.

Type	Description
0	Not Defined (May be a catalogue)
1	Catalogue
2	Object Code
3	Text
4	Configuration
5	Upper-Lower Case Text
6	Pep Object Code
7	Core Image (Overlay)
8	Command Table
9	Loader Data File
B	Access/Accounting Data
F	PCU Image File

Figure 7: CV File Types

```
Bit number      15                 8  7                  0
Binary          |1  0  0  0 | 0  1  0  1 | 0  0  0  0 | 0  0  0  0|
Hexidecimal          8           5            0            0
```

Bit	Meaning
15	file in use
14	open for modify in place
13	delete when closed iff word F=0
12	open for create
11	temp catalogue entry
10	open for read
9	open for short read
8	file is a catalogue

Figure 8: Status Word Bit Map

Only the 8 bits of the left byte are used. The right byte should always be zero. Some typical values for this word would be 0100, 8400, 8500. Looking at 8500 as an example the bits that set are 15, 10 and 8 meaning the file is in use, it is open for read and it is a catalogue. The FMCLEAR command operates on this word setting any bits to zero that indicate the file is open or in use.

SAMPLE TRACING THROUGH THE CATALOGUE STRUCTURE

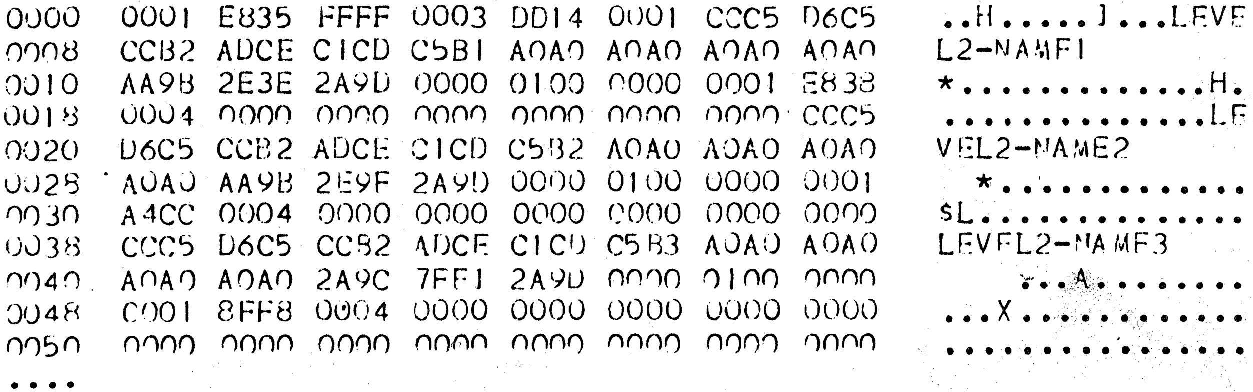
To see how these addresses and pointers interrelate we will look at all the catalogue levels of the file CD3PRTS.LEVEL1-NAME1.LEVEL2-NAME2.LEVEL3-NAME2. We can start anywhere within the catalogue structure and find our way backward to SYSCATG or forward to the COM and RG0 files.

We'll start with the LEVEL1-NAME1 subcatalogue. The command LISTCATCD3PRTS/LEVEL1-NAME1/FULL will give the following information about the subcatalogue:

```
CD3PRTS.LEVEL1-NAME1.
4-27-81 15:17:18 0001E834 0 0004 1 0000 0100
```

The entries are: the creation date and creation time, the sector address of the subcatalogue, a contiguous flag, the sector count of the first contiguous chunk of the subcatalogue, the file type, the protection group and the status word.

Using the PACHDISC command we can printout the contents of the sector located at 1E834. This is shown in figure 9.

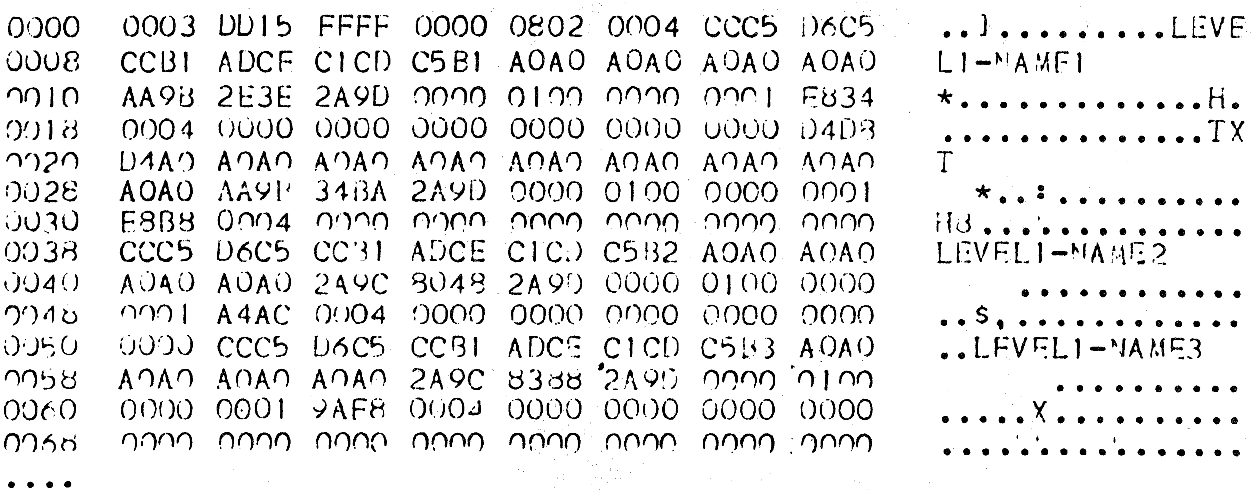


```
0000   0001 E835 FFFF 0003 DD14 0001 CCC5 D6C5    ..H.....]...LEVE
0008   CCB2 ADCE C1CD C5B1 A0A0 A0A0 A0A0 A0A0    L2-NAME1
0010   AA9B 2E3E 2A9D 0000 0100 0000 0001 E838    *.............H.
0018   0004 0000 0000 0000 0000 0000 0000 CCC5    ..............LE
0020   D6C5 CCB2 ADCE C1CD C5B2 A0A0 A0A0 A0A0    VEL2-NAME2
0028   A0A0 AA9B 2E9F 2A9D 0000 0100 0000 0001      *.............
0030   A4CC 0004 0000 0000 0000 0000 0000 0000    $L..............
0038   CCC5 D6C5 CCB2 ADCE C1CD C5B3 A0A0 A0A0    LEVEL2-NAME3
0040   A0A0 A0A0 2A9C 7FF1 2A9D 0000 0100 0000        ...A........
0048   C001 8FF8 0004 0000 0000 0000 0000 0000    ...X............
0050   0000 0000 0000 0000 0000 0000 0000 0000    ................
....
```

Figure 9: SYSCATLG.CD3PRTS.LEVEL1-NAME1.LEVEL2-NAME2.LEVEL3-NAME2

This is the LEVEL1-NAME1 subcatalogue. Within it are 3 entries, LEVEL2-NAME1, LEVEL2-NAME2 and LEVEL2-NAME3. The parent catalogue is located at 3DD14 and LEVEL1-NAME1 is the first entry in that catalogue (words 3, 4 & 5).

Looking at the contents of 3DD14, figure 10, we find 3 entries. The first, as expected, is LEVEL1-NAME1. Its address (words 16 & 17) is 1E834 which is where we just came from.



```
0000   0003 DD15 FFFF 0000 0802 0004 CCC5 D6C5    ..]...........LEVE
0008   CCB1 ADCE C1CD C5B1 A0A0 A0A0 A0A0 A0A0    L1-NAME1
0010   AA9B 2E3E 2A9D 0000 0100 0000 0001 E834    *.............H.
0018   0004 0000 0000 0000 0000 0000 0000 D4D8    ..............TX
0020   D4A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    T
0028   A0A0 AA9B 34BA 2A9D 0000 0100 0000 0001      *..:..........
0030   E8B8 0004 0000 0000 0000 0000 0000 0000    H8..............
0038   CCC5 D6C5 CCB1 ADCE C1CD C5B2 A0A0 A0A0    LEVEL1-NAME2
0040   A0A0 A0A0 2A9C 8048 2A9D 0000 0100 0000         ...........
0048   0001 A4AC 0004 0000 0000 0000 0000 0000    ..$,............
0050   0000 CCC5 D6C5 CCB1 ADCE C1CD C5B3 A0A0    ..LEVEL1-NAME3
0058   A0A0 A0A0 A0A0 2A9C 8388 2A9D 0000 0100          ..........
0060   0000 0001 9AF8 0004 0000 0000 0000 0000    .....X..........
0068   0000 0000 0000 0000 0000 0000 0000 0000    ................
....
```

Figure 10: SYSCATLG.CD3PRTS.LEVEL1-NAME1.LEVEL2-NAME2.LEVEL3-NAME2

The next up level catalogue is at location 802 (words 3 & 4) and the catalogue we're currently in will be the fourth entry (word 5) in 802.

Patching into 802, which is SYSCATLG, we find the fourth entry is CD3PRTS, figure 11.

```
0000   0000 0803 FFFF 0000 0800 FFFF C4C1 D4A0     .............DAT
0008   A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0
0010   AA87 95BB 2A9D 0000 0100 0000 0000 C8A4     *..!..........H$
0018   0004 0000 0000 0000 0000 0000 0000 C3C1     ..............CA
0020   C4A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0     D
0028   A0A0 2A8C 0018 2A9D 2001 0100 0000 0000       .............
0030   1FE4 0004 0000 0000 0000 0000 0000 0000     .D..............
0038   CBC5 D9C6 C9CC C5A0 A0A0 A0A0 A0A0 A0A0     KEYFILE
0040   A0A0 A0A0 2A8C 0079 2A9D 0000 0100 0000         ............
0048   0001 5F28 0004 0000 0000 0000 0000 0000     ................
0050   0000 C3C4 B3D0 D2D4 D3A0 A0A0 A0A0 A0A0     ..CD3PRTS
0058   A0A0 A0A0 A0A0 2A8C 0073 AA9D 0000 0100           ....*.....
0060   0000 0003 0014 0004 0000 0000 0000 0000     .....4....].....
0068   0000 0000 C3C9 C9CE A0A0 A0A0 A0A0 A0A0     ....CIIN
0070   A0A0 A0A0 A0A0 A0A0 2A8C 00E6 2A9D 3500             ...F....
0078   0100 0000 0000 CB64 0004 0000 0000 0000     ......K.........
0080   0000 0000 0000 C9CE D3C5 D2D4 A0A0 A0A0     ......INSEPT
0088   A0A0 A0A0 A0A0 A0A0 A0A0 2A8C 00F0 2A9D               ...P..
0090   0000 0100 0000 0001 6C60 0004 0000 0000     ................
```

Figure 11:SYSCATLG.CD3PRTS.LEVEL1-NAME1.LEVEL2-NAME2.LEVEL3-NAME2

Returning to figure 9 we will now trace downward to the COM and RG0 files. The second entry is LEVEL2-NAME2. Its address, words 2F and 30, is 1A4CC and the contents of that address are shown in figure 12.

```
0000   0001 A4CD FFFF 0001 E834 0002 CCC5 D6C5     ..$M....H...LEVE
0008   CCB3 ADCF C1CD C5B1 A0A0 A0A0 A0A0 A0A0     L3-NAME1
0010   AA98 2E9E 2A9D 0000 0100 0000 0001 E840     *..............H.
0018   0004 0000 0000 0000 0000 0000 0000 CCC5     ..............LE
0020   D6C5 CCB3 ADCE C1CD C5B2 A0A0 A0A0 A0A0     VEL3-NAME2
0028   A0A0 AA9B 309D 2A9D 0000 0100 0000 0001       *.............
0030   9704 0004 0000 0000 0000 0000 0000 0000     ................
0038   CCC5 D6C5 CCB3 ADCE C1CD C5B3 A0A0 A0A0     LEVEL3-NAME3
0040   A0A0 A0A0 AA9B 30B5 2A9D 0000 0100 0000         *..5........
0048   0001 E85C 0004 0000 0000 0000 0000 0000     ..H.............
0050   0000 0000 0000 0000 0000 0000 0000 0000     ................
....
```

Figure 12: SYSCATLG.CD3PRTS.LEVEL1-NAME1.LEVEL2-NAME2.LEVEL3-NAME2

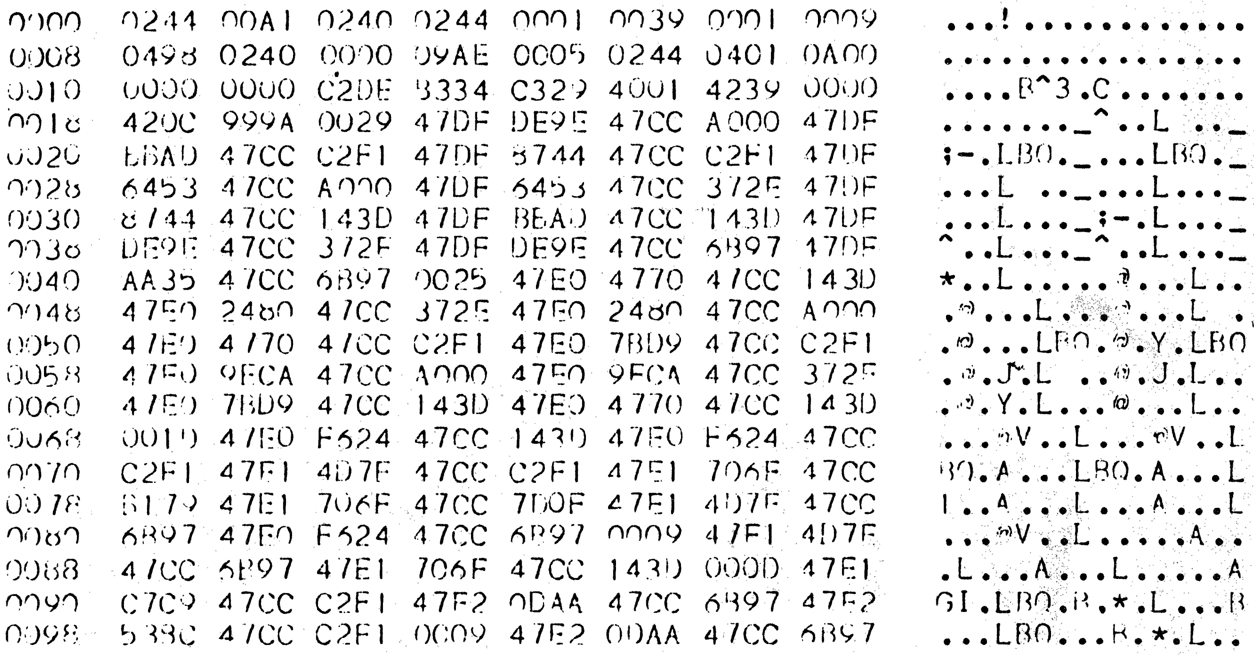
The second entry in this subcatalogue is LEVEL3-NAME2 and its address is 19704. At 19704 we find the listings shown in figure 13.

```
0000   0001 9705 FFFF 0001 A4CC 0002 D2C7 B0A0     .........SL..RGO
0008   A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0
0010   AA9C 07E1 2A9D 0000 0000 0000 0004 B2FC     *..A...........2\
0018   0258 0000 0000 0000 0000 0000 0000 C3CF     ..............CO
0020   CDA0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0     M
0028   A0A0 2A9C 88FF 2A9C 0000 0000 0000 0005           ..........
0030   9DC0 02B8 0000 0000 0000 0000 0000 0000     .?.8............
0038   0000 0000 0000 0000 0000 0000 0000 0000     ................
....
```

Figure 13: SYSCATLG.CD3PRTS.LEVEL1-NAME1.LEVEL2-NAME2.LEVEL3-NAME2

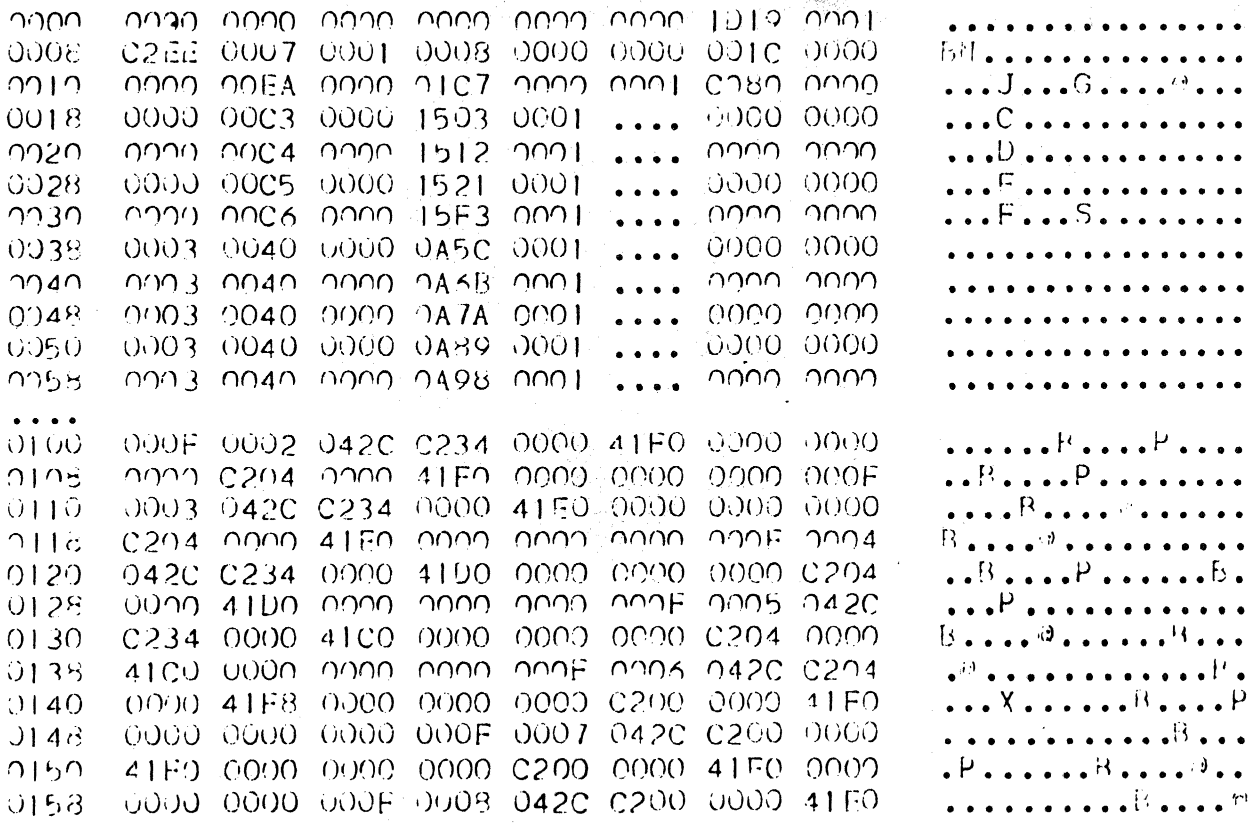
Here we find entries for the Regen and COM files. The actual Regen data starts at location 4B2FC, words 16 & 17, and requires 258 sectors, word 18. The com data starts at 59DCO, words 2F & 30, and requires 2B8 sectors, word 31.

Printouts of the actual Regen and COM data, figures 14 & 15, show how different their structure is.



```
0000   0244 00A1 0240 0244 0001 0039 0001 0009     ...!............
0008   0498 0240 0000 09AE 0005 0244 0401 0A00     ................
0010   0000 0000 C2DE 8334 C329 4001 4239 0000     ....B^3.C.......
0018   420C 999A 0029 47DF DE9E 47CC A000 47DF     ........_^..L .._
0020   EBAD 47CC C2F1 47DF 8744 47CC C2F1 47DF     ;-.LBO._...LBO._
0028   6453 47CC A000 47DF 6453 47CC 372E 47DF     ...L .._...L..._
0030   8744 47CC 143D 47DF BEAD 47CC 143D 47DF     ...L..._;-.L..._
0038   DE9E 47CC 372F 47DF DE9E 47CC 6B97 47DF     ^..L..._^..L..._
0040   AA35 47CC 6B97 0025 47E0 4770 47CC 143D     *..L.....@...L..
0048   47E0 2480 47CC 372E 47E0 2480 47CC A000     .@...L...@...L .
0050   47E0 4770 47CC C2F1 47E0 7BD9 47CC C2F1     .@...LBO.@.Y.LBO
0058   47E0 9ECA 47CC A000 47E0 9ECA 47CC 372F     .@.J.L ..@.J.L..
0060   47E0 7BD9 47CC 143D 47E0 4770 47CC 143D     .@.Y.L...@...L..
0068   0010 47E0 F624 47CC 143D 47E0 F624 47CC     ...@V..L...@V..L
0070   C2F1 47E1 4D7F 47CC C2F1 47E1 706F 47CC     BO.A...LBO.A...L
0078   8179 47E1 706F 47CC 7D0F 47E1 4D7F 47CC     I..A...L...A...L
0080   6B97 47E0 F624 47CC 6B97 0009 47E1 4D7F     ...@V..L.....A..
0088   47CC 6B97 47E1 706F 47CC 143D 000D 47E1     .L...A...L.....A
0090   C7C9 47CC C2F1 47F2 0DAA 47CC 6B97 47F2     GI.LBO.B.*.L...B
0098   538C 47CC C2F1 0C09 47E2 0DAA 47CC 6B97     ...LBO...B.*.L..
```

Figure 14: Typical RG0 File



```
0000   0030 0000 0000 0000 0000 0000 1D19 0001     ................
0008   C2EE 0007 0001 0008 0000 0000 001C 0000     BN..............
0010   0000 00EA 0000 01C7 0000 0001 C780 0000     ...J...G....@...
0018   0000 00C3 0000 1503 0001 .... 0000 0000     ...C............
0020   0000 00C4 0000 1512 0001 .... 0000 0000     ...D............
0028   0000 00C5 0000 1521 0001 .... 0000 0000     ...E............
0030   0000 00C6 0000 15F3 0001 .... 0000 0000     ...F...S........
0038   0003 0040 0000 0A5C 0001 .... 0000 0000     ................
0040   0003 0040 0000 0A6B 0001 .... 0000 0000     ................
0048   0003 0040 0000 0A7A 0001 .... 0000 0000     ................
0050   0003 0040 0000 0A89 0001 .... 0000 0000     ................
0058   0003 0040 0000 0A98 0001 .... 0000 0000     ................
....
0100   000F 0002 042C C234 0000 41F0 0000 0000     .......B....P....
0108   0000 C204 0000 41F0 0000 0000 0000 000F     ..B....P........
0110   0003 042C C234 0000 41F0 0000 0000 0000     ....B...........
0118   C204 0000 41F0 0000 0000 0000 000F 0004     B....@..........
0120   042C C234 0000 41D0 0000 0000 0000 C204     ..B....P......B.
0128   0000 41D0 0000 0000 0000 000F 0005 042C     ...P............
0130   C234 0000 41C0 0000 0000 0000 C204 0000     B....@......B...
0138   41C0 0000 0000 0000 000F 0006 042C C204     .@.............P.
0140   0000 41F8 0000 0000 0000 C200 0000 41F0     ...X.......B....P
0148   0000 0000 0000 000F 0007 042C C200 0000     .............B...
0150   41F0 0000 0000 0000 C200 0000 41F0 0000     .P.......B....@..
0158   0000 0000 000F 0008 042C C200 0000 41F0     ...........B....@
```

Figure 15: Typical COM File

INVALID POINTERS

As the subcatalogue listings show, addresses are stored in two 16 bit words. The number of distinct addresses available using 32 bits is 2^{32} or 4,294,967,296. Since a 150M word disc contains 508,702 sectors, only .012% of all possible addresses are valid; the other 99.988% are invalid.

If one of these invalid numbers is passed to a disc read utility routine, the system tries to find an auxiliary disc drive or a disc volume which would use the erroneous number as a valid address. Not finding it, the system gives a F006 error with the message "INVALID DRIVE NUMBER ARGUMENT", or "VOLUME NOT ON ASSIGNED DRIVE."

Addresses can become invalid by several means; a power surge or static discharge during a filing operation, a faulty bit in memory, a faulty disc controller, etc.

Both the hardware and the software in a CV system have many error checking routines but occasionally bad addresses will crop up. Once they do, the trick is to locate the bad address and return it to its proper state.

Any system command which accesses the file manager will trap to a F006 message if an invalid address is encountered. Particularly dangerous is the FMCLEAR command.

It is possible for this command to pick up an address that is within the valid range but points to the wrong place — into the O/S for instance. It then goes in and starts resetting what it thinks are status words and wipes out the O/S before it crashes all together. A cold start will recover the O/S but another FMCLEAR will wipe it out again.

RECOVERY PROCEDURE

This covers the preliminaries; the rest of the paper covers the meat of the topic — how to recover from a F006.

We must assume intitially that there is not a severe hardware problem — a head crash or broken cable for instance.

If the system is down, go as far as necessary in Procedure I to bring it up, then follow Procedure II.

Procedure I

1. Reboot but do not run FMCLEAR
2. Coldstart the system
3. Load a good disc pack into the drive and reboot. Press the CPU stop switch, power down the disc drive and put the bad disc back in, then press the CPU "continue" switch.
4. If step 3 won't bring the system up you have a serious hardware problem. Call the CV Response Center.

If the system is not down issue the DUMPERR command, and have all the users log off; then follow Procedure II to locate the bad address.

Procedure II

1. Issue the CHKDISC command. This step is optimal but it is useful in eliminating the possibility that a hard disc error is causing the corruption. If CHKDISC fails in the first 800 sectors of the disc (in the O/S), coldstart the system to bring it back up. If a coldstart won't work the disc must be reformatted or scrapped — after the files have been recovered; follow procedure I, step 3. If CHKDISC encounters a hard disc error in the file manager area it will eliminate the sector in which the error occurred by setting a bit in the "Cluster Allocation Table," a directory of available sectors.
2. Issue the FMCHECK command. If the command that evoked the F006 was printing out catalogue listings, the bad file manager area is easy to locate. Run FMCHECK cat.subcat on the next higher catalogue level than the one that crashed. If the specific file manager area is not known, run FMCHECK on SYSCATLG. FMCHECK will terminate when it encounters a bad address or pointer and print a message indicating the last file processed. The bad subcatalogue or file will be the next one in the catalogue tree structure.

3. Determine the bad address or pointer. This is the step in which magic, clairvoyance, luck and divine intervention are as helpful as a deep understanding of the file manager structure.

   Patch into the parent catalogue — use the LISTCAT command with the //FULL option to get the address and look for any of the following problems.

   The problem most often encountered will be a subcatalogue address pointing to something other than legitimate file manager data. If the correct address cannot be determined — look for single bit changes in the address — the entire subcatalogue can be deleted by setting all 25 words of the entry to zero. This, of course, will wipe out all files under the subcatalogue, so if these files are irreplaceable, considerable effort should be put into determining the correct address. Other areas where errors can cause a F006 are: subcatalogue sector count (word 19) should always be equal to the cluster size, a RIB address (words 20 & 21) can be invalid or indicate a RIB exists when there shouldn't be one, or RIB data can be invalid as can file types (word 25).

   Errors in the sector header can occur in the following places: Next sector address (words 0 & 1) can be wrong, the address of the parent catalogue (words 2 & 3) or the index into it (word 4) can be wrong.

Manually following the pointers, patching from one sector to the next will usually find the errant pointer and in most cases diligence will locate the correct pointer value. Occasionally an address will go bad just before a filing operation and a com file will be written right over 300 or 400 sectors of the catalogue structure. When this happens reconstruction and recovery are impossible. The FMINIT command should be issued and all parts and application software should be reloaded from backup.

This overwriting is a fairly rare problem, however. Should it occur with any frequency, more than once every 6 months for instance, a hardware problem exists and CV should be contacted. F006's will occur either very frequently or very infrequently. When their occurrence changes from infrequent to frequent, contact the response center.

Once the pointers have been reset or subcatalogues have been deleted, FMCHECK cat.subcat should be run to verify the success of the recovery. If the FMCHECK is successful, run FMCHECK on the entire SYSCATLG to determine if any other errors exist. If this is successful, run FMCLEAR,NOTINUSE.

At this point recovery is completed and any files which had to be deleted in the recovery process can be reloaded from backup tapes.

Or, if it was determined in procedure II step 1 that the pack was unusable, a full backup can be performed and a new disc pack can be loaded.

SUMMARY

F006 errors can be recovered, but unless you are familiar with and comfortable in the file manager and catalogue structure, recovery can be a long tedious process. Time spent learning your way around in the file manager before a F006 occurs will be paid back many times over when you need to recover 200 hours of production or design work, and the pressures to get the system back up are high.

So, to minimize loss of work, data and time, make daily backups and learn the file manager structure. It will pay off.

System Management

ARCHIVING CADDS PARTS AND AUTOMATIC UPDATING OF ON-LINE TEXTFILE DOCUMENTATION

Ronald C. Smith, Sr.
M&M/MARS
Highstreet
Hackettstown, NJ 07840
201-852-1000

ABSTRACT

The information contained in this paper outlines the following procedures:

(1) Semi-automatic offline storage of CADDS parts on magnetic tape;
(2) The automatic updating of textfiles which are used for documentation of parts on tape and for assistance in locating a part that must be restored to disc;
(3) The automatic updating of a master index of all parts that are on disc and archive tape;
(4) The required steps to prepare textfiles and new tapes for the proper execution of the programs.

