

COMPUTER PROGRAMMING SERIES

**DIGITAL**  
**COMPUTER GRAPHICS**

VOL. I

PUBLISHED BY  
**AMERICAN DATA PROCESSING, INC.**  
19830 MACK AVENUE  
DETROIT, MICHIGAN

© Copyright by American Data Processing, Inc.

PREPARED UNDER CONTRACT NO. NAS 8-21227  
By LOCKHEED MISSILES & SPACE COMPANY,  
SUNNYVALE, CALIFORNIA

## ABSTRACT

This document represents a study of data collection, application analysis, and user requirements for the following five computer graphics applications:

- Numerical Control
- Electrical Network Analysis
- Flight Mechanics
- Structural Analysis
- Engineering Drawing Retrieval

A discussion of "Study of Man-Machine Solution Optimization" and a hardware analysis report of the MSFC Univac 1108 computer complex using Univac specifications are also presented.

The results of a Computer-Aided Design/Computer Graphics literature search is compiled into an annotated bibliography as Volume II of this report. Conclusions and recommendations in respect to computer graphics and the various areas of computer graphic applications are summarized.

This study was prepared by Lockheed Missiles & Space Company under Contract NAS 8-21227, "Digital Computer Graphics," for the National Aeronautics and Space Administration, Marshall Space Flight Center.

## ACKNOWLEDGMENTS

The following LMSC personnel made major contributions to this document:

P. W. Card  
D. L. Drew  
W. R. Dunn  
S. Elster  
S. K. Ferriera  
K. R. Gielow  
J. F. Golden  
F. S. Kessler  
M. A. Krop  
D. C. Miller  
R. L. Moll  
N. E. Nielson  
W. C. Randels  
C. J. Skogh  
P. L. Taulbee  
D. L. Trask

Significant contributions were also provided by D. A. Biederman of Lockheed California Company and M. M. West of Lockheed Electronics Company.

## CONTENTS

Section		Page
	ABSTRACT	iii
	ACKNOWLEDGMENTS	iv
	ILLUSTRATIONS	ix
	TABLES	xiii
	INTRODUCTION	xv
1	NUMERICAL CONTROL	1-1
	1.1 Numerical Control (N/C) and Computer Graphics Involvement in the Industrial Process	1-2
	1.2 Feasibility of N/C - Computer Graphics at MSFC	1-16
	1.3 Numerical Control and N/C - Computer Graphics Specifications	1-20
2	ELECTRICAL NETWORK ANALYSIS	2-1
	2.1 Summary	2-2
	2.2 Network Analysis Program Requirements for the Graphics System	2-2
	2.3 Description of Available Network Analysis Programs	2-4
	2.4 Program Evaluation	2-11
	2.5 Program Selection	2-13
	2.6 Program Modifications	2-13
	2.7 User Graphic Procedures	2-15
	2.8 Network Analysis Graphics Applications at Lockheed	2-21
	2.9 Network Analysis Graphics Applications at Other Companies	2-22

Section		Page
	2.10 Programming Requirements	2-23
3	<b>FLIGHT MECHANICS</b>	3-1
	3.1 Mode of Application of Graphics	3-1
	3.2 Graphical-Computer System for Orbital Transfer Studies	3-4
	3.3 Requirements Specification for Orbital Transfer Studies	3-15
	3.4 Graphical-Computer System for Ascent Boost Optimization	3-29
	3.5 Requirements Specification for Ascent Boost Optimization	3-50
	3.6 Conclusions and Recommendations	3-55
	3.7 References	3-56
4	<b>STRUCTURAL DESIGN ANALYSIS</b>	4-1
	4.1 Background	4-1
	4.2 Introduction to Specifications	4-17
	4.3 Program Requirement Specifications	4-19
	4.4 Hardware/Software Requirements	4-32
	4.5 Program Specifications	4-32
	4.6 PEANUTS: An Example of Graphic Program Requirements and Specifications	4-70
	4.7 References	4-87
5	<b>ENGINEERING DOCUMENT RETRIEVAL</b>	5-1
	5.1 Introduction	5-1
	5.2 System Requirements	5-2
	5.3 Storage and Retrieval Systems	5-20
	5.4 Recommendation	5-37
	5.5 Hardware State-of-the-Art	5-78
	5.6 References	5-83

Section		Page
6	STUDY OF MAN/MACHINE SOLUTION OPTIMIZATION	6-1
	6.1 Introduction	6-1
	6.2 Description of Program	6-2
	6.3 Experiment Data and Operations	6-4
	6.4 Instructions	6-22
	6.5 Suggestions for More Efficient Division of Labor Between Man and Computer	6-23
	6.6 Simulation	6-24
7	HARDWARE ANALYSIS	7-1
	7.1 Introduction	7-1
	7.2 Hardware System Configuration	7-1
	7.3 Software System	7-3
	7.4 Display Operations	7-4
	7.5 Conversion Considerations	7-6
Appendix	GLOSSARY	A-1





## ILLUSTRATIONS

Figure		Page
1-1	Symbols Defined	1-38
1-2	Subroutine START	1-39
1-3	Subroutine POINT	1-40
1-4	Subroutine LINE-POINT-TO-POINT	1-41
1-5	Subroutine LINE-POINT DIRECTION	1-42
1-6	Subroutine CIRCLE	1-43
1-7	Subroutine PARALLEL	1-44
1-8	Subroutine FILLET	1-45
1-9	Subroutine RETOUCH	1-46
1-10	Subroutine LINE TYPE	1-47
1-11	Subroutine TRANSFORMATION	1-48
1-12	Subroutine NUMERICAL CONTROL	1-49
1-13	Subroutine DIMENSIONING	1-51
1-14	Subroutine MACRO	1-52
1-15	Local Area Layout of Display-Oriented Support Equipment	1-82
2-1	Typical Component Templates	2-17
2-2	Completed Schematic	2-18
2-3	Proposed Control Surface Layout	2-24
2-4	Proposed Keyboard Layout	2-26
2-5	Sample Component Group Structure	2-32
3-1	Orbital Transfer Display Scope	3-9
3-2	Ascent Boost Optimization REVIEW Display	3-39
3-3	Ascent Boost Optimization MONITOR Display	3-45
4-1	PEANUTS – Structural Design Evolution	4-12
4-2	Data Input Display	4-23

Figure		Page
4-3	Rectangular Communication and Data Display Layout	4-28
4-4	Square Communication and Data Display Layout	4-28
4-5	Shifted Square Communication and Data Display Layout	4-29
4-6	Communication Message Area Layout	4-29
4-7	Frame Types	4-31
4-8	Application Data Base Structure	4-34
4-9	Data Base Access Program	4-35
4-10	Resident Name Table (RNT)	4-37
4-11	Program ID Table (PIDT)	4-38
4-12	Case ID Table (CIDT)	4-39
4-13	Data Requirement Table (DRT)	4-40
4-14	Program Data Reference Table (PDRT)	4-42
4-15	Sample Data Set Format	4-43
4-16	Library Program	4-45
4-17	Sign-on Program	4-46
4-18	Stop Program	4-47
4-19	Program Selection Program	4-49
4-20	Program Next Case	4-50
4-21	Input-Output Flow for Card, Magnetic Tape and Data Base	4-51
4-22	Input Data Flow for Cathode-Ray Tube	4-52
4-23	Input Program	4-53
4-24	Card Input Program	4-55
4-25	Tape Input Program	4-56
4-26	Data Base Input Program	4-57
4-27	CRT Input Program	4-58
4-28	Compute Program	4-61
4-29	Standard Output Program	4-63
4-30	General Output Program	4-64
4-31	Void Output Data Program	4-65
4-32	Cathode-Ray Tube Layout	4-68

Figure		Page
4-33	Functional Block Diagram of an Ideal Structural Program	4-71
4-34	Functional Block Diagram of PEANUTS	4-72
4-35	Function Input	4-81
4-36	Single Item Input	4-82
4-37	Standard Output Display	4-86
5-1	Documentation Repository Input	5-7
5-2	Engineering Documentation Released by MSFC Release Center and Forwarded to Central Repository	5-8
5-3	Documentation Repository Output	5-14
5-4	Program Distribution of Manpower at MSFC	5-17
5-5	AAP Documentation Processed by MSFC Engineering Release Center Between June and December 1967	5-19
5-6	Documentation Flow	5-21
5-7	Sample of MSFC Form 2896 – Document Input Record	5-27
5-8	Sample of MSFC Form 433 – Request for Documentation	5-28
5-9	Sample of MSFC Form 2598 – Distribution Data Acquisition List	5-29
5-10	Release and Accounting Procedure Flow	5-31
5-11	Saturn SA-5 Main Assemblies	5-43
5-12	Breakdown of Saturn SA-5 Payload and Stage I Assemblies	5-45
5-13	Logical Flow Chart Segments SL1, SL2	5-56
5-14	Logical Flow Chart, Segment S	5-58
5-15	Logical Flow Chart, Segment D	5-60
5-16	Logical Flow Chart, Segment E	5-61
5-17	Logical Flow Chart, Segment U	5-63
5-18	Logical Flow Chart, Segment I	5-64
5-19	Logical Flow Chart, Segment A	5-66
5-20	Logical Flow Chart, Segment DL	5-68
5-21	Logical Flow Chart, Segment R	5-70
5-22	Logical Flow Chart, Segment IN	5-72
5-23	Typical Display – Interface Structure	5-73
5-24	Typical Display – Digitized Drawing	5-74

Figure		Page
5-25	Typical Display -- Structure Breakdown	5-75
6-1	Flow Diagram Used for the 144 Experiment	6-3
6-2	Complete Data Display	6-6
6-3	Display of Data Before Acceptance Into Data Set	6-7
6-4	Display of Functions of 10 Best Curves	6-8
6-5	Display of Best Curve	6-9
6-6	Display for User to Select a Function, Transform and Degree	6-12
6-7	Display of Chosen Curve From Box I, Fig. 6-1	6-13
6-8	Run 1, 144 Curve Fit Program, Preplot and Curve	6-16
6-9	Run 1, Computer Output	6-17
6-10	Run 2, 144 Curve Fit Program	6-18
6-11	Run 2, Computer Output	6-19
6-12	Run 3, 144 Curve Fit Program, Preplot and Curve	6-20
6-13	Run 3, Computer Output	6-21
7-1	Hardware Configuration	7-2
7-2	Display Console With Permanent and Programmed Controls Shown	7-5

## TABLES

Table		Page
1-1	Console Procedures for Start (Keyboard Oriented)	1-21
1-2	Console Procedures for Numerical Control	1-22
1-3	Console Procedures for Erase	1-24
1-4	Console Procedures for Line Type	1-24
1-5	Console Procedures for Fillet	1-25
1-6	Console Procedures for Rétouch	1-25
1-7	Console Procedures for Parallel	1-26
1-8	Console Procedures for Circle	1-27
1-9	Console Procedures for Point	1-28
1-10	Console Procedures for Line Through Point at Direction	1-28
1-11	Console Procedures for Line Point to Point	1-30
1-12	Console Procedures for Dimensioning Information	1-31
1-13	Console Procedures for Geometry Macro	1-34
1-14	Console Procedures for Transformation	1-36
2-1	Network Analysis Programs	2-5
2-2	Network Analysis Program Evaluation for Graphics	2-14
3-1	Sample Output of Orbital Transfer Program	3-19
3-2	Timing and Storage Requirements for Orbital Transfer System	3-23
3-3	Light Button Actions for Orbital Transfer System	3-27
3-4	Storage and Timing Requirements for Ascent Boost Optimization	3-53
4-1	Representative Listing of Structural Analysis Computer Programs	4-2
4-2	List of Input Requirements	4-22
4-3	Command Table	4-66
4-4	Register Table	4-69
4-5	Description of Input Quantities	4-73

Table		Page
4-6	List of Possible Output From PEANUTS	4-79
5-1	Summary of R&D and IO Usage	5-13
5-2	Six Most Active Requestors - October 1967	5-15
5-3	Machine Readable Documents	5-77
6-1	Time Elements	6-11
7-1	Hardware Comparison - CDC 273 vs. Univac Mariner Display	7-7

## INTRODUCTION

This country's scientific and technical environment increases in proportion to the design and complexity of our space program hardware. The resulting data which represent research, development, and design effort have become less accessible because of the sheer mass of information that must be stored, manipulated and retrieved. As a result, the data originator is literally separated from the information he needs and the cleavage increases as the data collection grows.

With the advent of on-line techniques and direct access devices the problem is diminished, but only in a limited manner in the area of alphanumerics, i. e. , the advisability of using on-line devices to examine mathematical equations is questionable when the changed data output must be transferred to hard copy and compared against graphic representations. This is a time-consuming iterative process. Graphics computer input devices that allow this iterative process on-line in an interactive, real-time mode are the keys to meaningful use of the computer as a tool to support the originators of information. Data capture at its initiating source rather than at the first paper disclosure as is presently inferred by the term "source data" is a prime goal in computer graphics. Some of the same benefits and advantages which are only promised in non-graphic, computer-aided design can actually be realized with the addition of graphics. These benefits include:

- Ability to observe the results in graphic form without having to resort to hard copy output
- Immediate program interaction capability
- Ability to revise data directly without a manual data reduction step by allowing users to work with problems in the graphics environment
- Ability to drive other output devices directly and produce hard copy or program tapes to operate such units as numerically controlled machine tools

Computer graphics at Lockheed has been conducted primarily on Lockheed independent development funds. While Lockheed interest in computer graphics dates back to 1958, the present operation, which began in 1962 at LMSC, is an outgrowth of an attempt to improve the design effort through computer graphics utilization.

Most of the early effort was concentrated on system evaluation and use of a printed circuit design program as the first application of software. Subsequent efforts have included such applications as reentry vehicle design, logic design, structural dynamics, and trajectory analysis.

Lockheed-Georgia Company and Lockheed-California Company have been actively engaged in this area on both independent and joint cooperative ventures within LAC. Lockheed Electronics Company (LEC) is examining applications presently relative to entry into graphics. Lockheed-Georgia Company has research-oriented applications in progress, both in computer-aided design and in computer graphics. The computer graphics programs include structural analysis, electrical circuit layout, and numerical control center path development for the operation of numerically controlled machine tools. At present, parts are being fabricated for the C5A program from input using a computer graphic terminal. Applications have been developed for both aircraft and shipbuilding operations. Lockheed-California Company has four major areas of computer graphics applications: design, design analysis, flight test, and numerical control of machine tools.

This report is segregated into eight major areas:

- Five computer graphic applications (see below), discussion of data collection and the application analysis, and a statement of user requirements ( a requirements specification) for each application:
  1. Numerical Control
  2. Electrical Network Analysis
  3. Flight Mechanics
  4. Structures Analysis
  5. Engineering Document Retrieval



- A discussion of "Study of Man-Machine Solution Optimization"
- A hardware analysis report of the MSFC Univac 1108 computer complex in MSFC Computation Laboratory taken from the Univac specifications supplied to LMSC especially for this study
- An LMSC Computer-Aided Design/Computer Graphics literature search bound separately as Volume II of this report

Each of the five computer graphic applications contains terms which are common to all along with special terms peculiar to the application. These are grouped together as a glossary for the total report in the appendix.

When applicable there is a summary of conclusions and recommendations, in each of the applications described herein.

The computer graphic hardware referred to in this report (and upon which all program applications are based) need not be of any specific manufacture or type, but generally consists of, as a minimum:

- A basic computing machinery (mainframe)
- A cathode-ray tube with a light-sensitive probe (light pen or pencil) with which the user points at the CRT and manipulates graphic lines or alphanumeric data plus certain auxiliary input devices such as a "button box" or keyboard (both of which may or may not be capable of input of alphanumerics or special graphic symbols and commands)
- Peripheral hardware for input and output such as a card punch, card readers, high speed printers, collators, sorters, tape drivers, etc.

The main consideration in all of the following is that the least complex hardware array allows the user/operator direct interaction with his program which is available as a CRT display.



Section 1  
NUMERICAL CONTROL

This section covers a portion of NASA-MSFC study contract NAS 8-21227. With deletions specified during negotiations, it is responsive as described in the LMSC proposal No. 699516, dated February, 1967.

The firms from which information for this study was gathered are:

- General Electric Industrial Process Control Division
- Gerber Scientific Instrument Company
- International Business Machines Corporation
- Lockheed Georgia Company
- Lockheed California Company
- Sundstrand Machine Tool Division of Sundstrand Corporation
- Bendix Industrial Controls Division
- Cincinnati Milling Machine Company
- Control Data Corporation
- Pratt & Whitney Machine Tool Division
- McDonnell - Douglas Aircraft Corporation
- Bunker Ramo Woolridge Company
- TRW, Incorporated
- Boeing Aircraft Company
- Kearney and Trecker
- Universal Drafting Machine Company

Society affiliations employed in this effort include the following:

- Aerospace Industrial Association
- Society of Automotive Engineers
- American Ordnance Association
- American Institute of Aeronautics and Astronautics

Another source of information was the Illinois Institute of Technology – Research Institute.

This document is intended to be comprehensive enough to facilitate successful implementation of Computer Graphics – Computer Aided Design – Numerical Control as one system when combined with an appropriate operating community and compatible hardware and software.

It should be recognized that some of the detailed information contained herein is in a state of change or expansion. The authors have attempted to provide enough information to enable creation of a working system, but they could not hope to foresee all future refinements.

## 1.1 NUMERICAL CONTROL (N/C) AND COMPUTER GRAPHICS INVOLVEMENT IN THE INDUSTRIAL PROCESSES

### 1.1.1 General Discussion

Numerically controlled machining, processing, and testing devices have become so sophisticated that many users now enjoy a high degree of precision. Also, the interfacing of computing equipment and computer graphic presentations to the N/C function allows man to develop a concept, review it, analyze it, modify it, and construct the manufacturing tool in a completely dynamic environment. It is also significant that, in addition to X-Y plotting equipment, another aid to man-machine communication is computer-interfacing, precision line drawing equipment, which enables a comfortable documentation transition period and completes the total industrial process.

The techniques of the use of computer graphics by the engineering community is the subject of a narrower field of study because of the many varying needs for data manipulation. However, the greatest realization of the potentials offered by computer graphics and numerical control will come from an industrial process that essentially eliminates dependency upon man for any interpretive roles beyond design completion and choice of production method. Realization of the full computer graphic/engineering

potential depends on the possession by the design engineer/operator of a complete understanding of the computer graphic environment. Only a few design engineers are now taking advantage of the existing new capabilities offered by data automation and computer graphic systems. It is therefore important that manufacturing and engineering personnel receive complete orientation in the latest disciplines.

A unique capability becomes available with use of the CRT and light pen for design engineering communication with the computer, and use of the same CRT and light pen for N/C parts programming, coupled with non-interpretive use of N/C graphics equipment, alphanumeric printers, punched tape, film, magnetics, electrostatics, etc., for documentation of any or all results.

One of the objectives of this study is to arrive at specifications which enable establishment of a working system to accomplish these ends. Existing technology provides the capability for implementing all the aforementioned technologies into one integrated system. It is improbable, under the present Engineering and Manufacturing organizational structure in the industry, that the cost-effectiveness of numerical control can be fully demonstrated. Consequently, a considerable amount of the computer graphics-engineering-N/C portion of this total NASA-MSFC study is devoted to the involvement in the industrial processes and to answering such questions as the following:

- What are the optimum techniques for training personnel in the use of these computer-aided capabilities?
- What organizational structures offer optimum realization of the potentials of these new technologies?
- What are the most advantageous approaches to linking together the computer, the N/C control system, and the N/C tool?
- What techniques for introduction of the computer-aided capabilities into the engineering/manufacturing/operational communities offer the greatest potential for early acceptance and use?
- What communication techniques between the users and the computer systems offer the greatest potential?

- What growth potential is necessary?
- What new computer-aided design techniques will be developed in the next three to five years?
- What is the value/need of software interchangeability between various facilities?
- What is the opportunity to take advantage of the economies offered by use of existing software?

### 1.1.2 Answers to Questions Previously Posed

Each of the previously listed questions will be dealt with in the following. Information has been gathered through visits and conferences with members of the industry, through searching literature, and through personal research.