This archiving system uses a series of EXECUTE and PAREX programs that are designed to enhance file managing.

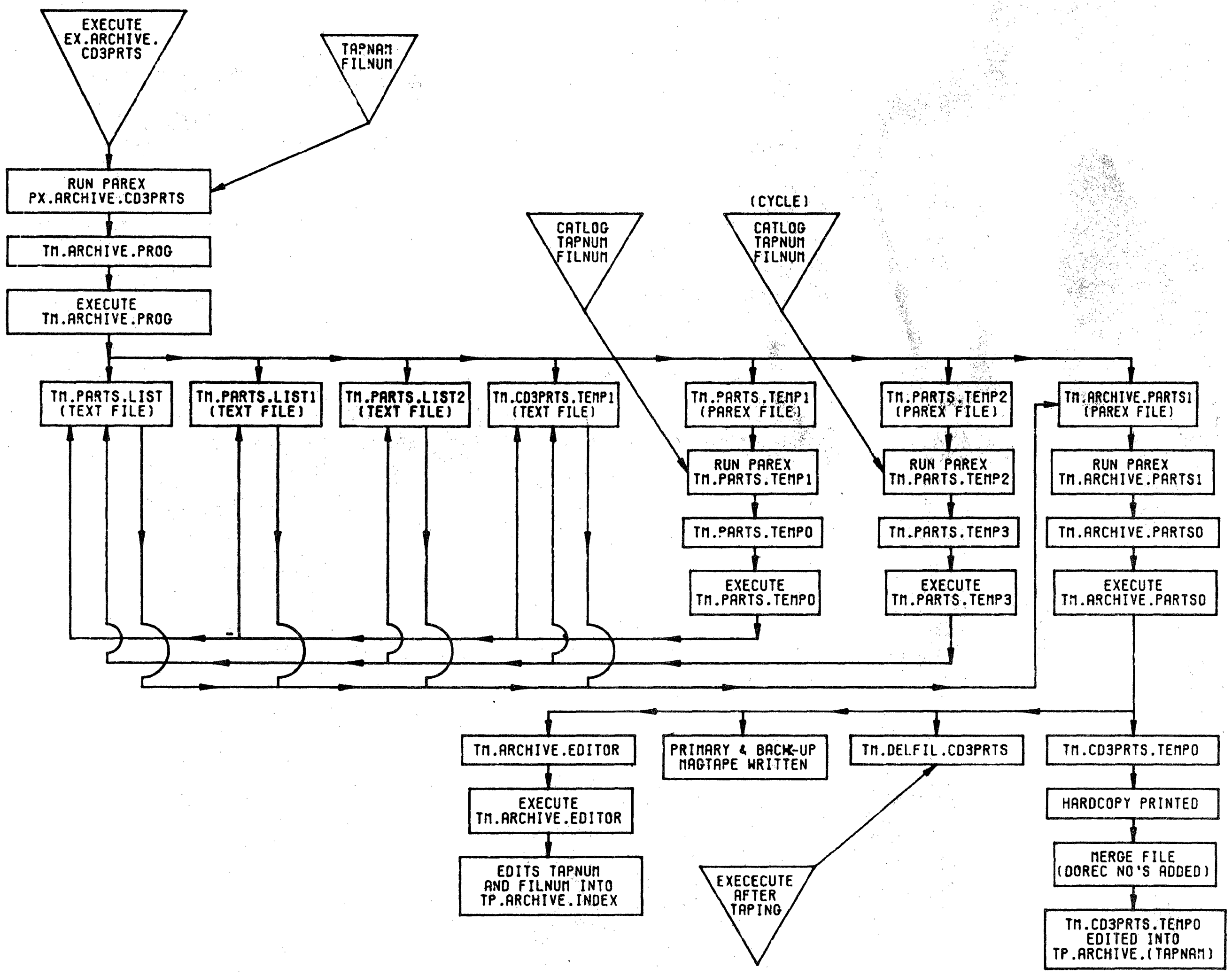
INDEX

Introduction

Shortly after the CADDS system went on stream at M&M/MARS, it became apparent that we should be writing a set of standard practices for naming files, archiving parts, and have a means to quickly find these parts to restore them to disc. Over a period of several months, many different naming conventions and taping procedures were tried and their inadequacies were soon discovered.

During this period it became apparent that we needed a system of file managing in which we could quiz the system (using n > SRCHBCD) to locate a given part and determine (1) if it were on disc or tape, and (2) if it were on tape, give us the SAVFIL number and the DOREC numbers to facilitate quick restoring to disc. We also wanted the maintenance of this file managing system to be automated to minimize errors and file managing time. Our archiving program (EX.ARCHIVE.CD3PRTS) has been developed to fulfill these requirements.

When EX.ARCHIVE.CD3PRTS is executed, parts are put to "PRIMARY" and "BACK-UP" magtapes for archival storage, and two textfiles (TP.ARCHIVE.INDEX and TP.ARCHIVE.(tape name)) are updated. The program also generates an execute program that, when executed, will delete the parts that have been put on tape.


EXECUTE EX.ARCHIVE. CD3PRTS
TAPNAM FILNUM
RUN PAREX PX.ARCHIVE.CD3PRTS
TM.ARCHIVE.PROG
EXECUTE TM.ARCHIVE.PROG
CATLOG TAPNUM FILNUM
(CYCLE)
CATLOG TAPNUM FILNUM
TM.PARTS.LIST (TEXT FILE)
TM.PARTS.LIST1 (TEXT FILE)
TM.PARTS.LIST2 (TEXT FILE)
TM.CD3PRTS.TEMP1 (TEXT FILE)
TM.PARTS.TEMP1 (PAREX FILE)
TM.PARTS.TEMP2 (PAREX FILE)
TM.ARCHIVE.PARTS1 (PAREX FILE)
RUN PAREX TM.PARTS.TEMP1
RUN PAREX TM.PARTS.TEMP2
RUN PAREX TM.ARCHIVE.PARTS1
TM.PARTS.TEMPO
TM.PARTS.TEMP3
TM.ARCHIVE.PARTSO
EXECUTE TM.PARTS.TEMPO
EXECUTE TM.PARTS.TEMP3
EXECUTE TM.ARCHIVE.PARTSO
TM.ARCHIVE.EDITOR
PRIMARY & BACK-UP MAGTAPE WRITTEN
TM.DELFIL.CD3PRTS
TM.CD3PRTS.TEMPO
EXECUTE TM.ARCHIVE.EDITOR
HARDCOPY PRINTED
EDITS TAPNUM AND FILNUM INTO TP.ARCHIVE.INDEX
EXECECUTE AFTER TAPING
MERGE FILE (DOREC NO'S ADDED)
TM.CD3PRTS.TEMPO EDITED INTO TP.ARCHIVE.(TAPNAM)

Section one of this paper discusses the procedures to run EX.ARCHIVE.CD3PRTS. Sections two through four will explain the necessary preliminary steps to prepare the system to run this program, suggestions for file naming conventions, formats, etc. and additional EXECUTE files to assist in file managing.

Section One

Procedures for running EX.ARCHIVE.CD3PRTS

Purpose—EX.ARCHIVE.CD3PRTS is executed to write CADDS parts on "PRIMARY" and "BACK-UP" magnetic for archival storage. The program also updates index textfiles and generates an EXECUTE file to delete archived parts from disc.

Procedure:

1. The operator determines what parts are to be archived and makes a list using as few catalog level names as possible to describe the part.

2. The operator then determines the tape name (TAPNAM), tape number (TAPNUM), and savfil number (FILNUM) that the parts are to be stored under.

3. Load the proper magtape on the magtape unit. (Be absolutely sure that the correct tapes are being used.)

4. At a terminal other than the system hardcopy device, and at system level, type in

```
n> EXECUTE EX.ARCHIVE.CD3PRTS ↲
```

5. When the program pauses for variable input and after typing a RETURN the operator is requested to enter:

TAPNAM—The third level (file name) of the tape name and supporting index text file.
FILNUM—The two digit number used to indicate the SAVFIL level on a stacked tape (Use leading zeros).
Type a CONTROL-X to continue the program.

6. When the system pauses a second time for variable input, the operator is requested to enter the following:

CATLOG—The catalog of CD3PRTS being archived (do not use more than sixteen characters to identify a past or group of parts).
TAPNUM—The four digit number of the magnetic tape that the parts are to be stored on (Use leading zeros).
FILNUM—refer to step five above.

7. When the system pauses the third time for variable input, the operator enters the values for the CATLOG, TAPNUM, and FILNUM parameters as in step number six (6) above, only this time he/she is requested to cycle the PAREX program for each additional CATALOG being ARCHIVED.

(The program prints "TYPE OK TO CYCLE")

Type a CONTROL-X to continue the program.

The program needs no further operator interaction, other than typing an occasional "CONTINUE" at a pause in the program.

Note: These "PAUSES", with the exception of the pause located between two SAVFILS, may be deleted if desired.

The program generates several text files and edits them into a specific format and the edits the results into the two permanent textfiles that serve as on line documentation tape indexes. A HARDCOPY list of parts saved is also printed out on the hardcopy device.

8. At the "PAUSE" after the primary savfil is complete, remove the primary tape and load the back-up tape. Type "CONTINUE" to complete the program.

If taping errors are encountered, extract the lines that contain all the taping commands and file names from TM.ARCHIVE.PARTS0 to create a new execute program.

After it has been determined what caused the error and corrections have been made, execute this new program. When both savfil tapes have been made without errors, execute TM.DELFIL.CD3PRTS to delete the archived parts from disc.

EXAMPLE :

NOTES:

ITALICIZED TEXT INDICATES COMMANDS AND TEXT WHICH MUST BE ENTERED BY THE OPERATOR.

THE FOLLOWING EXAMPLE IS A HARDCOPY OF THE AREAS IN THE PROGRAM WHICH REQUIRE OPERATOR INTERACTION ONLY.

```
3>EXECUTE EX.ARCHIVE.CD3PRTS


3>DELETE TM.ARCHIVE//NLEV
3>DELETE TM.PARTS//NLEV
3>DELETE TM.CD3PRTS//NLEV
3>DELETE TM.DELETE//NLEV

3>CADDS

PART NAME:TEMP
 CREATING NEW PART
***********************************************************************
****                                                               ****
****     TAPNAM = THIRD LEVEL (FILE) NAME OF THE TAPE PARTS        ****
****              ARE BEING SAVED ON AND THE SUPPORTING TEXT       ****
****              FILE NAME.                                       ****
****     FILNUM = 2 DIGIT MAGNETIC TAPE SAVFIL NUMBER              ****
****              (USE LEADING ZEROS)                              ****
****                                                               ****
***********************************************************************
*RUN PAREX FILE=PX.ARCHIVE.CD3PRTS NOGO TARGET TM.ARCHIVE.PROG

PARAMS: CONTROL-V   CARRIAGE-RETURN

  TAPNAM= CLASS-1-(3)  FILNUM= 02

VAR CMD*  CONTROL-X
*EXIT

FILE=F: DONT=D: REPLY=D
NEW PART=N: GOODBYE=G: REPLY=G


***********************************************************************
*****
*****      CATLOG = CATALOG OF CD3PRTS BEING ARCHIVED.
*****      TAPNUM = 4 DIGIT MAGNETIC TAPE NUMBER THAT
*****               THE PARTS ARE BEING SAVED ON.
*****               (USE LEADING ZEROS)
*****      FILNUM = 2 DIGIT MAGNETIC TAPE SAVFIL NUMBER
*****               (USE LEADIG ZEROS)
*****
*****      RECYCLE  AS MANY TIMES AS THERE ARE CATALOGS
*****      TO BE FILED. USE THE SAME TAPNUM AND FILNUM
*****      FOR EACH CATLOG.
*****
***********************************************************************
3>CADDS


PART NAME:ARCHIVE.DUMMY0
 CREATING NEW PART
*RUN PAREX FILE=TM.PARTS.TEMP1 TARGET TM.PARTS.TEMP0 NOGO
```

```
PARAMS: CONTROL-V

  CATLOG=N.DIA.DAS  TAPNUM= 0028  FILNUM= 02 (CARRIAGE-RETURN)

*RUN PAREX FILE=TM.PARTS.TEMP2 TARGET TM.PARTS.TEMP3 CYCLE NOGO

  CATLOG=N.CHT.JHB  TAPNUM= 0028  FILNUM= 02 (CARRIAGE-RETURN)

TYPE OK TO RECYCLE
>>OK  (CARRIAGE-RETURN)
PARAMS: (CARRIAGE-RETURN)

  CATLOG=N.GRF.RJE  TAPNUM= 0028  FILNUM= 02 (CARRIAGE-RETURN)

TYPE OK TO RECYCLE
>> CARRIAGE RETURN

VAR CMD* CONTROL-X
*EXIT

FILE=F; DONT=D; REPLY=D
NEW PART=N; GOODBYE=G; REPLY=G



3>ATTACH MT.TAPE
TAPO OF TYPE TAPE ATTACHED AS MT
3>REWIND
3>FUTIL
* COPY CD3PRTS.N.DIA.DAS,:MT02//NLEV
* ADD CD3PRTS.N.CHT.JHB//NLEV
* ADD CD3PRTS.N.GRF.RJE//NLEV
*
3>REWIND
3>CHECK :MT02
3>REWIND


**                MOUNT THE BACKUP TAPE......
3>PAUSE
TYPE "CONTINUE" WHEN READY
   .        2         .         .         .         .         .


CONTINUE


3>REWIND
3>FUTIL
* COPY CD3PRTS.N.DIA.DAS,:MT02//NLEV
* ADD CD3PRTS.N.CHT.JHB//NLEV
* ADD CD3PRTS.N.GRF.RJE//NLEV
*
3>REWIND
3>CHECK :MT02
3>REWIND
3>DETACH MT
```

```
EX.ARCHIVE.CD3PRTS
 5-25-81  8:32:28
```

```
<*     FILE:          EX.ARCHIVE.CD3PRTS
<*     CREATED:       22 MAY 1981
<*     PROGRAMMER:    RONALD SMITH
<*
<*
<*     PURPOSE:
<*             EX.ARCHIVE.CD3PRTS IS EXECUTED TO WRITE CLASS-I A
<*             CLASS-II CADDS PARTS ON A PRIMARY AND BACK-UP TAPE
<*             FOR ARCHIVAL STORAGE.  THIS PROGRAM ALSO UPDATES TWO
<*             (2) TEXT FILES (TP.ARCHIVE.INDEX AND TP.ARCHIVE.(TAPENAM).
<*             THESE FILES ARE USED FOR RAPID SEARCH AND LOCATION OF ANY
<*             CADDS PART THAT IS ON DISK OR THAT HAS BEEN ARCHIVE.
<*             THIS PROGRAM ALSO CREATES AN EXECUTE FILE THAT WHEN
<*             EXECUTED, WILL DELETE THE ARCHIVED PARTS.
<*
<*
<*     PROCEDURE:
<*             1.     DELETE ALL TEMPORARY FILES USED DURING THE PREVIOUS
<*                    ARCHIVAL TAPING.
<*
<*             2.     RUN PAREX FOR TAPNAM AND FILNUM (OPERATOR INPUT).
<*
<*             3.     TEMPORARY FILES CREATED FOR FUTURE USE.
<*
<*             4.     RUN PAREX FOR CATLOGS,TAPNUM, AND FILNUM
<*                    (OPERATOR INPUT)
<*
<*             5.     PARTS LIST AND EDIT COMMANDS CREATED AND
<*                    ADDED TO TEMP FILES FOR FUTURE USE.
<*
<*             6.     A HARDCOPY LIST OF PARTS BEING ARCHIVED
<*                    IS PRINTED OUT.
<*
<*             7.     PRIMARY AND BACK-UP SAVFIL TAPES MADE.
<*
<*             8.     TP.ARCHIVE.(TAPNAM) AND TP.ARCHIVE.INDEX
<*                    IS UPDATED.
<*
<*             9.     A MESSAGE TO EXECUTE TM.DELFIL.CD3PRTS
<*                    IS EXECUTED.
<*
<*
<*
<*     REVISIONS:
<*
<*             R1     22MAY81     CREATED FROM EX.ARCHIVE.CD3PRTS
<*                                UPDATED TO RUN WITH CGOS200/10
<*                                OPERATING SYSTEM.
<*
<*
```

```
EX.ARCHIVE.CD3PRTS
    5-25-81  8:32:28
```

```
<*-----------------------------------------------------------------
DELETE TM.ARCHIVE//NLEV
DELETE TM.CD3PRTS//NLEV
DELETE TM.PARTS//NLEV
DELETE TM.DELFIL//NLEV
*******************************************************************
***                                                            ****
***     TAPNAM = THIRD LEVEL (FILE) NAME OF THE TAPE PARTS     ****
***              ARE BEING SAVED ON AND THE SUPPORTING TEXT    ****
***              FILE NAME.                                    ****
***     FILNUM = 2 DIGIT MAGNETIC TAPE SAVFIL NUMBER           ****
***              (USE LEADING ZEROS)                           ****
***                                                            ****
*******************************************************************
CADDS <CR>TEMP
RUN PAREX PX.ARCHIVE.CD3PRTS NOGO TARGET TM.ARCHIVE.PROG
<VAR>
EXIT<CR>DG
EDIT TM.ARCHIVE.PROG
S11-102,/XY/,/&/
F
PAUSE
EXECUTE TM.ARCHIVE.PROG
********************************************************************
***                                                             ****
***      IF NO ERROR HAVE BEEN ENCOUNTERED AND IF BOTH BACK-UP ****
***      AND BACK-UP TAPES HAVE BEEN MADE SUCCESSFULLY, EXECUTE****
***      THE FOLLOWING COMMAND:                                 ****
***                                                             ****
***            EXECUTE TM.DELFIL.CD3PRTS                        ****
***                                                             ****
********************************************************************
```

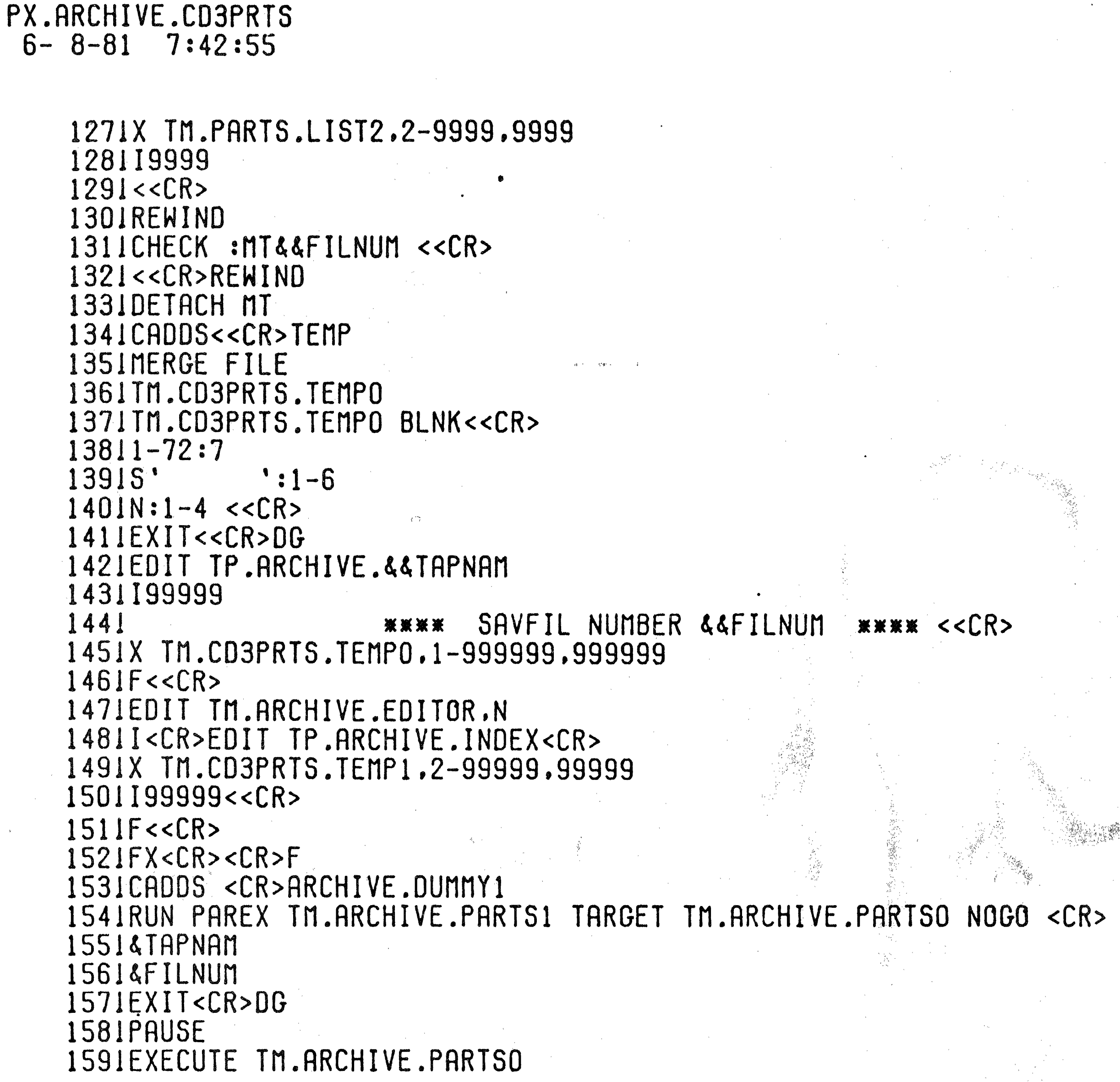
```
PX.ARCHIVE.CD3PRTS
 6- 8-81  7:42:55
```

```
&&
2 TAPNAM
2 FILNUM
&&
EDIT TM.PARTS.LIST.N
I
|||||||||||||||||||||||||||||| <CR>
F
EDIT TM.PARTS.LIST1.N
I
|||||||||||||||||||||||||||||| <CR>
F
EDIT TM.PARTS.LIST2.N
I
```

```
|||||||||||||||||||||||||||||| <CR>
F
EDIT TM.CD3PRTS.TEMP1,N
I
|||||||||||||||||||||||||||||| <CR>
F
EDIT TM.PARTS.TEMP1,N
I
&&XY
2 CATLOG
2 TAPNUM
2 FILNUM
&&XY
EDIT TM.PARTS.LIST1
I99
COPY CD3PRTS.&&CATLOG,:MT&&FILNUM//NLEV <<CR>
F<<CR>
EDIT TM.PARTS.LIST
I99
CD3PRTS.&&CATLOG//NLEV,TYPE=0<<CR>
F<<CR>
EDIT TM.CD3PRTS.TEMP1
I99999
S1-9999,/*        .&&CATLOG/,/*&&TAPNUM.&&FILNUM.&&CATLOG/<<CR>
F<<CR> <CR>
F
EDIT TM.PARTS.TEMP2,N
I
&&XY
2 CATLOG
2 TAPNUM
2 FILNUM
&&XY
EDIT TM.PARTS.LIST
I99999
CD3PRTS.&&CATLOG//NLEV,TYPE=0<<CR>
F<<CR>
EDIT TM.PARTS.LIST2
I9999
ADD CD3PRTS.&&CATLOG//NLEV <<CR>
F<<CR>
EDIT TM.CD3PRTS.TEMP1
I9999
S1-9999,/*        .&&CATLOG/,/*&&TAPNUM.&&FILNUM.&&CATLOG/<<CR>
F<<CR> <CR>
F
*********************************************************************
*****
*****      CATLOG = CATALOG OF CD3PRTS BEING ARCHIVED.
*****      TAPNUM = 4 DIGIT MAGNETIC TAPE NUMBER THAT
*****               THE PARTS ARE BEING SAVED ON.
*****               (USE LEADING ZEROS)
*****      FILNUM = 2 DIGIT MAGNETIC TAPE SAVFIL NUMBER
*****               (USE LEADIG ZEROS)
*****
*****      RECYCLE  AS MANY TIMES AS THERE ARE CATALOGS
*****      TO BE FILED. USE THE SAME TAPNUM AND FILNUM
*****      FOR EACH CATLOG.
```

PX.ARCHIVE.CD3PRTS
 6- 8-81 7:42:55

```
*****
***************************************************************
CADDS <CR> ARCHIVE.DUMMY
RUN PAREX TM.PARTS.TEMP1 TARGET TM.PARTS.TEMPO NOGO
<VAR>
RUN PAREX TM.PARTS.TEMP2 TARGET TM.PARTS.TEMP3 CYCLE NOGO
<VAR>
EXIT<CR>DG
PAUSE
EXECUTE TM.PARTS.TEMPO
EXECUTE TM.PARTS.TEMP3
<*
<*
EDIT TM.ARCHIVE.PARTS1,N
I
&&XY
2 TAPNAM
2 FILNUM
&&XY
GENCOM TM.CD3PRTS.TEMPO
<<CR>$2,-1,0$ <<CR> <CR>
X TM.PARTS.LIST,2-9999,9999
I9999
<<CR><<CR>F
GENCOM TM.DELFIL.CD3PRTS
<<CR> DELETE $1,-2,$/$-1$ <<CR> <CR>
X TM.PARTS.LIST,2-9999,9999
I9999
<<CR><<CR>F
*
*
*  A HARDCOPY OF THE FILE NAMES BEING ARCHIVED
*     IS NOW BEING PRODUCED
*
*
EDIT TM.CD3PRTS.TEMPO
N3
TH <<PRINT>
Q
ATTACH MT,TAPE
REWIND
FUTIL<CR>
X TM.PARTS.LIST1,2-99,9999
X TM.PARTS.LIST2,2-9999,9999
I9999
<<CR>
REWIND
CHECK :MT&&FILNUM <<CR>
<<CR>REWIND
****     MOUNT THE BACK-UP TAPE     ****
PAUSE
REWIND
FUTIL<CR>
X TM.PARTS.LIST1,2-99,9999
```



```
PX.ARCHIVE.CD3PRTS
 6- 8-81  7:42:55


    127|X TM.PARTS.LIST2,2-9999,9999
    128|I9999
    129|<<CR>
    130|REWIND
    131|CHECK :MT&&FILNUM <<CR>
    132|<<CR>REWIND
    133|DETACH MT
    134|CADDS<<CR>TEMP
    135|MERGE FILE
    136|TM.CD3PRTS.TEMPO
    137|TM.CD3PRTS.TEMPO BLNK<<CR>
    138|1-72:7
    139|S'      ':1-6
    140|N:1-4 <<CR>
    141|EXIT<<CR>DG
    142|EDIT TP.ARCHIVE.&&TAPNAM
    143|I99999
    144|                 ****  SAVFIL NUMBER &&FILNUM  **** <<CR>
    145|X TM.CD3PRTS.TEMPO,1-999999,999999
    146|F<<CR>
    147|EDIT TM.ARCHIVE.EDITOR,N
    148|I<CR>EDIT TP.ARCHIVE.INDEX<CR>
    149|X TM.CD3PRTS.TEMP1,2-99999,99999
    150|I99999<<CR>
    151|F<<CR>
    152|FX<CR><CR>F
    153|CADDS <CR>ARCHIVE.DUMMY1
    154|RUN PAREX TM.ARCHIVE.PARTS1 TARGET TM.ARCHIVE.PARTSO NOGO <CR>
    155|&TAPNAM
    156|&FILNUM
    157|EXIT<CR>DG
    158|PAUSE
    159|EXECUTE TM.ARCHIVE.PARTSO
```

Section Two
Preliminary Preparation

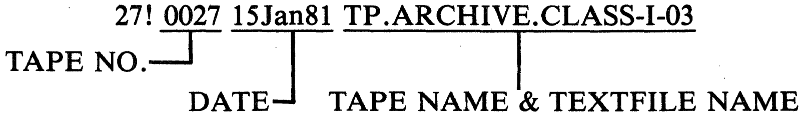
Prior to setting up the archive taping and file managing program, it is strongly recommended that standard procedures for naming CADDS parts and textfiles be put into practice. This will facilitate file transfer and file identification. A file naming standard also helps in quickly identifying files that do not belong on disc.

A CADDS part name should briefly describe the part and should also be of a format that will categorize similar parts, projects, plant areas, etc. Part names should also have a format that is easy to type.

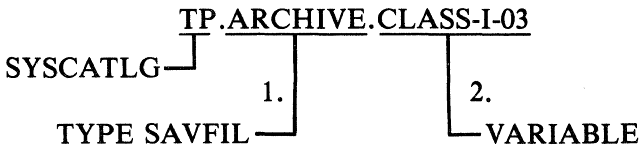
At M&M/MARS we use two different naming formats for CADDS parts. The first (Class-I) for training parts and scratch parts, working drawings, nonparts, and library parts. The second (Class-II) is for formal drawings that have been checked and approved for release. A chart showing these conventions may be found in the appendix.

The naming convention that we use for textfiles is somewhat simpler, the only requirement being that the "SYSCATLG" portion (first level) describes the function of the file (i.e., EX for EXECUTE files, PX for PAREX files, TP for TAPE RECORD files, TM for TEMPORARY files, etc. The remaining levels of the name are left to the discretion of the operator.

A tape numbering and naming convention is also required for the archiving system to function properly. As a new tape is put into use, a four digit number is assigned to it as well as a name. The tape name is also used as the textfile name for on-line documentation. All textfiles containing tape indexes and listings are filed in syscatlg "TP" (as noted above) and a record of numbered tapes is filed in TP.INDEX.MAGTAPE. A line from this file may appear as follows:


27! 0027 15Jan81 TP.ARCHIVE.CLASS-I-03
TAPE NO.
DATE
TAPE NAME & TEXTFILE NAME

The tape naming convention used at M&M/MARS is as follows:



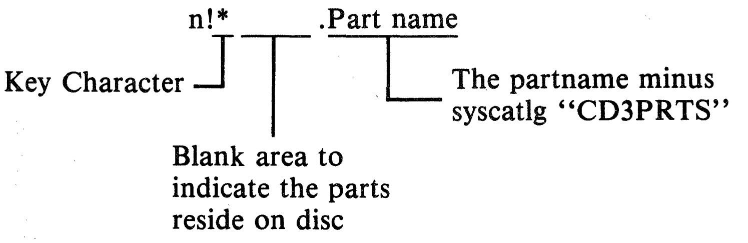
1. Type Savfil — ARCHIVE — for all archive tapes
PERSONAL — for personal tapes
DUMP — for daily system back-up tapes

2. Variable Variable tape name.

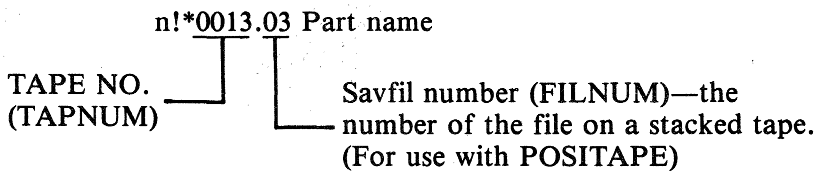
The next step in preparing the system for the archiving program is the creation of a textfile for the master index. The name of our master index is TP.ARCHIVE.INDEX. This file contains a list of CADDS parts currently on disc and archive tape. This index should be updated daily with the names of the parts that have been created or filed since a previous update. It may be done automatically when a daily back-up is done or it may be executed manually. (A program to update the index appears in section three.)

The line format of the master index file is as follows:

Phase 1:


n!* .Part name
Key Character
The partname minus syscatlg "CD3PRTS"
Blank area to indicate the parts reside on disc

Phase 2:


n!*0013.03 Part name
TAPE NO. (TAPNUM)
Savfil number (FILNUM)—the number of the file on a stacked tape. (For use with POSITAPE)

Note: When the tape number and Savfil number are present, the file is on archive tape and should not be on disc.

This file (TP.ARCHIVE.INDEX) must be on disc prior to executing the updating and archiving programs. This file is created as follows:

```
n  EDIT TP.ARCHIVE.INDEX,N ↵
#I ↵
!!!!!!!!!!!!!!!! ↵
↵
#F ↵
```

The part names will be included in the index after executing an update program using a very old "SINCEDATE" (EX.INDEX.UPDATE.CD3PRTS) will be discussed in Section Three.

New tapes and their supporting textfiles are prepared in the following manner:

A. New Tapes
   1. Assign a number to two tapes (a primary and a backup).
   2. Assign a name to the tapes (same name on both tapes). Note: Self-adhesive tape labels are strongly suggested.
   3. Load the tape on the magtape unit.
   4. Execute the following commands:

```
n>  ATTACH MT, TAPE ↵
n>  COPY TX.EMPTYFIL,:MT//NLEV, CHECK, LABEL ↵
type 6 character tape volume ID:  0025 ↵ *
```

NOTE: TX.EMPTYFIL must be on disk

*Use the tape number that has been assigned the tape.

This procedure writes a "Vol 1" label on the tape. (Do Step 4 for each tape.)

B. New textfiles
   1. Execute the following commands:

```
n>  EDIT TP.ARCHIVE.(Tape name),N ↵
#I ↵
        ***SAVFIL 01    *** ↵
EMPTY SAVFIL   EMPTY SAVFIL ↵
#F ↵
```

This procedure creates a textfile and enters the appropriate text into the file.

The above procedures may also be completed by executing EX.ARCHIVE.PREP.CR3PRTS. (A copy may be found in the appendix.)

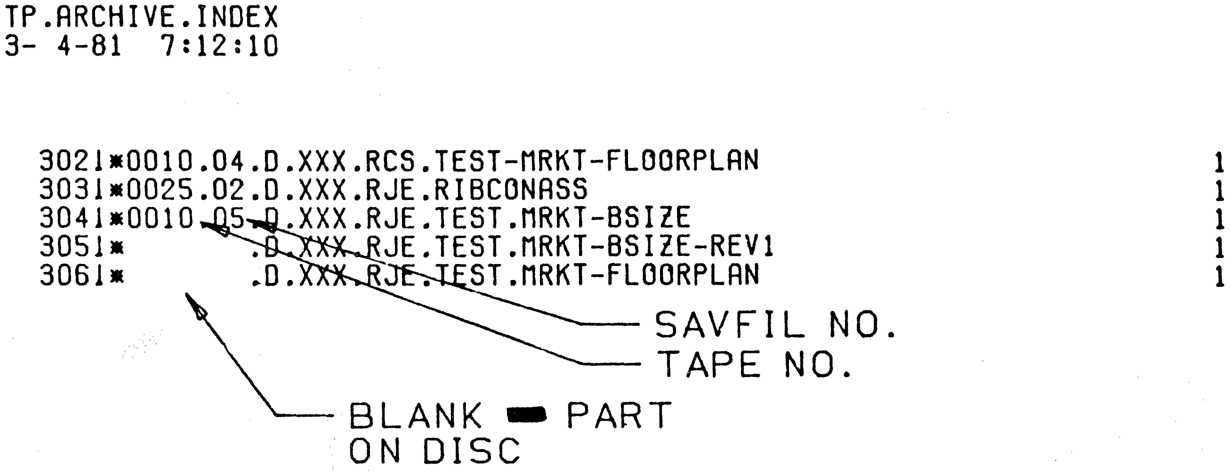
The final step in the preparation for executing the archive program is to establish a file for the "SINCEDATE" (TP.INDEX.SINCEDATE.PRT). Format for this file, as with all supporting files, should be strictly adhered to.

Create the file as follows:

```
        n>  EDIT TP.INDEX.SINCEDATE.PRT,N ↵
        #I ↵
        2-20-81        7:16:12 ↵
DATE                          TIME (hh:MM:SS)
TWO SPACES
        ↵
        #F ↵
```

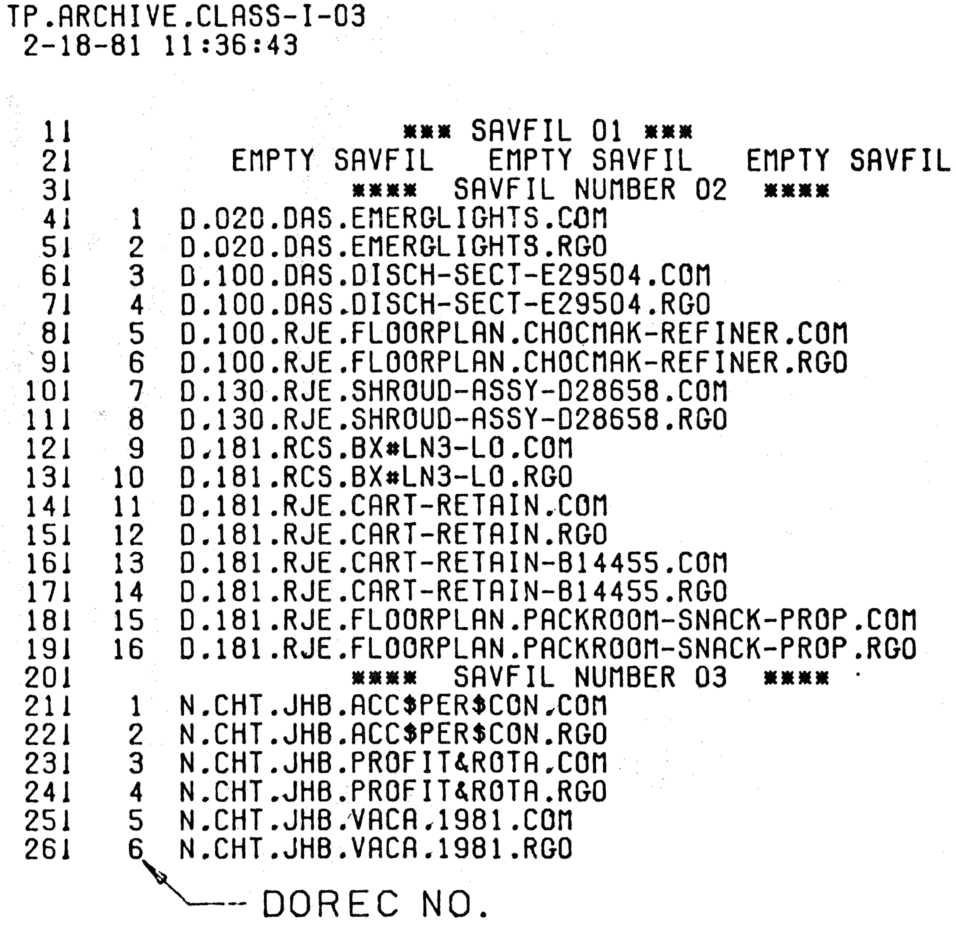
NOTE: As mentioned above, the SINCEDATE for the initial running of the update program should be very old to assure that all parts that are on disc will be included in the index.

Examples of the on-line documentation files follow on the next page.

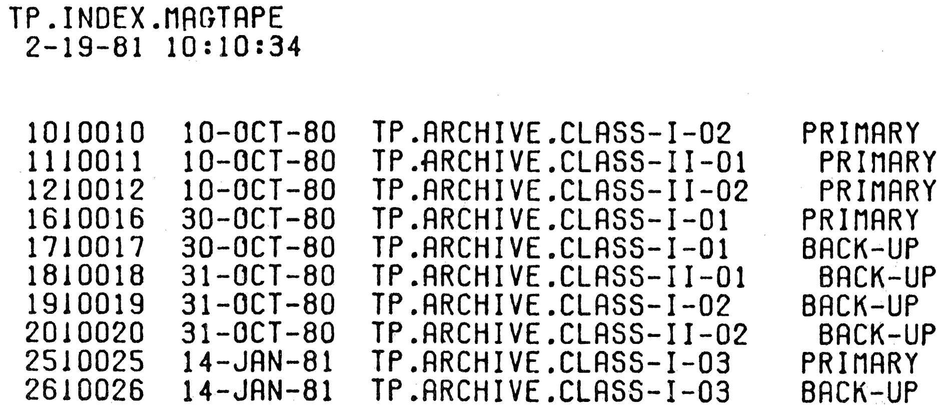


```
TP.ARCHIVE.INDEX
3- 4-81  7:12:10


  3021*0010.04.D.XXX.RCS.TEST-MRKT-FLOORPLAN                              1
  3031*0025.02.D.XXX.RJE.RIBCONASS                                        1
  3041*0010.05.D.XXX.RJE.TEST.MRKT-BSIZE                                  1
  3051*         .D.XXX.RJE.TEST.MRKT-BSIZE-REV1                           1
  3061*         .D.XXX.RJE.TEST.MRKT-FLOORPLAN                            1
```

SAVFIL NO.
TAPE NO.
BLANK ■ PART
ON DISC



```
TP.ARCHIVE.CLASS-I-03
 2-18-81 11:36:43

  11                      *** SAVFIL 01 ***
  21          EMPTY SAVFIL    EMPTY SAVFIL    EMPTY SAVFIL
  31                   ****  SAVFIL NUMBER 02  ****
  41    1  D.020.DAS.EMERGLIGHTS.COM
  51    2  D.020.DAS.EMERGLIGHTS.RGO
  61    3  D.100.DAS.DISCH-SECT-E29504.COM
  71    4  D.100.DAS.DISCH-SECT-E29504.RGO
  81    5  D.100.RJE.FLOORPLAN.CHOCMAK-REFINER.COM
  91    6  D.100.RJE.FLOORPLAN.CHOCMAK-REFINER.RGO
 101    7  D.130.RJE.SHROUD-ASSY-D28658.COM
 111    8  D.130.RJE.SHROUD-ASSY-D28658.RGO
 121    9  D.181.RCS.BX*LN3-LO.COM
 131   10  D.181.RCS.BX*LN3-LO.RGO
 141   11  D.181.RJE.CART-RETAIN.COM
 151   12  D.181.RJE.CART-RETAIN.RGO
 161   13  D.181.RJE.CART-RETAIN-B14455.COM
 171   14  D.181.RJE.CART-RETAIN-B14455.RGO
 181   15  D.181.RJE.FLOORPLAN.PACKROOM-SNACK-PROP.COM
 191   16  D.181.RJE.FLOORPLAN.PACKROOM-SNACK-PROP.RGO
 201                   ****  SAVFIL NUMBER 03  ****
 211    1  N.CHT.JHB.ACC$PER$CON.COM
 221    2  N.CHT.JHB.ACC$PER$CON.RGO
 231    3  N.CHT.JHB.PROFIT&ROTA.COM
 241    4  N.CHT.JHB.PROFIT&ROTA.RGO
 251    5  N.CHT.JHB.VACA.1981.COM
 261    6  N.CHT.JHB.VACA.1981.RGO
```



```
TP.INDEX.MAGTAPE
 2-19-81 10:10:34


 1010010  10-OCT-80  TP.ARCHIVE.CLASS-I-02    PRIMARY
 1110011  10-OCT-80  TP.ARCHIVE.CLASS-II-01    PRIMARY
 1210012  10-OCT-80  TP.ARCHIVE.CLASS-II-02    PRIMARY
 1610016  30-OCT-80  TP.ARCHIVE.CLASS-I-01    PRIMARY
 1710017  30-OCT-80  TP.ARCHIVE.CLASS-I-01    BACK-UP
 1810018  31-OCT-80  TP.ARCHIVE.CLASS-II-01    BACK-UP
 1910019  31-OCT-80  TP.ARCHIVE.CLASS-I-02    BACK-UP
 2010020  31-OCT-80  TP.ARCHIVE.CLASS-II-02    BACK-UP
 2510025  14-JAN-81  TP.ARCHIVE.CLASS-I-03    PRIMARY
 2610026  14-JAN-81  TP.ARCHIVE.CLASS-I-03    BACK-UP
```

Section Three

Updating the CD3PRTS Index

A key element of the file managing system is the daily updating of the master index (TP.ARCHIVE.INDEX). This index is designed to include all CADDS parts that are currently on disc and CADDS parts that have been put to magnetic tape. As mentioned previously, this program (EX.INDEX.UPDATE.CD3PRTS) may be executed either manually or automatically with the daily back-up taping or any other suitable means to execute the program automatically. Once executed, the program requires no further response from the operator. The program proceeds as follows:

1. A temporary execute program is created containing the "GENCOM" command and its related command and file skeletons.

2. CVTECO is used to edit the sincedate into the temporary execute file.

3. "HARDFILE" is toggled to update the sincedate file.
(TP.INDEX.SINCEDATE.PRT)

4. The temporary execute file is executed to create a temporary list of parts filed since the last update.

5. "CD3PRTS" is edited out of the temporary parts list and replaced with "*   ".

6. The temporary index is next edited into TP.ARCHIVE.INDEX.

7. The two temporary files are deleted.

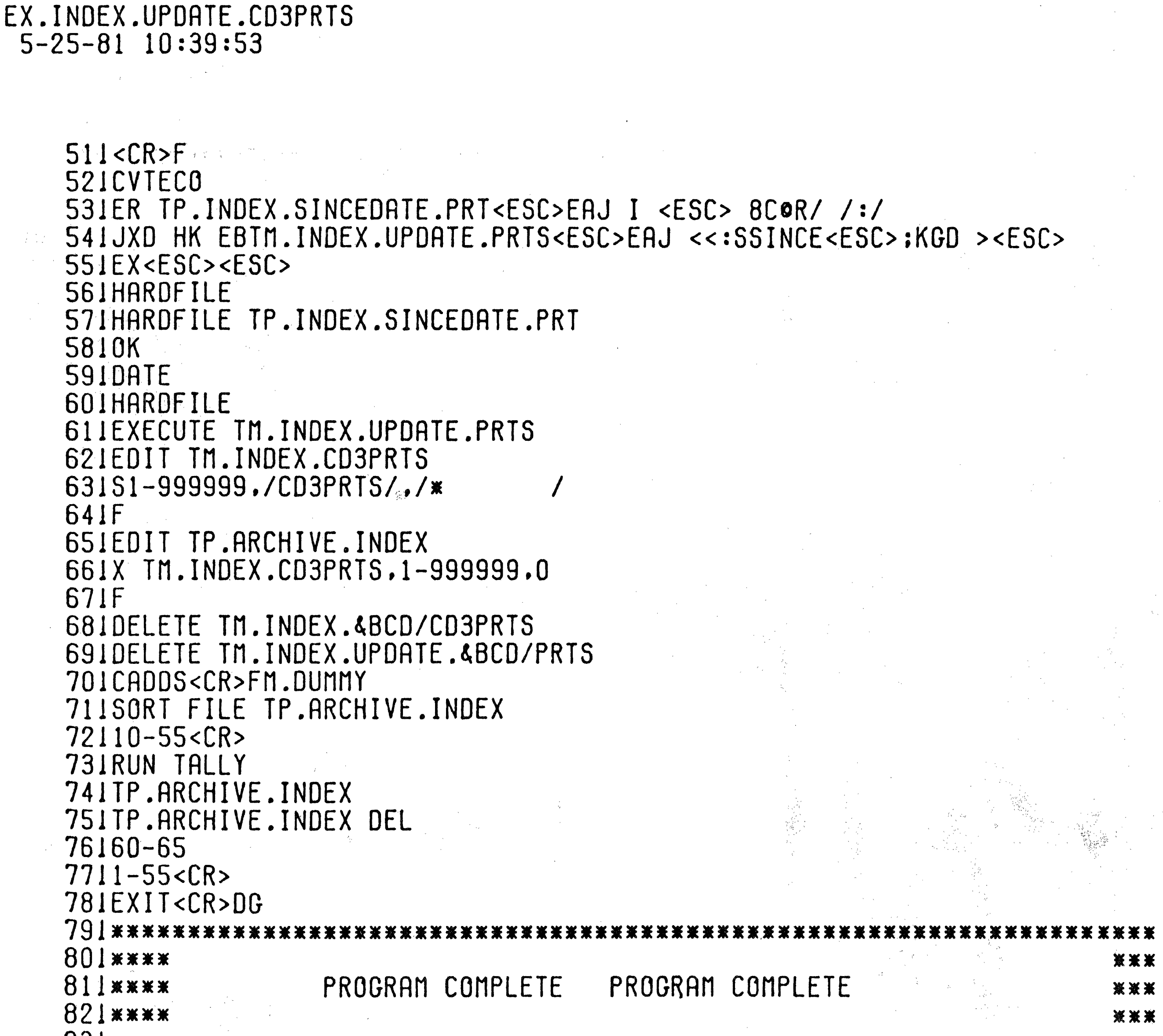
8. TP.ARCHIVE.INDEX is sorted and tallied to eliminate duplicate file names.

9. The program finishes by exiting from CADDS.

A copy of this program can be found on page 463 and 464 of this section.

EX.INDEX.UPDATE.CD3PRTS
5-25-81 10:39:53

```
<********************************************************************
<***                                                              ***
<***          FILE:        EX.INDEX.UPDATE.CD3PRTS
<***          PROGRAMMER:  RONALD SMITH                           ***
<***          DATE:        14 OCTOBER 1980                        ***
<***                                                              ***
<********************************************************************
<*    PURPOSE-
<*       TO UPDATE THE MASTER CADDS PARTS INDEX (TP.ARCHIVE.INDEX)
<*       WITH THE NAMES OF THE PARTS CREATED SINCE EXECUTING THE
<*       UPDATE.
<*
<*    EXECUTED FROM-
<*                     EX.DUMP.0830
<*
<*    PROCEDURE-
<*       1)  A TEMPORARY EXECUTE PROGRAM IS CREATED CONTAINING THE
<*           GENCOM COMMAND AND ITS RELATED COMMAND AND FILE SKELETONS.
<*
<*       2)  CVTECO IS USED TO ENTER THE SINCE DATES INTO THE
<*           TEMPORARY FILES.
<*
<*       3)  HARDFILE IS USED TO UPDATE THE SINCEDATE.
<*           (TP.INDEX.SINCEDATE.PRT)
<*
<*       4)  THE TEMPORY EXECUTE FILE IS EXECUTED.
<*           (TM.INDEX.UPDATE.PRTS)
<*
<*       5)  'CD3PRTS' IS EDITED OUT AND REPLACED
<*           WITH '*      '.
<*
<*       6)  TM.INDEX.CD3PRTS IS INSERTED INTO TP.ARCHIVE.INDEX
<*
<*       7)  TP.ARCHIVE.INDEX IS SORTED AND TALLIED TO
<*           ELIMINATE DUPLICATE TITLES.
<*
<*       8)  THE TWO TEMPORY FILES ARE DELETED.
<*
<*
<*    REVISIONS-
<*       R1  25MAY81    UPDATED FOR CGOS 200/10 O/S
<*
<*
<********************************************************************
EDIT TM.INDEX.UPDATE.PRTS,N
I
GENCOM TM.INDEX.CD3PRTS
<<CR>$1,-2,-1$ <<CR>
CD3PRTS/COM/NLEV,TYPE=0, SINCE  1- 1-80:12:00:00
<<CR><<CR>F
```



```
EX.INDEX.UPDATE.CD3PRTS
 5-25-81 10:39:53


    51|<CR>F
    52|CVTECO
    53|ER TP.INDEX.SINCEDATE.PRT<ESC>EAJ I <ESC> 8C@R/ /:/
    54|JXD HK EBTM.INDEX.UPDATE.PRTS<ESC>EAJ <<:SSINCE<ESC>;KGD ><ESC>
    55|EX<ESC><ESC>
    56|HARDFILE
    57|HARDFILE TP.INDEX.SINCEDATE.PRT
    58|OK
    59|DATE
    60|HARDFILE
    61|EXECUTE TM.INDEX.UPDATE.PRTS
    62|EDIT TM.INDEX.CD3PRTS
    63|S1-999999,/CD3PRTS/,/*        /
    64|F
    65|EDIT TP.ARCHIVE.INDEX
    66|X TM.INDEX.CD3PRTS,1-999999,0
    67|F
    68|DELETE TM.INDEX.&BCD/CD3PRTS
    69|DELETE TM.INDEX.UPDATE.&BCD/PRTS
    70|CADDS<CR>FM.DUMMY
    71|SORT FILE TP.ARCHIVE.INDEX
    72|10-55<CR>
    73|RUN TALLY
    74|TP.ARCHIVE.INDEX
    75|TP.ARCHIVE.INDEX DEL
    76|60-65
    77|1-55<CR>
    78|EXIT<CR>DG
    79|**********************************************************************
    80|****                                                               ***
    81|****          PROGRAM COMPLETE   PROGRAM COMPLETE                  ***
    82|****                                                               ***
    83|**********************************************************************
```

The master index should be checked periodically to assure that it contains (1) the names of the parts that are on disc, or (2) that names of parts that have been deleted and not archived have been removed from the disc. This check can be done by executing an execute program (EX.INDEX.STATUS.CD3PRTS) that lets the system make a comparison between a list of parts that should be on disc and a list of parts that actually are on disc.

The program first generates a list of the parts currently on disc. It then generates a list of parts that are supposed to be on the disc by extracting a list of all the parts without the tape number and savfil numbers from the master index. These lists are then edited so that they have identical formats. The final step uses the system level command "COMPARE" to list and report any discrepancies between the two files.

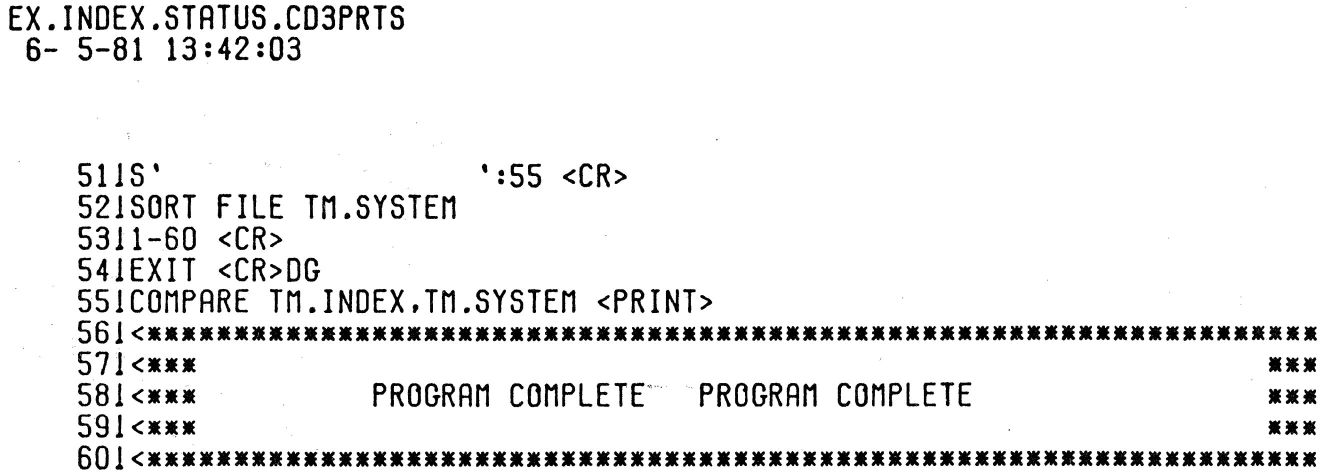
NOTE: The techniques used in the attached program duplicate those in a paper presented by David A. Quinn, Supervisor, Computer Aided Design; G.T.E. Lighting Products; Estes Street; Ipswich, Mass., at the Second Annual International Computervision User Conference. I have changed the merge key elements and the transfer elements in the merge file command and the column position numbers in the sort file command to suit the format of our files.

Once executed, the program does not require any further interaction with the operator.

A copy of this program can be found on page 465 of this section.