1.1.2.1 What are the Optimum Techniques for Training Personnel in the Use of These Computer-Aided Capabilities? Initially, those who need training must be identified; and the extent to which they will be trained must be agreed upon. Then the varieties of techniques can be evaluated to determine which offers the highest potential for success.

The degree of success of employment of a CG-CAD-N/C (Computer Graphics – Computer Aided Design-Numerical Control) system will be in direct proportion to the degree of accurate knowledge of the system held by: (a) manufacturing, (b) engineering, and (c) management. The weakest of the three will be the limiting factor for full realization of the potentials of the CG-CAD-N/C system. The technique of training each group will be different, and yet the overlap of understanding must be broad.

- a. The N/C manufacturing community (even more than the conventional) could much more easily and accurately program the creation of a part if the designers limited themselves to geometrics employing the locus of straight lines, circle sections, other conic sections, and oft-repeated special curves.

- b. Engineering personnel generally do not realize the creativity offered by an N/C manufacturing facility.
- c. Management must realize the necessity of the timely, dedicated support necessary to gain maximum potential from CG-CAD-N/C as a system.

Thus it is that Manufacturing, Engineering, and Management must each understand the role not only they will have, but in a large sense, the role the others will have also. This is the general training task that must not be overlooked as definition of skill-developing training commences.

It has been demonstrated that young men fresh out of school can most quickly become proficient with N/C-CG parts programming as well as with CG as a geometry defining tool. Therefore, it is desirable to concentrate on the young for training in these new technologies; however, consideration of the following elements causes the "educate only the youngsters" approach to be very suspect: 1) the environments into which they will be placed (controlled by the older generation), 2) the support that is needed to make them effective, 3) the indispensable need for acceptance by the established communities, 4) the wealth of practical experience held by the established communities, 5) the investments in each area. In looking around the country at several of N/C installations, it also becomes apparent that as the young men become proficient in this new technology, they are moving on to loftier goals and higher rewards, necessitating a continuous training program and continuing reliance upon recent arrivals.

Thus it becomes important to ensure that training in these new technologies is directed toward the established community as well as the young men.

The answer to the question posed, then, must factor in the foregoing. There must be at least three specifically tailored, comprehensive training and education efforts in each of three areas that obviously interlock with each other. Manufacturing, Engineering, and Management must be trained in:

- a. An understanding of the other's problems, needs, and capabilities
- b. Skill in use of these capabilities for older personnel

This training and education effort must be tailored to complement the recipient's background, study habits, capabilities, and motivations to realize a high degree of effectiveness. A modicum of proficiency with the hardware involved must be realized at the same time as the capabilities and known techniques of use are portrayed to those involved in the training and education effort. The most effective training techniques employ several of the presently used tools such as manuals, lecture-demonstrations, and a wide variety of proprietary and vendor-supplied audio-visual aids. At least one system also employs the tutorial dialogue in which a statement is displayed upon the face of the cathode-ray tube describing the next piece of information needed by the system as the logical progression from the last information or command given.

1.1.2.2 Which Organizational Structures Offer Optimum Realization of the Potentials of These New Technologies? At the time any industrial facility involved in manufacturing, engineering, and computing initiates a capital investment in CG-CAD-N/C they should also endeavor to develop a fresh reevaluation of their approach to the total industrial process. The organizational structure that would be optimum 10 years from now is unlikely to be detailed today. It is also unlikely that many firms would undertake to restructure in one large step. Summing up these factors leads to the conclusion that continued success will require frequent reevaluation and faster response to new needs than many organizations have either given or experienced.

In studying the methods that effect organizational structure changes, it becomes axiomatic that the larger the organization the less likely it is to accept a radical change over a very short span of time. It is important, also, that existing product flow not be impaired or interrupted to any significant extent. It has already been demonstrated by the N/C users alone that those who make a strong, fast move to the highest possible (actual) use of their N/C tools realize a faster profit on their investment.

An example of the misconceptions existing in some parts of the N/C community is the idea that "N/C is unprofitable in small-quantity machining." The facts show that it is most profitable for short runs (less than 10 items) in small shops; and in large shops it can be shown to be at least as profitable if all costs of conventional manufacture are



equated objectively with costs of N/C manufacture. Support organizations must be correctly oriented in their roles in the CG-CAD-N/C implementation.

The computer sciences, engineering, and manufacturing disciplines must be encouraged to work together to realize maximum benefit from technical knowledge and economy of operation.

A vast difference exists between: (1) the current employment of a "batch mode" wherein time intervals between input and output of minutes, hours, and even days are tolerated (because nothing better was available) and (2) necessary insistence upon response in seconds or fractions of seconds by both Engineering and Manufacturing. A relatively small portion of the computer world has actually experienced the rigors and pressures of involvement in the real-time environment; and most of the systems that have experienced these rigors are special-purpose, specifically designed systems, owned and operated by independent organizations.

Of equal significance is the impact on engineering methods, including accustomed time frames and techniques of differentiation between design engineering, engineering support, administration, and such functions as manufacturing, engineering, engineering standards, specifications, preferred parts, and procurement.

It is possible for engineering personnel to train their own parts programmers and thus deliver to Manufacturing the tapes for the N/C tools, tool and fixture instructions, and material specifications. It is also possible for Manufacturing to take an engineering design of a piece part and redesign that part with superior functional weight, strength, cost, and schedule results using manufacturing techniques about which Engineering has no prior knowledge. Elimination of, or a significant reduction in, such lack of communication between manufacturing and engineering is an undeniable effect of a properly structured CG-CAD-N/C industrial process. As in Manufacturing, Engineering and support organizations must be reexamined with respect to their roles in new industrial processes.

**1.1.2.3 What are the Most Advantageous Approaches to Linking Together the Computer, the N/C Control System, and the N/C Tool?** Computer graphics is progressing toward a capability for enabling a design engineer's rapid communication with a computer for design concept evaluation and a resultant improvement in optimization before design release. Use of computer graphics for numerical control parts programming in a profiling operation (where depth is anything but curvilinear) has in 1967 become operational. The outputs of the CG-N/C parts programming are either basic tool movement geometry definitions or completely processed for particular N/C tools and ready-to-cut parts (in the case of N/C machining centers). This information can be in any of the several computer output forms such as punched tape, magnetic device, or alphanumeric printout. The control systems for N/C tools are generally of various levels of sophistication and accept input data from devices such as magnetic tape, punched tape, and, under recent development, direct computer-processor interface.

There are a number of efforts in progress to identify a computer for the N/C operation. Certain installations (Bendix, G. E., Bunker-Ramo-Woolridge, and Boeing) have been visited and were found to be in some phase of study or implementation of on-line computer/processor running the N/C tools. They fall into two distinct categories:

- A special control system that relies upon a central computer to do the interpolating and that runs the machine tool from that point forward
- A much simpler device that interfaces with a complete, conventional N/C tool control system where a tape reader would normally interface and that uses the local control system's interpolating capabilities

There are several advantages of the dedicated, on-line computer to manufacturing. They fall into these categories:

- Scheduling
- Elimination of routine use of punched tape
- Calibration and fault isolation
- Status reporting
- Local computation for simpler part programming
- Rapid turnaround

In most installations, no one of the above advantages will pay for a computer/processor capable of running 10 or more three-axis N/C tools simultaneously. A control system can be estimated at \$10,000 per axis, and a properly sized processor with necessary peripherals cannot be purchased for less than \$350,000. However, a system is established which can by-pass the punched tape reader, do calibration and fault isolation and some of the part programming work now being paid for elsewhere; improvements in the N/C tool utilization can be realized which can approximate the costs of computer lease and operation. By adding to this the worth of accurate, flexible scheduling and near-instantaneous statusing, it can be shown to be profitable to have a dedicated computer system for an N/C manufacturing facility employing 10 or more three-axis tools.

The initial configuration of the system to be profitably employed would operate on a three- or four-level priority interrupt basis — a software-hardware based technique interfacing with as many complete, local N/C tool control systems as there are tools. The interface would be at the output of the tape readers (which should be left switchable on the N/C tool for backup). The interrupt to the computer would be generated in the same way as the punched tape advance is initiated when buffer storage is unloaded. The computer would draw its increments of data from a combination of long- and short-term storage media such as magnetic tape and core. When not honoring interrupt commands, fault isolation routines might be going on. When neither N/C tool-generated interrupts nor fault isolation is going on, next-job orders might be prepared for moving tools, fixtures, materials, and instructions to arrive with a setup crew at an appropriate time before the previous job is completed. If these direct support functions of the N/C shop do not need to be done at any particular moment, the computer might honor a request for status or a parts program request and, if no other operational requirements are levied, scheduling information can be factored in to accumulate experience-updated commitments on every N/C tool. This technique of loading the computer has been demonstrated in other applications, and hardware for implementation has been in development for some time. The basic processor/computer systems are exemplified by the IBM 1800, the CDC 1700, and the GE 4020.

The initial tie-in of such a dedicated, on-line N/C computer to a computer used for computer graphics is obviously through hard-line disc, magnetic tape, or punched tape data transfer. Subsequent developments in the linking of the computer graphics to the

N/C tool are in various stages of development and experimentation. Generally quiet activities are going on with joint participation between computer-processor manufacturers, control system manufacturers, and N/C tool users.

1.1.2.4 What Techniques for Introduction of the Computer-Aided Capabilities Into the Engineering/Manufacturing/Operational Communities Offer the Greatest Potential for Early Acceptance and Use? In the preceding answers, part of the answer to this question has been given. Actual experience has shown a slow-rising exponential curve when linearly plotting elapsed time against effective use of the computer-aided capabilities.

The introduction of these new capabilities into the established community will best be done by establishing the strongest possible ties with familiar entities. The computer graphics hardware used by the various engineering and manufacturing personnel should be in or adjacent to the areas in which they already work.

The introduction must also be preceded by careful indoctrination and education of the management community.

The answers to the question of introduction techniques to be employed grows out of the need for a professionally administered, deliberate effort at pre-selling the capabilities at all levels, followed immediately by successful demonstrations by persons already known and respected by the audience in environments in which they feel secure. All of this is necessary to overcome the almost universal resistance to change:

1.1.2.5 What Communication Techniques Between the Users and the Computer Systems Offer the Greatest Potential? Experience with computer systems in operational situations has clearly demonstrated the value of communication with computer systems in a way that uses established languages and fast response. In terms of encouraging wide-spread use of computer systems, development of user-oriented languages has been most beneficial. The next most advantageous characteristic is that of the user's getting instantaneous responses to his requests of and instructions to the computer.

The employment of computer graphics as a communication medium with a computer system has already been demonstrated to be a very powerful technique. This is not surprising since man's most effective communication techniques involve some form of graphic symbolism.

Those who have researched the problems connected with communication between man and man, and between man and computer, conclude that man has significant advantages over a computer as a recipient of information. This advantage lies in the ability of man to (1) compensate for a faulty input by drawing upon partially related information and (2) bring more than one type of sensor to bear upon a phenomenon. Man also has an ability to question the sender when reception is not clear. The more nearly the man - computer communication approaches a man - man communication, the more reliance men will place upon the computer to do those things best done by computing equipment.

Awareness of the advantages of this approach has led to the development of "questioning" and trouble-shooting software routines (as accessories to basic software packages) which are automatically called up when the computer cannot continue with a process. Another technique of improving the communication calls for in-depth training of persons who will use a computer to learn a language the computer accepts.

In the use of computer graphics, two basically different philosophies were early employed for user communication with the computer. One used an elaborate accessory keyboard and the other used a simple keyboard and a listing (menu) flashed on the face of the cathode-ray tube from which the next desired information was manipulated by means of the light pen. Experience with these two and addition of some special features has resulted in a movement by each group toward a mix of menu and keyboard.

To use these systems, instructional reference manuals are being created. Early samples are enclosed with this study report.

Another technique that simulates man's preconditioning for the next sensation is the use of a "tutorial" system which displays on the CRT the next information needed by the computer to act upon the previously communicated information.

There is no doubt that a combination graphic-symbolic communication medium that preconditions both the man and the computer offers the greatest potential. Many isolated, and a few integrated, techniques for accomplishing this are being employed by groups realizing significant improvements in communication accuracy and efficiency.

In the case of N/C part programming, communication with the computer through CRT and light pen has proven effective from the viewpoints of cost, time, and accuracy. It's obvious superiorities over manuscript-card-tape communication techniques are today limited to applications involving two-dimensional curvilinear and third-dimensional fixed depth and slope activities. It is expected, however, that third axis curvilinear definition can be accomplished in 1968.

1.1.2.6 What Growth Potential is Necessary? From a management position, the answer to this question is important for capital needs and budgetary apportionment. It is obvious that once the capability of the new techniques of CG-CAD-N/C is known, there will be an increase in demand for system time. It is also obvious that it would be extremely costly to provide every man with a CRT and light pen at his desk (at least at present). Thus, a firm's most creative talent should be given initial access to the system so that their creativity can be expanded at the earliest possible moment.

Several factors have not yet been accurately determined or are still far from reaching a plateau of development in their area. These factors include the following:

- The amount of time a man remains productive when paced by a system that can consume and process data faster than man can conceive it – this raises the question of number of men per console.
- The number of CRTs (with light pens) that can practically be tied into one system.

- The continuing improvements in hardware and software enabling efficiencies that expand effective power without change in main frame hardware.
- Innovations such as data compression that expand the effective power of the system.

1.1.2.7 What New Computer-Aided Design Techniques Will be Developed in the Next Three to Five Years? In numerical control, it is not possible to deny the value of direct processor control of many machine tools simultaneously. With that advent will come fault isolation, machine tool loading, scheduling, and accounting as well as statusing, preventive maintenance direction and other functions better handled by machinery than men. Another area that will interface with, and be aided by, the dedication of a computer to the N/C operation will be adaptive control.

Because part of this study was devoted to ascertaining and understanding the highly probable forthcoming developments, it was found necessary to consider adaptive control; at least three companies are already deeply involved in its development. In brief, adaptive control involves sensing the cutting edge - work piece position relationships to enable near instantaneous modifications to motion - for the purpose of safeguards and more precision in manufacture. Sensors in this field are not yet capable of ideally accurate definition of position; but all the approaches being taken make a very intelligent approximation based upon sensing of a variety of reactions within the system, and then selecting an appropriate, preconditioned, immediate response from the machine and its control system. The potential for expansion of the sophistication of the adaptive control's response by employing a processor or computer with significantly more capability than the normal N/C control system computer's capability is apparent. The ability to achieve a significant improvement in precision in positioning and movement of an N/C tool (over today's state of the art) is a direct dependent upon more accurate, timely definition of true position relationship between tool and work piece. Thus, a manufacturing-dedicated computer/processor suggests a path that can lead to the inevitable demands for greater accuracies.

Tying an N/C manufacturing-dedicated computer to a computer graphics computer will enable greater accuracies and economies through reduced handling of data. The ties will initially be of the telephone grade transmission line category.

1.1.2.8 What is the Value/Need of Software Interchangeability Between Various Facilities? In many computer-employing installations, a lack of communication between developers of software results in duplications that are three-fold wasteful: (1) repetitious effort, (2) waste of creativity, and (3) restrictions on growth and improvement. Often continuance of this lack of communication or support for this continued duplication can be traced to hardware and software differences. Most experienced users of computers now realize the time and dollars of software conversion and thus make an attempt at hardware standardization; however, when this is done, management often still hears that employment of one group's software package by another group as an accessory to existing software (even though hardware is identical) can't be done without conversion costs. Searching out the reasons for this statement reveals that basic operating software used in the two or more installations is not identical even though objectives of data handling were stated to be the same. All of this has been, is being, and will continue to be experienced by computer users until a learned, well-defined, well-disciplined management of software/hardware systems is instituted.

The advantages of software interchangeability are many and include (1) effective extension of creativity, (2) conservation of computer resources, (3) reduction in programming costs, (4) better understanding by users of computer capability and needs because of need for tighter specifications to enable conformance to interface requirements, (5) back-up by multiple installations, thereby counteracting down time and work load peaks.

Much of this applies directly to N/C tools. It is desirable to be able to quickly define alternate tools upon which work can be done when the preferred tool is down or overloaded. There is no well established reason why post processing for more than one machine tool cannot be an automatic function in those designs that lend themselves



to computer graphic definition. It would be beneficial to have backup to parts programs to allow use of N/C tools located in another facility to which you can gain access.

Computer graphics software should be capable of interfacing with material displayed or produced in other facilities to enable efficient communication and eventual centralized call-up of most programs or drawings.

1.1.2.9 What are the Opportunities to Take Advantage of the Economics Offered by Use of Existing Software? There are so many considerations that must be satisfied to enable an overall working system of people, machines, and software such as CG-CAD-N/C that a considerable advantage can be gained by evaluating existing working systems before moving into new areas. Well documented, fully modularized software will reduce implementation time.

The value of early demonstration of an expandable, but minimally satisfying, overall system is difficult to ascertain. There are many advantages to employing selected existing software packages, at least at the outset, even though eventual modification or replacement of some of those packages is initially understood.

## 1.2 FEASIBILITY OF N/C-COMPUTER GRAPHICS AT MSFC

### 1.2.1 General

Consideration of N/C-Computer Graphics at MSFC falls into the following three categories:

- Economic feasibility
- Technical feasibility
- Functional feasibility

Economics will be discussed in the following subsection. Technical and functional feasibility considerations are discussed elsewhere in this document.

### 1.2.2 Economics

Numerical control parts programming by use of scope and light pen by itself allows a small percentage of improvement over conventional techniques; however, it can show a considerable time advantage over parts programming when used in conjunction with APT or some similar computer-employing, batch-process technique. The time taken for a man to copy a print by manually exercising a light pen is, of course, eliminated when the original design is developed using a light pen. In addition, it will be feasible to eliminate the many interpretive errors and dimensioning omissions now experienced in entering data from design sketch through engineering drawings to scope and light pen re-draw of original design.

### 1.2.3 Facets of CG - CAD Applicable to N/C Operations

Other sections of this document deal with the potentials of the applications of computer graphics to a variety of analyses and design efforts. In most cases, the end product of the analysis programs result in hardware of some type. The very fact that these hardware configurations are digitized will aid in the eventual conversion of the manufacturing processes to support this type of hardware to N/C. Three major areas are identifiable now that lend themselves today to a total CG-CAD-N/C system. They are described in succeeding paragraphs.

1.2.3.1 Electronics. Electronics will have considerable CG-CAD-N/C potential in several areas in which N/C manufacture can be employed. Those areas include:

- o Discrete component/solid state
- o Integrated circuit/microelectronic
- o Interconnections

Suggestions of the N/C applications in each of those specific areas are:

- a. Discrete component/solid state:
  - 1. Total etched board manufacturing processes
  - 2. Component selection and insertion
  - 3. Soldering/welding operations
  - 4. Subassembly integration
  - 5. Testing
  - 6. Enclosing, packaging, etc.
  - 7. Storing, retrieving, etc.
- b. Integrated circuit/microelectronics:
  - 1. Total photo-optic and mechanical operations in the manufacture of discrete components
  - 2. Integration of discrete components into logic configuration
  - 3. Integration of logic configurations into functional configuration
  - 4. Welding and soldering control by many precision techniques such as conventional micro point, unconventional electron beam, and laser beam
  - 5. Testing at many steps through functional system build up and final acceptance
  - 6. Enclosing, packaging, etc.
- c. Interconnections:
  - 1. Wire/cable sizing, selection, cutting, trimming, and tinning
  - 2. Connector selection, mating, welding/soldering, etc.

3. Continuity and insulation testing
4. Labeling
5. Large cable build up, forming, lacing, etc.

#### 1.2.3.2 Enclosures.

- a. Sheet metal forming, punching, bending, cutting, fastening, welding, etc.
- b. Solid material machining by cutting, electro-discharge, chem mill, etc.
- c. Plastic molding, curing, cutting, etc.

#### 1.2.3.3 Structural Members.

- a. All machining operations
- b. All welding operations
- c. Many assembly operations
- d. Many testing operations
- e. Raw material storage, retrieval, and sizing
- f. Finished part packaging, storage, and retrieval

#### 1.2.4 MSFC Computer Involvement in N/C

It will be a long time before computer graphics at MSFC is capable of N/C parts programming and another lengthy time period will follow that before all design and N/C parts programming is done with computer graphics. During this interval, the NASA-MSFC computer support of N/C operations can significantly affect the successful employment of N/C tooling. Card punch, printing, tape punch and computing operations are, by necessity, sequential operations. It is most desirable for NASA-MSFC Computer Operations to enable the shortest practical time spans between parts programmers' attention to each job. Both off-line and on-line activities are involved and special attention should be given to their needs.