EX.INDEX.STATUS.CD3PRTS
6- 5-81 13:42:03

```
<*************************************************************
<***                                                       ***
<***          FILE:        EX.INDEX.STATUS.CD3PRTS
<***          PROGRAMMER:  RONALD SMITH                    ***
<***          DATE:        15 JAN 1981                     ***
<***                                                       ***
<*************************************************************
<*
<*    PURPOSE:
<*          TO SCAN AN INDEX OF PART NAMES (TP.ARCHIVE.INDEX).
<*          DETERMINE WHICH PARTS ARE SUPPOSED TO BE ON
<*                THE SYSTEM.
<*          COMPARE THAT LIST WITH WHAT ACTUALLY IS ON THE
<*                SYSTEM AND NOTIFY OF THE DIFFERENCES.
<*
<*    PROCEDURE:
<*          EXECUTE EX.INDEX.STATUS.CD3PRTS WHICH
<*                1.   CREATES A LIST OF PARTS ON THE SYSTEM
<*                     USING GENCOM.
<*                2.   SEARCHES THE INDEX FOR PARTS SUPPOSED
<*                     TO BE ON THE SYSTEM.
<*                3.   COMPARES BOTH &BCD FILES USING COMPARE.
<*
<*
<*    REVISIONS:
<*          R1  25MAY81    UPDATED FOR CGOS 200/10 O/S
<*
<*
<*************************************************************
DELETE TM.&BCD/SYSTEM,INDEX,TMP
GENCOM TM.SYSTEM <ECHO>
||||||||||||||||||||<CR>
$2,-2,$<CR>
CD3PRTS/COM/NLEV<CR><CR><CR>F
HARDFILE
HARDFILE TM.INDEX
EDIT TP.ARCHIVE.INDEX <ECHO>
T1
L,/*       /
Q
HARDFILE
EDIT TM.INDEX <ECHO>
S1-2,/*/,//
F
COPY TM.&BCD.INDEX,TM.&BCD.TMP
CADDS <CR> TEMPFILE
MERGE FILE TM.TMP PAIR BLNK
TM.INDEX PAIR BLNK
1 <CR>
17-70:1-54
```



```
EX.INDEX.STATUS.CD3PRTS
 6- 5-81 13:42:03


    51|S'                      ':55 <CR>
    52|SORT FILE TM.SYSTEM
    53|1-60 <CR>
    54|EXIT <CR>DG
    55|COMPARE TM.INDEX,TM.SYSTEM <PRINT>
    56|<**********************************************************************
    57|<***                                                                ***
    58|<***           PROGRAM COMPLETE   PROGRAM COMPLETE                 ***
    59|<***                                                                ***
    60|<**********************************************************************
```

Section Four
Locating an Archived Part

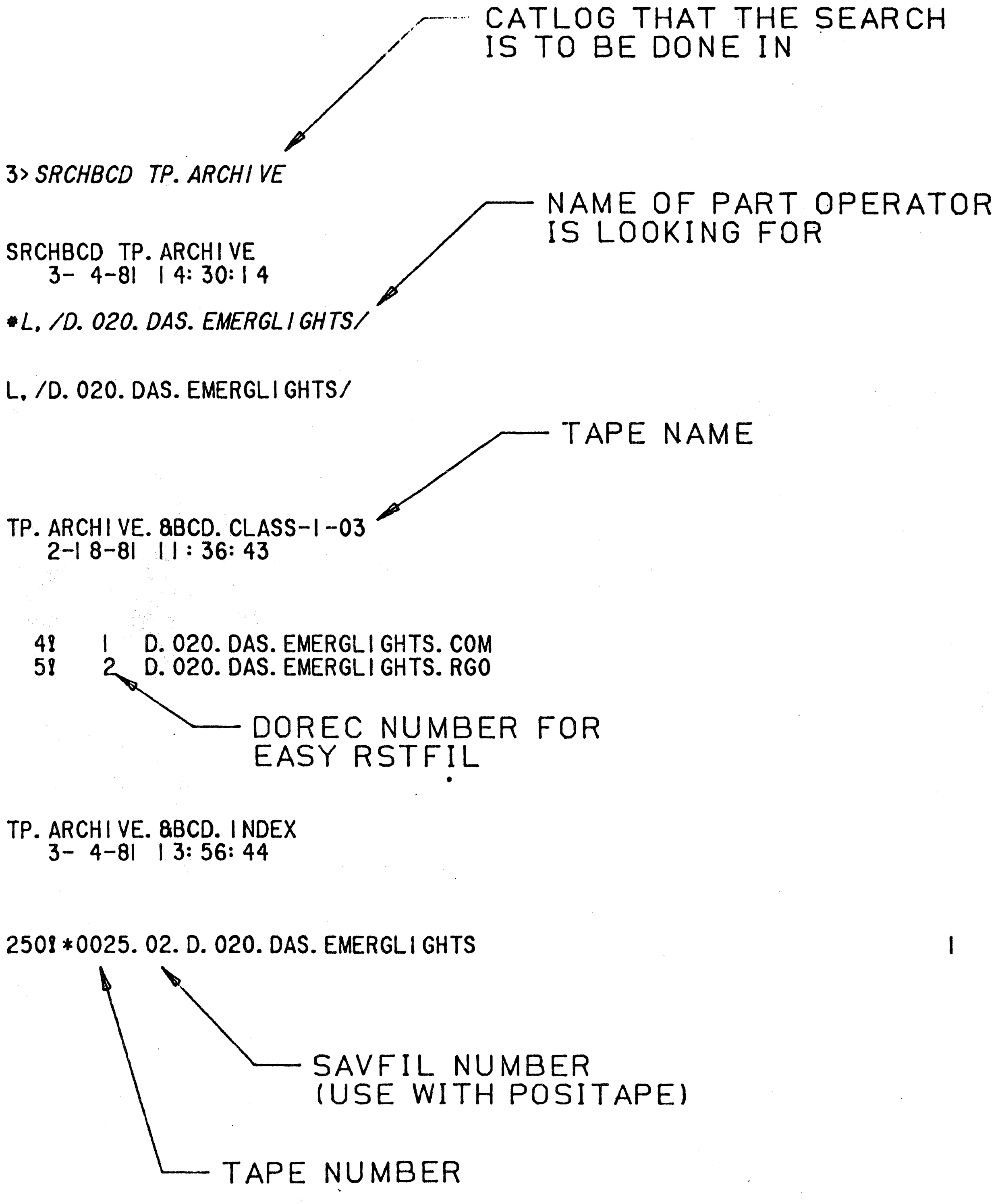
As mentioned in the introduction, one of the functions of the archiving system is to be able to locate any CADDS part that is on disc or archive tape as quickly as possible. This can be archived by using the system command "SRCHBCD". The procedure for the search is as follows:

```
n> SRCHBDC  TP.ARCHIVE ↲
#L,/(Part name)/ ↲
```

An example of a typical search will be found on the next page.

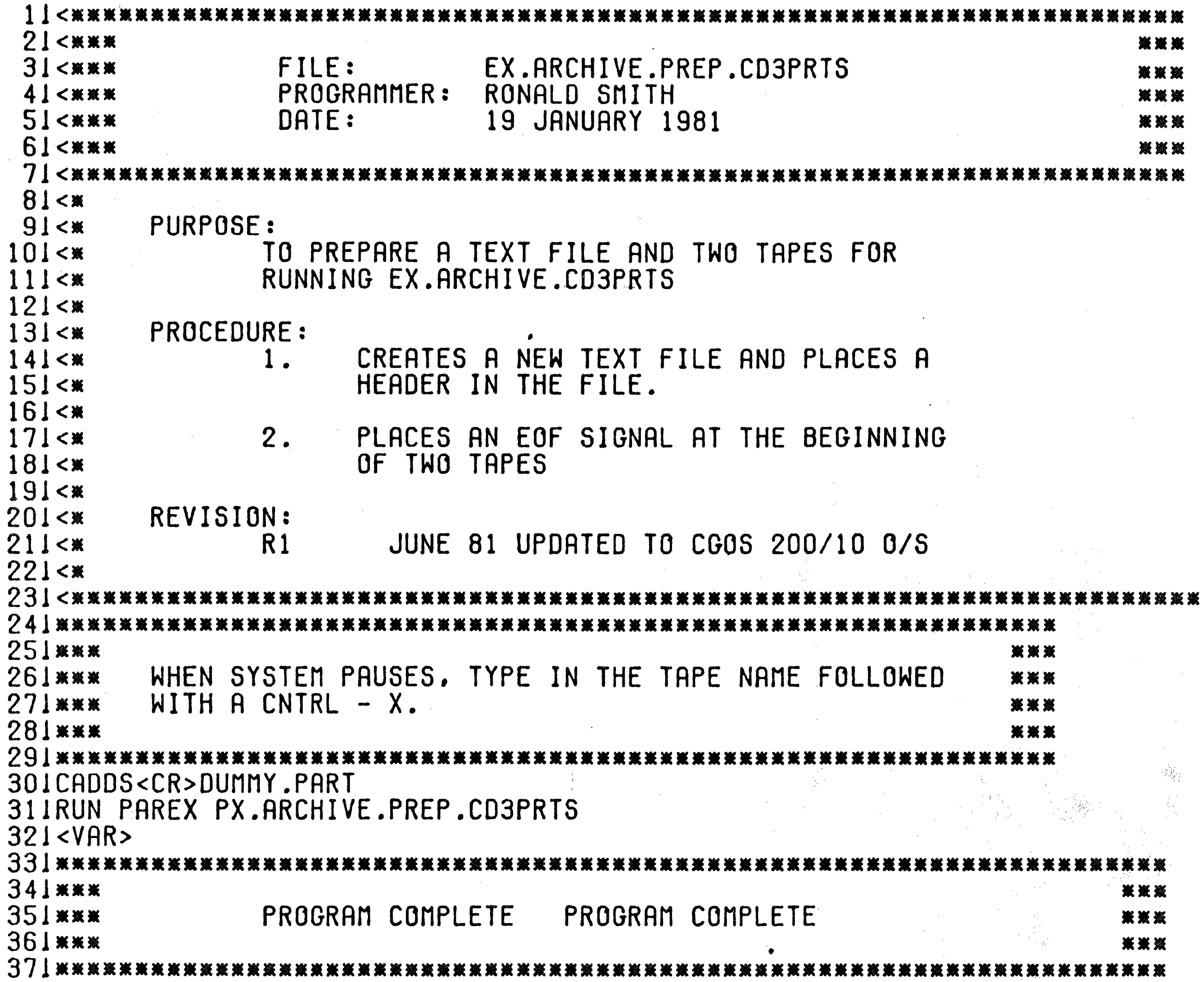
THIS IS AN EXAMPLE OF A TYPICAL SEARCH FOR A CADDS PART.

NOTE:
ITALICIZED TEXT INDICATES OPERATOR INPUT

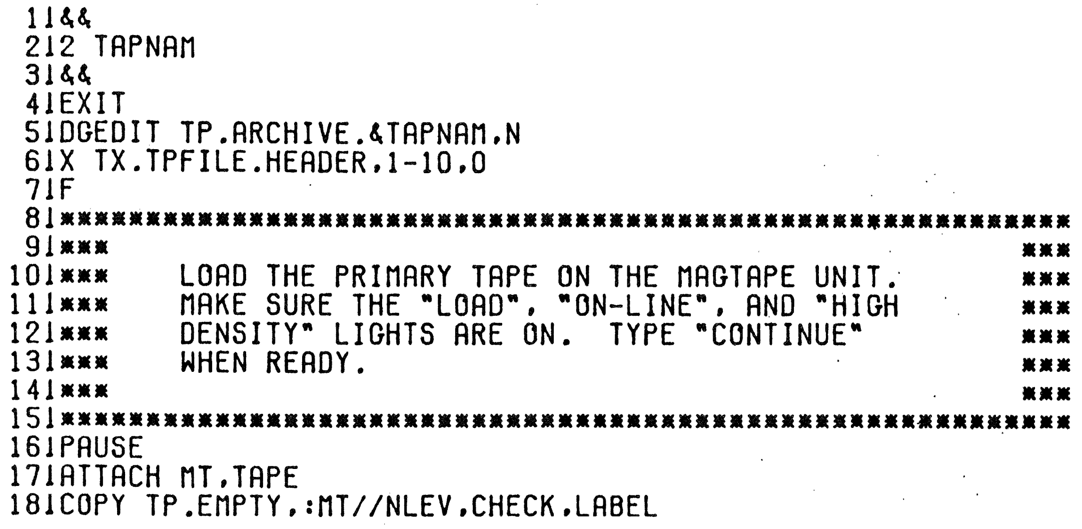


```
                                  CATLOG THAT THE SEARCH
                                  IS TO BE DONE IN

3>SRCHBCD TP.ARCHIVE
                                     NAME OF PART OPERATOR
SRCHBCD TP.ARCHIVE                   IS LOOKING FOR
   3- 4-81  14:30:14

*L,/D.020.DAS.EMERGLIGHTS/

L,/D.020.DAS.EMERGLIGHTS/

                                      TAPE NAME

TP.ARCHIVE.&BCD.CLASS-1-03
   2-18-81  11:36:43


  4!    1   D.020.DAS.EMERGLIGHTS.COM
  5!    2   D.020.DAS.EMERGLIGHTS.RGO

                       DOREC NUMBER FOR
                       EASY RSTFIL

TP.ARCHIVE.&BCD.INDEX
   3- 4-81  13:56:44


250!*0025.02.D.020.DAS.EMERGLIGHTS                               1


                          SAVFIL NUMBER
                          (USE WITH POSITAPE)


                     TAPE NUMBER
```

APPENDIX



```
<**********************************************************************
<***                                                                ***
<***          FILE:        EX.ARCHIVE.PREP.CD3PRTS                  ***
<***          PROGRAMMER:  RONALD SMITH                             ***
<***          DATE:        19 JANUARY 1981                          ***
<***                                                                ***
<**********************************************************************
<*
<*    PURPOSE:
<*           TO PREPARE A TEXT FILE AND TWO TAPES FOR
<*           RUNNING EX.ARCHIVE.CD3PRTS
<*
<*    PROCEDURE:
<*           1.    CREATES A NEW TEXT FILE AND PLACES A
<*                 HEADER IN THE FILE.
<*
<*           2.    PLACES AN EOF SIGNAL AT THE BEGINNING
<*                 OF TWO TAPES
<*
<*    REVISION:
<*           R1      JUNE 81 UPDATED TO CGOS 200/10 O/S
<*
<***********************************************************************
***************************************************************
***                                                         ***
***   WHEN SYSTEM PAUSES, TYPE IN THE TAPE NAME FOLLOWED    ***
***   WITH A CNTRL - X.                                     ***
***                                                         ***
***************************************************************
CADDS<CR>DUMMY.PART
RUN PAREX PX.ARCHIVE.PREP.CD3PRTS
<VAR>
**********************************************************************
***                                                                ***
***          PROGRAM COMPLETE   PROGRAM COMPLETE                   ***
***                                                                ***
**********************************************************************
```



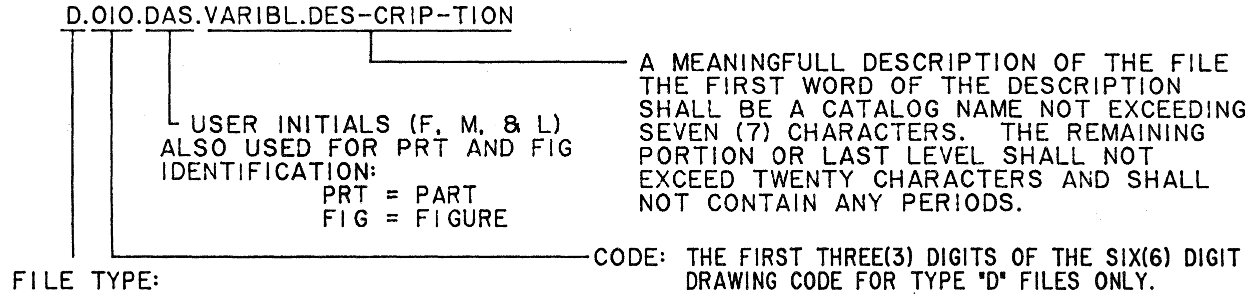
```
&&
2 TAPNAM
&&
EXIT
DGEDIT TP.ARCHIVE.&TAPNAM,N
X TX.TPFILE.HEADER,1-10,0
F
***********************************************************
***                                                     ***
***    LOAD THE PRIMARY TAPE ON THE MAGTAPE UNIT.       ***
***    MAKE SURE THE "LOAD", "ON-LINE", AND "HIGH       ***
***    DENSITY" LIGHTS ARE ON.  TYPE "CONTINUE"         ***
***    WHEN READY.                                      ***
***                                                     ***
***********************************************************
PAUSE
ATTACH MT,TAPE
COPY TP.EMPTY,:MT//NLEV,CHECK,LABEL
```

```
**************************************************************
***                                                        ***
***    LOAD THE BACK-UP TAPE ON THE MAGTAPE UNIT.          ***
***    MAKE SURE THE "LOAD", "ON-LINE", AND "HIGH          ***
***    DENSITY" LIGHTS ARE ON.  TYPE "CONTINUE"            ***
***    WHEN READY.                                         ***
***                                                        ***
**************************************************************
PAUSE
COPY TP.EMPTY,:MT//NLEV,CHECK,LABEL
DETACH MT
```

FILE NAMING CONVENTION FOR CD3PRTS

- CLASS I

FOR TRAINING,SCRATCH,DRAWINGS,AND LIBRARY PARTS AND FIGURES



D.010.DAS.VARIBL.DES-CRIP-TION

A MEANINGFULL DESCRIPTION OF THE FILE THE FIRST WORD OF THE DESCRIPTION SHALL BE A CATALOG NAME NOT EXCEEDING SEVEN (7) CHARACTERS. THE REMAINING PORTION OR LAST LEVEL SHALL NOT EXCEED TWENTY CHARACTERS AND SHALL NOT CONTAIN ANY PERIODS.

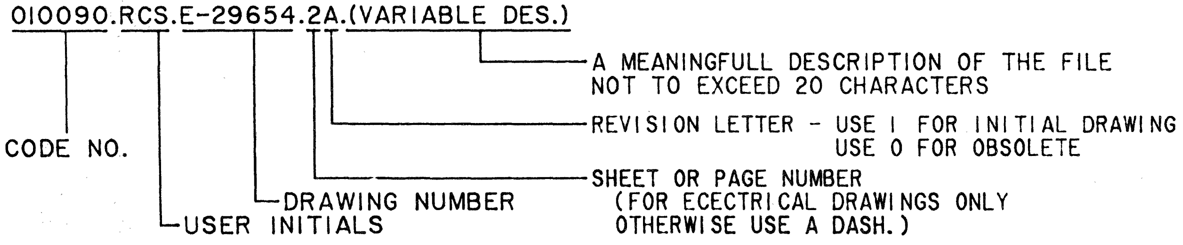
USER INITIALS (F, M, & L)
ALSO USED FOR PRT AND FIG
IDENTIFICATION:
PRT = PART
FIG = FIGURE

CODE: THE FIRST THREE(3) DIGITS OF THE SIX(6) DIGIT DRAWING CODE FOR TYPE "D" FILES ONLY.
"CHT" FOR CHARTS
"GRF" FOR GRAPHS
"ART" FOR ART WORK
"XXX" FOR FILES NOT REQUIRING A CODE
"DIA" FOR DIAGRAMS
"FOR" FOR FORMS

FILE TYPE:
T = TRAINING
S = SCRATCH
D = FORMAL DRAWING
N = NON-PART
M = MECHANICAL PARTS (LIBRARY)
E = ELEC. SCHEMATIC FIG. LIBRARY
P = PRINTED CIRCUIT BOARD FIG. LIBRARY
W = WIRING DIAGRAM FIG. LIBRARY
O = MACHINE PART OUTLINES

- CLASS II

FOR APPROVED FORMAL DRAWINGS



010090.RCS.E-29654.2A.(VARIABLE DES.)

A MEANINGFULL DESCRIPTION OF THE FILE NOT TO EXCEED 20 CHARACTERS

REVISION LETTER - USE I FOR INITIAL DRAWING
USE O FOR OBSOLETE

SHEET OR PAGE NUMBER
(FOR ECECTRICAL DRAWINGS ONLY
OTHERWISE USE A DASH.)

CODE NO.

DRAWING NUMBER

USER INITIALS

AN ALTERNATE ACCOUNTING PACKAGE

Curt Strobel
Southern California Edison
100 Long Beach Boulevard
Long Beach, California 90801
213-435-1121 x325

TABLE OF CONTENTS

ABSTRACT

This paper describes the accounting package used at Southern California Edison in place of Computervision's. The Fortran program is utilized in a closed shop atmosphere on multiple systems.

Individual station/operator information is captured on standard Computervision text files. These files are then summarized and used as input to a performance tracking program on an IBM mainframe.

INTRODUCTION

Southern California Edison is an electric utility serving some 3,500,000 meters in Southern California. Its service territory covers approximately 50,000 square miles, of which 25,000 is mapped. Edison first acquired a Computervision system in 1976 for testing, evaluation, and mapping. It was expanded to the current system in 1978.

The systems were acquired primarily for the purpose of digitizing and regularly maintaining some 100,000 maps, of which some 30,000 are digitized. In addition, other engineering drafting applications are phased in after testing and evaluation. The systems are operated on a 3 shift, 5 day a week basis employing approximately 48 operators. Accurate and complete records are kept on all activities to monitor costs, truput, employee performance, and control scheduling. Prior to the implementation of this program we judiciously used the Computervision-supplied accounting package. Some shortcomings in the package caused us to develop this substitute. We have found it more flexible, more reliable, easier to customize to our needs, and faster operation-wise.

OPERATOR COMMANDS

The SCE accounting package "PUTACT" is a Fortran program compiled in an ONLY.PUTACT file. The package is entered by typing in *PUTACT* at system level. The system responds with the message: ENTER NAME, NUMBER, (PART): The operator can then respond with either one of the following replies:

1. His name in the standard SCE format, a comma, a number between 1 and 20, another comma, and optionally up to 37 characters. This last entry is usually the part name with special codes for the specific activities. The system will verify the name against a text file containing the authorized names. This is not done so much for security as to assure that the operator doesn't forget this name during the shift.

The number is also validated and, it if is acceptable, the entry is generated in the appropriate text file. If there are any errors, then the program will return to the ENTER. . . step. The previous accounting entry and the new one will be outputted to the station's CRT. This enables the operator to re-verify his entry and correct it if necessary.

2. The other reply is "CF..". This feature enables the operator to close the previous record without opening a new one. This effectively puts the file into a no operation mode and is used when the task is idle for lunch, breaks, and shift changes.

The entry outputted to the operator will only be the entry closed.

When the operator again logs in after using a "CF..", the system will create a "GENERATED RECORD" entry to cover the time lost.

SYSTEM COMMANDS

Two commands are available for rearranging the individual accounting files into a summary format.

The first one is "CL..". This command will combine designated files into a combined file "&ACT.ddd.TASKXX". A "QUIT" will terminate the command.

The second one is "TL...". This will take a sorted file and generate a summary file "&SUM.ddd.TASKXX" containing the number of entries and total time for each entry. In addition, totaling by task number, activity number, and system number is made. A "QUIT" will also terminate this command.

By using these two commands weekly, monthly, and yearly, summaries can be easily obtained.

FILE DESCRIPTION

The following is a layout of the individual accounting entry. The configuration number is what is used for the CGP number. A new file is automatically opened at the start of our new workday, which is 11:00 p.m. This enables us to process each previous day's entries without disturbing the current accounting package or logging everybody out.

AAAAAA,BB,CCC ---- (37 characters) CCC,-S,P,TT,------,DDD,-HH,MM.M(-EEEEE.E)

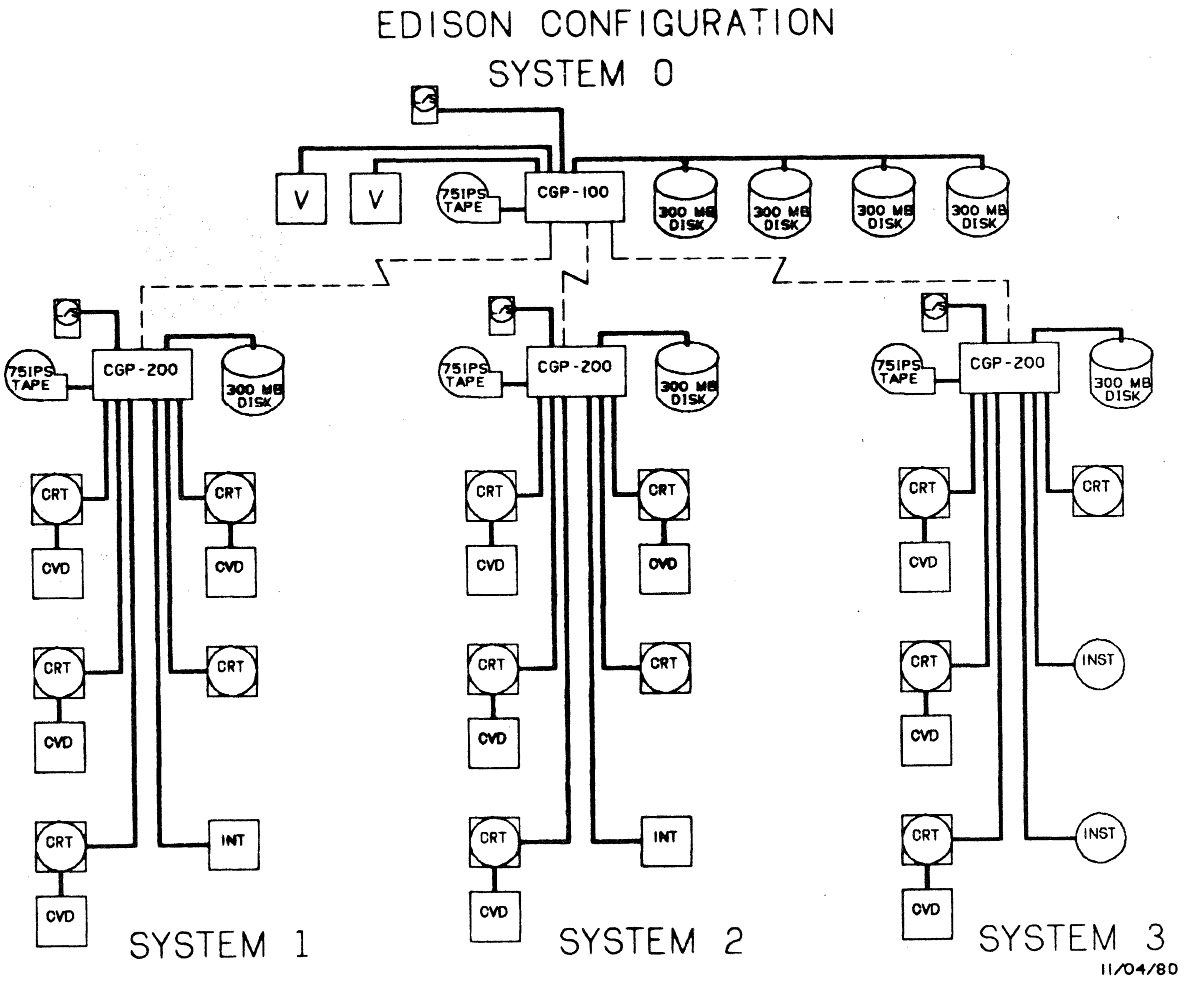
A — Operator's name—6 characters
B — Activity number—2 digits (1-20)
C — Part name or activity code—37 characters

S — Shift—1 digit, graveyard = 1
Note shift changes at 7:00 a.m., 7:00 p.m., and 11:00 p.m.
The new day (file) starts at 11:00 p.m.
P — CGP number—1 digit
T — Task number—2 digits
D — Date—3 digits
H — Hour entry created—2 digits
M — Minute entry created in tenths of minutes—3 digits
E — Elapsed time in tenths of minutes—6 digits

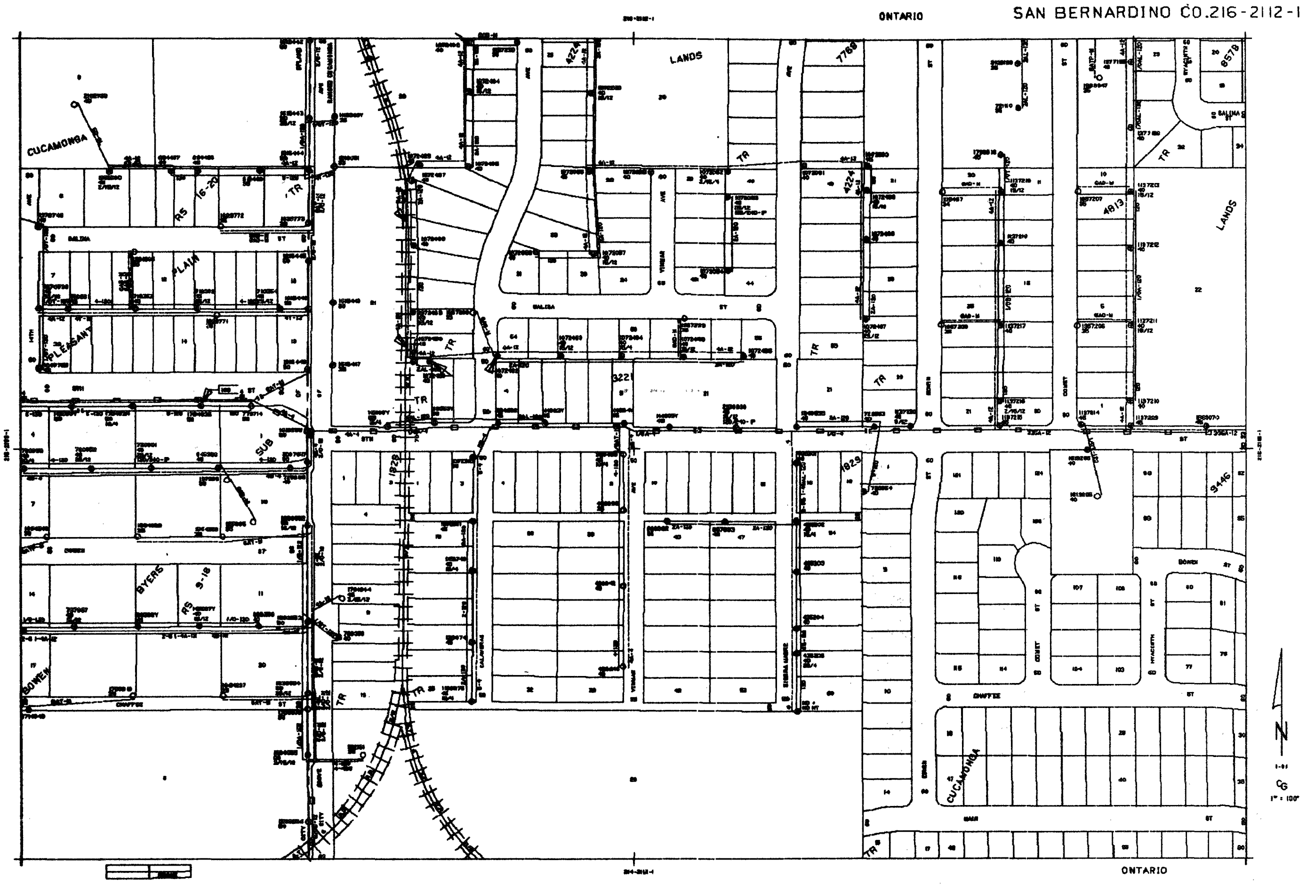
CONCLUSION

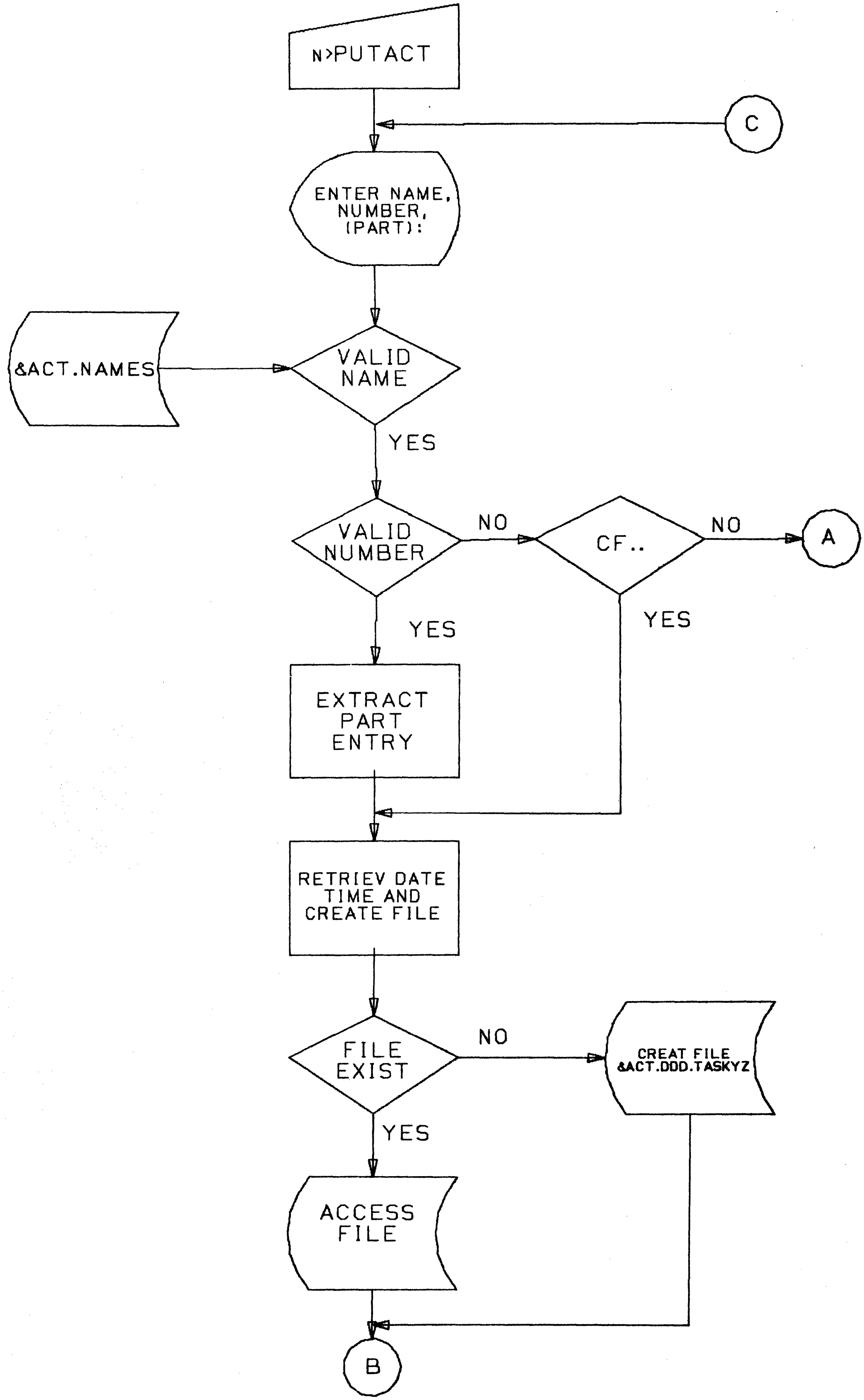
This accounting package, although tailored somewhat to our needs, is still fairly flexible and general. It is easy and quick for the operators to use and they get an immediate feedback from it, so they do judiciously use it. The maintaining of individual files for each task provides complete flexibility and assures that only the minimum of data may be lost. We have had no problems with file security and do not use Computervision's authorize or accounting options.

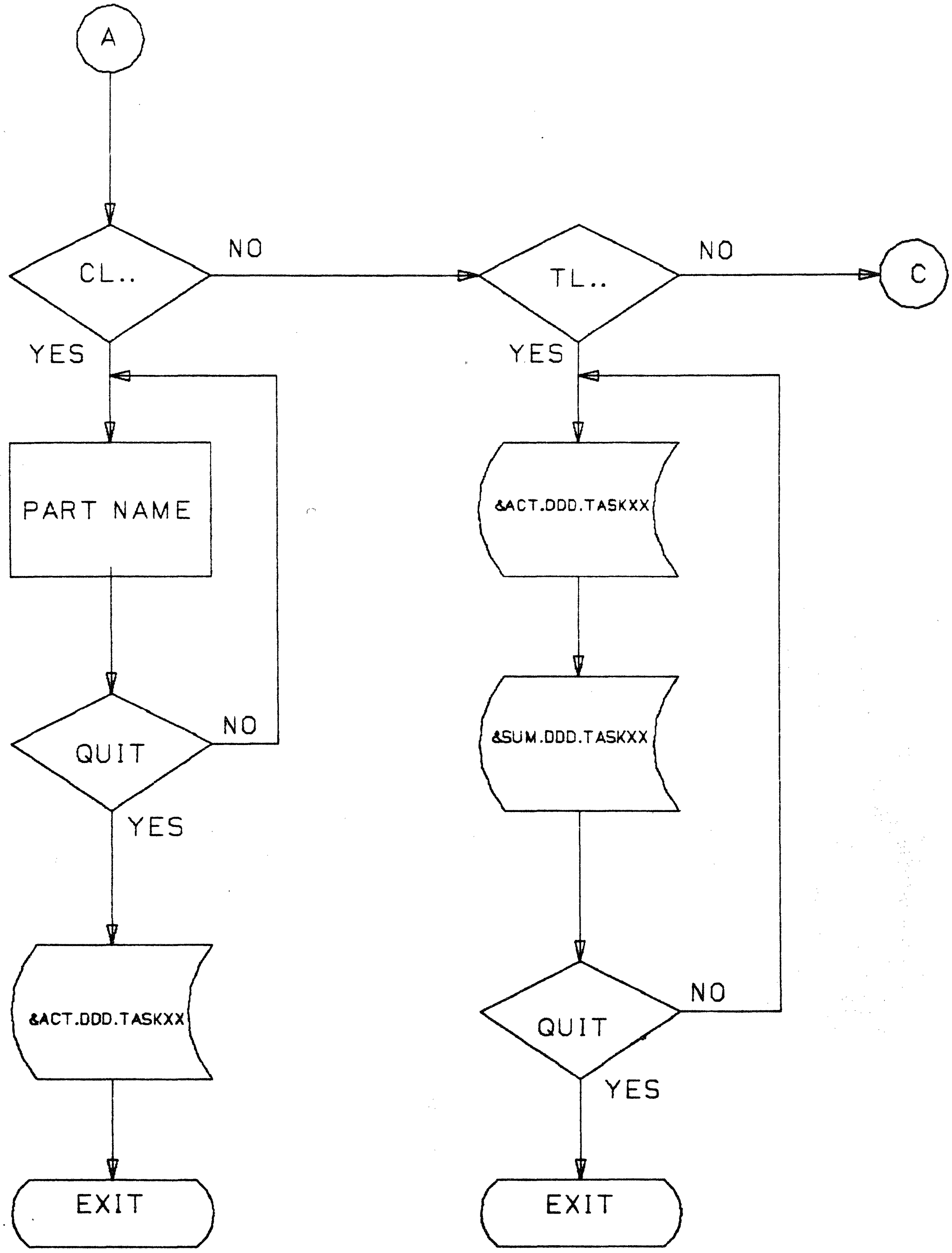
Our future plans are to do more checking and validating of the activity code entries to help ensure more valid ones. And to work this package into the current Computervision file security and accounting schemes.

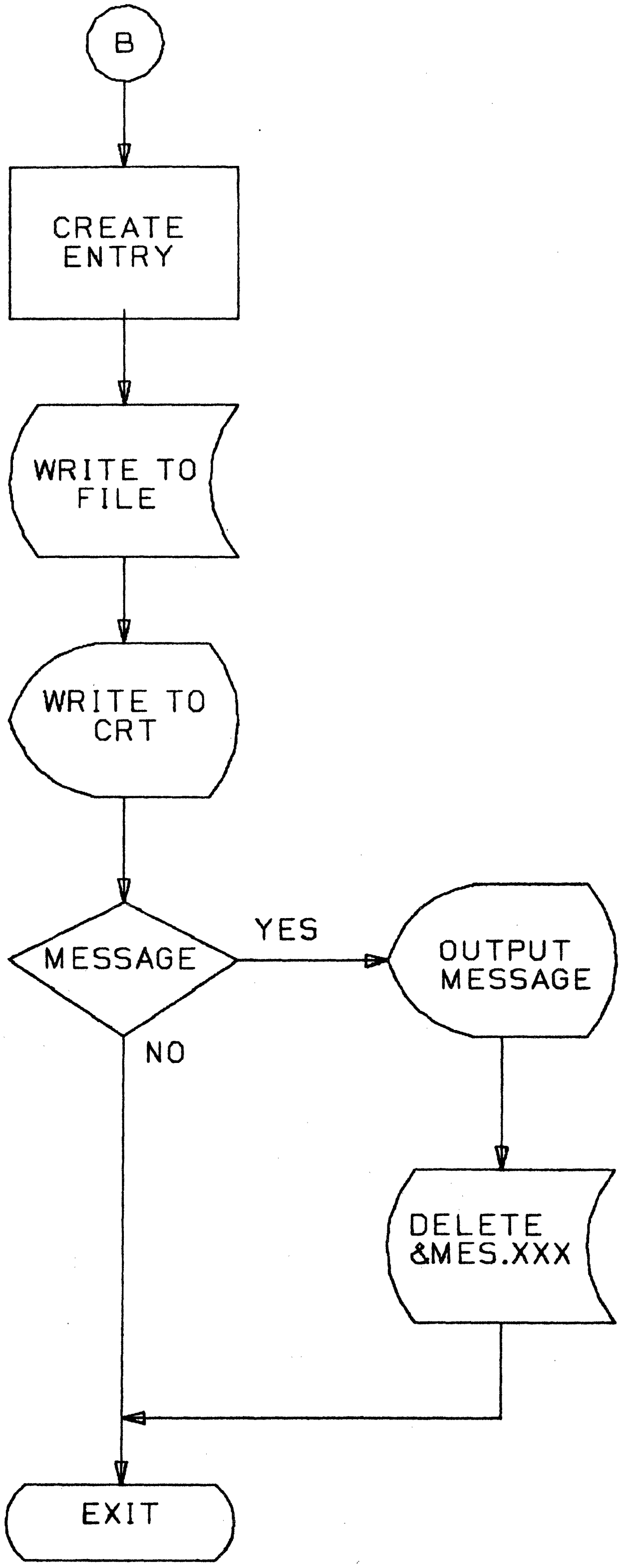


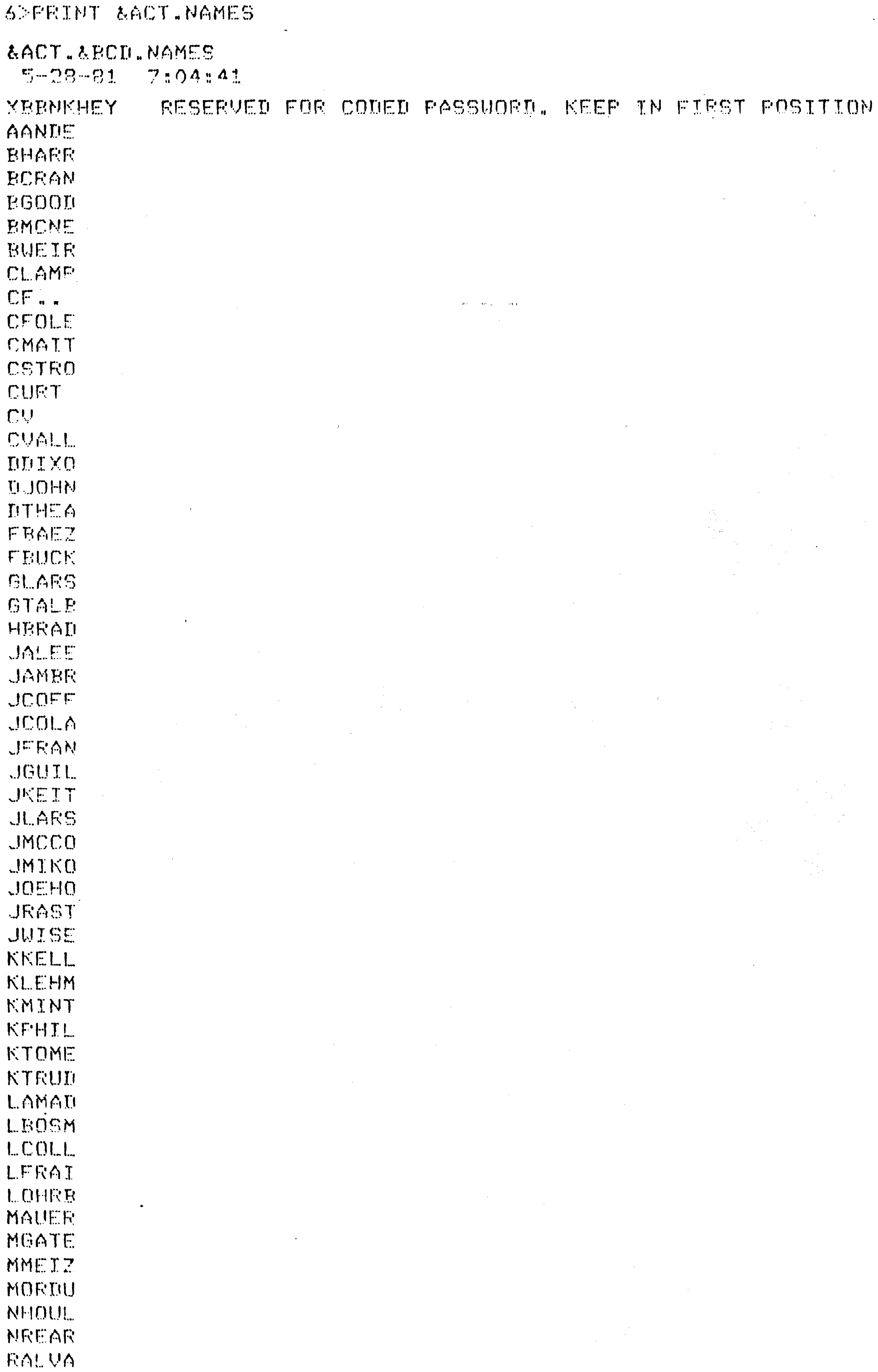
SAN BERNARDINO CO.216-2112-1




N>PUTACT
C
ENTER NAME, NUMBER, (PART):
&ACT.NAMES
VALID NAME
YES
VALID NUMBER
NO
CF..
NO
A
YES
YES
EXTRACT PART ENTRY
RETRIEV DATE TIME AND CREATE FILE
FILE EXIST
NO
CREAT FILE &ACT.DDD.TASKYZ
YES
ACCESS FILE
B


A
CL..
NO
TL..
NO
C
YES
YES
PART NAME
&ACT.DDD.TASKXX
&SUM.DDD.TASKXX
QUIT
NO
YES
&ACT.DDD.TASKXX
QUIT
NO
YES
EXIT
EXIT

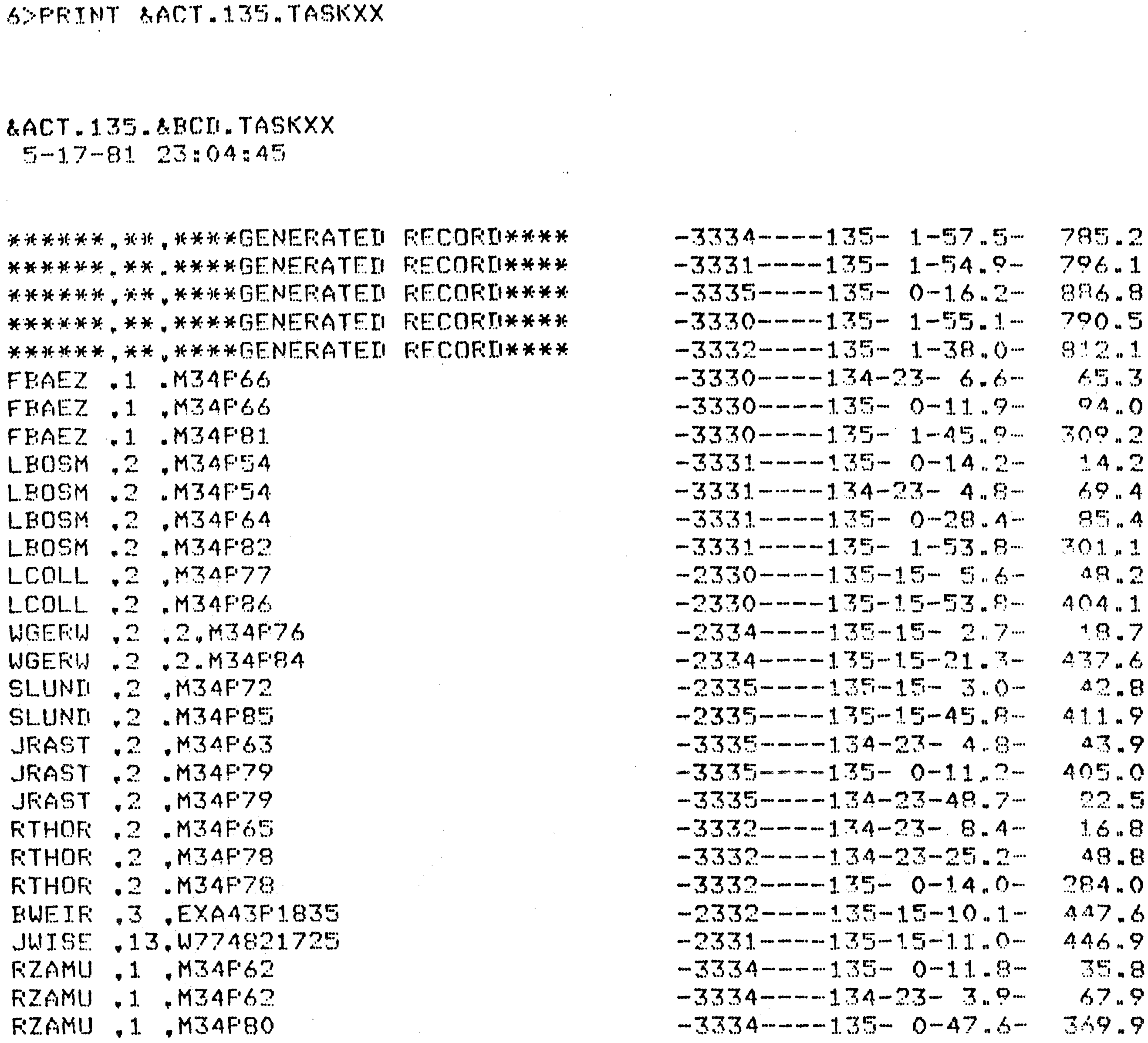

B
CREATE ENTRY
WRITE TO FILE
WRITE TO CRT
MESSAGE
YES
NO
OUTPUT MESSAGE
DELETE &MES.XXX
EXIT



```
6>PRINT &ACT.NAMES

&ACT.&BCD.NAMES
 5-28-81  7:04:41
XBBNKHEY   RESERVED FOR CODED PASSWORD. KEEP IN FIRST POSITION
AANDE
BHARR
BCRAN
BGOOD
BMCNE
BWEIR
CLAMP
CF..
CFOLE
CMAIT
CSTRO
CURT
CV
CVALL
DDIXO
DJOHN
DTHEA
FBAEZ
FBUCK
GLARS
GTALB
HBRAD
JALEE
JAMBR
JCOFF
JCOLA
JFRAN
JGUIL
JKEIT
JLARS
JMCCO
JMIKO
JOEHO
JRAST
JWISE
KKELL
KLEHM
KMINT
KPHIL
KTOME
KTRUD
LAMAD
LBOSM
LCOLL
LFRAI
LOHRB
MAUER
MGATE
MMEIZ
MORDU
NHOUL
NREAR
RALVA
```

```
6>
6>
6>
6>   THIS IS AN EXAMPLE OF ADDING A NEW ENTRY
??
6>
6>
6>PUTACT
ENTER NAME.NUMBER,(PART): CSTRO,12,THIS IS A NEW ENTRY...
 CSTRO .13.THIS IS THE NEXT ENTRY...     -1336----148-12-26.4-    0.2
 CSTRO ,12,THIS IS A NEW ENTRY...        -1336----148-12-26.6--------
6>
6>
6>
6>  THIS IS AN EXAMPLE OF CLOSING AN ENTRY USING CF..
??
6>
6>
6>PUTACT
ENTER NAME.NUMBER,(PART): CF..
 CSTRO ,12,THIS IS A NEW ENTRY...        -1336----148-12-26.6-    0.1
6>
6>
6>
6>  THIS IS AN EXAMPLE OF ADDING AN ENTRY AFTER A CLOSED ONE
??
6>
6>
6>PUTACT
ENTER NAME.NUMBER,(PART): CSTRO,13,THIS IS THE NEXT ENTRY...
 ******,**,****GENERATED RECORD****      -1336----148-12-26.7-    0.1
 CSTRO ,13,THIS IS THE NEXT ENTRY...     -1336----148-12-26.8--------



==  LIST  ==



&ACT.135.&BCD.TASK30
 5-17-81 23:04:29


FBAEZ ,1 ,M34P66                         -3330----134-23- 6.6-   65.3
FBAEZ .1 .M34P66                         -3330----135- 0-11.9-   94.0
FBAEZ ,1 ,M34P81                         -3330----135- 1-45.9-  309.2
******,**.****GENERATED RECORD****       -3330----135- 1-55.1-  790.5
LCOLL ,2 ,M34P77                         -2330----135-15- 5.6-   48.2
LCOLL .2 .M34P86                         -2330----135-15-53.8-  404.1
```

```
&ACT.135.&BCD.TASK31
 5-17-81 23:04:30


LBOSM ,2 ,M34P54                       -3331----134-27- 4.8-   69.4
LBOSM .2 .M34P54                       -3331----135- 0-14.2-   14.2
LBOSM ,2 ,M34P64                       -3331----135- 0-28.4-   95.4
LBOSM .2 .M34P82                       -3331----135- 1-53.8-  301.1
******,**,****GENERATED RECORD****     -3331----135- 1-54.9-  796.1
JWISE ,13,W774821725                   -2331----135-15-11.0-  446.9


&ACT.135.&BCD.TASK32
 5-17-81 23:04:31


RTHOR ,2 ,M34P65                       -3332----134-27- 8.4-   16.8
RTHOR .2 .M34P78                       -3332----134-23-25.2-   48.8
RTHOR ,2 ,M34P78                       -3332----135- 0-14.0-  294.0
******,**,****GENERATED RECORD****     -3332----135- 1-38.0-  912.1
BWEIR ,3 ,EXA43P1835                   -2332----135-15-10.1-  447.6


&ACT.135.&BCD.TASK34
 5-17-81 23:04:32


RZAMU ,1 ,M34P62                       -3334----134-23- 3.9-   67.9
RZAMU .1 .M34P62                       -3334----135- 0-11.8-   35.8
RZAMU ,1 ,M34P80                       -3334----135- 0-47.6-  369.9
******,**,****GENERATED RECORD****     -3334----135- 1-57.5-  785.2
WGERJ ,2 ,2,M34P76                     -2334----135-15- 2.7-   18.7
WGERJ ,2 ,2.M34P84                     -2334----135-15-21.3-  437.6


&ACT.135.&BCD.TASK35
 5-17-81 23:04:33


JRAST ,2 ,M34P63                       -3335----134-23- 4.8-   43.9
JRAST ,2 ,M34P79                       -3335----134-23-48.7-   22.5
JRAST ,2 ,M34P79                       -3335----135- 0-11.2-  405.0
******,**,****GENERATED RECORD****     -3335----135- 0-16.2-  886.8
SLUND ,2 ,M34P72                       -2335----135-15- 3.0-   42.8
SLUND ,2 .M34P85                       -2335----135-15-45.8-  411.9
```



```
6>PRINT &ACT.135.TASKXX


&ACT.135.&BCD.TASKXX
 5-17-81 23:04:45


******,**,****GENERATED RECORD****      -3334----135- 1-57.5-  785.2
******.**.****GENERATED RECORD****      -3331----135- 1-54.9-  796.1
******,**,****GENERATED RECORD****      -3335----135- 0-16.2-  886.8
******,**,****GENERATED RECORD****      -3330----135- 1-55.1-  790.5
******,**,****GENERATED RECORD****      -3332----135- 1-38.0-  812.1
FBAEZ .1 .M34P66                        -3330----134-23- 6.6-   65.3
FBAEZ ,1 ,M34P66                        -3330----135- 0-11.9-   94.0
FBAEZ .1 .M34P81                        -3330----135- 1-45.9-  309.2
LBOSM ,2 ,M34P54                        -3331----135- 0-14.2-   14.2
LBOSM ,2 .M34P54                        -3331----134-23- 4.8-   69.4
LBOSM ,2 ,M34P64                        -3331----135- 0-28.4-   85.4
LBOSM .2 .M34P82                        -3331----135- 1-53.8-  301.1
LCOLL ,2 ,M34P77                        -2330----135-15- 5.6-   48.2
LCOLL ,2 .M34P86                        -2330----135-15-53.8-  404.1
WGERW ,2 ,2,M34P76                      -2334----135-15- 2.7-   18.7
WGERW ,2 ,2.M34P84                      -2334----135-15-21.3-  437.6
SLUND ,2 ,M34P72                        -2335----135-15- 3.0-   42.8
SLUND ,2 .M34P85                        -2335----135-15-45.8-  411.9
JRAST ,2 ,M34P63                        -3335----134-23- 4.8-   43.9
JRAST ,2 .M34P79                        -3335----135- 0-11.2-  405.0
JRAST ,2 ,M34P79                        -3335----134-23-48.7-   22.5
RTHOR ,2 .M34P65                        -3332----134-23- 8.4-   16.8
RTHOR ,2 ,M34P78                        -3332----134-23-25.2-   48.8
RTHOR ,2 .M34P78                        -3332----135- 0-14.0-  284.0
BWEIR ,3 ,EXA43P1835                    -2332----135-15-10.1-  447.6
JWISE ,13,W774821725                    -2331----135-15-11.0-  446.9
RZAMU ,1 ,M34P62                        -3334----135- 0-11.8-   35.8
RZAMU ,1 ,M34P62                        -3334----134-23- 3.9-   67.9
RZAMU ,1 ,M34P80                        -3334----135- 0-47.6-  369.9
```

```
6>PRINT

&SUM.135.&BCD.TASKXX
 5-17-81 23:04:45

******,**,****GENERATED RECORD****        -3334----   5----------  4070.7
FBAEZ .1 .M34P66                          -3330----   2----------   159.3
FBAEZ .1 ,M34P81                          -3330----   1----------   309.2
LBOSM .2 .M34P54                          -3331----   2----------    83.6
LBOSM .2 ,M34P64                          -3331----   1----------    85.4
LBOSM .2 .M34P82                          -3331----   1----------   301.1
LCOLL ,2 ,M34P77                          -2330----   1----------    48.2
LCOLL .2 .M34P86                          -2330----   1----------   404.1
WGERW ,2 ,2,M34P76                        -2334----   1----------    18.7
WGERW ,2 .2.M34P84                        -2334----   1----------   437.6
SLUND ,2 ,M34P72                          -2335----   1----------    42.8
SLUND .2 .M34P85                          -2335----   1----------   411.9
JRAST ,2 ,M34P63                          -3335----   1----------    43.9
JRAST .2 .M34P79                          -3335----   2----------   427.5
RTHOR ,2 ,M34P65                          -3332----   1----------    16.8
RTHOR ,2 .M34P78                          -3332----   2----------   332.8
BWEIR ,3 ,EXA43P1835                      -2332----   1----------   447.6
JWISE ,13.W774821725                      -2331----   1----------   446.9
RZAMU ,1 ,M34P62                          -3334----   2----------   103.7
RZAMU ,1 .M34P80                          -3334----   1----------   369.9
NUMBER   1---     15-42.1
NUMBER   2---     44-14.4
NUMBER   3---      7-27.6
NUMBER  13---      7-26.9
SYSTEM   3---    142-41.7
GRAND     ---    142-41.7

6>PRINT

CGP.&BCD.GET-ACT
 5-27-81 23:09:47

* POWER-ON THE PRINTER ........TYPE CONTINUE WHEN READY...........
PAUSE
PUTACT
CL..
&ACT.147.TASK30
&ACT.147.TASK31
&ACT.147.TASK32
&ACT.147.TASK33
&ACT.147.TASK34
&ACT.147.TASK35
QUIT
SORTFILE &ACT.147.TASKXX/FIELD=1,46
PUTACT
TL..
&ACT.147.TASKXX
QUIT
TOF
PRINT &SUM.147.TASKXX//DIGITS=0,FILEHEAD=NO
TOF
```

```
                                           R/W ENGINEERING DIVISION STATUS REPORT
                             AUTOMATED DRAFTING SECTION    JOBS COMPLETED        PERIOD  4/ 1/81 TO  4/30/81
                                                           EMPLOYEE: JCOFF                                                      J.COFFEY
DIST.NO.    MAP NUMBER    NUMBER        DATE          DATE        SYSTEM HOURS        SYSTEM HOURS                  NON-SYSTEM  SYSTEM
                          OF MAPS       ASSIGNED      COMPLETED   THIS PERIOD           TO DATE                       HOURS     HRS/MAP


ANNEX MAPS
      39    ANXA 77 5        1          4/22/81       4/23/81         3.0                 3.0                        1.1       3.0
      39    ANXA 78 10       1          4/21/81       4/29/81         2.5                 2.5                        0.9       2.5
      39    ANXIS21          1          4/20/81       4/22/81         3.0                 3.0                        1.1       3.0
      39    ANXVOSS A        1          4/21/81       4/22/81         2.3                 2.3                        0.9       2.3
            ANX77 4          0          4/23/81       0/ 0/ 0         2.8                 2.8                        1.0       0.0
TOTAL                        4                                       10.9                10.9                        4.0       2.7

BASE MAPS
      39    B00269           6          4/14/81       4/16/81         0.5                 0.5                        0.2       0.1
      30    B00273           1          4/27/81       4/28/81         3.9                 3.9                        1.4       3.9
      49    B00279           1          4/20/81       4/22/81         0.8                 0.8                        0.3       0.8
      36    B00280           5          4/27/81       4/28/81         2.5                 2.5                        0.9       0.5
      47    B00289           1          4/20/81       4/20/81         2.0                 2.0                        0.7       2.0
      47    B00290           1          4/17/81       4/20/81         0.3                 0.3                        0.1       0.3
      49    B00294           4          4/16/81       4/16/81         3.5                 3.5                        1.3       0.9
      30    B00295           4          4/17/81       4/20/81         2.5                 2.5                        0.9       0.6
      30    B00296           1          4/17/81       4/20/81         1.0                 1.0                        0.4       1.0
      26    B00302           1          4/17/81       4/20/81         1.1                 1.1                        0.4       1.1
      32    B00304           1          4/17/81       4/20/81         0.7                 0.7                        0.3       0.7
            B00310           0          4/27/81       0/ 0/ 0         0.7                 0.7                        0.3       0.0
      51    B00312           2          4/17/81       4/20/81         0.9                 0.9                        0.3       0.5
      32    B00317           1          4/17/81       4/20/81         0.3                 0.3                        0.1       0.3
      77    B00329           2          4/16/81       4/20/81         2.2                 2.2                        0.8       1.1
      30    B00338           1          4/14/81       4/16/81         1.7                 1.7                        0.6       1.7
      35    B00341           2          4/14/81       4/16/81         1.3                 1.3                        0.5       0.7
      40    B00347           2          4/17/81       4/20/81         1.3                 1.3                        0.5       0.6
      51    B00353           6          4/15/81       4/16/81         6.6                 6.6                        2.5       1.1
      39    B00383           2          4/28/81       4/29/81         5.9                 5.9                        2.2       2.9
TOTAL                       44                                       39.1                39.1                       14.5       0.9

EXHIBIT A MAPS
      33    EXAJ2112         1          4/23/81       4/27/81         1.2                 1.2                        0.4       1.2
      26    EXAK2578         1          4/23/81       4/27/81         3.3                 3.3                        1.2       3.3
TOTAL                        2                                        4.5                 4.5                        1.7       2.2

SV PLANNER MAPS
      46    TRALAANA         1          4/14/81       4/15/81         2.1                 2.1                        0.8       2.1
      79    TRCASTNO         1          4/24/81       4/27/81         5.5                 5.5                        2.1       5.5
            TRREGENTSP       0          4/24/81       0/ 0/ 0         0.3                 0.3                        0.1       0.0
            TR11209          0          4/28/81       0/ 0/ 0         5.0                 5.0                        1.9       0.0
            TR12322          0          4/30/81       0/ 0/ 0         1.2                 1.2                        0.4       0.0
            TR14731          0          4/30/81       0/ 0/ 0         1.5                 1.5                        0.6       0.0
      79    TR17294          1          4/14/81       4/15/81         2.3                 2.3                        0.9       2.3
            TR36075          0          4/27/81       0/ 0/ 0         1.7                 1.7                        0.6       0.0
      26    TR40765          1          1/ 2/81       4/30/81         4.7                 4.7                        1.8       4.7
TOTAL                        4                                       14.6                14.6                        5.5       3.7

CONVERSIONS
      76    6301611          1          4/ 9/81       4/15/81         2.5                 2.5                        0.9       2.5
      76    6301614          1          4/ 9/81       4/15/81         3.0                 3.0                        1.1       3.0
      76    6401596          4          4/ 2/81       4/15/81        29.1                29.1                       10.9       7.3
      76    6641608          1          3/16/81       4/ 9/81         0.0                 1.7                        0.6       1.7
      76    6641611          5          3/16/81       4/ 9/81         0.0                27.8                       10.4       5.6
      76    6661608          3          3/16/81       4/ 9/81        10.8                10.8                        4.0       3.6
      76    6661611          5          3/16/81       4/ 9/81         0.0                35.8                       13.4       7.2
TOTAL                       20                                       45.4               110.6                       41.4       5.5

TOTAL MAPS                  74                                      114.5 HRS           179.8 HRS                   67.1 HRS  2.4 AVERAGE
TOTAL TIME                                                          127.7 HRS           192.9 HRS                   72.0


                                    PERFORMANCE RECORD - SYSTEM HOURS PER MAP
                                  JAN  FEB  MAR  APR  MAY  JUN  JUL  AUG  SEP  OCT  NOV  DEC
    ANNEX MAPS         HRS/MAP    0.0  0.0  0.0  2.7  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
                          MAPS      0    0    0    4    0    0    0    0    0    0    0    0
    BASE MAPS          HRS/MAP    0.0  0.0  0.0  0.9  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
                          MAPS      0    0    0   44    0    0    0    0    0    0    0    0
    EXHIBIT A MAPS     HRS/MAP    0.0  0.0  0.0  2.2  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
                          MAPS      0    0    0    2    0    0    0    0    0    0    0    0
    SV PLANNER MAPS    HRS/MAP    0.0  0.0  0.0  3.7  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
                          MAPS      0    0    0    4    0    0    0    0    0    0    0    0
    CONVERSIONS        HRS/MAP    4.7  6.4  7.2  5.5  2.9  3.7  7.1  1.7  4.7  3.5  7.8  4.3
                          MAPS     16   22    6   20   35   15    6   50   14   34    5   26
    OTHER MAPS         HRS/MAP    0.0  0.0  6.0  0.0  0.0  1.6  1.6 12.9  0.0  0.0  0.0  0.0
                          MAPS      0    0    1    0    0   45   29    5    0    0    0    0

TIME-OFF CODES                HOURS
       0                        0.0
TOTAL TIME-OFF                  0.0
TOTAL NON-SYSTEM TIME          47.6
TOTAL SYSTEM TIME             114.5
TOTAL TIME                    162.1
```