If we hypothesize a situation in which a remote terminal with reasonable main frame interrupt capability and appropriate peripheral equipment is located in the N/C parts

programming area, some comparisons with conventional techniques of computer-employing parts programming can be shown. Assume that the remote terminal is a Univac 1004 remote to an 1108 which in turn has full APT and post processor capability. Let us further suppose that the 1004 has as its own, local peripherals: card read and punch, magnetic tape and/or magnetic disc read and write, and punched tape (both paper and mylar) read and write. In addition, a key punch would be set up in the same area as the parts programmer and the 1004 and its peripherals which, in turn, are located adjacent to the N/C tool area. With an interrupt system into the 1108 which essentially handles complete batches on a first in/first out basis, and with sufficient storage space in the 1004, jobs needing 1108 power can be loaded into the 1004, processed a few minutes later by the 1108 and output back into the 1004 for the format desired. Tape punching, card punching, alphanumeric printing, magnetic tape or disk-pack-to-punched-tape conversions, and verifications can all be done as needed by the 1004 without waiting for 1108 availability.

With this hypothetical facility, it becomes possible for parts programmers to reduce the number of parts programs they have in work from a high of 6 to 15 month to an average of three to four. The reasons for this lie in the ability to get key punch, computer runs, printouts, paper tapes, and all computer or computer peripheral-dependent products returned, completed in a few minutes or few hours as a routine instead of the present (without special priorities) minimum of one day for each activity that now exists in many installations. This kind of support will greatly enhance the posture of the Computer Services organization. It will demonstrate the problem to be expected in servicing dedicated, on line operations. It will demonstrate the worth of quick response to requests for off-line work. It will enable manufacturing to effect significant economies in the N/C organizations by cutting programming and tool try costs as well as lead times to finished programs.

## 1.3 NUMERICAL CONTROL AND N/C-COMPUTER GRAPHICS SPECIFICATIONS

### 1.3.1 Console Procedures and Flow Diagrams for Keyboard-Oriented Systems.

The console procedures (Tables 1-1 through 1-14) and flow diagrams (Figs. 1-1 through 1-14) are mutually complementary. A good basic understanding of the graphics system user's viewpoint and function can be obtained from these data. A glossary of terms used in the tables appears after Table 1-14.

The console procedures and flow diagrams herein form a solid basis for developing a final graphics system configuration. The configuration shown is a minimum basic package that can be used for engineering production design and drafting and numerical control programming applications.

The console hardware configuration for this system requires, in addition to the cathode-ray tube, three input devices: a function keyboard, a typewriter, and a light pen.

Console procedures and flow diagrams should be completed and agreed upon between the graphics user and the graphics software programmer prior to any coding. These will eliminate much misunderstanding by the user, serve as a basis for instructing users, and eliminate much costly reprogramming of software.

The most successful graphics systems have been derived by user-programmer teams developing and checking the applications programs jointly.

Table 1-1

CONSOLE PROCEDURES FOR START  
(KEYBOARD ORIENTED)

Step	User Action	Machine Reaction
1.	<u>FK START</u>	<p style="text-align: center;"><u>MENU DISPLAYED</u></p> <p><b>CALL</b> - Retrieve file (drawing) from storage device</p> <p><b>FILE</b> - File drawing on storage device</p> <p><b>ID</b> - Key in ID for starting new dwg</p> <p><b>CKPT</b> - (Checkpoint) file model on storage device for recall in case of bomb off</p> <p><b>RECALL</b> - Recall model from storage device after bomb off</p>
2. 3.	<p>LPD <u>CALL</u></p> <p>Key or pick <u>ID</u></p>	<p><b>CALL</b> underlined</p> <p>Message - KEY ID</p> <p>Retrieves drawing from storage device and displays drawing</p>
2. 3.	<p>LPD <u>FILE</u></p> <p>Key or pick <u>ID</u></p>	<p><b>FILE</b> underlined</p> <p>Message - KEY ID</p> <p>Files dwg on storage device under ID keyed in</p>
2.	LPD <u>CKPT</u>	<p><b>CKPT</b> underlined</p> <p>Outputs model on storage device</p>
2.	LPD <u>RECALL</u>	<p><b>RECALL</b> underlined</p> <p>Retrieves and displays last model checkpointed</p>

Table 1-2

CONSOLE PROCEDURES FOR NUMERICAL CONTROL

Step	User Action	Machine Reaction
1.	<u>FK NUMERICAL CONTROL</u>	<p><u>MENU DISPLAYED</u></p> <p>SETPT - Key in or LPD cutter starting position</p> <p>ZDIM - Key Z axis, value of bottom of cutter</p> <p>FEDRAT - Key or pick feed rate (velocity in inches per minute)</p> <p><u>POST</u> (Postprocess) Retrieve CL data from bottom of model and write APT CLTAPE.</p> <p><u>ABORT</u> - Abort present cutter path and exit from N/C mode</p>
2. 3. 4. 5.	<p>LPD <u>SETPT</u></p> <p>LPD point from display</p> <p>Pick or key <u>SETPT Z DIM</u></p> <p>Pick or key <u>CUTTER DIA</u></p>	<p>SETPT underlined</p> <p>Message - SETPTX =</p> <p>Message - SETPT X = XXXXX SETPT Y = XXXX SETPT Z =</p> <p>Message - SETPT Z = XXXXX CUTTER =</p> <p>Message - CUTTER = XXXXX</p>
2. 3. 4. 5. 6.	<p>LPD <u>SETPT</u></p> <p>Key <u>SETPT X DIM</u></p> <p>Key <u>SETPT Y DIM</u></p> <p>Key <u>SETPT Z DIM</u></p> <p>Key or pick <u>CUTTER DIA</u></p>	<p>SETPT underlined</p> <p>Message - SETPT X =</p> <p>Message - SETPT X = XXXXX SETPT Y =</p> <p>Message - SETPT Z =</p> <p>Message - SETPT Z = XXXX CUTTER =</p> <p>Message - CUTTER = XXX</p> <p>*SELECT DRIVE SURFACE</p>

\*MESSAGE - Select PT/ELEM



Table 1-2 (Cont.)

Step	User Action	Machine Reaction
2.	LPD <u>Z DIM</u>	Z DIM underlined Message - Z DIM =
3.	Pick or key <u>Z DIM</u>	Message - Z DIM = XXXXX *SELECT DRIVE SURFACE
2.	LPD <u>FEDRAT</u>	FEDRAT underlined Message - FEDRAT =
3.	Pick or key <u>FEDRAT</u>	Message - FEDRAT = XX *SELECT DRIVE SURFACE
2.	LPD <u>POST</u>	(APT INTERFACE) POST underlined Message - MACHIN/
3.	Pick or key <u>MACHIN STATEMENT</u>	Message - MACHIN/XXXXX
3A.	FK <u>ACCEPT</u>	SCALE =
4.	Pick or key <u>SCALE</u>	Message - SCALE = XXXX Writes CLTAPE to input to APT system
2.	LPD <u>ABORT</u>	ABORT underlined All N/C output deleted from the model - exit from N/C mode
2.	LPD <u>PT</u>	Cutter displayed at PT selected CL placed in model *SELECT DRIVE SURFACE
2.	LPD <u>ELEM</u>	Message - SELECT - CKSURF
3.	LPD <u>CKSURF</u>	Cutter displayed at intersection of elements Message - SELECT - CKSURF

\*MESSAGE - Select PT/ELEM

Table 1-3

CONSOLE PROCEDURES FOR ERASE

Step	User Action	Machine Reaction
1.	<u>FK ERASE</u>	Message - LPD ELEMENT
2.	<u>LPD ELEMENT</u>	Removes element from display and model

Table 1-4

CONSOLE PROCEDURES FOR LINE TYPE

Step	User Action	Machine Reaction
1.	<u>FK LINE TYPE</u>	<u>MENU DISPLAYED</u>  SOLID HIDDEN CENTERLINE PHANTOM
2. 3.	<u>LPD SOLID</u> <u>LPD ELEMENT</u>	Message - LPD ELEMENT Displays element as solid line
2. 3.	<u>LPD HIDDEN</u> <u>LPD ELEMENT</u>	Message - LPD ELEMENT Displays element as hidden line
2. 3.	<u>LPD CENTERLINE</u> <u>LPD ELEMENT</u>	Message - LPD ELEMENT Displays element as centerline
2. 3.	<u>LPD PHANTOM</u> <u>LPD ELEMENT</u>	Message - LPD ELEMENT Displays element as a phantom line

Table 1-5

CONSOLE PROCEDURES FOR FILLET

Step	User Action	Machine Reaction
1.	FK <u>FILLET</u>	Message - KEY RADIUS
2.	Key or pick <u>RADIUS</u>	Message - LPD ELEMENT 1
3.	LPD <u>1ST ELEMENT</u>	Message - LPD ELEMENT 2
4.	LPD <u>2ND ELEMENT</u>	Message - LPD LOCATION OF FILLET CENTER
5.	LPD <u>SIDE OF FILLET CENTER</u>	Displays fillet tangent to elements selected

Table 1-6

CONSOLE PROCEDURES FOR RETOUCH

Step	User Action	Machine Reaction
1.	FK <u>RETOUCH</u>	<u>MENU DISPLAYED</u>  EXPAND CONTRACT BREAK
2.	LPD <u>EXPAND</u>	Message - LPD ELEMENT
3.	LPD <u>ELEMENT</u>	Message - LPD EXPANDED DISTANCE LPD ELEMENT TO EXPAND TO
4.	LPD <u>ELEMENT OR DISTANCE</u>	Expands element
2.	LPD <u>CONTRACT</u>	Message - LPD ELEMENT
3.	LPD <u>ELEMENT</u>	Message - LPD END TO BE CONTRACTED
4.	LPD <u>END</u>	Message - LPD LOCATION TO CONTRACT TO LPD ELEMENT TO CONTRACT TO
5.	LPD <u>LOCATION OR ELEMENT</u>	Contracts element

Table 1-6 (Cont.)

Step	User Action	Machine Reaction
2.	LPD <u>BREAK</u>	Message - LPD ELEMENT
3.	LPD <u>ELEMENT</u>	Message - LPD LOCATION OF BREAK
4.	LPD <u>LOCATION</u>	Breaks element at point normal to LPD location. Element is now stored as two separate entities in the model.

Table 1-7

CONSOLE PROCEDURES FOR PARALLEL  
(STORES NEW ELEMENT PARALLEL TO OLD ELEMENT)

Step	User Action	Machine Reaction
1.	FK <u>PARALLEL</u>	Message - LPD ELEMENT
2.	LPD <u>ELEMENT</u>	Message - LPD SIDE NEW ELEMENT TO BE
3.	LPD <u>SIDE</u>	Message - KEY NORMAL DISTANCE
4.	Key <u>DISTANCE</u>	Displays parallel element

Table 1-8

## CONSOLE PROCEDURES FOR CIRCLE

Step	User Action	Machine Reaction
1.	FK <u>CIRCLE</u>	Message - KEY X DIMENSION Message - LPD POINT
2.	Key or pick <u>X DIMENSION</u>	Message - KEY Y DIMENSION
3.	Key or pick <u>Y DIMENSION</u>	Message - KEY RADIUS
4.	Key or pick <u>RADIUS</u>	Displays circle
2.	LPD <u>POINT</u>	Message - KEY RADIUS LPD POINT 2
3.	Key or pick <u>RADIUS</u>	Displays circle
2.	LPD <u>POINT</u>	Message - KEY RADIUS LPD POINT 2
3.	LPD <u>POINT 2</u>	Message - LPD CENTER DIRECTION LPD POINT 3
4.	LPD <u>CENTER DIRECTION</u>	Message - KEY RADIUS
5.	Key or pick <u>RADIUS</u>	Displays circle
2.	LPD <u>POINT</u>	Message - KEY RADIUS LPD POINT 2
3.	LPD <u>POINT 2</u>	Message - LPD CENTER DIRECTION LPD POINT 3
4.	LPD <u>POINT 3</u>	Displays circle

Table 1-9

CONSOLE PROCEDURES FOR POINT

Step	User Action	Machine Reaction
1.	FK <u>POINT</u>	Message - KEY X DIMENSION Message - LPD PHYSICAL LOCATION Message - LPD 1ST ELEMENT OF INTERSECTION
2.	Key or pick <u>X DIMENSION</u>	Message - KEY Y DIMENSION
3.	Key or pick <u>Y DIMENSION</u>	Displays point
2.	<u>LPD LOCATION OF POINT</u>	Displays point
2.	<u>LPD 1ST ELEMENT</u>	Message - LPD 2ND ELEMENT OF INTERSECTION
3.	<u>LPD 2ND ELEMENT</u>	*Displays point

\*If one or both elements are circles or curves an additional LPD is required to indicate which of multiple intersections the point is to be stored.

Table 1-10

CONSOLE PROCEDURES FOR LINE THROUGH POINT AT DIRECTION

Step	User Action	Machine Reaction
1.	FK <u>LINE THRU POINT AT DIRECTION</u>	<u>MENU DISPLAYED</u> HORIZONTAL VERTICAL PARALLEL PERPENDICULAR AT ANGLE

Table 1-10 (Cont.)

Step	User Action	Machine Reaction
2.	LPD <u>HORIZONTAL</u>	Message - KEY X DIMENSION Message - LPD POINT Message - LPD TANGENT CIRCLE
3. 4.	Key or pick <u>X DIMENSION</u> Key or pick <u>Y DIMENSION</u>	Message - KEY Y DIMENSION *Displays Line
3.	LPD <u>POINT</u>	Displays Line
3. 4.	LPD <u>TANGENT CIRCLE</u> LPD <u>TANGENT SIDE OF CIRCLE</u>	Message - LPD SIDE OF CIRCLE THAT LINE IS TANGENT *Displays line
2.	LPD <u>VERTICAL</u> Procedure same as HORIZONTAL	
2.	LPD <u>PARALLEL</u>	Message - KEY X DIMENSION Message - LPD POINT Message - LPD TANGENT CIRCLE
3. 4. 5.	Key or pick <u>X DIMENSION</u> Key or pick <u>Y DIMENSION</u> LPD <u>PARALLEL LINE</u>	Message - KEY Y DIMENSION Message - LPD PARALLEL LINE *Displays line
3. 4.	LPD <u>POINT</u> LPD <u>PARALLEL LINE</u>	Message - LPD PARALLEL LINE *Displays line
3. 4. 5.	LPD <u>TANGENT CIRCLE</u> LPD TANGENT SIDE OF CIRCLE LPD PARALLEL LINE	Message - LPD SIDE OF CIRCLE THAT LINE IS TANGENT Message - LPD PARALLEL LINE *Displays line

\*Program will set value of line length to some constant. Retouch function will be used to trim line to desired configuration.

Table 1-11

CONSOLE PROCEDURES FOR LINE POINT TO POINT

Step	User Action	Machine Reaction
1.	FK <u>LINE POINT TO POINT</u>	Message - KEY X DIMENSION Message - LPD POINT Message - LPD TANGENT CIRCLE
2.	Key using typewriter or pick with light pen <u>X DIMENSION</u>	Message - Key Y DIMENSION
3.	Key or pick <u>Y DIMENSION</u>	Message - Key X DIMENSION Message - LPD POINT Message - LPD TANGENT Message - CIRCLE
4.	Repeat steps 2 and 3	Displays line
4.	LPD <u>PREVIOUSLY STORED POINT</u>	Displays line
4.	LPD <u>CIRCLE</u>	Message - LPD SIDE OF CIRCLE THAT LINE IS TANGENT
5.	LPD <u>SIDE OF CIRCLE LINE IS TANGENT TO</u>	Displays line
2.	LPD <u>POINT</u>	Message - KEY X DIMENSION Message - LPD POINT Message - LPD TANGENT CIRCLE
3.	LPD <u>2ND POINT</u>	Displays line
3.	Key or pick <u>X DIMENSION</u>	Message - KEY Y DIMENSION
4.	Key or pick <u>Y DIMENSION</u>	Displays line
3.	LPD <u>TANGENT CIRCLE</u>	Message - LPD SIDE OF CIRCLE THAT LINE IS TANGENT
4.	LPD <u>TANGENT SIDE OF CIRCLE</u>	Displays line



Table 1-11 (Cont.)

Step	User Action	Machine Reaction
2.	LPD <u>TANGENT CIRCLE</u>	Message - LPD SIDE OF CIRCLE THAT LINE IS TANGENT
3.	LPD <u>TANGENT SIDE OF CIRCLE</u>	Message - KEY X DIMENSION Message - LPD POINT Message - LPD TANGENT CIRCLE
4.	LPD <u>TANGENT CIRCLE</u>	Message - LPD SIDE OF CIRCLE THAT LINE IS TANGENT
5.	LPD <u>TANGENT SIDE OF CIRCLE</u>	Displays line
4.	LPD <u>POINT</u>	Displays line
4.	Key or pick <u>X DIMENSION</u>	Message - KEY Y DIMENSION
5.	Key or pick <u>Y DIMENSION</u>	Displays line

Table 1-12

CONSOLE PROCEDURES FOR DIMENSIONING INFORMATION

Step	User Action	Machine Reaction
1.	FK <u>DIMENSION</u>	<u>MENU DISPLAYED</u>  HORIZONTAL VERTICAL PARALLEL OFFSET ANGLE RADIUS DIAMETER CURVE

Table 1-12 (Cont.)

Step	User Action	Machine Reaction
2.  3. 4. 5.	LPD <u>HORIZONTAL</u>  LPD <u>1ST POINT</u> LPD <u>2ND POINT</u> LPD <u>LOCATION OF DIMENSION</u>	HORIZONTAL underlined Message - SELECT POINT 1 Message - SELECT POINT 2 Message - LPD LOCATION OF DIMENSION Displays dimensions, dimension lines and arrowheads (they are stored in the model to be used for hard copy if required)
2.	LPD <u>VERTICAL</u>  Same as HORIZONTAL except dimension is vertical between points selected	
2.  3. 4.	LPD <u>PARALLEL</u>  LPD <u>1ST POINT</u> LPD <u>2ND POINT</u>	PARALLEL underlined Message - SELECT POINT 1 Message - SELECT POINT 2 Message - SELECT ELEMENT THAT DIMENSION IS PARALLEL TO
5.  6.	LPD <u>ELEMENT THAT DIM IS PARALLEL TO</u> LPD <u>LOCATION OF DIMENSION</u>	Message - LPD LOCATION OF DIMENSION  Displays dimension, dimension lines and arrowheads
2.  3. 4. 5.	LPD <u>OFFSET</u>  LPD <u>1ST ELEMENT</u> LPD <u>2ND ELEMENT</u> LPD <u>LOCATION OF DIMENSION</u>	OFFSET underlined Message - SELECT 1ST ELEMENT Message - SELECT 2ND ELEMENT Message - LPD LOCATION OF DIMENSION Displays dimension, dimension lines and arrowheads

Table 1-12 (Cont.)

Step	User Action	Machine Reaction
2. 3. 4. 5.	<p>LPD <u>ANGLE</u></p> <p>LPD <u>1ST ELEMENT</u></p> <p>LPD <u>2ND ELEMENT</u></p> <p>LPD <u>LOCATION OF DIMENSION</u></p>	<p><u>ANGLE</u> underlined</p> <p>Message - SELECT 1ST ELEMENT</p> <p>Message - SELECT 2ND ELEMENT</p> <p>Message - LPD LOCATION OF DIMENSION</p> <p>Displays dimension in degrees, minutes, and seconds, dimension lines and arrowheads</p>
2. 3. 4.	<p>LPD <u>RADIUS</u></p> <p>LPD <u>CIRCLE OR ARC</u></p> <p>LPD <u>LOCATION OF DIMENSION</u></p>	<p><u>RADIUS</u> underlined</p> <p>Message - SELECT CIRCLE OR ARC</p> <p>Message - LPD LOCATION OF DIMENSION</p> <p>Displays dimension with R, RAD, or RADIUS following, dimension line and arrowhead</p>
2. 3. 4.	<p>LPD <u>DIAMETER</u></p> <p>LPD <u>CIRCLE</u></p> <p>LPD <u>LOCATION OF DIMENSION</u></p>	<p><u>DIAMETER</u> underlined</p> <p>Message - SELECT CIRCLE</p> <p>Message - LPD LOCATION OF DIMENSION</p> <p>Displays dimension with D or DIA following, dimension line, and arrowheads</p>
2. 3. 4.	<p>LPD <u>CURVE</u></p> <p>LPD <u>CURVE</u></p> <p>LPD <u>LOCATION OF DIMENSION</u></p>	<p><u>CURVE</u> underlined</p> <p>Message - SELECT CURVE</p> <p>Message - LPD LOCATION OF DIMENSION</p> <p>Displays dimension, dimension lines and arrowheads, parallel to curve</p>

Table 1-13

CONSOLE PROCEDURES FOR GEOMETRY MACRO

Step	User Action	Machine Reaction
1.	<u>FK MACRO</u>	<u>MENU DISPLAYED</u> START ABORT RETURN TO DRAWING DISPLAY MACRO CALL STORE
2.	LPD <u>START</u>	START underlined Message - SELECT ELEMENT Message - SELECT FK (This instruction is to be followed only if the console operator is constructing the macro from previously stored geometry)
3.	LPD <u>ALL GEOMETRIC ELEMENTS TO BE USED IN MACRO</u>	Message - SELECT ELEMENT
4.	LPD <u>DISPLAY MACRO</u>	DISPLAY MACRO underlined; computer displays only those elements selected. Operator can then see if he has completed the macro Message - SELECT PIVOT POINT
5.	LPD <u>RETURN TO DRAWING</u>	RETURN TO DRAWING underlined Message - SELECT ELEMENT
6.	LPD additional elements & repeat step 4 until macro is complete	Message - SELECT ELEMENT

Table 1-13 (Cont.)

Step	User Action	Machine Reaction
7	LPD <u>DISPLAY MACRO</u>	DISPLAY MACRO underlined (same as step 4)
8	LPD a point (becomes pivot point)	Message - SELECT PIVOT LINE (pivot point and pivot line will be used to locate geometry on face of drawing-location and rotation)
9	LPD a line (becomes pivot line)	Message - SELECT STORE OR ABORT
10	LPD <u>STORE</u>	STORE underlined; stores macro
11	LPD <u>RETURN TO DRAWING</u>	Returns original drawing to display
3	Select any function key	Display goes blank
4	Construct geometry	Geometry stored in model as macro
5	Repeat step 1	
6	LPD <u>DISPLAY MACRO</u>	DISPLAY MACRO underlined Message - SELECT PIVOT POINT - RETURN TO DRAWING
7	LPD <u>a point</u>	Message - SELECT PIVOT LINE
8	LPD <u>a line</u>	Message - SELECT STORE OR ABORT
9	LPD <u>STORE</u>	STORE underlined; stores macro
2	LPD <u>DISPLAY MACRO</u>	Displays 1st macro stored
3	LPD <u>DISPLAY MACRO</u>	Displays 2nd macro
4	LPD <u>DISPLAY MACRO</u>	Displays Nth macro stored
5	LPD <u>CALL</u>	CALL underlined (th Nth macro which was displayed when call was selected is the macro being called). Original drawing is displayed Message - SELECT PIVOT

Table 1-13 (Cont.)

Step	User Action	Machine Reaction
6. 7.	LPD <u>a point</u> LPD <u>a line</u>	Message - SELECT PIVOT LINE Macro geometry is displayed in conjunction with original drawing (macro geometry is stored in model as part of original geometry)
Any LPD ABORT Any + 1 Select any function key to get out of macro subrouting		ABORT underlined

Table 1-14

CONSOLE PROCEDURES FOR TRANSFORMATION

Step	User Action	Machine Reaction
1.	FK <u>TRANSFORMATION</u>	<u>MENU DISPLAYED</u> MOVE ROTATE SCALE
2. 3. 4.	LPD <u>MOVE</u> Key or pick <u>X TRANSLATION</u> Key or pick <u>Y TRANSLATION</u>	MOVE underlined Message - KEY X Message - KEY Y Display is moved X&Y dimensions keyed will be new position of the window center
2. 3.	LPD <u>ROTATE</u> Key or pick <u>ANGLE</u>	ROTATE underlined Message - KEY ANGLE Display rotates to angle keyed in
2. 3.	LPD <u>SCALE</u> Key <u>SCALE</u>	SCALE underlined Message - KEY SCALE Display is scaled by amount keyed in

## GLOSSARY OF TERMS

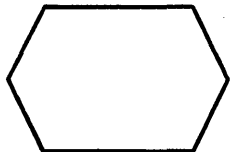
FK	Function key
LPD	Light pen detect
Pick	Select item from display with light pen
Window	CRT is considered a window through which we view the drawing
Key	Key in data with typewriter keyboard
Drive surface	Surface the cutter is to drive along
Check surface	Surface cutter stops against
Perp	Perpendicular



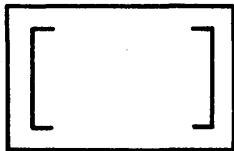
= USER ACTION



= MACHINE DISPLAY = MESSAGE OR MENU



= MACHINE ACTION



= FUNCTION - FUNCTION SHOWN IN  
BOX REMAINS IN THIS UPPER L.H.  
CORNER OF DISPLAY UNTIL ANOTHER  
FUNCTION KEY IS DEPRESSED