SYSTEM UTILIZATION TRACKING BY JOB NUMBER

by
Brent Graham and Bob Rutherford
Boeing Commercial Airplane Company
M/S 72-08 P.O. Box 3707
Seattle, Washington 98124
206-237-7075 773-9736

ABSTRACT

In order to effectively manage a CV system, the personnel responsible need timely, accurate and accessible information. Such information would include job description, time spent on that job, estimated time to completion, date due, job number, operator, etc. The ability to collect and output this information will aid management in allocation of resources, scheduling, tracking of possible late releases, backlog, and work forecasting. This paper describes the methods used to collect and arrange the information in a usable format.

OBJECT

The object of this paper is to describe the methods used to collect the information from the CV system and put it into a usable format. To identify procedures and CV system capabilities that will aid other CV system users in producing a usable accounting system. The flow chart, Figure 1, shows how each of these programs fit together and a brief description of what each program accomplishes. To collect this information as easily as possible, we will use a variety of system capabilities such as PEP, data extract, execute file, sort and tally programs.

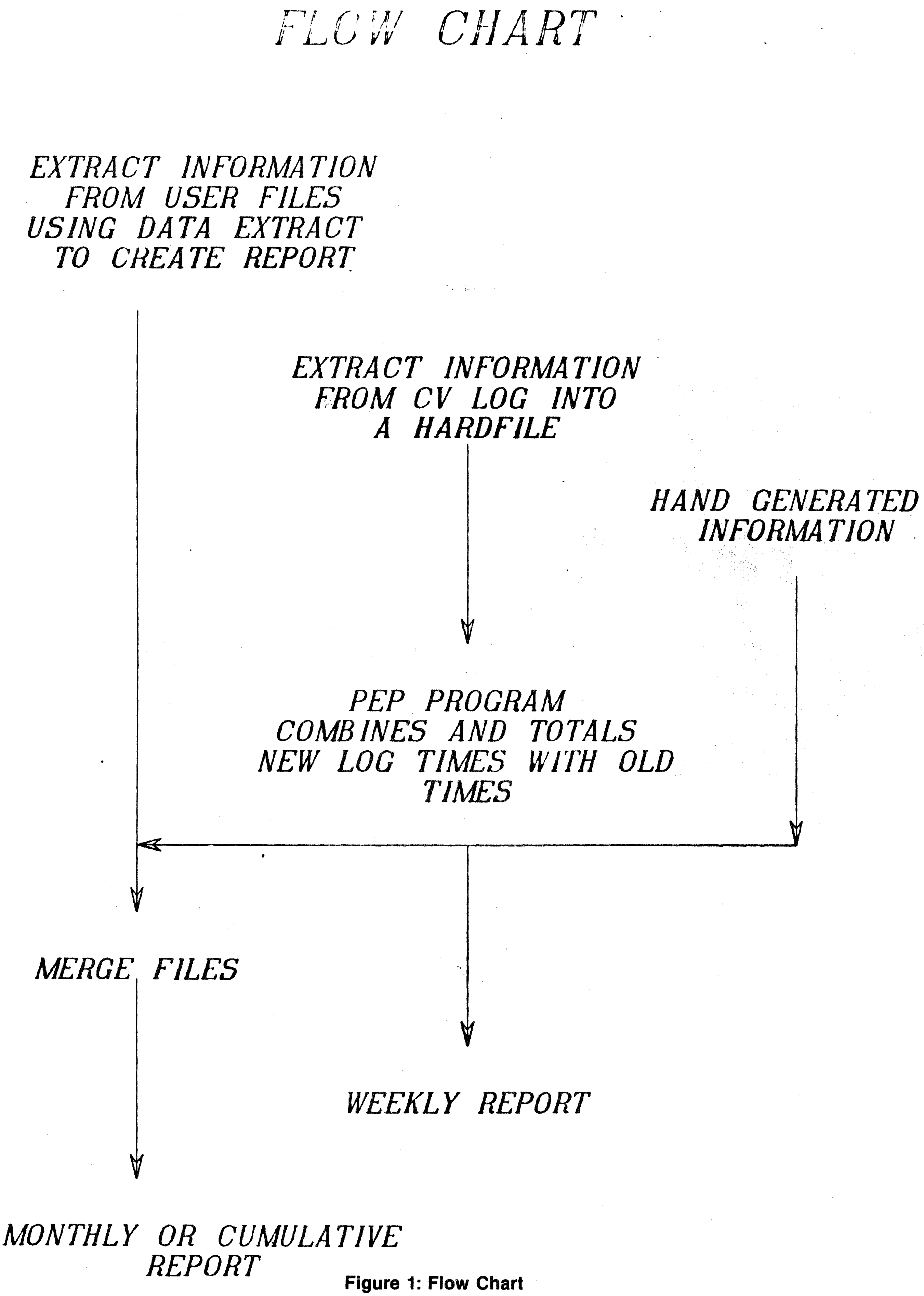

FLOW CHART
EXTRACT INFORMATION FROM USER FILES USING DATA EXTRACT TO CREATE REPORT
EXTRACT INFORMATION FROM CV LOG INTO A HARDFILE
HAND GENERATED INFORMATION
PEP PROGRAM COMBINES AND TOTALS NEW LOG TIMES WITH OLD TIMES
MERGE FILES
WEEKLY REPORT
MONTHLY OR CUMULATIVE REPORT

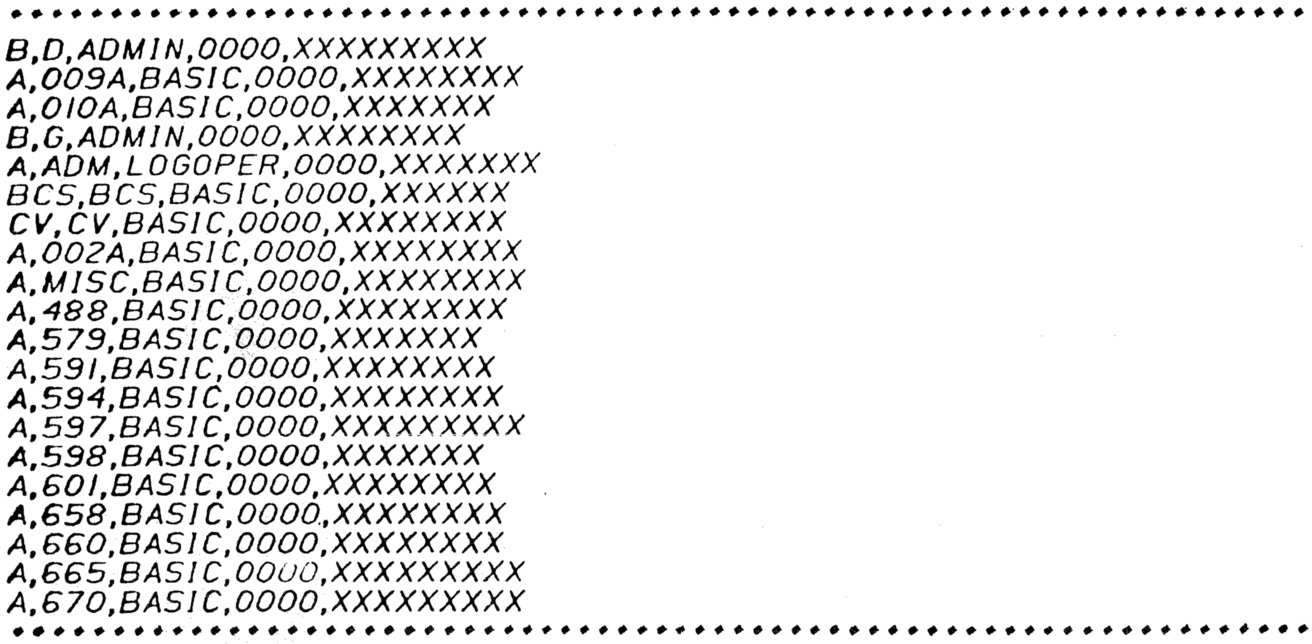
Figure 1: Flow Chart

PROCEDURE

Several reports are generated: a short report showing the weekly system usage, a report showing total times, and a longer more detailed report showing a cumulation of work done on the system by job title, job number, operator, and time spent.

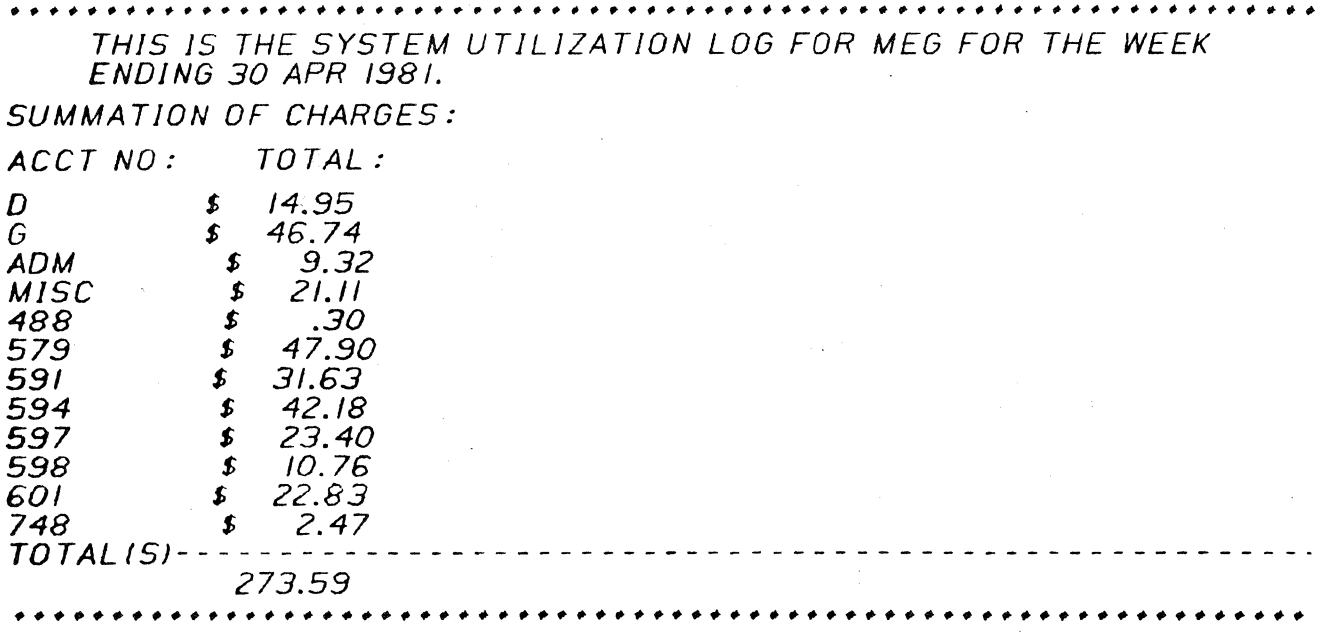
The first report is generated using the accounting tables, which contain the active job numbers. In order to track the time spent on a job the user is required to log onto the system using a name, job number, and then a password. An example of the accounting table is Figure 2. This procedure will allow the CV log system to track the time spent on a task and associate that time with a job number.

The information is extracted using a hardfile to save the oplog. It is now relatively easy to produce a short report that shows the time spent that week on each job. In our case, we used the CVTECO editor to produce the report shown in Figure 3. The report could also have been produced by using an execute file and the text editor.



```
B,D,ADMIN,0000,XXXXXXXXX
A,009A,BASIC,0000,XXXXXXXX
A,010A,BASIC,0000,XXXXXXX
B,G,ADMIN,0000,XXXXXXXX
A,ADM,LOGOPER,0000,XXXXXXX
BCS,BCS,BASIC,0000,XXXXXX
CV,CV,BASIC,0000,XXXXXXXX
A,002A,BASIC,0000,XXXXXXXX
A,MISC,BASIC,0000,XXXXXXXX
A,488,BASIC,0000,XXXXXXXX
A,579,BASIC,0000,XXXXXXX
A,591,BASIC,0000,XXXXXXXX
A,594,BASIC,0000,XXXXXXXX
A,597,BASIC,0000,XXXXXXXXX
A,598,BASIC,0000,XXXXXXX
A,601,BASIC,0000,XXXXXXXX
A,658,BASIC,0000,XXXXXXXX
A,660,BASIC,0000,XXXXXXXX
A,665,BASIC,0000,XXXXXXXXX
A,670,BASIC,0000,XXXXXXXXX
```

Figure 2



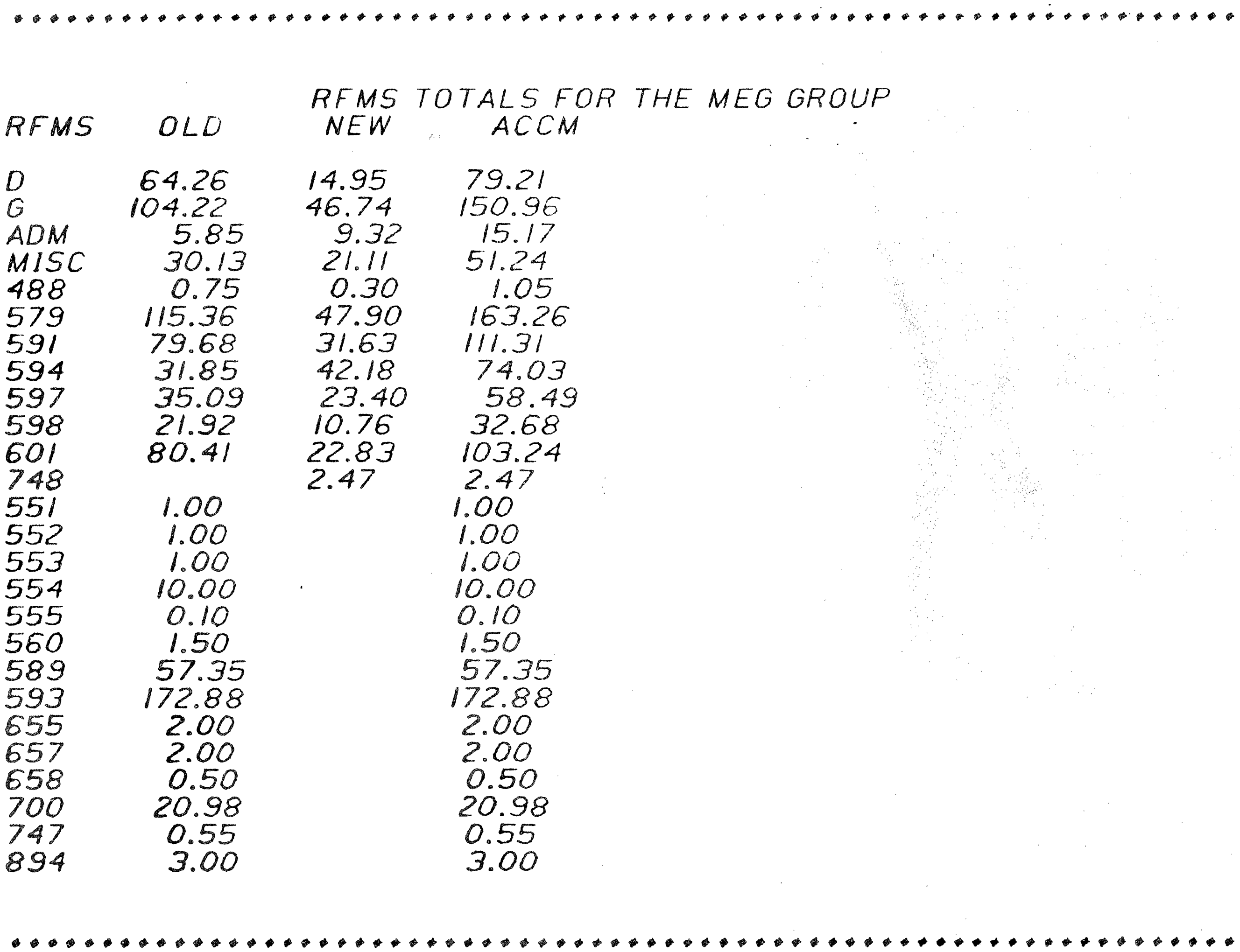
```
     THIS IS THE SYSTEM UTILIZATION LOG FOR MEG FOR THE WEEK
     ENDING 30 APR 1981.
SUMMATION OF CHARGES:
ACCT NO:      TOTAL:
D          $   14.95
G          $   46.74
ADM         $    9.32
MISC        $   21.11
488         $     .30
579         $   47.90
591        $   31.63
594         $   42.18
597         $   23.40
598         $   10.76
601        $   22.83
748         $    2.47
TOTAL(S)-----------------------------------------------------
               273.59
```

Figure 3

The next step is to add that time to the old time to produce a cumulative total. See Figure 3.

This report was done by comparing job numbers and adding the times together if the job numbers matched. If there was no matching job number, the value was entered into the new column and added to the totals.

The second report was produced by data extraction. The data extract section of the collection of programs is really very simple in principle and structure. The data.extract skeleton file on the system was used to create the working file. The properties used could have been any of those ready on the system, but we chose to create our own in order to avoid confusion to the users. These new properties were added to the data.property file in the area reserved by Computervision for user use.



```
                       RFMS TOTALS FOR THE MEG GROUP
RFMS       OLD          NEW       ACCM

D          64.26       14.95      79.21
G         104.22       46.74     150.96
ADM         5.85        9.32      15.17
MISC       30.13       21.11      51.24
488         0.75        0.30       1.05
579       115.36       47.90     163.26
591        79.68       31.63     111.31
594        31.85       42.18      74.03
597        35.09       23.40      58.49
598        21.92       10.76      32.68
601        80.41       22.83     103.24
748                     2.47       2.47
551         1.00                   1.00
552         1.00                   1.00
553         1.00                   1.00
554        10.00                  10.00
555         0.10                   0.10
560         1.50                   1.50
589        57.35                  57.35
593       172.88                 172.88
655         2.00                   2.00
657         2.00                   2.00
658         0.50                   0.50
700        20.98                  20.98
747         0.55                   0.55
894         3.00                   3.00
```

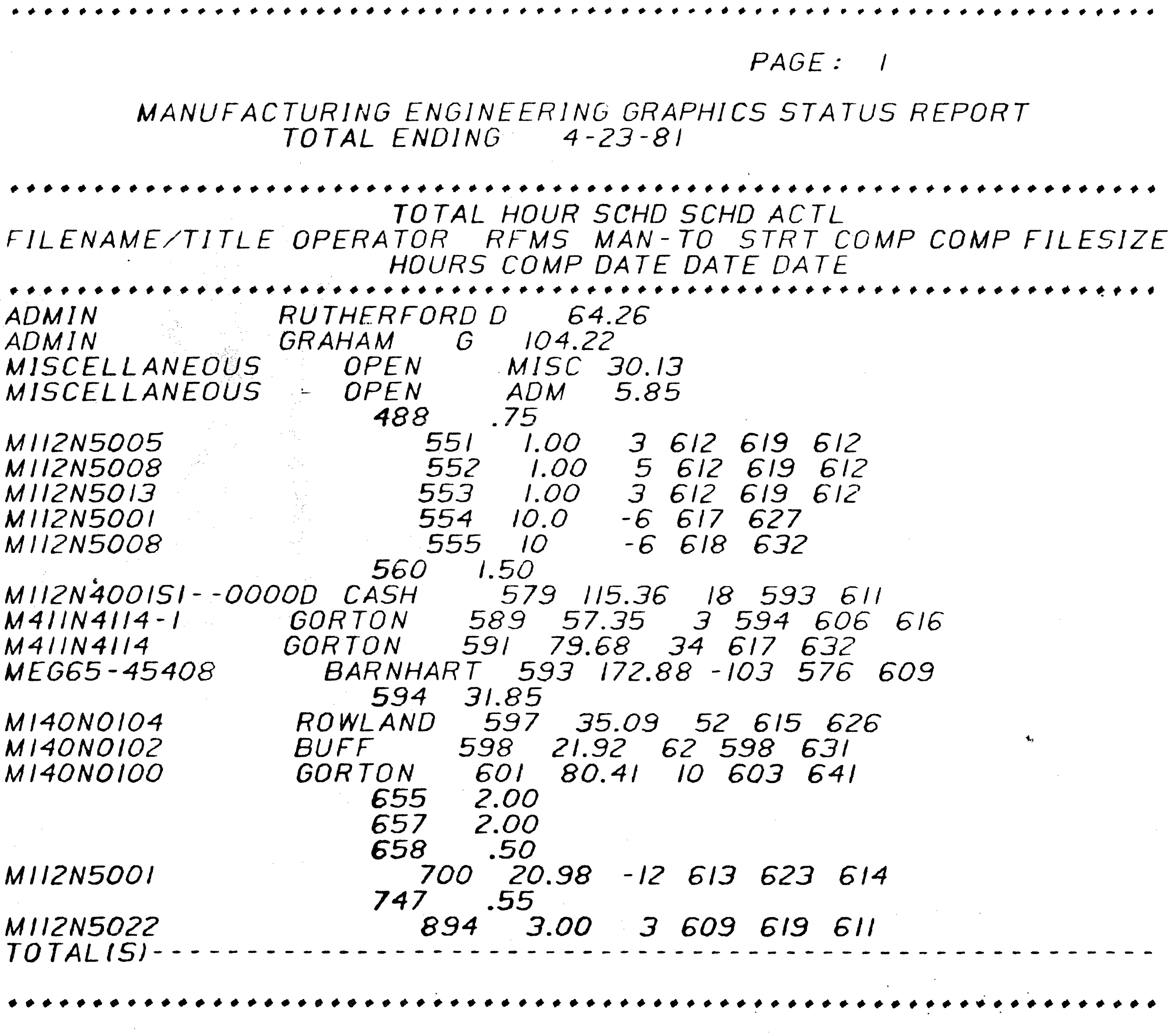
Figure 4

For our application, we chose to use props for their universal properties. We wanted each production dataset to carry the name of the creator, the RFMS number (request for MEG service), the scheduled start date, the scheduled completion date, the estimated hours to completion, and the actual completion date. These last two entries would be updated as the hours to completion changed, and when the job was completed.

The initial inputs of creator, RFMS number, scheduled start date, and scheduled completion date are only entered once at the time the job is activated. These inputs are done through use of either the menu keyfile or through execute files at the CADDS level. Any changes to these properties are also done through these files. The operator assigned the RFMS is responsible for all inputs and is charged with keeping the information current. All inputs are either text entries or integers.

The data extraction file is quite simple and straight forward. We use the standard format. The program sorts on the creator and then the RFMS number. There are three fields in the report which are left open for data to be merged from other programs. Once the extraction has been run, the report can be re-sorted as needed by any of the field headings through the use of the CADDS level command, "sort file."

In order to obtain the titles for jobs that were not included in the data extraction, another PEP file was created that, when executed, builds a text file that could be merged with the report. This report contains information about jobs that were not run directly on the CV system but were related to it. For example see Figure 5.



```
♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦

                                                      PAGE :   1

          MANUFACTURING ENGINEERING GRAPHICS STATUS REPORT
                         TOTAL ENDING     4-23-81

♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦
                              TOTAL HOUR SCHD SCHD ACTL
FILENAME/TITLE OPERATOR  RFMS  MAN-TO  STRT COMP COMP FILESIZE
                              HOURS COMP DATE DATE DATE
♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦
ADMIN            RUTHERFORD D    64.26
ADMIN            GRAHAM    G   104.22
MISCELLANEOUS      OPEN      MISC  30.13
MISCELLANEOUS   -  OPEN      ADM    5.85
                        488    .75
MI12N5005                 551   1.00   3  612  619  612
MI12N5008                 552   1.00   5  612  619  612
MI12N5013                 553   1.00   3  612  619  612
MI12N5001                 554  10.0   -6  617  627
MI12N5008                 555  10     -6  618  632
                        560   1.50
MI12N4001SI--0000D  CASH     579  115.36   18  593  611
M411N4114-1        GORTON    589  57.35   3  594  606  616
M411N4114          GORTON    591  79.68  34  617  632
MEG65-45408          BARNHART  593  172.88 -103  576  609
                        594   31.85
MI40N0104          ROWLAND   597   35.09  52  615  626
MI40N0102          BUFF      598   21.92  62  598  631
MI40N0100          GORTON    601   80.41  10  603  641
                        655   2.00
                        657   2.00
                        658    .50
MI12N5001                 700   20.98  -12  613  623  614
                        747   .55
MI12N5022                 894   3.00   3  609  619  611
TOTAL(S)----------------------------------------------------------

♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦
```

Figure 5

CONCLUSION

Once this system is implemented, it is easy for the system administrator to update the files to reflect changing needs. The reports generated can be disposed to line printers and routed to affected management. The examples shown in this paper show our needs and our naming convention. Each CV system administrator will need to determine his tracking needs and develop his accounting systems accordingly. We hope that some of the concepts discussed will be of use in setting up your accounting system.

Software

PEST: AN INTERACTIVE GRAPHICS LANGUAGE FOR THE CV USER

S. C. Morris
Applications Manager
Elstree Computing Limited
London, U.K.
1-959-3636

John Laing Limited, one of the United Kingdom's largest civil engineering contractors, installed its first Computervision equipment in 1975. It was located in the offices of John Laing Design Associates (JLDA), the Laing Group's own multidisciplinary design office for building and civil engineering projects.

The main objectives of this installation were two-fold:

1. to undertake a detailed assessment of the feasibility (from both the client and computing points of view) of developing a Computer Aided Design and Pricing application (CADAP) for projects for which Laing undertook both the design and construction. In the envisaged application, the CV system would be used to:

   i) develop a co-ordinated design for the project in question,
   ii) produce the required working drawings from the agreed design,
   iii) generate from it files of basic identification and dimensional data.

   These CV-generated files would then be transferred to the Laing Group's twin IBM370/145 mainframes for further processing (quantification, sorting and collating) before being made available to an existing on-line estimating system running on the IBM370s and used by estimators at the London head office and at 12 regional offices throughout the UK.

2. The second objective of the CV installation was to develop in parallel with the above study a wide range of computer-aided drafting techniques to meet the requirements of the Laing Group in general and JLDA in particular.

In both cases, the necessary computing expertise was provided by staff from Elstree Computing Limited (ECL), a wholly-owned subsidiary of the Laing Group, with a staff of approximately 150, who provide a complete computing service to the Laing Group and, increasingly, to a number of other external organizations.

Since the latter half of the 1970s, there has been a steady decline in the UK construction workload, adversely affecting the "design and construct" market in particular. Consequently, the CADAP application has not been implemented although a large part of the necessary application design and costings have been completed. Instead, effort has been concentrated on the development of computer-aided drafting techniques. A user-oriented graphics language, developed by ECL for use in this application area, is the subject of this paper.

Our initial drafting experience was gained with CADDS3, releases 3 and 4. This proved that CADDS has the graphic capabilities to enable typical building and civil engineering drawings to be produced on the system. However, it was not possible to construct these economically from either a time or cost standpoint because of time-consuming and complex input procedures.

We briefly considered using PEP but at this time (1976) it did not offer interactive input facilities at run time and did not meet our users' requirements. A decision was, therefore, taken to develop our own routines to provide the facilities we needed, and the idea of PEST was conceived.

PEST (Parameterised Entity Symbol Translator) was designed to meet the following basic requirements:

1. It should be a simple programming language easily used by draftsmen with no special programming knowledge.

2. In use, it should:

   a. allow a symbol to be specified in terms of a number of parameters and the relationships between them, with the actual value of these parameters and other dimensions being input at run time;

   b. allow the user to specify all the appropriate interactive dialogue invoked at run time to secure the necessary parameter values;

   c. appear to the user to operate as part of the basic operating system, in this case CADDS;

   d. allow a symbol, once created, to be added to the graphics database, subject to the normal graphics operations;

   e. operate at a speed acceptable to the user.

The initial CV version of PEST was completed very quickly and passed to the JLDA draftsmen for assessment and comment. In its earliest form, PEST simply permitted the specification of a symbol in terms of various straight lines, the relationships between these lines, and the input of the various linear dimensions from the keyboard at run time.

Despite the limitations that this basic system imposed, it was clear from the enthusiastic response from the users that this type of software went a long way towards meeting JLDA requirements, both in terms of approach computer-aided drafting and cost- and time-effective drawing creation. This acceptance was further underlined to ECL by the arrival from JLDA of a long list of suggested enhancements to the basic version of PEST then in use.

Since that time, PEST has been continually developed by ECL to meet JLDA requirements. Four major releases have been developed and installed.

It is now considered that the major development of CV-based PEST is complete, and that the original objectives have been achieved with a level of success undreamt of at the outset.

PEST can not be regarded rightfully as offering comprehensive graphics language and processing facilities. In addition to those features already mentioned, PEST provides for:

—parameter value input via the digitizer as well as via the keyboard;

—both compulsory and optional parameters, with default values for optional parameters;

—a range of 30 different language statement types;

—processing loop facility;

—decision statements;

—full mathematical functions and the ability to handle expressions;

—symbol modifiers to control
—angle of rotation
—origin
—scaling
—layer specification
—mirroring
—arrays of symbols with or without variable spacing.

Our most recent development includes the provision of links both to and from FORTRAN—a facility now being used with increasing frequency and cost effectiveness.

The PEST software consists of three phases, as follows:

1. **The Compiler**
   The Compiler behaves like a conventional language compiler. It reads the PEST symbol source code produced by the user and checks its syntax. From this source code, the Compiler produces an object code which is designed to be obeyed by the Language Processor. Once successfully compiled, the object code of a particular symbol is stored within the CV/PEST library from which it is retrieved whenever required for insertion into a drawing.

2. **The Language Processor**
   The Language Processor retrieves the object code from the library and undertakes functions such as prompts for input/calculations specified in the user's CV/PEST symbol source code. In the CV version of PEST, standard CADD3 utilities are called in order to process any input statements in the PEST source code which require keyboard or digitizer input. During processing of the commands contained within the symbol, the output to generate the graphics is written to a temporary file for input to the Display Processor.

3. **The Display Processor**
   The Display Processor modifies the output from the Language Processor to take account of user-selected origin, scale, rotation and mirroring parameters. The Processor then writes the resulting graphics to the CV database. Handling the common operations of rotating, mirroring, scaling, and positioning in this way makes PEST programs simpler to write and more powerful. That's because this logic does not need to be included in each program.

Our PEST library currently contains some 350 PEST symbols. They cover all aspects of our CV drafting activities from simple 2D grids and drawing sheet outlines through a range of architectural, civil engineering and piping layout symbols, to complex 3D symbols relating to excavation quantities for quarrying and open-cast mining.

Since our purpose is not to provide instruction in the use of PEST, no further technical details of PEST are given. The remainder of the paper discusses in detail the application of PEST to some of the graphics-computing activities of a major Construction Group.

TYPES OF PEST SYMBOLS

The PEST symbols that we have developed can be conveniently classified into three categories:

1. **Basic Symbols**

These symbols normally require only one piece of additional input from the user—he must digitize the position at which the symbol is to be drawn.

Examples of this type of symbol are:

company logos
drawing sheet outlines

Essentially they are very simple symbols which do not vary in size or proportion.

2. **Interactive Symbols**

This group represents not only the largest number of individual symbols but also those which provide the greatest increase in productivity and hence, cost benefit. In use, these symbols require additional user input to create additional graphics either directly or as a result of some further calculation based upon the user-provided input. Examples of interactive symbols will be discussed later.

3. **Non-Graphics Symbols**

Symbols in this group do not generate any graphics output but are used to perform various functions related to the use of another graphical, normally interactive, symbol or symbols. Typical examples are a simple design calculation or the preparation of a material schedule.

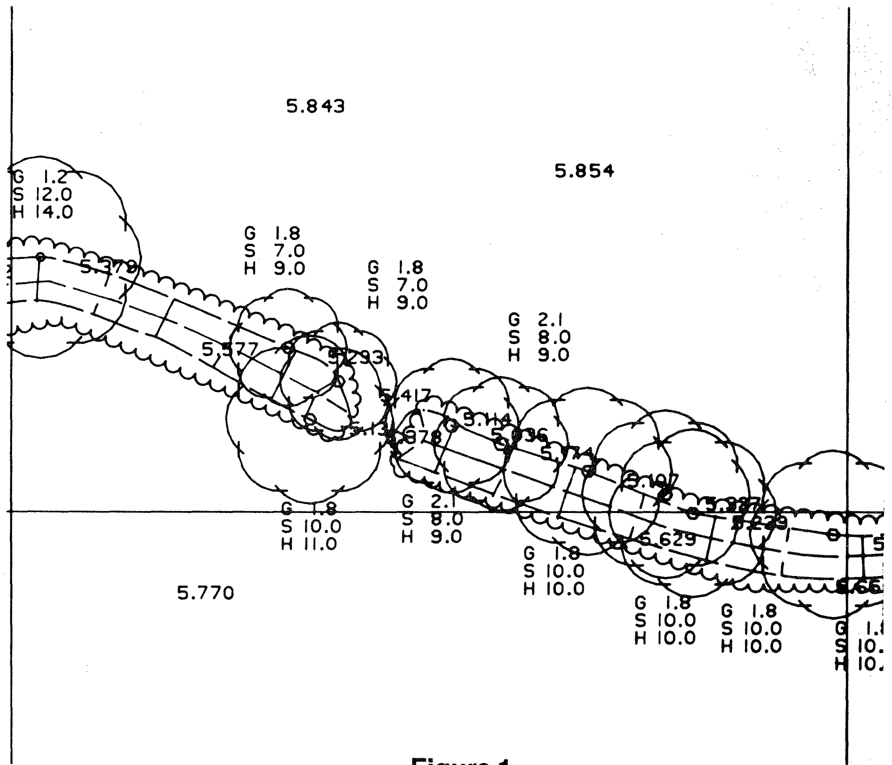


Figure 1

APPLICATIONS OF PEST

1. Land Surveying

We use our CV installation to prepare final survey drawings as a result of our field surveys. Our raw survey data (points, etc.) is processed initially on our 370 (though this work will shortly be moved to our in-house PRIME 550). The completed survey or point plot is then passed via magnetic tape to the CV installation.

In completing the survey drawing, we use a number of relatively simple PEST symbols to insert the various features (trees, hedges, ditches, etc.) onto the plot. Figure 1 shows an extract from such a plot with trees and hedges.

In the relatively simple interactive symbol for the tree, the user is prompted to supply the following:

Tree girth in metres
Tree spread in metres (but can be digitized)
Tree height in metres
Tree type

The tree symbol is then drawn to scale in the position selected by the operator, and the appropriate text identifying the tree is also drawn alongside the symbol in the position identified.

The hedge shown is also another symbol inserted where required at the width given by the draftsman. At this stage in the preparation of the survey drawing, the draftsman may well be working direct from the site surveyor's notebook.

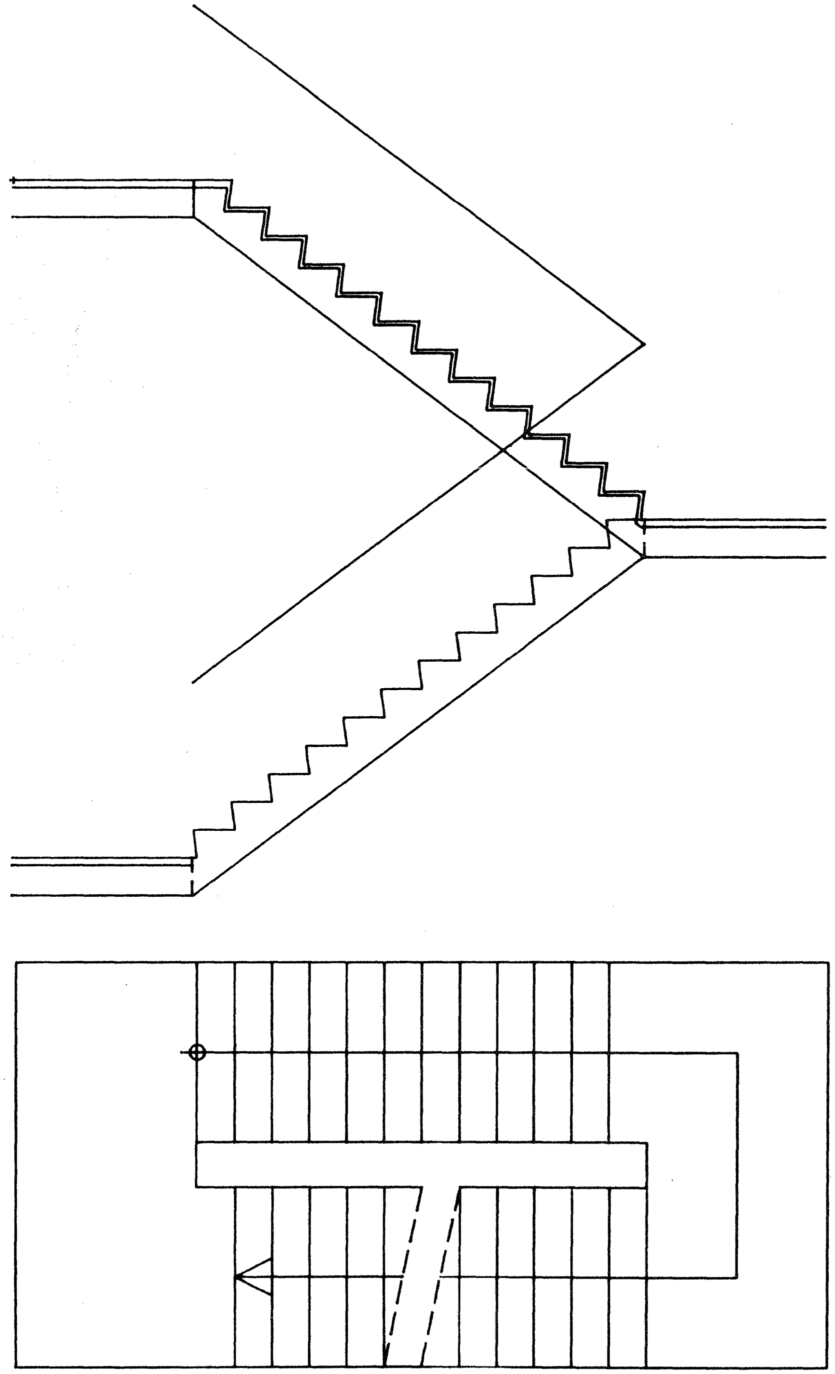
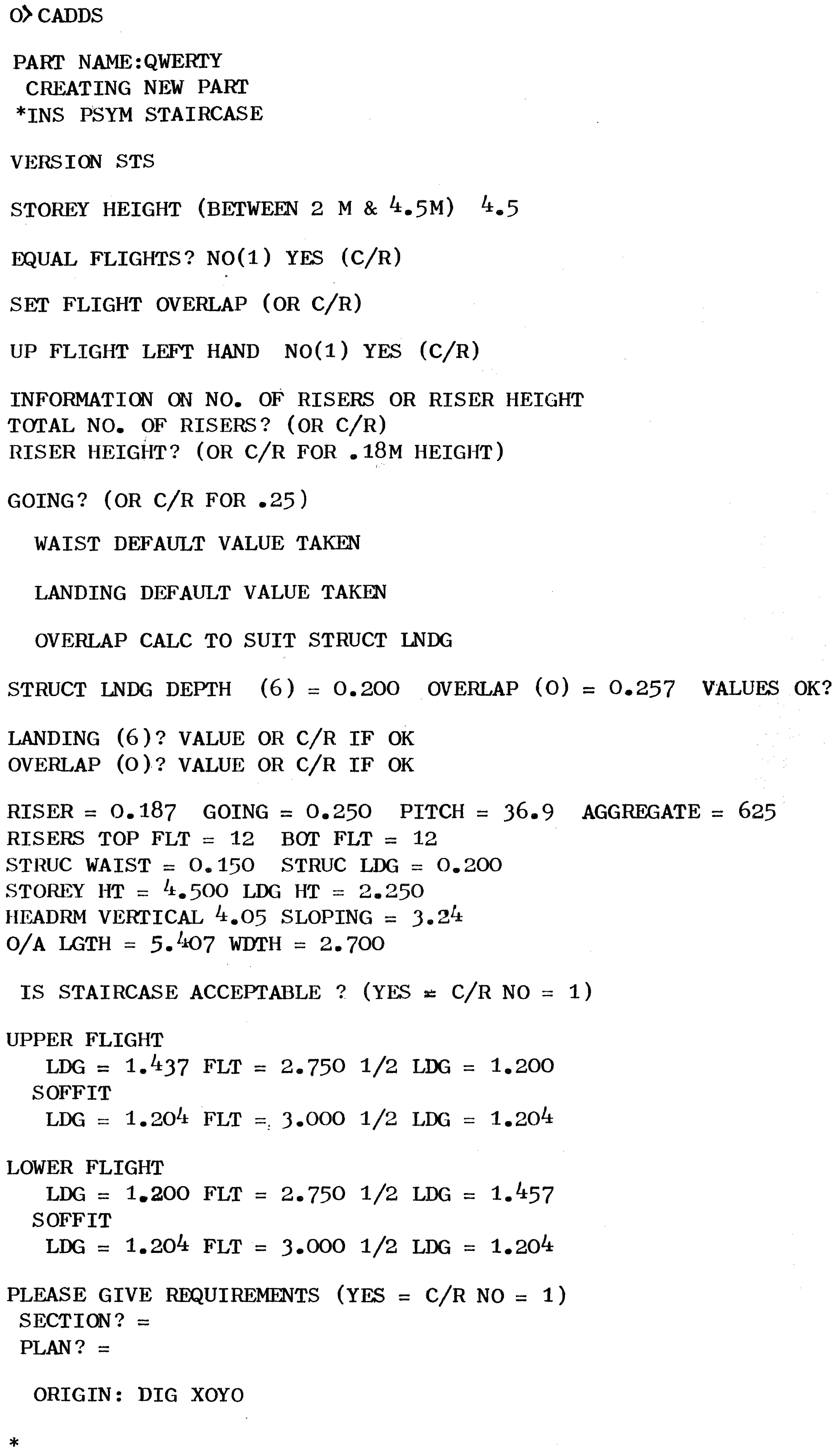


Figure 2.1



```
O> CADDS

 PART NAME:QWERTY
  CREATING NEW PART
 *INS PSYM STAIRCASE

VERSION STS

STOREY HEIGHT (BETWEEN 2 M & 4.5M)  4.5

EQUAL FLIGHTS? NO(1) YES (C/R)

SET FLIGHT OVERLAP (OR C/R)

UP FLIGHT LEFT HAND  NO(1) YES (C/R)

INFORMATION ON NO. OF RISERS OR RISER HEIGHT
TOTAL NO. OF RISERS? (OR C/R)
RISER HEIGHT? (OR C/R FOR .18M HEIGHT)

GOING? (OR C/R FOR .25)

  WAIST DEFAULT VALUE TAKEN

  LANDING DEFAULT VALUE TAKEN

  OVERLAP CALC TO SUIT STRUCT LNDG

STRUCT LNDG DEPTH  (6) = 0.200  OVERLAP (O) = 0.257  VALUES OK?

LANDING (6)? VALUE OR C/R IF OK
OVERLAP (O)? VALUE OR C/R IF OK

RISER = 0.187  GOING = 0.250  PITCH = 36.9  AGGREGATE = 625
RISERS TOP FLT = 12  BOT FLT = 12
STRUC WAIST = 0.150  STRUC LDG = 0.200
STOREY HT = 4.500 LDG HT = 2.250
HEADRM VERTICAL 4.05 SLOPING = 3.24
O/A LGTH = 5.407 WDTH = 2.700

 IS STAIRCASE ACCEPTABLE ? (YES = C/R NO = 1)

UPPER FLIGHT
   LDG = 1.437 FLT = 2.750 1/2 LDG = 1.200
  SOFFIT
   LDG = 1.204 FLT = 3.000 1/2 LDG = 1.204

LOWER FLIGHT
   LDG = 1.200 FLT = 2.750 1/2 LDG = 1.457
  SOFFIT
   LDG = 1.204 FLT = 3.000 1/2 LDG = 1.204

PLEASE GIVE REQUIREMENTS (YES = C/R NO = 1)
 SECTION? =
 PLAN? =

  ORIGIN: DIG XOYO

*
```

Figure 2.2

```
#RUN PSYM 3D-B69T

DEPTH OF HOUSE 7

WIDTH OR FRONTAGE 5

HEIGHT TO EAVES 4.2

DEPTH OF PORCH 1

WIDTH OF PORCH 1

HEIGHT OF PORCH TO EAVES 1

DEPTH OF GARAGE 6

WIDTH OF GARAGE 2.3

HEIGHT OF GARAGE 2.2

DIG POSITION DIG d
GIVE LEVEL OF GROUND FLOOR SLAB 1

LEFT (1) OR RIGHT (2) HAND PORCH

GARAGE YES

DIG POSITION DIG d
GIVE LEVEL OF GROUND FLOOR SLAB 2

LEFT (1) OR RIGHT (2) HAND PORCH

GARAGE NO

DIG POSITION DIG


# DEF VIEW AX=65AY15 SEL

AS VIEW 7

#SEL VIEW 7
```

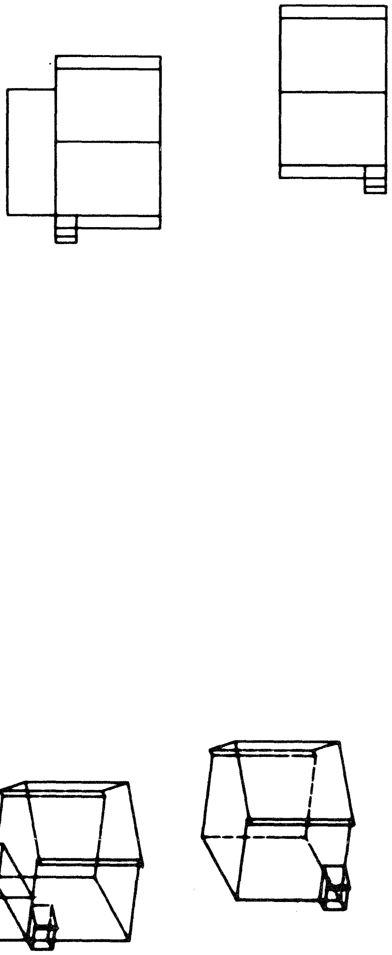


Figure 3

2. Architectural Detailing

In this application area, I have selected a staircase as a representative symbol.

The setting out and drawing of staircases in buildings is a time-consuming and tedious task. This early PEST symbol has removed the problem. Now the insertion of staircases in architectural or civil engineering drawings is simple, straightforward and, above all, fast. Figure 2 shows a typical staircase produced by the symbol in both elevation and plan, together with details of the associated prompts and user responses. After the user has entered all the appropriate parameter values, and all the necessary calculations have been performed, PEST outputs details of the staircase under design for user approval, before insertion in the drawing. If the user is not satisfied, the parameter values input may be changed and the staircase redesigned. Only when the user is satisfied with the arrangement of the staircase is it inserted into the drawing.

In the early days of PEST, it was the requirement for a staircase symbol that highlighted our need to be able to input parameter values by direct digitizing. In that way we could handle situations where in certain locations in a building exact story heights, stairwell sizes, etc. were not immediately obvious to the draftsman.

Over the years we have established a large number of architectural symbols covering not only the normal fittings (WCs, washbasins, etc.) but also door and window opening details, etc.

3. Estate Layouts

During the past two years an important part of our work has been concerned with estate layout procedures. We now have a comprehensive estate layout "package" which covers all aspects of this work including road and footpath layout, horizontal and vertical alignment, drainage detailing, and the layout of houses, garages, etc. In this work, significant use is made of PEST.

The example shown in Figure 3 relates to one particular house type which can vary in size, has a number of options (garage, porch, etc.) and can be built as a LH or RH version. The user is prompted for various parameters as shown, and the house plan is drawn accordingly.

Another feature of PEST is that a symbol can be repeated simply by changing those parameters which vary, and so a second house may be simply inserted omitting the garage and reversing the hand.

We are extending our estate layout work to include full 3D modelling. This particular PEST symbol will, as shown, automatically generate a 3D view of the houses previously specified.

Figure 4 shows part of a typical housing estate layout produced by our Estate Layout System.

The procedures outlined briefly above are, of course, equally applicable to non-housing estates, e.g. industrial layouts, airports, etc.

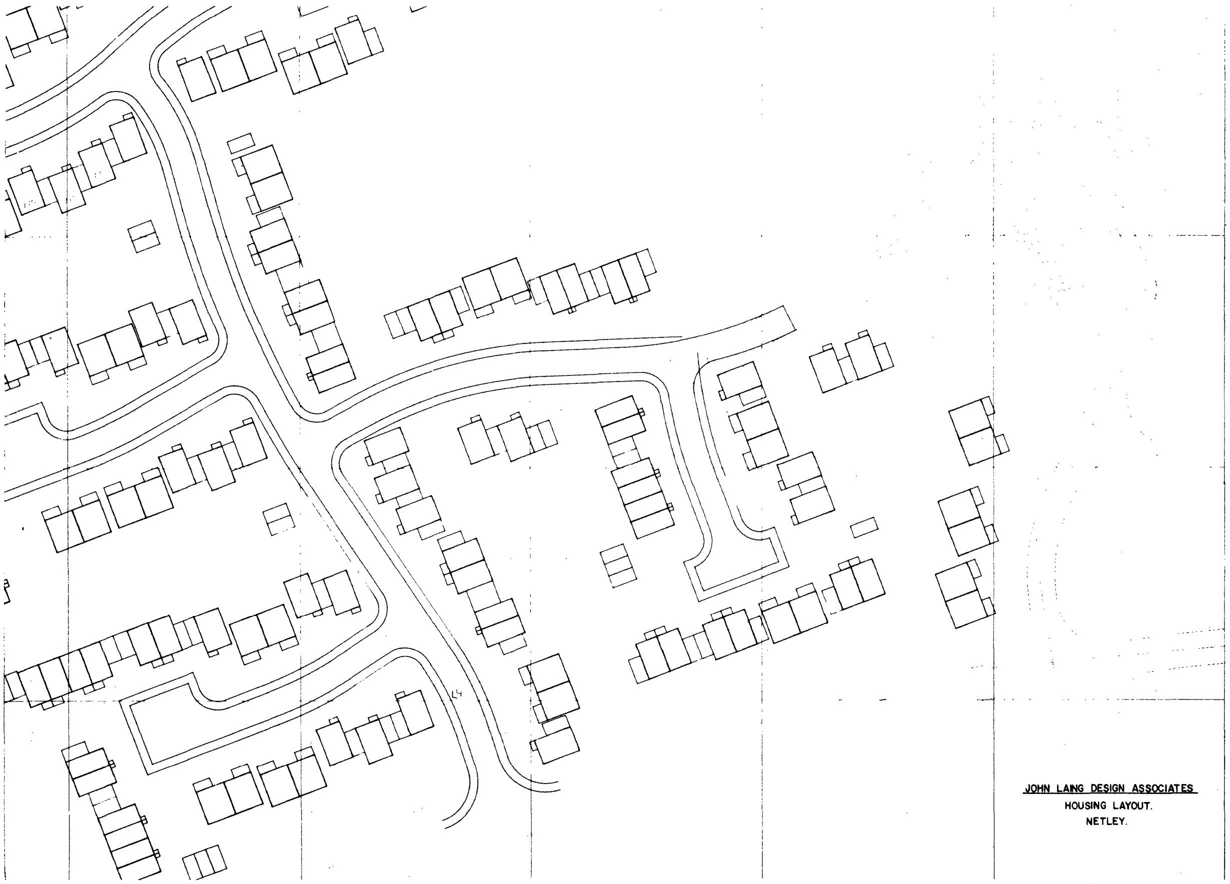


Figure 4

4. Housing Components

A significant proportion of the houses now being built in the UK are of timber frame construction. The symbol shown in Figure 5 was developed solely to explore the possibility of using our CV installation to assist in the layout of timber panels in a house and to produce at the end a schedule of the panel types and quantity in a particular house type.