Fig. 1-1 Symbols Defined



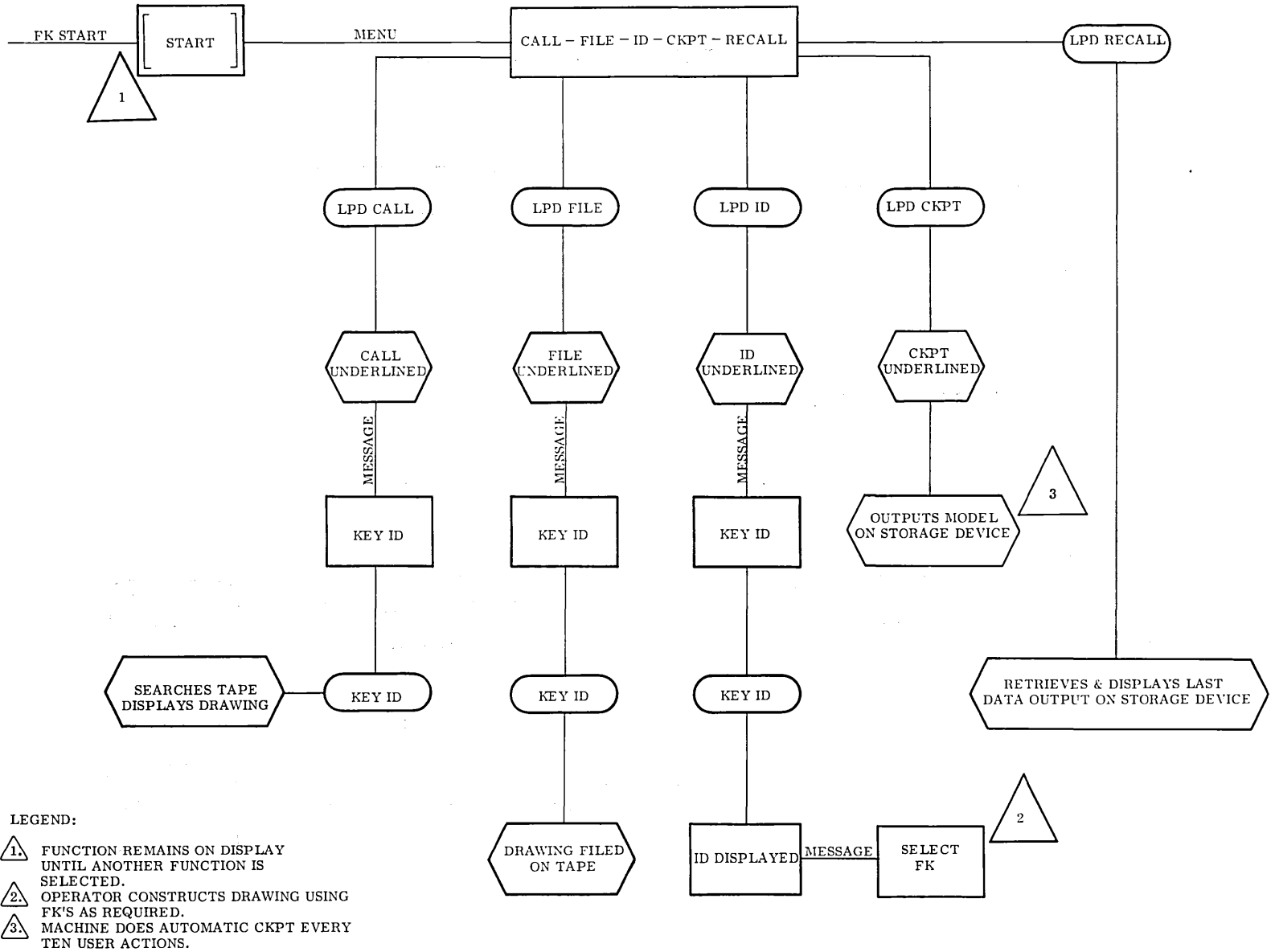


Fig. 1-2 Subroutine START

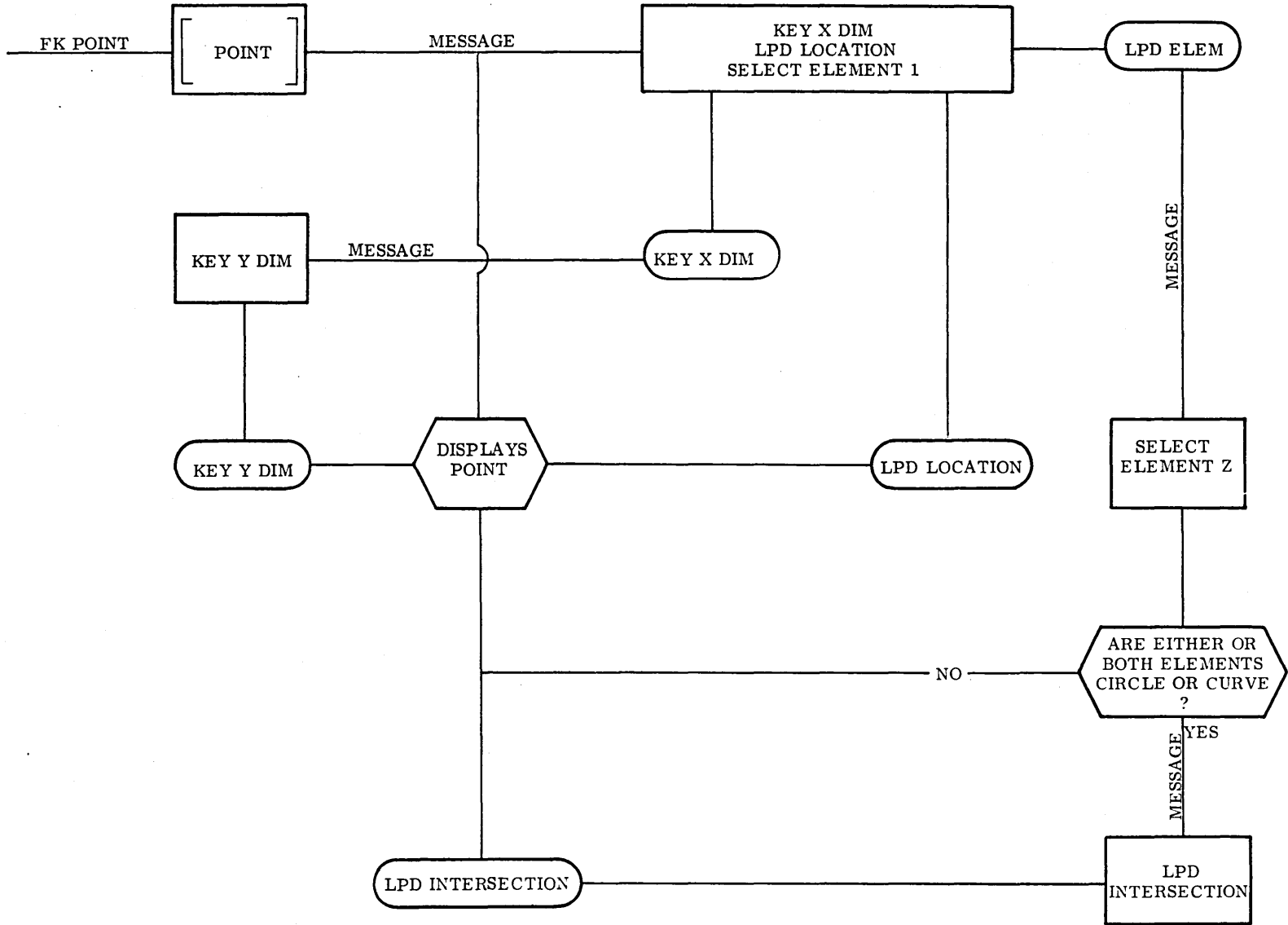


Fig. 1-3 Subroutine POINT

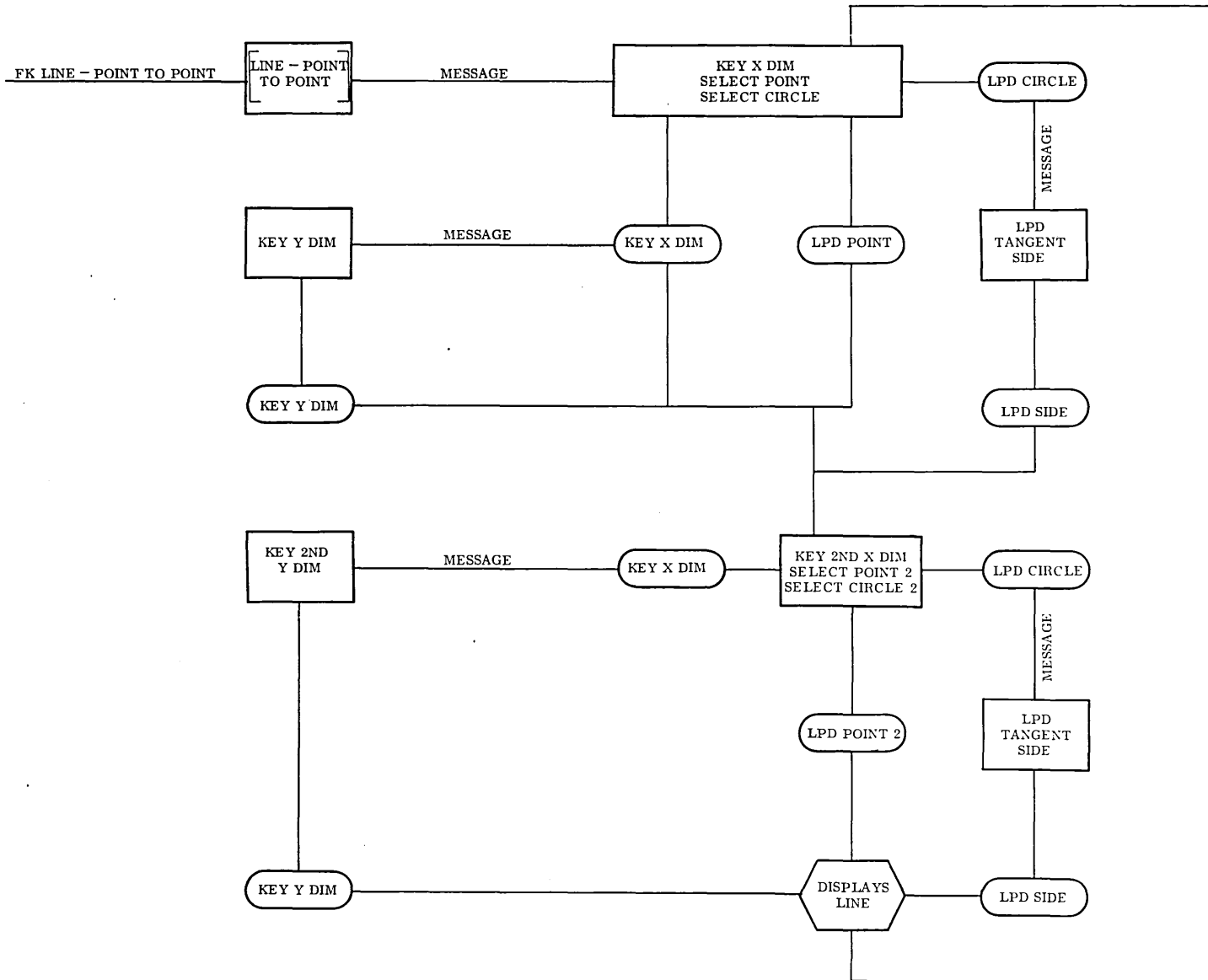


Fig. 1-4 Subroutine LINE-POINT-TO-POINT

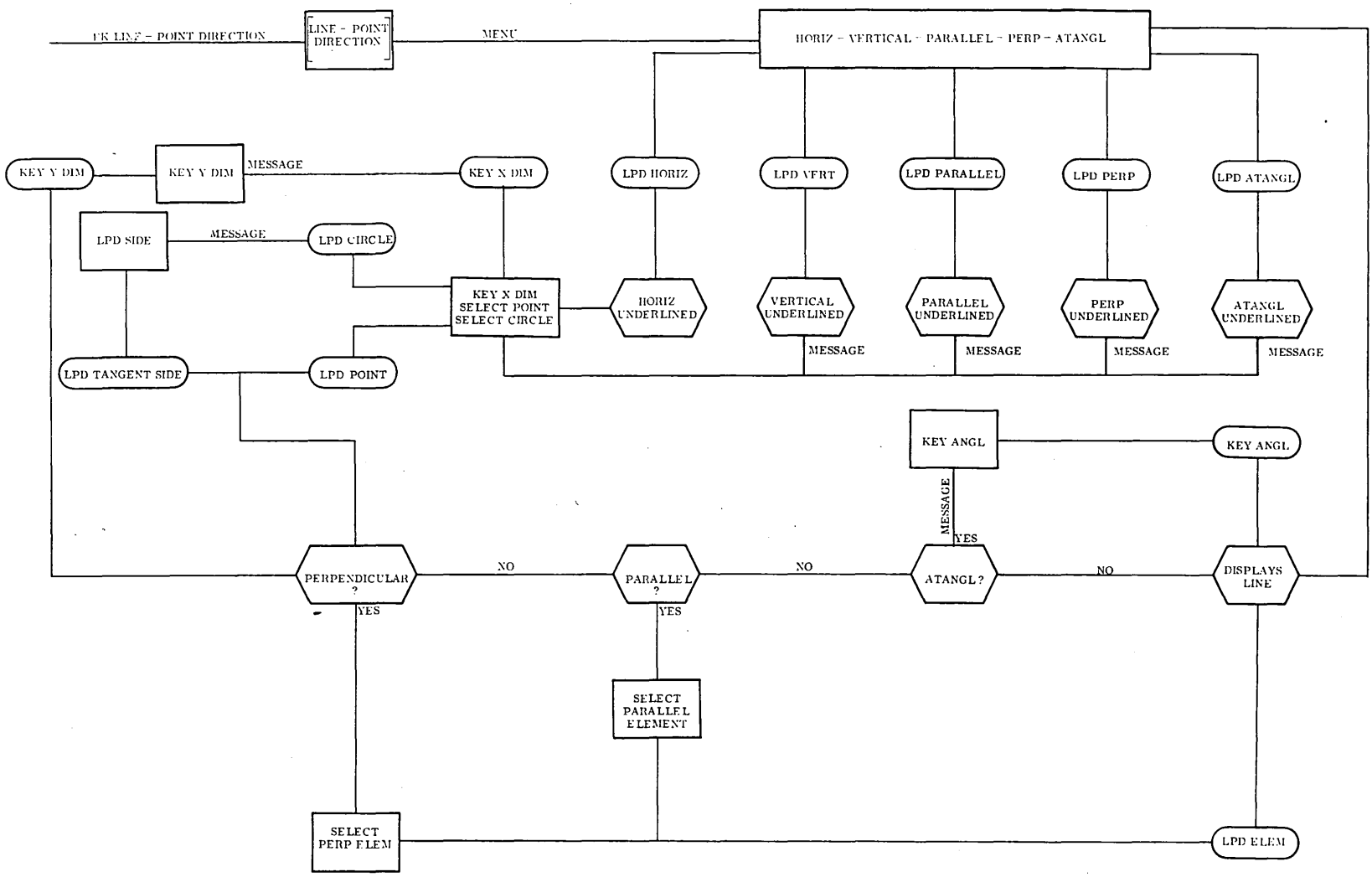
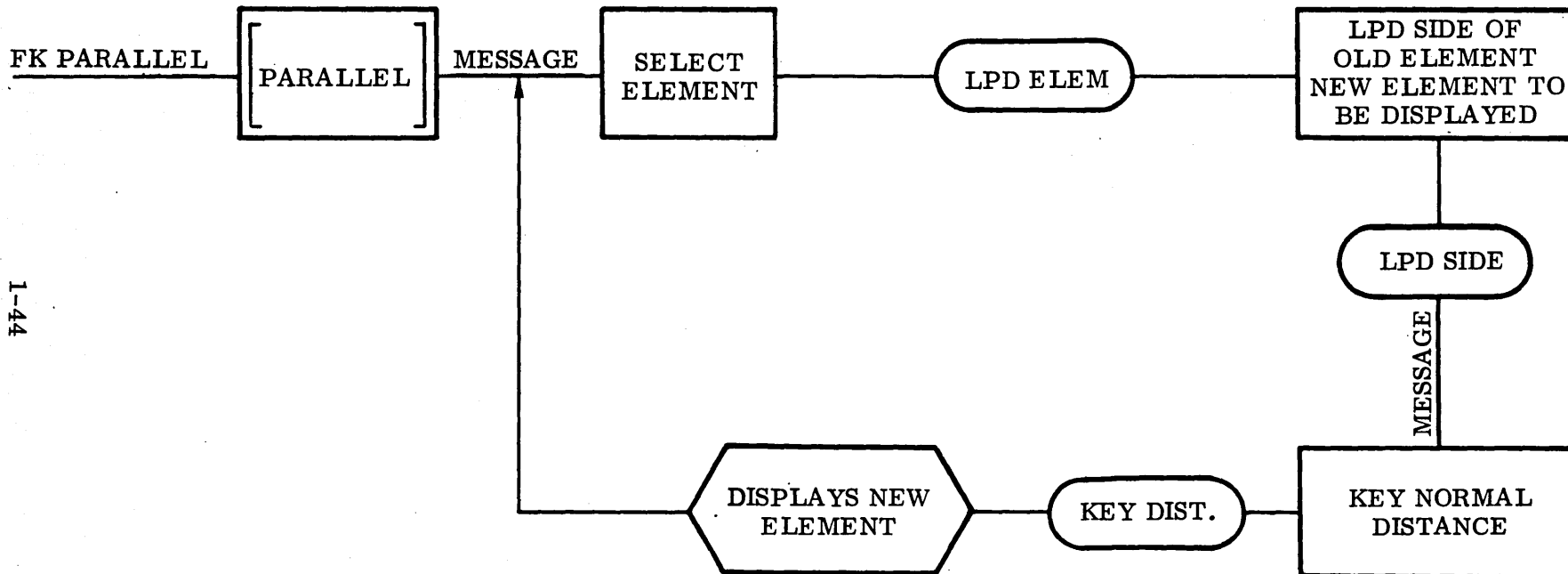


Fig. 1-5 Subroutine LINE-POINT DIRECTION





1-44

Fig. 1-7 Subroutine PARALLEL

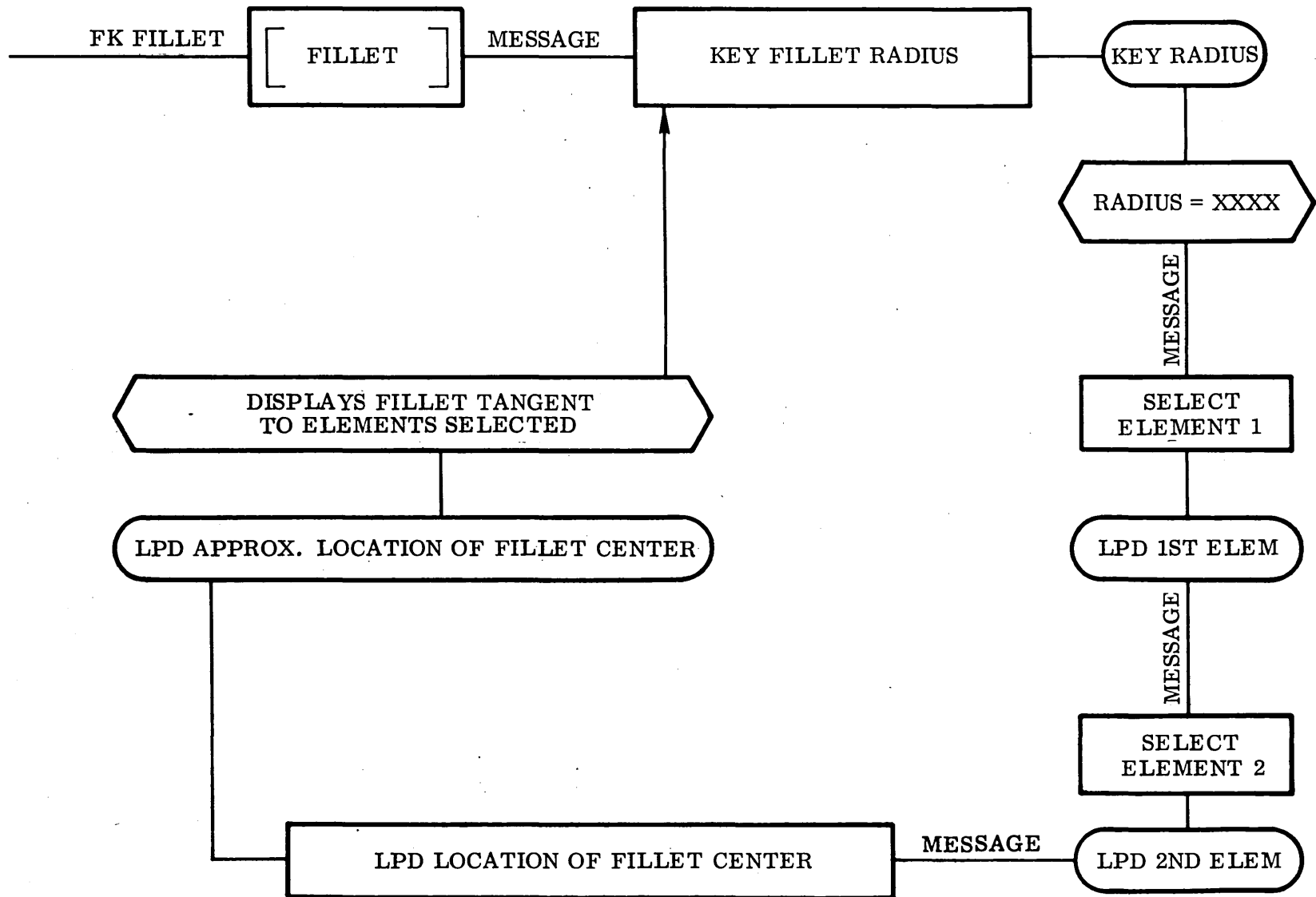


Fig. 1-8 Subroutine FILLET

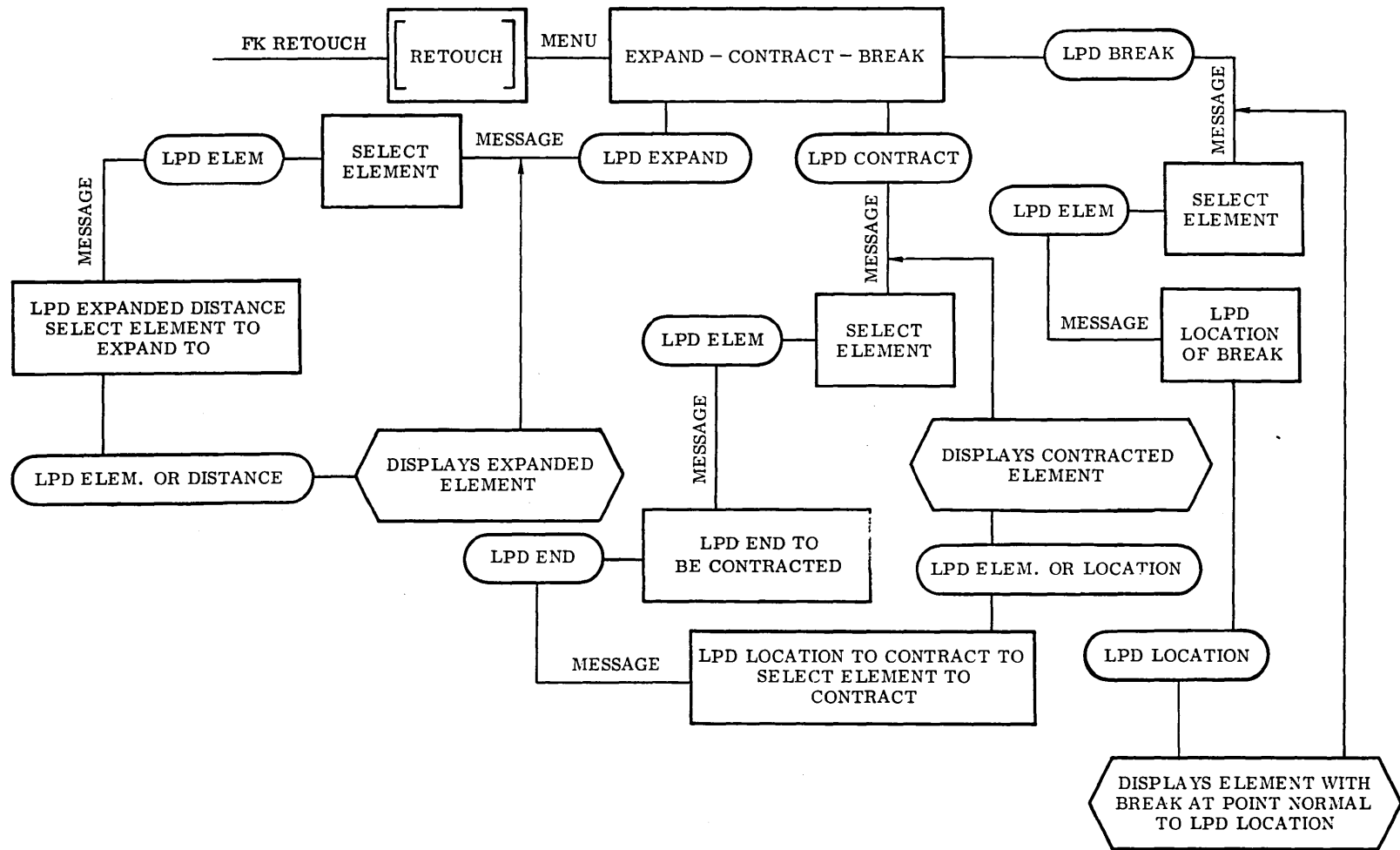
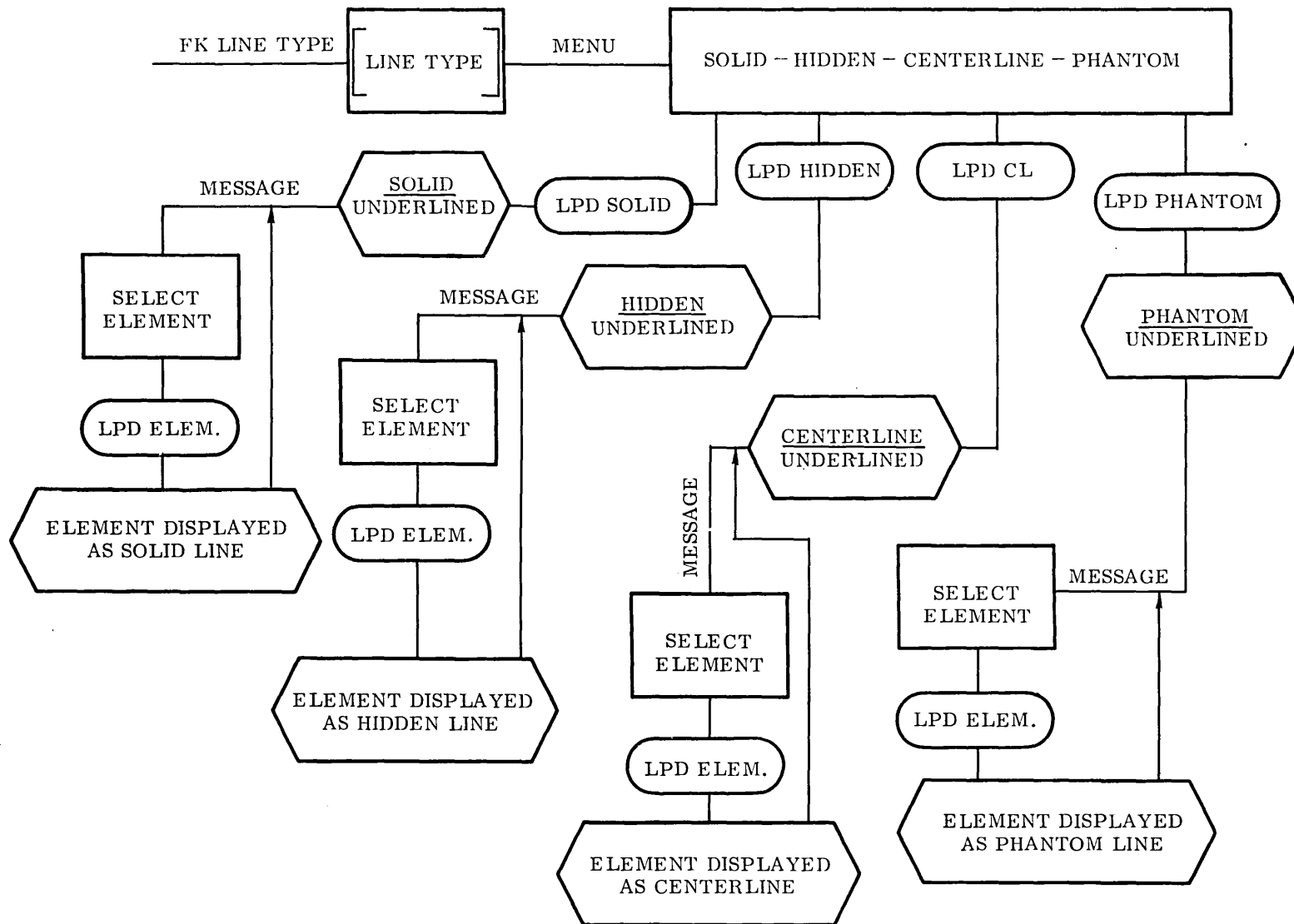


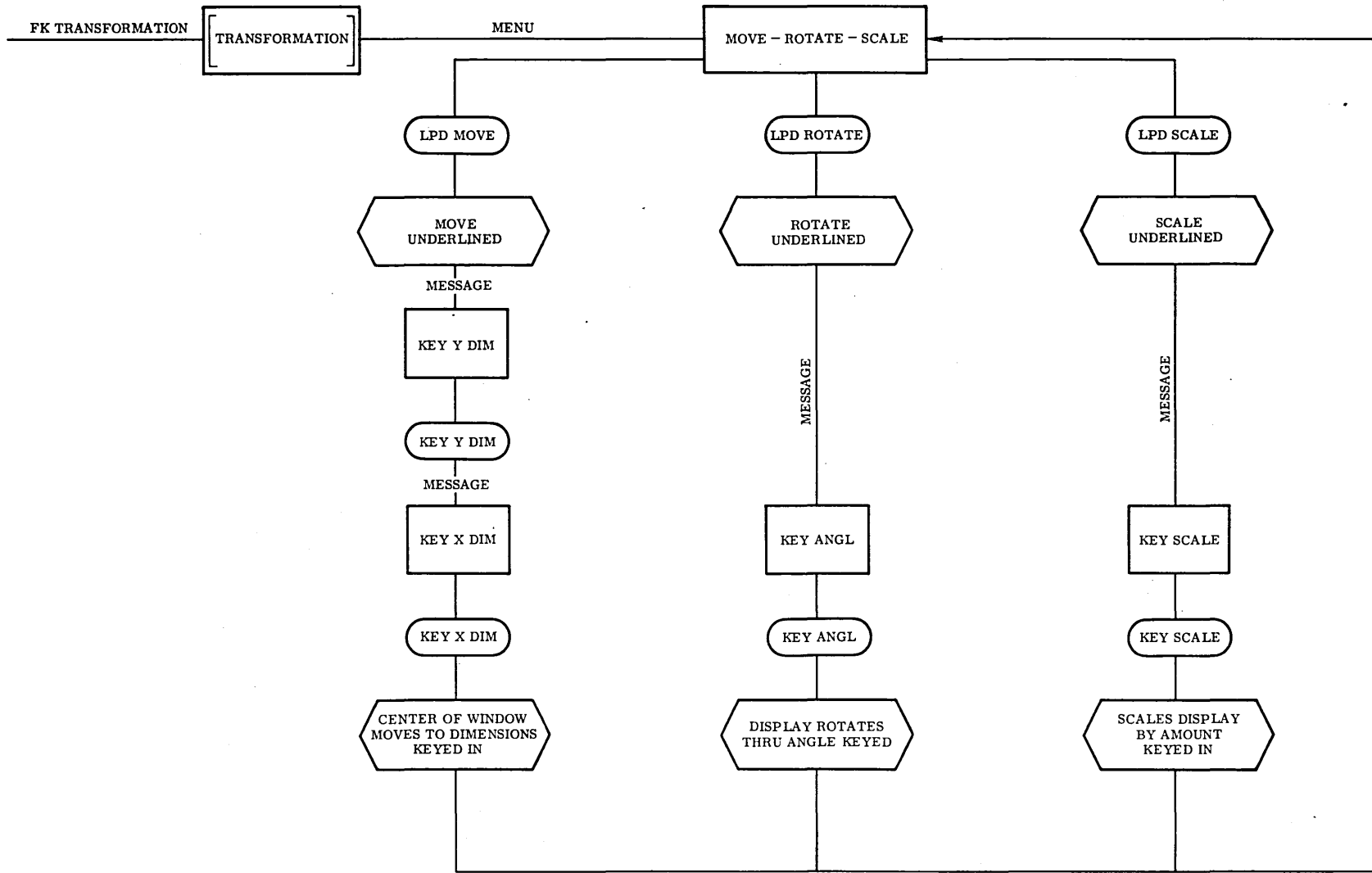
Fig. 1-9 Subroutine RETOUCH





1-47

Fig. 1-10 Subroutine LINE TYPE



1-48

Fig. 1-11 Subroutine TRANSFORMATION

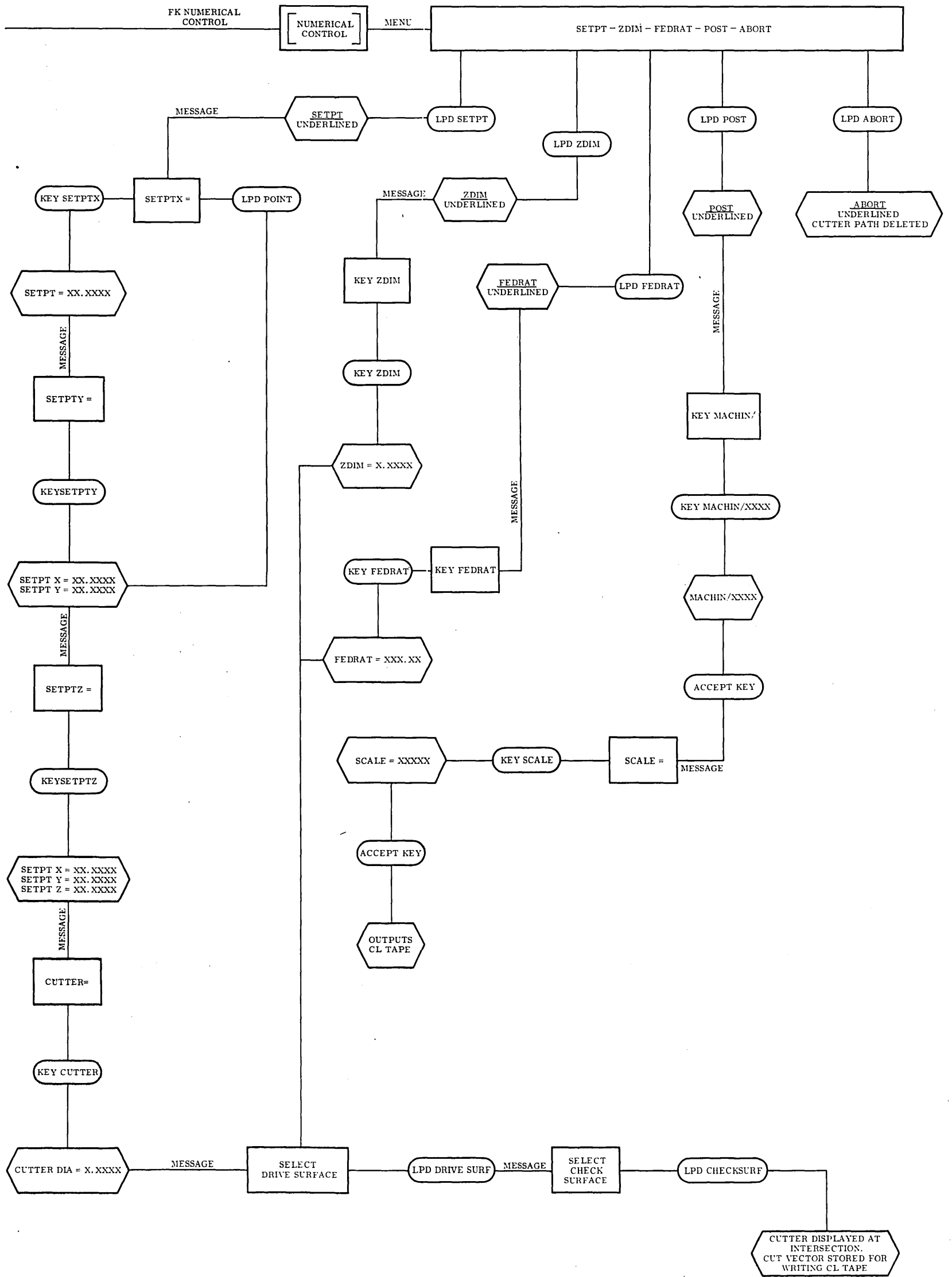


Fig. 1-12 Subroutine NUMERICAL CONTROL

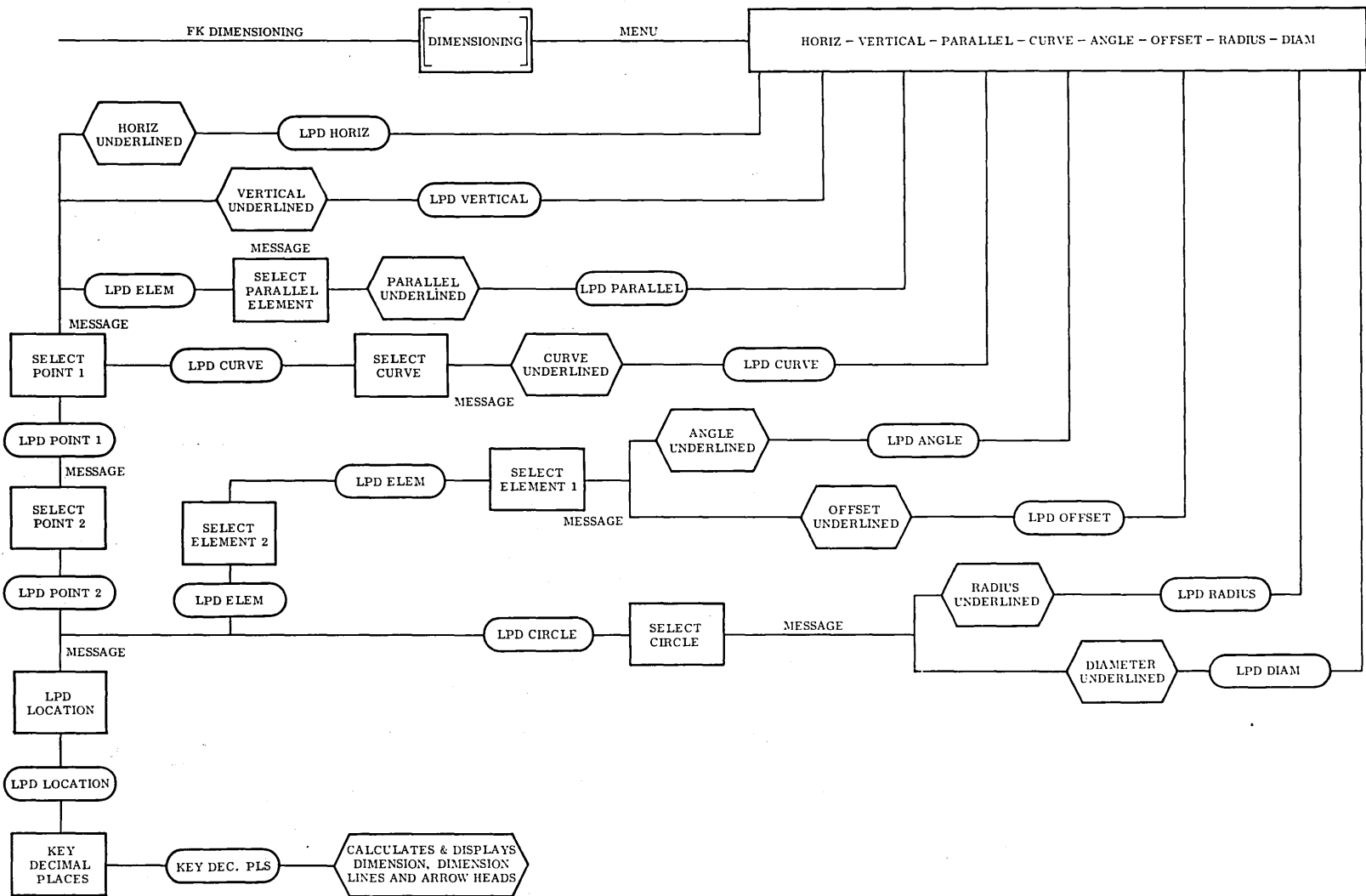


Fig. 1-13 Subroutine DIMENSIONING

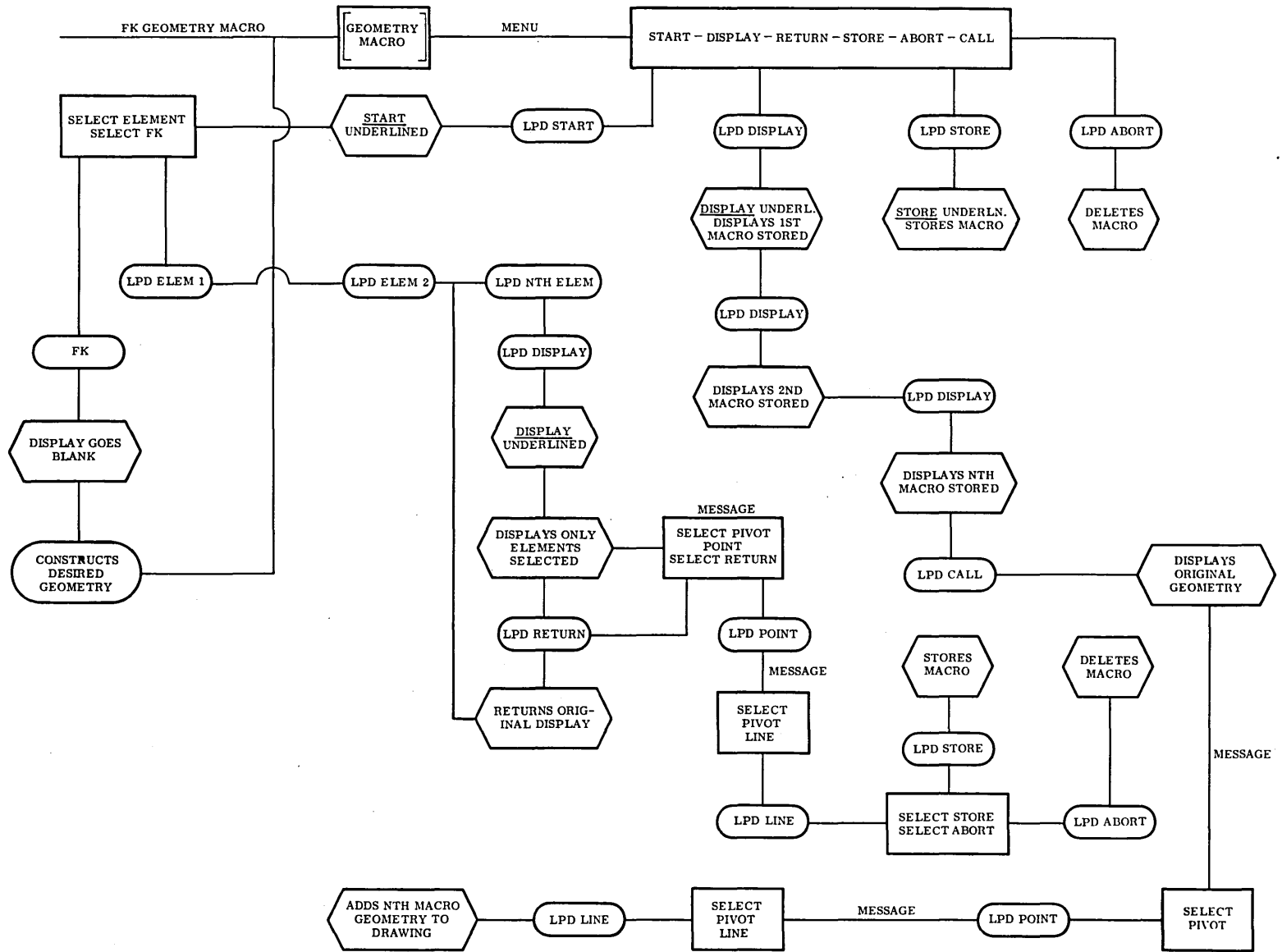


Fig. 1-14 Subroutine MACRO

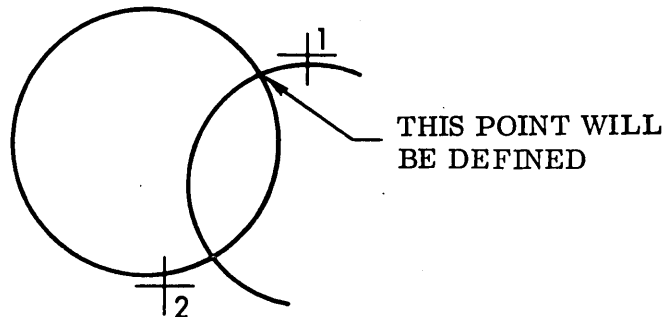
## 1.3.2 Console Procedure for Display-Oriented Systems

### 1.3.2.1 Points. Methods of definition of points are as follows:

1. Using the Computer Console Typewriter:
  - a. Push the "typewriter" button.
  - b. Type P √ X √ Y (now space several times).
  - c. Push FINISH on the computer console.
2. Using the Alphanumeric Pick Table:
  - a. Push the PICK button.
  - b. Pick P √ X √ Y
  - c. Push the NO MORE button.
3. Positioning the Tracking Cross:
  - a. Push the POINT button  
or  
pick the POINT function
  - b. Position the tracking cross where the point is desired.
  - c. Push the ACCEPT TC button  
or  
pick the ON function.
4. Intersections of Surfaces:
  - a. Pick the INTERSECT function.
  - b. Position the tracking cross on the first surface.
  - c. Push the THIS SURFACE button  
or  
Pick the TO or PAST function.
  - d. Position the tracking cross on the second surface.
  - e. Push the THIS SURFACE button  
or  
Pick the TO or PAST function.

**NOTE:**

Where multiple intersections exist, the point nearest the first tracking cross location will be defined.



**1.3.2.2 Lines.** Methods of definition of lines are as follows:

1. Using the Computer Console Typewriter:
  - a. Push the TYPEWRITER button.
  - b. Type HL √ K<sub>1</sub> (now several spaces)  
or  
VL √ K<sub>2</sub> (now several spaces)
  - c. Push FINISH on the computer console.

**NOTE:**

This format defines horizontal or vertical lines extending to the boundaries of the display area.

2. Using the Computer Console Typewriter:
  - a. Push the TYPEWRITER button.
  - b. Type - L √ X<sub>1</sub> √ Y<sub>1</sub> √ X<sub>2</sub> √ Y<sub>2</sub> (now several spaces)
  - c. Push FINISH on the computer console.
3. Using the Alphanumeric Pick Table:
  - a. Push the PICK button.
  - b. Pick HL √ K<sub>1</sub>  
or  
VL √ K<sub>2</sub>

- c. Push the NOMORE button.
4. Using the Alphanumeric Pick Table:
    - a. Push the PICK button.
    - b. Pick  $L \vee X_1 \vee Y_1 \vee X_2 \vee Y_2$
    - c. Push the NOMORE button.
  5. Two Tracking Cross Locations:
    - a. Push the LINE button.  
or  
Pick the LINE function.
    - b. Position the tracking cross.
    - c. Push the ACCEPT TC button.  
or  
Pick the ON function.
    - d. Reposition the tracking cross.
    - e. Push the ACCEPT TC button.  
or  
Pick the ON function.
  6. Through a Point and Tangent to an Arc:
    - a. Push the LINE button.  
or  
Pick the LINE function.
    - b. Position the tracking cross for the point condition.
    - c. Pick the THRU function.
    - d. Reposition the tracking cross near the desired tangency point.
    - e. Pick the TANGENT function.
  7. Tangent to Two Arcs:
    - a. Push the LINE button.  
or  
Pick the LINE function.
    - b. Position the tracking cross on the arc near the desired point of tangency.
    - c. Pick the TANGENT function.
    - d. Position the tracking cross on the second arc near the desired point of tangency.
    - e. Pick the TANGENT function.



8. Through a Point at Some Angle to a Line:
  - a. Push the LINE button.  
or  
Pick the LINE function.
  - b. Position the tracking cross for the point condition.
  - c. Pick the THRU function.
  - d. Position the tracking cross on the line.
  - e. Pick the AT ANGLE function.
  - f. Pick the angle value in decimal degrees.
  - g. Push the NOMORE button.

NOTE:

If the point falls on the line a degenerate line display will be given. The display is from the point to the line.

9. Parallel to a Line by a Given Normal Distance:
  - a. Pick the LINE function.
  - b. Position the tracking cross on one side of a line.
  - c. Pick the PARALLEL function.
  - d. Pick the normal distance.
  - e. Push the NOMORE button.

NOTE:

Connecting the end points of the two parallel lines will form a parallelogram. The parallel lines are the same length.

1.3.2.3 Arcs. Methods of definition of arcs are as follows:

1. Using the Computer Console Typewriter:
  - a. Push the TYPEWRITER button.
  - b. Type A / X<sub>c</sub> / Y<sub>c</sub> / X / Y (several spaces)  
or  
A  $\underbrace{X_c / Y_c}_{\text{center point}} / \underbrace{X_b / Y_b}_{\text{beginning point}} / \underbrace{X_e / Y_e}_{\text{end point}}$  (several spaces)

- c. Push FINISH on the computer console.

NOTE:

The arc will be generated clockwise from beginning point to end point.

2. Using the Alphanumeric Pick Table:

- a. Push the PICK button.
- b. Pick  $A \sqrt{X_c} \sqrt{Y_c} \sqrt{r}$   
or  
 $A \sqrt{X_c} \sqrt{Y_c} \sqrt{X_b} \sqrt{Y_b} \sqrt{X_e} \sqrt{Y_e}$
- c. Push the NOMORE button.

3. By Two Tracking Cross Locations:

- a. Push the ARC button  
or  
Pick the ARC function.
- b. Position the tracking cross.
- c. Push the ACCEPT TC button  
or  
Pick the ON function.
- d. Reposition the tracking cross.
- e. Push the ACCEPT TC button.  
or  
Pick the ON function.

4. By a Center Point and a Radius:

- a. Push the ARC button  
or  
Pick the ARC function.
- b. Position the tracking cross to establish the center point.
- c. Push the ACCEPT TC button  
or  
Pick the ON function.
- d. Push the PICK button.
- e. Pick  $R \sqrt{r}$ .
- f. Push the NOMORE button.

5. By a Center Point and a Line to Which it is Tangent:
  - a. Push the ARC button  
or  
Pick the ARC function.
  - b. Position the tracking cross to establish the center point.
  - c. Push the ACCEPT TC button.  
or  
Pick the ON function.
  - d. Position the tracking cross on the line.
  - e. Pick the TANGENT function.
  
6. Passing Tangent to Two Lines With a Given Radius:
  - a. Push the ARC button  
or  
Pick the ARC function.
  - b. Position the tracking cross on the first line.
  - c. Pick the TANGENT function.
  - d. Position the tracking cross on the second line.
  - e. Pick the TANGENT function.
  - f. Pick the RADIUS value.
  - g. Push the NOMORE button.

NOTES:

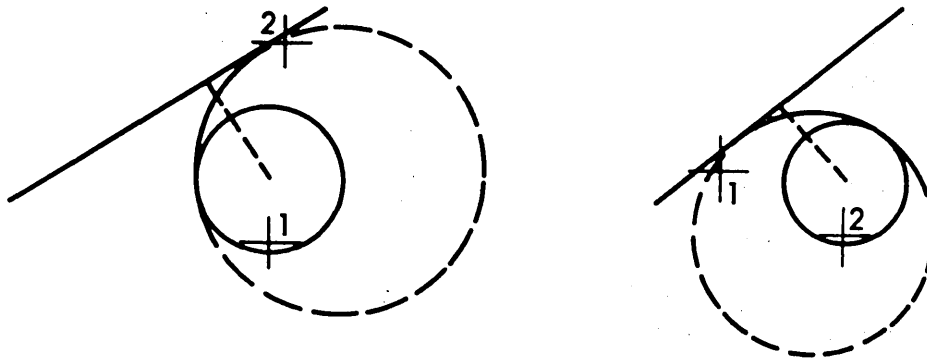
1. The lines need not visually intersect.
  2. It is necessary to position the tracking crosses on the appropriate sides of the lines.
  3. The arc will be generated in a clockwise direction from the first tangent line to the second.
- 
7. Passing Tangent to a Line and an Arc With a Given Radius:
    - a. Push the ARC button  
or  
Pick the ARC function.
    - b. Position the tracking cross on the line or arc.
    - c. Pick the TANGENT function.
    - d. Position the tracking cross on the remaining surface.
    - e. Pick the TANGENT function.
    - f. Pick the RADIUS value.

g. Push the NOMORE button.

NOTES:

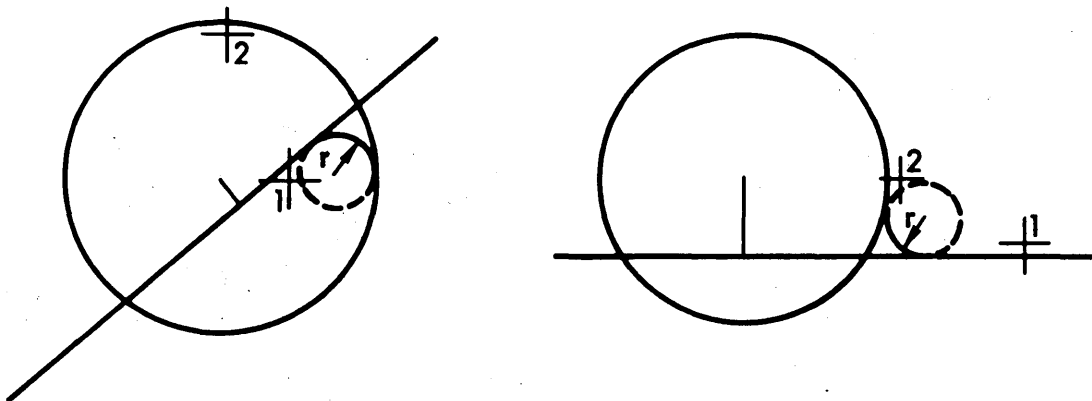
The arc will be generated clockwise from the first condition to the second.