This symbol generates the required house outline. The user is then prompted to insert panels of different types and sizes (plain, corner, window, door, etc.) into a wall of the house.

As this is being done, the graphics displayed by the PEST symbol are modified to show cumulatively the panels that have been inserted. In addition, for the wall in which the panels are currently being inserted, the remaining length into which panels may be fitted is continuously adjusted and displayed to the user. As a panel is inserted into a wall, it is written away to a temporary queue. On completion of the layout of the panels, it is picked up from the queue and accessed by another PEST symbol which produces what is currently a fairly crude schedule. (Figure 5.2) It is worth noting that these two symbols, developed essentially for research purposes, took less than 2½ days to code, debug and get operational.

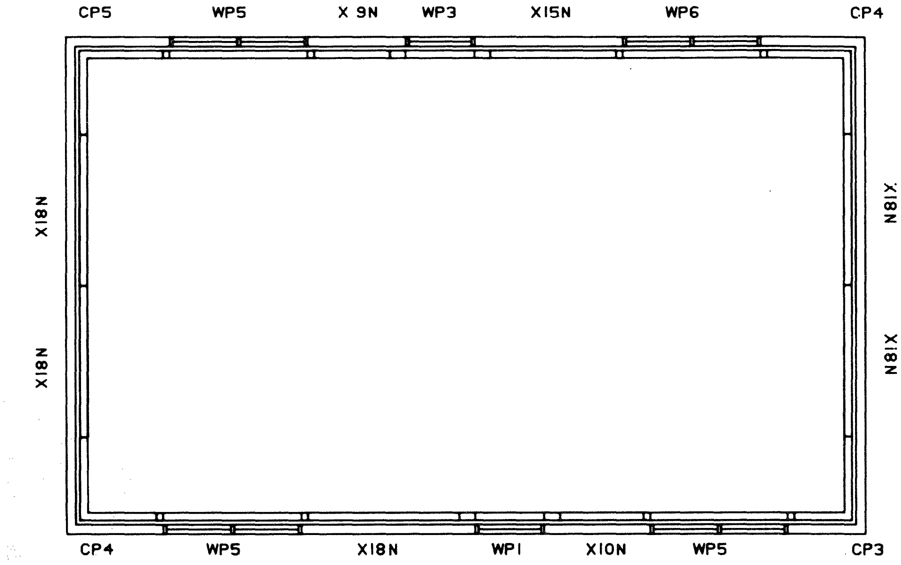


Figure 5.1

```
#RUN PSYM SUPAN

GIVE O/ALL LENGTH 9.514

GIVE O/ALL DEPTH 5.914

DIG POSITION GROUND FLOOR PLAN DIG d
DIG POSITION 1ST FLOOR PLAN DIG d
POSITION PLAN WITH ZWIN DIG ZWINdd

GIVE DISTANCE TO FIRST PANEL .9

DISTANCE REMAINING 8.290
IS PANEL WINDOW (1) OR PLAIN (2) 1

GIVE WINDOW PANEL TYPE 1,2,3,4,5,6 5

DISTANCE REMAINING 6.490
IS NEXT PANEL A CORNER YES (1) NO(2) 2

IS PANEL WINDOW (1) OR PLAIN (2) 2

GIVE PANEL WIDTH 1.8   etc. . .

#RUN PSYM SCHED

ORDER NO. 1234

CUSTOMER NAME ELSTREE COMPUTING
```

Figure 5.2

(continued)

```
PLACE OF DELIVERY BOREHAMWOOD

ORDER NO. 1234
CUSTOMER NAME ELSTREE COMPUTING
PLACE OF DELIVERY BOREHAMWOOD

BIN              PART            DESCRIPTION            MANUF/        QUANTITY
NO.              NO.                                    BOUGHT        REQUIRED
                 WP  1        WINDOW                      B              1
                 WP  3        WINDOW                      B              1
                 WP  5        WINDOW                      B              3
                 WP  6        WINDOW                      B              1
                 CP  3        CORNER PANEL                B              1
                 CP  4        CORNER PANEL                B              2
                 CP  5        CORNER PANEL                B              1
                 XP  9        EXTERNAL PANEL              B              2
                 XP 15        EXTERNAL PANEL              B              1
                 XP 18        EXTERNAL PANEL              B              5
#ZOM ALL
```

Figure 5.2 (continued)

5. Civil Engineering

One of our main interests in this area is the detailing of drawings of reinforced concrete structures. We have been using PEST symbols for some time in connection with this work. As contractors as well as designers, we prefer certain typical arrangements of reinforcing bars in various situations, from both the design and construction view points. PEST is ideally suited to such cases and we have established a number of PEST symbols to reflect this method of working. The PEST symbol generates the necessary graphics on the basis of parameter values both as input by the draftsman and as calculated by a separate design program resident on the CV installation.

There is another advantage to this approach using linked, but nevertheless separate, design and graphics programs—a single PEST symbol can be linked to a number of different design programs each complying with a specific Code of Practice (e.g., UK, USA or European).

The example shown is of a reinforced concrete pile cap. (Figure 6).

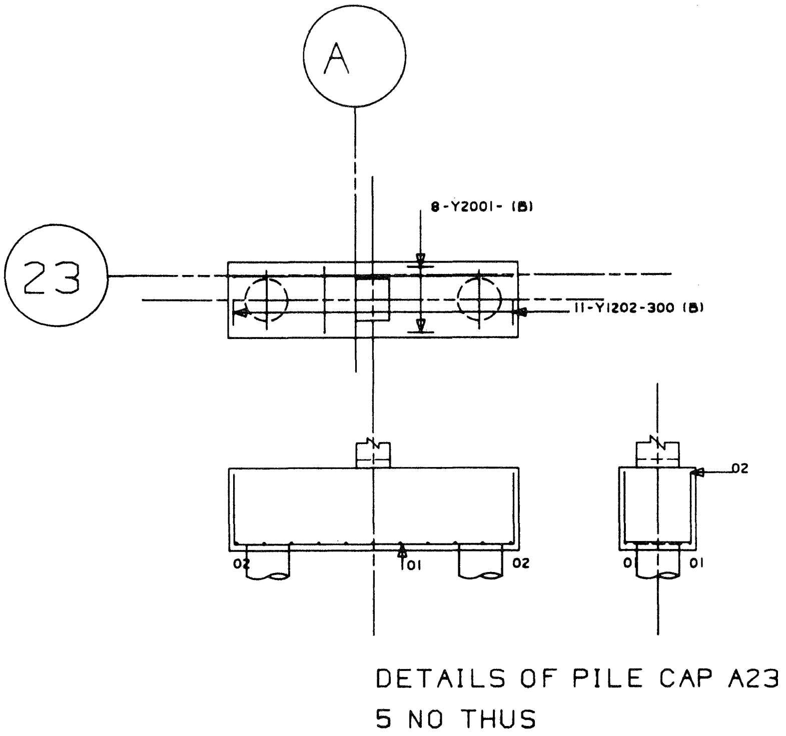


Figure 6

6. Piping

One of the Laing Group's associated companies has recently been using our CV installation for the preparation of piping drawings. They have established some 30 separate symbols which are used in the conventional PEST manner (i.e., the user supplying the parameter values—diameter, length, capacity, etc.) to produce the symbol size and shape for insertion at the desired part of the drawing.

In addition, a number of other PEST symbols have been developed for vessels such as filters. These symbols perform all the necessary calculations to determine flow rate, filter characteristics, etc. Then, after the operator has confirmed his acceptance of the results, the symbols produce the required graphics output.

It is worth noting that the drawings shown in Figures 7 and 8, and all the symbols contained therein, were produced by one particular draftsman whose total experience in computer-aided drafting (i.e., gaining familiarity with CADDS and PEST) was approximately half-time over a period of some six months.

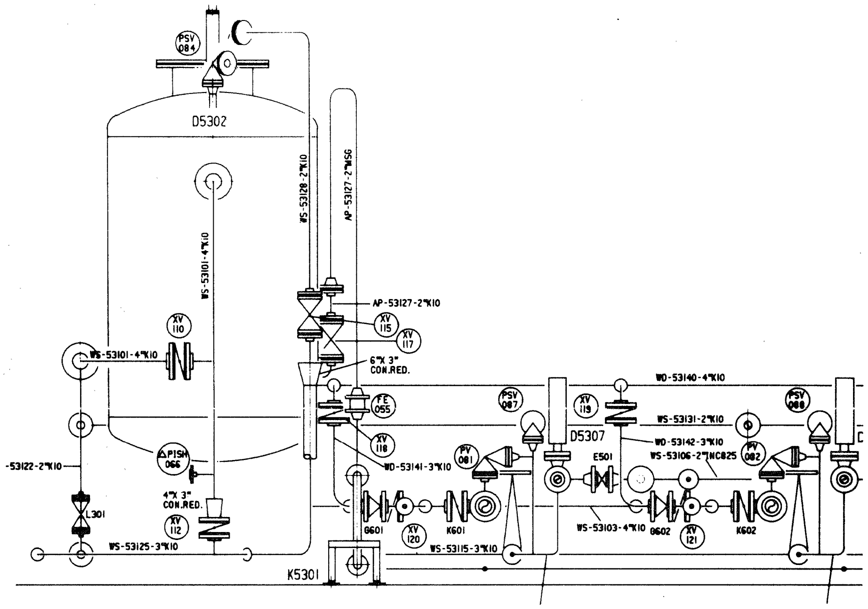


Figure 7

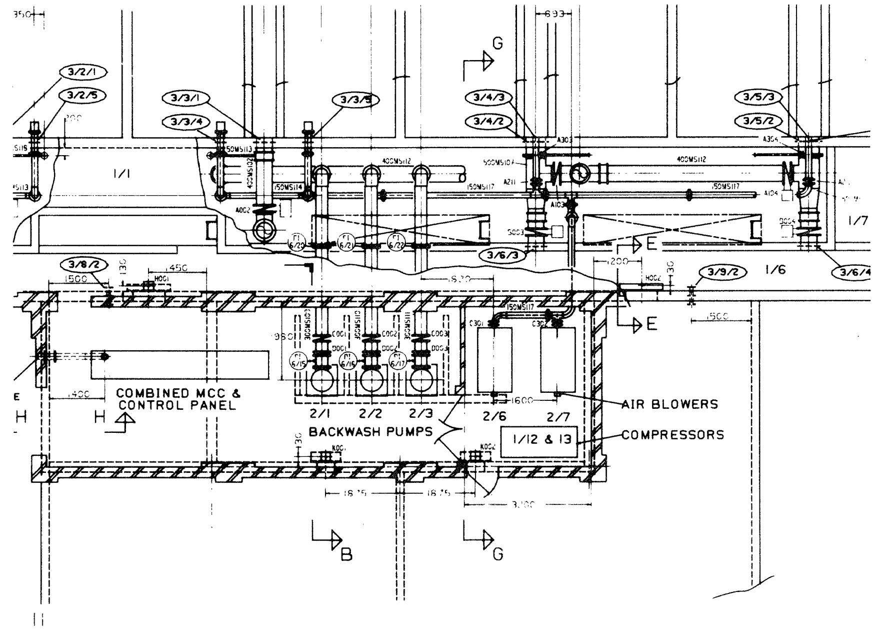


Figure 8

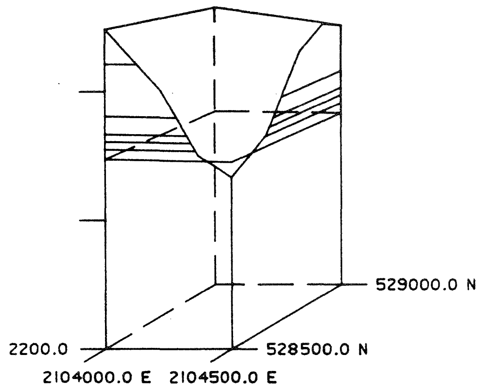
7. Earthwork Calculations

It is often necessary within the construction industry to prepare bids in what appears to be an impossibly short period. An advantage of PEST is that it enables specific computer solutions or aids to be developed in a timescale typically less than half of that required by FORTRAN.

Recently the Laing Group were involved in preparing a bid for a large open-cast, coal-mining project. An initial and crucial part of the work involved determining, from borehole data and contour maps, the volume of coal to be extracted from two seams between 5 and 10 feet thick, and the amounts of over and interburden to be removed to get at the coal. In certain areas of the site, the overburden/coal ratio might make it uneconomic to extract the coal. Therefore, it was necessary to isolate those parts of the site. To facilitate this, the site was overlaid with a 500 foot grid, and calculations of volumes were then undertaken individually for each 500 foot-square block. The main output required was a diagrammatic 3D representation of single or groups of blocks showing the surface of the land, the coal seams and over the interburdens, together with the calculated volumes of coal and spoil. In this application, the PEST-FORTRAN links were used. FORTRAN was used mainly for establishing the files of borehole and gridded contour data and carrying out the necessary volumetric calculations and level interpolations. PEST was used for generating the necessary graphics output from the appropriate data transferred from the FORTRAN files.

An example of the typical PEST output for a particular block is shown opposite in Figure 9. The volumetric data to the right of the three-dimensional block diagram could also, if required, be output to the telewriter without graphics. An initial, abbreviated, pilot scheme to demonstrate to our estimators the feasibility of this scheme was completed by ECL and JLDA in three days. Following their acceptance of this approach, a full production version containing a number of other features not described above was completed and operational in less than three weeks.

We have used very similar techniques in preparing numerous cross-sections for quantification in canal and highway construction projects.



```
          VOL 1      VOL 2      MEAN (CU.YDS.)

OB1. =    17814.8    70925.9    44370.4
OB2. =   269560.2   272083.3   270821.8
OB3. =    99606.5    98472.2    99039.3
M.K. =    44893.5    44120.4    44507.0
I.B. =    48643.5    47796.3    48219.9
L.K. =    63055.5    63203.8    63129.7

TOTAL COAL =  120553.0 TONS
TOTAL OB.  =  462451.4 CU. YDS.
OB/COAL    =       3.8 CU.YDS/TONS.

FILL VOL   =     5495.4     5213.0     5354.2
```

BLOCK D/II

Figure 9

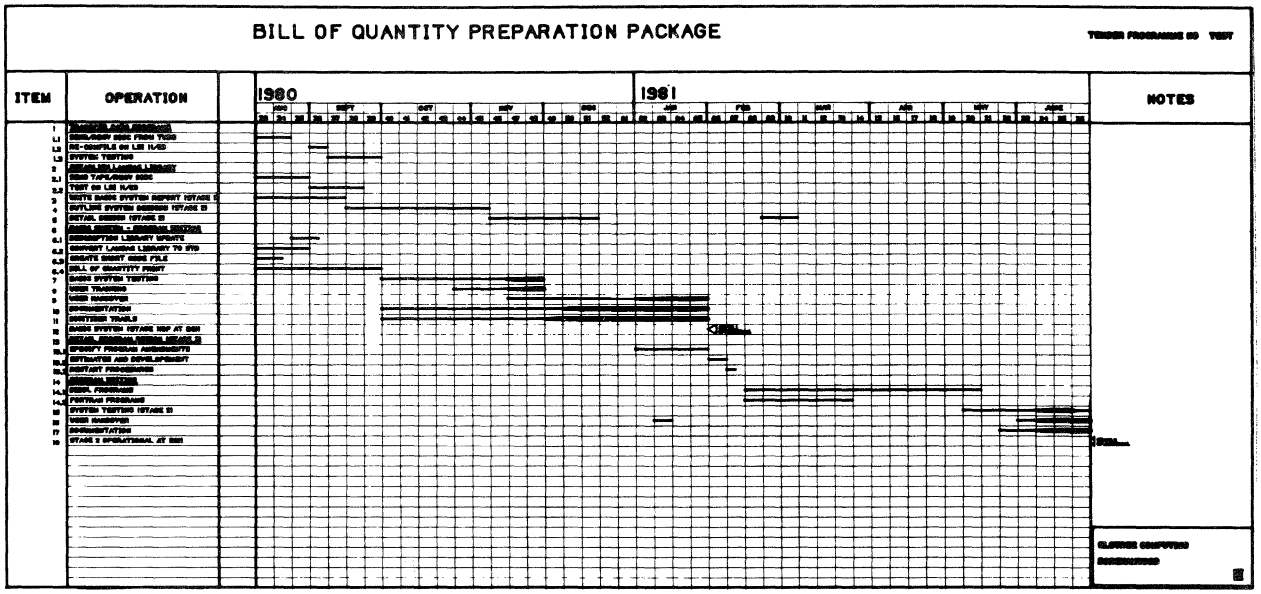


Figure 10

8. Project Bar Charts

A PEST symbol has recently been produced by the JLDA draftsmen to facilitate the rapid production of project bar charts as shown opposite. It is necessary for the Group to produce a number of these charts each week to a consistent high standard for bid submissions, etc.

The user is prompted to enter the obvious project variables:

name
start week number
duration
holiday weeks

and then for each activity:
description
start week
finish week

A typical bar chart such as that shown opposite in Figure 10 with approximately twenty activities can be completed in about 6–8 minutes from the basic details provided.

SUMMARY

The development of PEST over the past four years has been a highly successful project, contributing significantly to the success of our CV installation. It is now the foundation upon which all our computer-aided drafting activity is based. A major factor in the acceptance of PEST by the user, apart from its speed of operation, has been the ease with which draftsmen with no previous programming experience have been able to develop their own particular PEST symbols. Typically, within three days of starting to learn PEST, a graphics draftsman will have successfully completed his first, albeit relatively simple, interactive PEST symbol. We find that once a draftsman passes this stage and has accepted and understood the way PEST operates, he quickly attains skill in developing complex symbols.

Consider speed of operation. Generally speaking, the formation of the graphics is almost instantaneous. Obviously where the symbol is extremely complex in terms of calculations or the amount of graphics to be generated, insertion of the symbol takes longer. It is not possible to give any comparative figures based on our own observations as we do not have PEP, VARPRO or PAREX. However, PEST is now in use in two other CV installations in the UK.

They tell us that PEST typically achieves a 10 to 1 saving in the generation of a symbol at run time compared with previous methods. Also the coding of PEST symbols is considerably simpler and less time-consuming.

But perhaps the greatest benefit from PEST lies in its versatility. Not only can the software deal efficiently with a wide range of applications as shown, but it also increases the versatility of our draftsmen. Through the use of PEST, they too are becoming skilled in the cost-effective application of graphics over an increasingly wide range of applications. That has to be good news for both the Laing Group as a whole and for the draftsmen themselves.

ADVANCED PROGRAMING TECHNIQUES UTILIZING EXECUTE AND PAREX

John Richards
Boeing Commercial Airplane Company
Valley Office Park
1801 Lind Avenue S.W.
Renton, WA 98055

Abstract

The subject of this paper is Advanced Programing Techniques and it is not my intention to describe detailed programs. I shall be describing techniques which can be utilized to achieve program functions that Execute and Parex are not generally thought capable of achieving. Anyone wishing to discuss detailed ideas may contact me and I will be pleased to assist in whatever way possible.

By way of introduction, I will describe some basic ground rules for approaching the subject of programing effort on CV. This approach has been used with great success during my association with the Boeing Company, and the potential for time and cost savings is enormous. The measure of a user's success in programing efforts will be a direct result of the choice of personnel for the task and the method of utilizing these personnel.

The CV CADDS software is an extremely versatile package, which coupled with the simplicity of Execute/Parex, produces an exceptionally powerful tool. At this juncture, I note that PEP, which is another whole field of techniques, some superior and some inferior, is not considered in this present paper.

The essence of advanced programing is "know your system." Programmers must be experienced and familiar with the software, both CADDS and operating system. They should also have an operator's knowledge of the problems they are trying to solve, and should develop a broad knowledge of the commands from areas other than their own specialty. Furthermore, they must know their terminal and its capabilities intimately. Most of all, they must be interested and enthusiastic and will probably already have some good programs under their belts.

Their personnel should be situated near each other and encouraged to share ideas and problems. Their ideas and suggestions should be considered carefully and when necessary they should be given the task of producing simple demonstration programs to prove basic ideas and requirements within reasonably light but flexible time scales.

Finally, they must never pass over error messages, but always investigate their mistakes. Sometimes, errors are not great and in any case can often be useful features. Contrary to popular belief, the software rarely, if ever, makes an error and almost everything works correctly if one knows how to use it. I would also note that there are a lot of unknown and undocumented features to some commands.

There will almost certainly be a host of personal programs on most systems; however, for programs to be suitable for use by all operators of varying familiarity and ability, they must have a single, friendly operator interface. The Lear Siegler terminal is probably the most effective for this, expecially with the simple face field cursor conversion.

Various techniques may be used in Execute/Parex to write to the terminal from CADDS or the operating system. Asterisks and bells are a pain and a bore to those trying to use a program and their neighboring workers, so avoid them. Programs must give instructions and messages and the best method is to have a message text file. To print a message, merely print those lines required with any suitable modifiers to give a good clear readable message. An ASCII control-Z may be patched into a file to echo a shift-clear to the beginning of a message. This looks good to the operator and avoids confusing text remaining and paging up the screen. One advantage of the message file is that it allows the messages to be edited and structured neatly and easily without the danger of messing up the program at the same time, and may be completed after the program itself. The select noecho and echo features are very useful here because they allow a message to remain on the screen to be read while the program is doing something useful, and since the baud rate of the terminal is much slower than the CPU, it also speeds execution.

Messages may also be used to expand Parex prompts by printing instructions and questions such that the cursor may be moved up the screen (control-K and control-J do this) to make a subsequent Parex file line up its prompts with the questions. The first line of a Parex program can be a shift-clear to blank the screen after the last parameter entry. To execute a shift-clear in CADDS all that is required is * ∧ Z.

Progressing on to more complex programing, the obvious omission from Execute/Parex is conditional statements. There is one true conditional and that is the X OPTION command in the operating system, but this has distinct limitations. However, conditionals can be built by other means. Since Execute/Parex programs can be nested (and therefore parameters can be passed down from one Parex to another) the simplest conditional is to have a Parex parameter as a file name to be executed. This, again, has limitations and often degenerates into menu picking of numbers, which is wearing and unnatural to CV oriented operators.

There are five modifiers to run Parex; they are NFPLACES, which everyone knows, TARGET, NOGO, CYCLE and PRINT. The last four are extremely useful. Programers must get to know these and their combined effects. Since "TARGET" allows Parex to make a hardfile, a data text file can be produced instead of a parameterised execute program. This can store input for use later in a program as data or conditional information. Extraction of this information can be done in CADDS by merge file and run rearrange, etc., or in the operating system by the editor or CVTECO. Note that Teco is worthwhile learning, as it is very powerful and has many conditional features of its own. Therefore, by using data text files and conditionally transferring information to other files, one can very effectively modify later nested programs, create them from scratch or produce a conditional branch by executing a file later in the main program which may by then be blank or execute a different file. This sounds rather complex but is in fact simple and amazingly fast and efficient.

There are two important things to know at this point. One is never to try to execute a file name that does not exist, otherwise CADDS error 135 may bomb the program; always ensure a dummy blank file exists. Secondly, it is very simple to delete a text file from CADDS level. This is achieved by merely running any Parex file with a target name of the file to be deleted. After typing OK to replace and obtaining the PARAMS: prompt, the process is aborted with a control-E. This also serves to illustrate the point that many commands can be used quite legally to achieve functions they were not designed for, provided the operator knows how they work.

Another common requirement for programs is the extraction of data from a CADDS post. This can be achieved in many ways. Data Extract is far from ideal for many user's requirements, partly because it is not foolproof, requiring correct and often complex part construction to start with. Use of Hardfile will allow almost all data that can be written to the terminal to be extracted and used under programs controlled by merge file, edit, Teco, etc. At present, it is necessary to exit to the operating system for this and there is a desperate need for the facility to toggle a hardfile in CADDS 3. Not only will this allow a program to extract CADDS part data quickly, but also time, date, task number, etc. Some powerful accounting data can be created in this manner.

Having introduced some basic techniques, I shall now briefly discuss Parex before continuing with a description of some actual programs to illustrate what can be done. Parex is a very clever piece of software. The total documentation available does not describe one quarter of its features. (I have a sneaking suspicion that the programer left before he documented it!) There is not time to give a detailed full description here; perhaps I will find the time to write a T.I.P.S. sheet on the subject. However, suffice it to say that, in addition to the features I have already mentioned, parameters may be entered by name and value after the first carriage return and a comment may also be entered in quotes. It should be noted here that text storage, comments, file names and printouts of files all echo the full ASCII character set within the limitations of the terminal type. Another way to obtain comments in Parex parameter entry is by using dummy text parameters, thus Control-Z and Control-J allow clear single question messages in the center of a blank screen.

Utilizing these and other techniques has made it possible to write some very sophisticated and cost effective programs during my association with Boeing Company. These programs are usually a combined effort by several people and as such should not be credited to any one individual. Users will find that this team approach is most effective even though one person usually does most of the actual writing of a program.

One of these programs will insert a drawing border format correctly on a CADDS post, first time, by giving trial ghost images. This is achieved with Parex to input drawscale to the getdata calculator in a verify position command. This command draws a screen trace between positions and is then aborted by ∧ E before outputting the position data. (There are better ways of achieving screen display vectors but they require fortran programs not available outside Boeing.) This command can be used out of a keyfile by calling tablet raster positions to give a screen image which disappears on reprinting and is independent of the screen display scale. By this means a menu picture can be displayed, which overlays the tablet keyfile positions, wihtout the need to call up a CADDS part.

My second program example is a CADDS file management system. This utilizes a similar screen image to identify ten keys. It stores the current definitions in ten of the thirty six spare keys. (The manual is wrong; a total of 256 keys may be defined.) The program then gets keys on a ten key keyfile and before the program ends, it returns the operator's original key definitions. Incidentally, get keys is only a glorified execute command, and other commands may be inserted in a text keyfile before the edit statement and after the Q, and will be executed during the get keys, for example, an automatic 'sav key'. The CADDS file manager maintains a list of operators with information about them which may be listed. It incorporates new users automatically and assigns them an ID number. It creates, lists and manipulates an indexed list of part files with drawscale and archive information for each operator, and can call up any of these parts or identify files to be archived, all by using only the index numbers. It can produce paper documentation for retrieval of archived files and cause execution of any optional programs the user wishes to add and pass information about parts and users down to that program.

The third example is a program that will create and fix out a full drawing border format, file the part under a special file name convention, calculate all the information required to transmit and store the file on a mainframe database, and produce all the company documentation required to release the drawing. These documents include data set release forms, drawing release forms, design change notes, parts lists (without data extract) and wet ink plot requests forms. The program asks the operator which of these features he requires and asks up to fifteen questions. At that point, it goes on to produce all the documents and a versatec check plot of the drawing. All the other information is extracted from the CADDS file manager, the part file and data text files. The parts listing feature is not currently completed, but the technique used is to insert primary drawing callout labels with a parex file. This attaches properties to the label. Hardfiling a listing of these properties by windowing the entire drawing and using the Teco character editor with the merge, sort, tally and total commands produces a parts list without the need for nodal entities or figures, and the list matches the drawing callouts.

Other programs being written produce a two-man day duct drawing in about twenty minutes from center line, diameter and bend radius information from a mainframe printout, and select make-from parts from a parameter input list.

These are examples of what can be achieved. The programs are fast, accurate, and above all, time saving. Some users produce large fortran programs to achieve similar results. The time and cost involved in debugging, proving and updating these is often questionable. A good approach which has been highly successful in practice is to write the programs in execute/parex/PEP, then prove and update them, and above all use them immediately. Only then, when a known and proven set of requirements is available, spend time and money on a fortran program to replace the other version when it is available and if it is still justified. There is another use for fortran programing where it is available. That is to produce simple building brick routines for use in execute/parex programs to avoid complex routines and save time.

My aim in writing this paper was to introduce a few advanced techniques, and some basic advice on implementation of programing efforts together with some examples of what can be done to whet the users' appetites. Some users will take up the challenge, and some already have. I hope we will contact each other, if only informally, and soon I expect to be learning from someone else's discoveries as well as teaching them mine. I do not buy a dog to bark myself, so I want to see my system work for me and I hope to have inspired others to do the same. When we see smoke rising from the hardware, we will know we are getting maximum returns from our equipment! We will never get that far, but let us keep trying.

Next Conference

TOWN & COUNTRY HOTEL
SAN DIEGO, CALIFORNIA

September 13-17, 1982

COMPUTERVISION
100 Commerce Way, Woburn, MA 01801
(617) 935-9723