Case I: Where the line does not intersect the arc, positioning the tracking cross inside the arc will indicate one of the possibilities illustrated below.



Positioning the tracking cross on the line determines which of these two possibilities is selected.

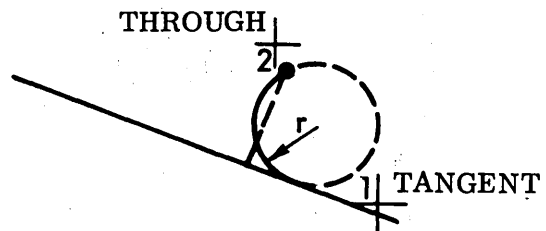
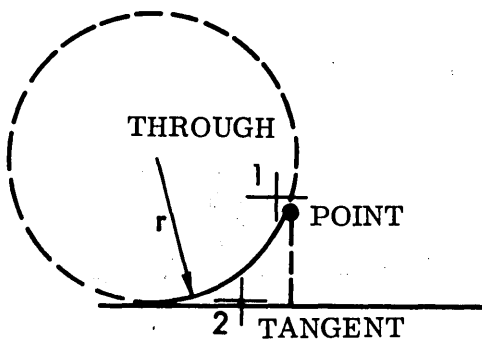
Case II: Where the line and arc intersect, positioning the tracking cross inside the arc indicates an internal tangency and positioning outside indicates an external tangency.



8. By a Center Point and an Arc to Which it is Tangent:
  - a. Push the ARC button  
or  
Pick the ARC function.
  - b. Position the tracking cross to establish the center point.
  - c. Push the ACCEPT TC button  
or  
Pick the ON function.
  - d. Position the tracking cross on the arc near the desired point of tangency.
  - e. Pick the TANGENT function.
  
9. Passing through a Point and Tangent to a Line With a Given Radius:
  - a. Push the ARC button  
or  
Pick the ARC function.
  - b. Position the tracking cross to establish the THRU point.
  - c. Pick the THRU function.
  - d. Position the tracking cross on the line.
  - e. Pick the TANGENT function.
  - f. Pick the radius value.
  - g. Push the NOMORE button.

NOTE:

Due to the clockwise convention of arc generation, steps b. through d. must be reversed for certain possibilities (see illustrations below).

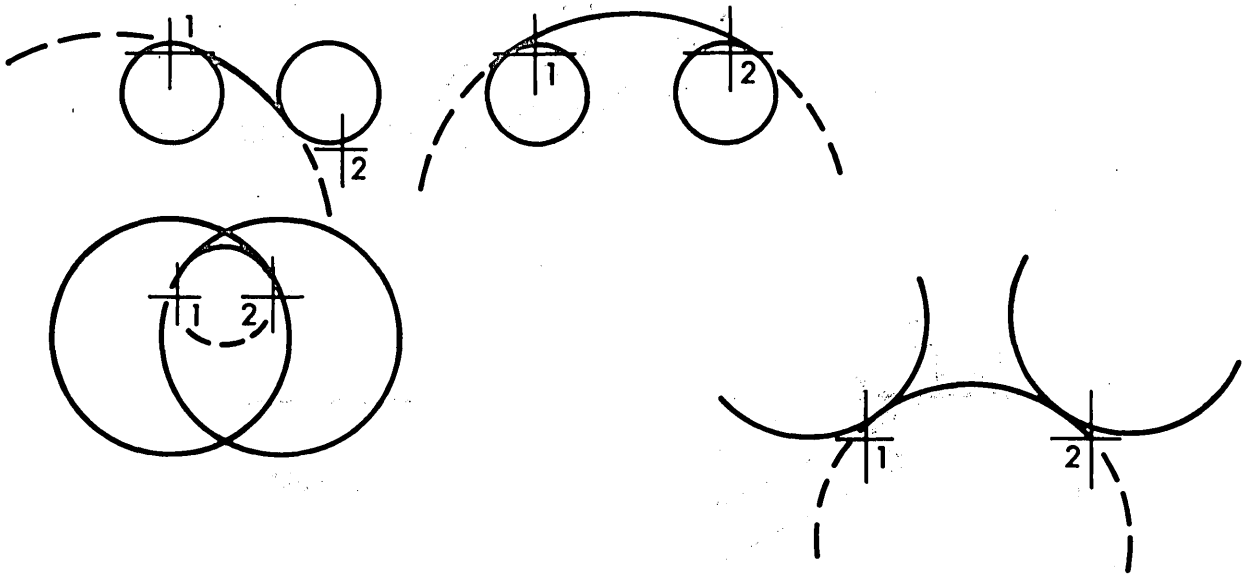


10. Passing Tangent to Two Arcs With a Given Radius:

- a. Push the ARC button  
or  
Pick the ARC function.
- b. Position the tracking cross on the first arc.
- c. Pick the TANGENT function.
- d. Position the tracking cross on the second arc.
- e. Pick the TANGENT function.
- f. Pick the radius value.
- g. Push the NOMORE button.

NOTE:

Always position the tracking crosses near the desired points of tangency. Refer to the illustrations below for internal and external tangency conditions.



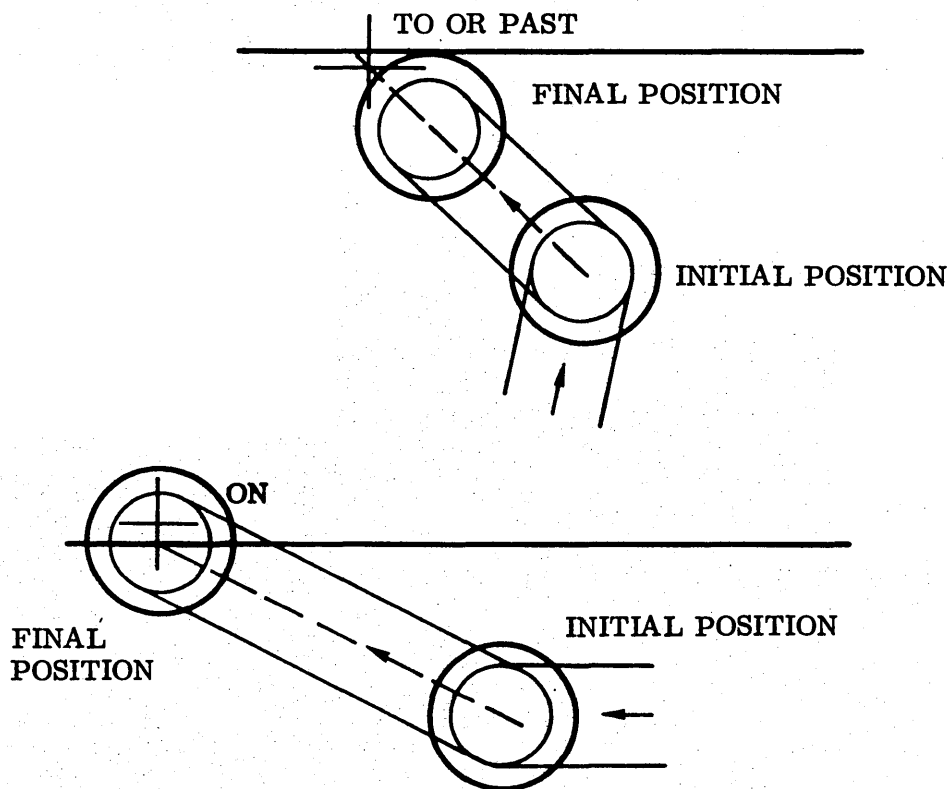
1.3.2.4 Motion Commands. Motion commands are described in the following paragraphs.

One-Surface Startup.

- a. Push PT-TO-PT MODE button.
- b. Pick the CKSURF function.
- c. Position the tracking cross near the surface.
- d. Pick TO or PAST or ON function.

NOTE:

Step a. is modal until changed.

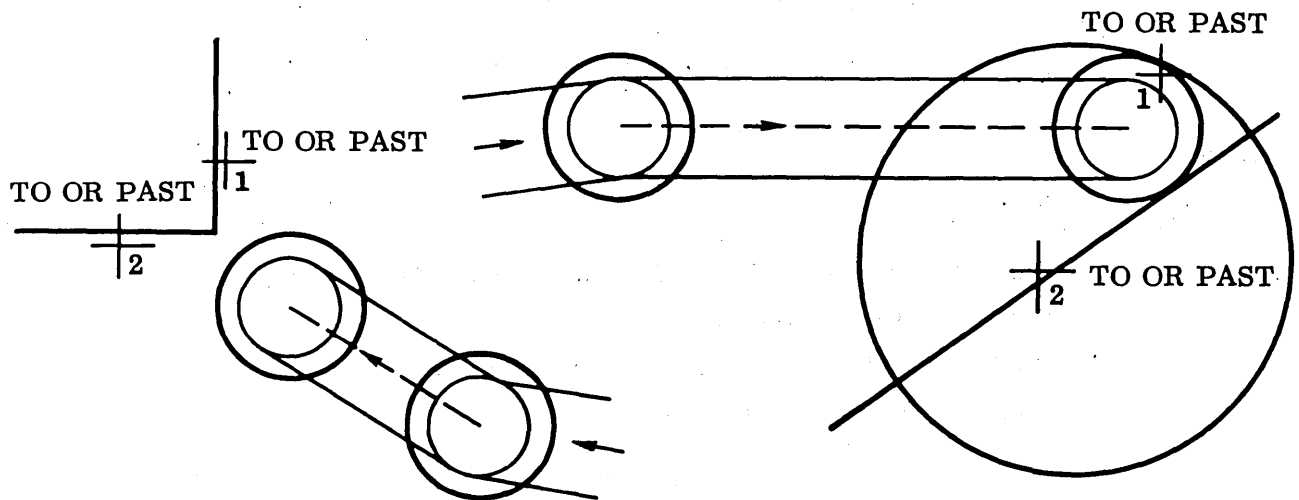


### Two-Surface Startup.

- a. Push PT-TO-PT MODE button (modal until changed).
- b. Pick DIRECTION function
- c. Position the tracking cross on first surface.
- d. Pick TO or PAST or ON function.
- e. Position the tracking cross on second surface.
- f. Pick TO or PAST or ON function.

#### NOTES:

1. The second surface indicated is a check surface and must be the next drive surface if peripheral mode is contemplated.
2. In the case of arcs, TO, ON or PAST conditions will limit the two surface startup to two possibilities. The first tracking cross location must be nearer the desired location.



### TC-to-TC Mode.

- a. Push TC-TO-TC MODE button (modal until changed).
- b. Pick CKSURF function.
- c. Position the tracking cross as desired.



- d. Pick the TO or PAST function.

NOTE:

This mode operates independently of the "gravity field". The cutter path is from one tracking cross location to another.

PT-to-PT Mode

- a. Push PT-TO-PT MODE button (modal until changed).
- b. Pick the CKSURF function.
- c. Position the tracking cross as desired.
- d. Pick the TO or PAST function.

NOTES:

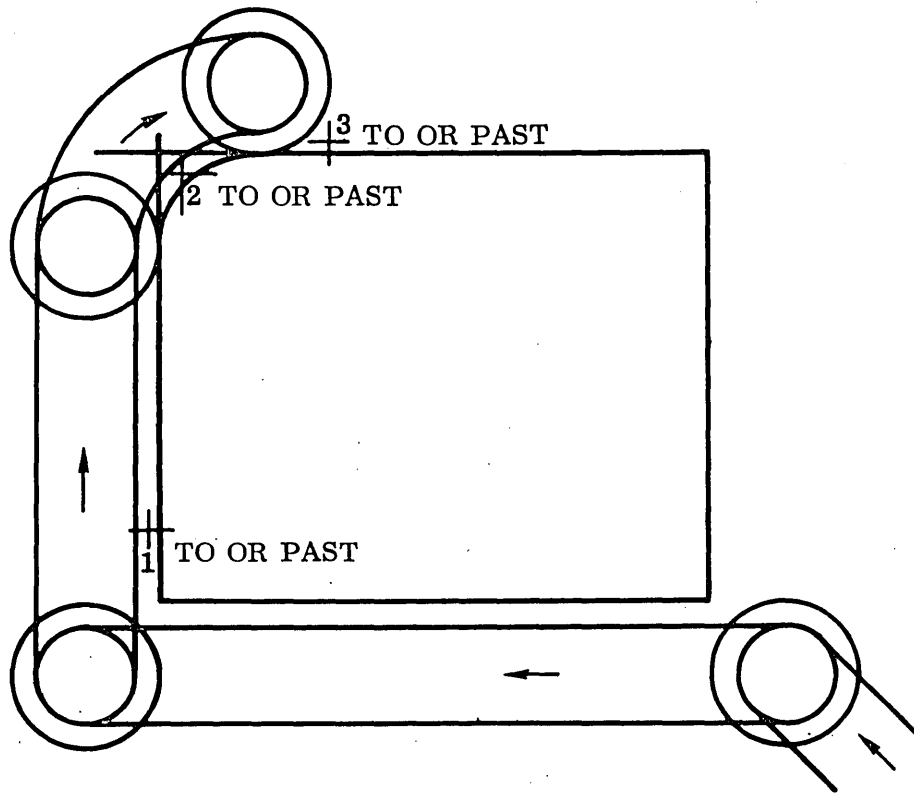
1. Major use will be for drilling, boring, and reaming operations. However, it can be used to offset surfaces in the same way a one surface startup is used. Reference the one surface and two surface startup.
2. This mode is modal within the system.

Peripheral Mode

1. Step Method:
  - a. Push PERIPHERAL MODE button (modal until changed).
  - b. Pick the CKSURF function.
  - c. Position the tracking cross near a surface.
  - d. Pick TO or PAST or ON function.

NOTES:

1. This method will generate one cutter path at a time. It essentially steps from one surface to the next, as indicated.
2. A startup is required prior to using the peripheral mode.



2. Automatic Method:

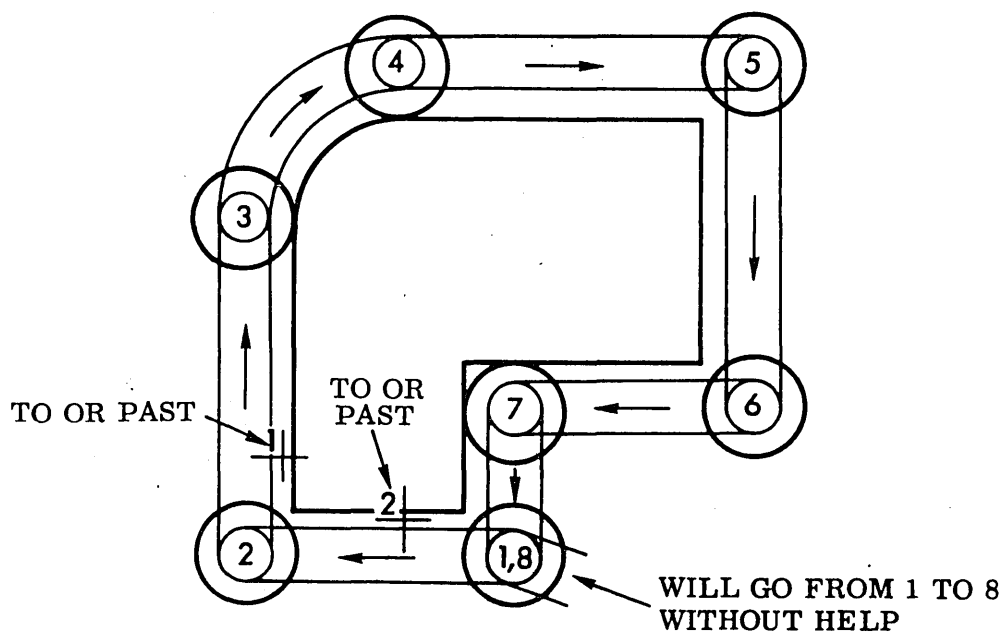
- a. Push PERIPHERAL MODE button (modal until changed).
- b. Pick DIRECTION function.
- c. Position the tracking cross near the next surface.
- d. Pick TO or PAST or ON function.
- e. Position the tracking cross near the final surface.
- f. Pick TO or PAST or ON function.

NOTES:

1. All surfaces must be connected by end points to form a continuous string of lines and arcs.
2. Step c. must be the next line or arc.

### NOTES (Cont.)

3. There must be no ambiguous connections; that is, there cannot be more than one connected path to the final surface.
4. This routine should be avoided because it has not been debugged.



#### Go To Point.

- a. Push the PICK button.
- b. Pick G  $\checkmark$  X  $\checkmark$  Y  $\checkmark$  Z.
- c. Push the NOMORE button.

#### Go Forward.

- a. Pick GOFWD function.
- b. Position the tracking cross.
- c. Pick the TO or PAST or ON function.

NOTE:

This feature overrides the convention which states that the last CKSURF will be the next drive surface. The last drive surface will be retained and motion made to another CKSURF.

Direction. Direction is used in three ways:

- a. As the first surface in a two surface startup.
- b. To indicate the direction of movement in automatic peripheral mode.
- c. To establish direction of motion around arc drive surfaces. By convention, direction of motion is established by measuring the angle of sweep between the cutter center and the tracking cross position for the CKSURF. Movement is in the direction of the small sweep angle.

This convention may be overridden by establishing a small sweep angle with judicious positioning of the DIRECTION tracking cross.

AUTOCKSURF. When operating in tracking cross and peripheral mode, repetitive picks of the CKSURF button are required. AUTOCKSURF eliminates this repetitive pick by reactivating the CKSURF routine after each movement.

Reject Cutter Path. The last, but only the last, item of motion data may be rejected. This item might be a PARTNO, Z-VALUE, cutter path, mode button, etc.

Number of Cutter Paths. The number of cutter positions which are displayed is modally set at four. This may be altered at any time by:

- a. Pushing the PICK button
- b. Picking NCP / K
- c. Pushing the NOMORE button

Where K = the number of cutter positions to be retained on display.

Check Points. CL-DATA may be segmented by entering check points from time to time. This permits faster access to portions of CL-DATA deep within a part

program. Thus, when creating a program, the more check points which are entered the more directly accessible positions within CL-DATA are available when replaying.

- a. Push the PICK button.
- b. Pick CKPNT.
- c. Push the NOMORE button.

1.3.2.5 Machine Functions. Machine functions are described in the following paragraphs.

Part No.

- a. Push the PICK button.
- b. Pick PARTNOXXXXXXXXXXXXX.
- c. Push the NOMORE button.

NOTES:

1. A PARTNO should not exceed 68 characters in length.
2. A PARTNO is required to dump to magnetic tape.

Cutter.

- a. Push the PICK button.
- b. Pick CU  $\sqrt{K_1} \sqrt{K_2}$ .
- c. Push the NOMORE button.

NOTE:

$$K_1 = \text{Dia, } K_2 = \text{Corner Radius}$$

Z-Value.

- a. Push the PICK button.
- b. Pick Z  $\sqrt{K}$ .
- c. Push the NOMORE button.

NOTE:

K is a positive or negative number.

Feed Rate.

- a. Push the PICK button.
- b. Pick  $FR \sqrt{K_1} \sqrt{K_2}$ .
- c. Push the NOMORE button.

NOTE:

$K_1$  = Desired feedrate in inches per minute.

$K_2$  = 88 for auto acceleration and deceleration.

$K_2$  = 92 for step acceleration and deceleration.

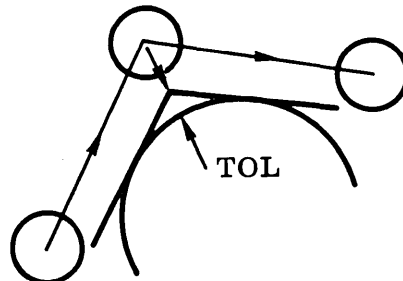
If  $K_2$  is not given, the modal value in postprocessor will be used.

Tolerance.

- a. Push the PICK button.
- b. Pick  $TOL \sqrt{K}$ .
- c. Push the NOMORE button.

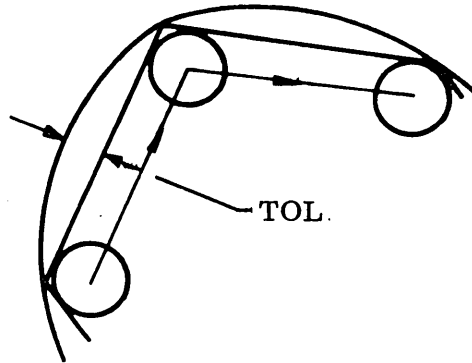
NOTES:

1. For external profiling around arcs the K value represents the maximum tangential deviation from circular.



NOTES (Cont.)

2. For internal profiling around arcs the K value represents the maximum cordal deviation from circular.



Machine Tolerance.

- a. Push the PICK button.
- b. Pick MCHTOL  $\checkmark$  K.
- c. Push the NOMORE button.

STOP.

- a. Push the PICK button.
- b. Pick STOP.
- c. Push the NOMORE button.

Reference Point.

- a. Push the PICK button.
- b. Pick REFPT  $\checkmark$  K.
- c. Push the NOMORE button.

Coolant Control.

- a. Push the PICK button.
- b. Pick  $\emptyset$ NKUL or pick  $\emptyset$ FFKUL.
- c. Push the NOMORE button.

### Spindle Speed.

- a. Push the PICK button.
- b. Pick  $\emptyset$ NSPIN  $\vee$  K  
or  
pick  $\emptyset$ FSPIN.
- c. Push the NOMORE button.

#### NOTE:

K = spindle speed in rpm.

Positive values of K mean clockwise rotation.

Negative values of K mean counterclockwise rotation.

### REWIND.

- a. Push the PICK button.
- b. Pick REWIND.
- c. Push the NOMORE button.

### FINI.

- a. Push the FINI button.  
A message will appear in the upper left-hand corner above the frame stating:  
"Verify readiness of proper dump tape."
- b. Communicate via the intercom with the computer operator and verify that the paper tape is mounted on Channel O, Unit K, where K is the console number of the requesting station.
- c. Push the FINI button again.  
All data relative to the job on Console K will be written on magnetic tape.  
This is the master record of replay tape to be labeled.

### CL Tape.

- a. With the FINI procedure complete, push CL-TO-PP button.  
A message will appear stating: "Verify readiness of proper postprocessor tape."



- b. Communicate via the intercom with the computer operator and verify that the proper tape is mounted on Channel 1, Unit 2.
- c. Push the CL-TO-PP button again (Ref. Operations Section for Tape Control Procedures).

1.3.2.6 Special Features. Special features are described in the following paragraphs.

Arc Centers. Picking the Arc Centers button will cause displayed center points to disappear and undisplayed center points to appear.

CANON. CANON is used to obtain canonical forms of elements.

- a. Pick the CANON function.
- b. Position the tracking cross on the desired element.
- c. Push the THIS SURFACE button  
or  
pick the TO or PAST function.
- d. See Note 5 below.

NOTES:

1. The canonical form is displayed above the upper left hand edge of the frame.
2. The canonical form of a point is POINT = X √ Y (coordinates).
3. The canonical form of a line has the end points.  
Line = X<sub>1</sub> √ Y<sub>1</sub> √ X<sub>2</sub> √ Y<sub>2</sub>.
4. The canonical form of an arc is the center point and radius.  
ARC = X √ Y √ R.
5. After observing the canonical form of the first element, other elements may be interrogated by repeating steps b. and c. above. The feature is terminated by picking the CANON function again.

Delete Geometry. This feature is used to delete from display and memory all record of the geometric element being removed.

- a. Pick the DELETE GEOM function.
- b. Position the tracking cross on the desired element.
- c. Push the THIS SURFACE button  
or  
pick the TO or PAST function.

The tracking cross will go away and reappear. If it is desired to delete additional elements repeat steps b. and c. above.

- d. Pick DELETE GEOM function again. This terminates delete and removes the display.

NOTE:

Do not delete an element used as a DIRECTION surface or a CHECK surface of motion command. They are required for REPLAY.

DUMP. This feature is used to periodically store on magnetic tape the information relative to a specific console.

- a. Push the DUMP button.  
A message will appear above the left hand edge of the frame stating, "Verify readiness of proper dump tape."
- b. Communicate via the intercom with the computer operator and verify that the proper tape is mounted on Channel 1, Unit K, where K is the console number of the requesting station.
- c. Push the DUMP button again (Ref. Operations Section Tape Control).

ERASE. This feature is used to abbreviate the display of lines and arcs to provide clarity and/or connectivity.

- a. Pick the ERASE function.
- b. Position the tracking cross on the surface to be shortened.
- c. Push the THIS SURFACE button  
or  
pick the TO or PAST function.

- d. Position the tracking cross in space, near an end point or on a surface.
- e. If an end point or tracking cross position:
  - Push the ACCEPT TC button
  - or
  - pick the ON function.
 If a surface:
  - Push the THIS SURFACE button
  - or
  - pick the TO or PAST function.
- f. Repeat Steps d. and e. for a final condition.

NOTES:

- 1. For arcs, ERASE functions in a clockwise direction from the first surface or point to the second.
- 2. ERASE should not be used in place of DELETE.

EXTEND. This feature is used to lengthen the display of lines or arcs for clarity and/or connectivity.

- a. Pick the EXTEND function.
- b. Position the tracking cross on the surface to be lengthened.
- c. Push the THIS SURFACE button
  - or
  - pick the TO or PAST function.
- d. Position the tracking cross near the end point to be affected.
- e. Push the ACCEPT TC button
  - or
  - pick the ON function.
- f. Position the tracking cross in space or near a surface.
- g. If in space:
  - Push the ACCEPT TC button
  - or
  - pick the ON function.
 If near a surface:
  - Push the THIS SURFACE button
  - or
  - pick the TO or PAST function.

NOTE:

EXTEND functions from one end point of the surface to a tracking cross location or to another surface.

INPUT TAPE. Any feature which may be input via the alphanumeric pick table may be read in from magnetic tape. Punched cards, beginning in Column 1 and using the identical formats which would be used if entered via the alphanumerics, are put on magnetic tape. The last card must be double punched 7 and 8 in Column 1. Several groups of input cards may be put on the same magnetic tape if separated by the double punched 7,8 card. The order and number of groups must be known by the part programmer.

- a. Push the INPUT TAPE button.
- b. Verify, via intercom, that the proper tape is mounted on Channel 1, Unit O.
- c. Push the INPUT TAPE button again.

Invisible Geometry. This feature is used to temporarily remove from display geometric surfaces in order to provide clarity or ease in "attaching".

- a. Pick the INVIS GEOM function.
- b. Position the tracking cross on the surface desired.
- c. Push the THIS SURFACE button  
or  
pick the TO or PAST function.
- d. Repeat Steps b. and c. as many times as desired.
- e. Pick the INVIS GEOM function again.

JUMP TO CKPNT. This feature is used to replay through CL-DATA without performing all calculations and displays. You may jump to check points and begin stepping through all operations. This feature shortens the required time to make corrections or check certain areas for errors if you are replaying and in tracking cross mode.

- a. Push the JUMP TO CKPNT button.
- b. Pick the CKPNT number.
- c. Push the NOMORE button.

NOTES:

1. If the last CKPNT number is not known, a very large number (i.e., 1000) will jump to FINI.
2. The last cutter display will appear and the last Z-VALUE, feedrate value, mode, etc. will be activated.
3. After stepping beyond the PARTNO, the JUMP TO CKPNT feature must be used only when in tracking cross mode.

Light Pen. The light pen may be activated by depressing the switch on the pen or by depressing the light pen button on the button box. This button further serves to indicate an active pen by remaining lighted when switch is depressed.

LOAD. This feature recalls from magnetic tape storage only geometry data. Thus, if a REPLAY TAPE is read in using the LOAD feature, no CL-DATA data will be entered.

- a. Push the LOAD button.
- b. Verify readiness of proper LOAD tape.
- c. Push the LOAD button again.

MASTER REPLAY. This feature is used to select and load a REPLAY TAPE from a magnetic file containing numerous REPLAY tapes.

- a. Push the MASTER REPLAY button.
- b. Verify readiness of proper LOAD tape.
- c. Push the MASTER REPLAY button again.
- d. Pick the desired file number.
- e. Push the NOMORE button.

NOTES:

1. See Operations for further information on master files.
2. Do not use MASTER REPLAY unless the file number is small or requirement is imperative.

OBTAIN REFSYS. This feature is used to reenter a closed reference system in order to REFSYS additional geometric inputs.

- a. Pick the OBTAIN REFSYS function.
- b. Position the tracking cross on the REFSYS axis indicator.
- c. Push the THIS SURFACE button  
or  
Pick the TO or PAST function.

Origin. This feature moves the coordinate system about the face of the scope in order to view different portions of the display.

- a. Push the PICK button.
- b. Pick  $\emptyset \vee X \vee Y$ .
- c. Push the NOMORE button.

NOTE:

X and Y are values from the center of the scope to the coordinate system.

Origin and Scale. This feature combines the ORIGIN feature with a scaling feature. The ORIGIN shift is performed first then the scaling is performed about the center of the scope.

- a. Push the PICK button.
- b. Pick  $\emptyset \vee X \vee Y \vee S$ .
- c. Push the NOMORE button.

NOTE:

S is the desired scale value.

Reference System. This feature allows the temporary establishment of a second (or more) coordinate system.

- a. Push the PICK function.

- b. Pick RFS  $\checkmark$  X  $\checkmark$  Y  $\checkmark$   $\alpha$ 
  - or
  - RFS  $\checkmark$  X  $\checkmark$  Y
  - or
  - RFS  $\checkmark$   $\alpha$ .
- c. Push the NOMORE button.

NOTES:

1. A half arrow is displayed from the origin of the reference system along the + X axis to visually denote its orientation.
2. If a second reference system is input with the first one active this second REFSYS will be with respect to the first REFSYS. The first will become dormant and the second active.
3. No geometric formats which require tracking cross assistance (i.e., a line parallel to another line) may be used with a REFSYS active.
4. To deactivate a REFSYS:
  - Pick RFS.
  - Push the NOMORE button.

Reinitialize. This feature restores the console to its initial state removing any record of previous work done. All modal values are reinstated and the console set up as if at the sign-on stage.

- a. Push the REINIT button.

NOTE:

Do not use REINIT until all information is safely stored on magnetic tape.

See All Invisibles. This feature recalls the display of all surfaces previously made invisible.

- a. Pick the SEE ALL INVIS button.

Scale. This feature is used to reduce or expand the size of the display. The change of scale is accomplished about the center of the scope.

- a. Push the PICK button.
- b. Pick  $S \sqrt{K}$ , where  $K$  is the desired scale value.
- c. Push the NOMORE button.

System Dump. This feature is used to store on magnetic tape all information necessary to reestablish the present condition of the system. If a computer stop occurs, the last System Dump may be read into the computer and work may continue from that point. This feature is an expansion of the DUMP feature in that it dumps all data for all consoles.

Every 30 minutes the computer operator will interrupt console operation to take a system dump. No action of the part programmer is required. After the system dump has been taken the computer operator will notify the part programmers that they may continue.

Tracking Cross. If for some reason the system fails to return a tracking cross when required, one may be obtained by pushing the Tracking Cross button.

REPLAY. The REPLAY feature is used to:

- Re-enter an incomplete part program in order to complete it
  - Check a part program
  - Make corrections to a part program
  - Make additional "dash-numbered" parts from existing part programs
- a. Push the REPLAY button.
  - b. Verify readiness of proper load tape.
  - c. Push the REPLAY button again.

The geometry will appear in the configuration existing at the time of dumping or FINI'ing.

- d. Push the STEP button.

The PARTNO will appear to further verify that the proper tape was loaded.



- **STEP:** If you desire to check the entire cutting sequence, push the STEP button. If at any time you wish to alter CL-DATA you may pick REJECT C/P which will reject the last STEP performed. If at any time you wish to add motions or CL-DATA, you may do so by normal procedures. Additions must eventually connect logically with the function of the next STEP so that you may continue stepping through the previously programmed CL-DATA. A small X-mark is displayed to show the original position of the tracking cross and serves to indicate the CKSURF used to perform motions.
- **JUMP TO CKPNT:** If you do not desire to step through all previously programmed CL-DATA and the replay tape has check points you may accept, without replaying, portions of this data by using the JUMP-TO-CKPNT feature. The CL-LISTING or preprogramming notes provide a numerical listing of the CKPNT segmentation.
  - a. Push the JUMP-TO-CKPNT button.
  - b. Pick the CKPNT value.
  - c. Push the NOMORE button.

After using JUMP-TO-CKPNT, the stepping feature may again be used.

**NOTE:**

JUMP TO CKPNT may be used if the drive surface is not a line or arc. Therefore check points should follow tracking cross movements or movements to displayed points.

- **SKIP:** If when replaying you wish to omit one or more steps of CL-DATA, depress the SKIP button rather than the STEP button. If the skipped data was motion, the X-marker will advance to the original tracking cross location but no cutter motion will be performed. If the skipped data was machine function or monitored parameter, the message will appear then be automatically rejected.

**1.3.2.7 Operations.** Operations are described in the following paragraphs.

Operating System. The system can be defined as a three-console, on-line, time-shared, real-time, dedicated, graphical part programming system. On-line indicates that the part programmer is inputting directly to the computer. Time-shared means that each console shares the time of the computer. Real time indicates an on-line simulation of some exercise. Dedicated indicates that the computer is required to do nothing but respond to console requests.

Hardware. The hardware or computing machinery and necessary peripheral equipment consists of a computing system and a three-console computer graphics system.

- Central Processor: The central processor and necessary controllers to perform computations and process data
- Disk: A peripheral storage device which is used to store program and CL-DATA.
- Drum: A peripheral storage device used to store program and display data (the drum maintains the display)
- Controller: The communications link between the display consoles and the computer
- Display Consoles: The display unit consisting of a CRT display, light pen, and button box
- Tape Transports: The devices used to read or write information on magnetic tape. This equipment is required using the DUMP, FINI, INPUT TAPE, REPLAY, MASTER REPLAY, LOAD, SYSTEM DUMP, and CL-TO-PP features.
- Computer Console: Work area for computer operator
- Disk: A bulk storage device used to store program data
- Magnetic Tape Files: Storage bins used to file REPLAY tapes and MASTER REPLAY tapes

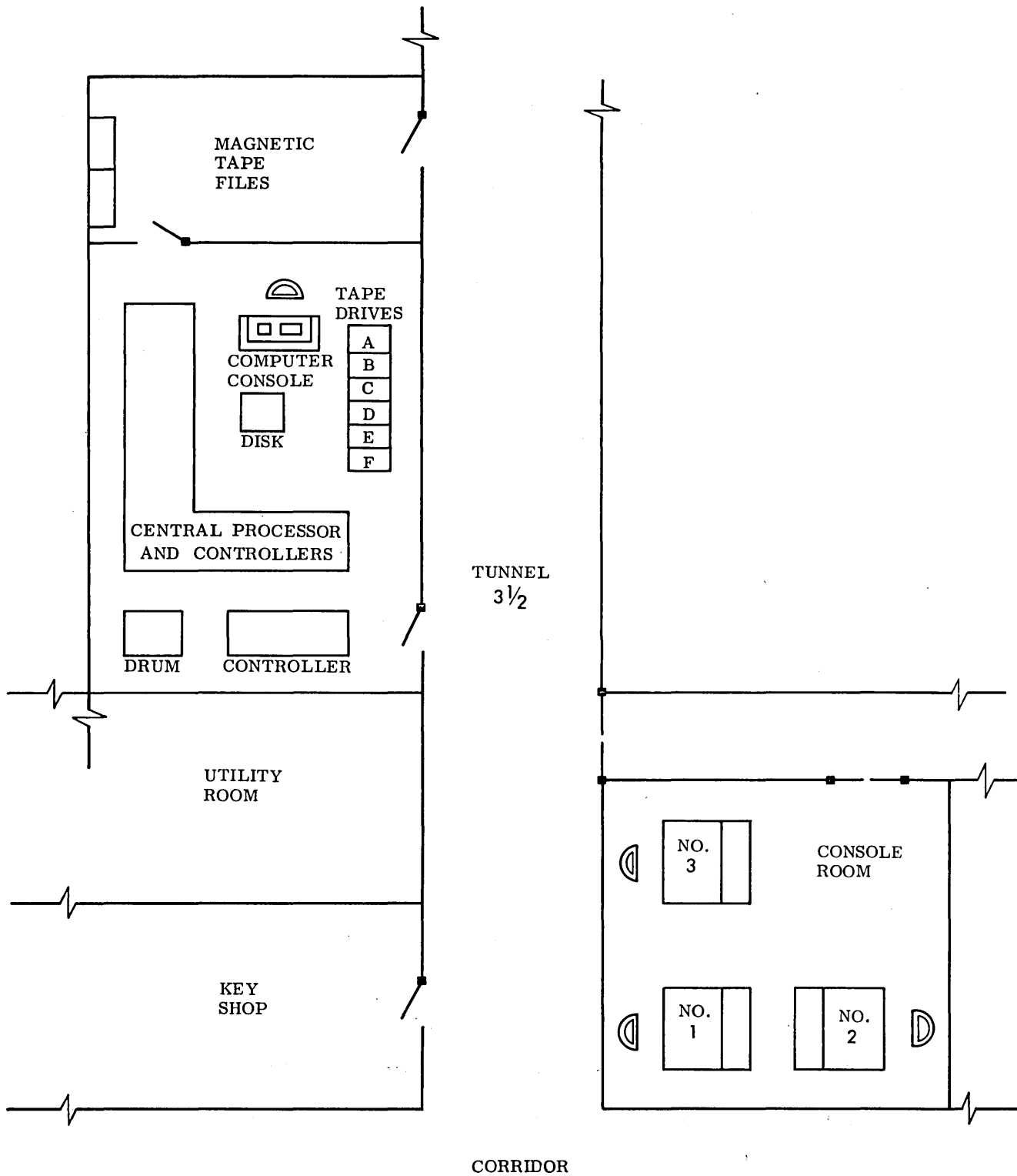


Fig. 1-15 Local Area Layout of Display-Oriented Support Equipment

### 1.3.3 Software Specifications and Descriptions

1.3.3.1 Drafting and Dimensioning Package Specifications. Package specifications are described below.

#### Capabilities.

a. Geometry Storage and Display:

POINT

1. Light pen detect (LPD) location
2. Keyboard input or pick function the location
3. Intersection of geometric elements

LINE

1. Through two points
2. Through point and tangent to circle
3. Tangent to two circles
4. Through point and at angle
5. Parallel to other line through a point
6. Parallel to other line and tangent to circle
7. Through a point perpendicular to a geometric element
8. Tangent to a circle and perpendicular to a geometric element

CIRCLE

1. Keyboard or pick function input of center and radius
2. Through two points with keyboard or pick input of the radius
3. Through three points on the periphery

FILLET

1. Key or pick fillet radius, indicate elements and location of fillet with light pen

CURVES

1. Spline input by keyboard, pick or light pen detection of point locations
2. Conics
  - Equation input

- Five elements
  - Two angles - three points
  - One angle - four points
  - Five points

3. Cubics

Equation input

RETOUCH

1. Expand and contract lengths of lines to specific locations
2. Contract curves to specific locations
3. Break elements into separate entities
4. Contract broken elements (circles, curves)

- b. Dimensioning: The computer calculates and displays dimension requested including dimension lines and arrowheads to acceptable drafting quality.

Dimension types:

- Horizontal
- Vertical
- Parallel
- Offset
- Angle
- Radius
- Diameter
- Curve

Hard Copy. Output elements and alphanumeric characters (shown on CRT) on tape in numerical control drafting machine-compatible language.

1.3.3.2 Numerical Control and APT Interface Specifications. Specifications are described below.

Geometric Construction. Geometric construction will utilize all capabilities of the engineering drafting package.

Response Time. Average response time is not to exceed two seconds.

Statistics. Utilize the statistics output of the engineering drafting package:

User Response

Min-Max-Mean

System Response

Min-Max-Mean

Run Time

APT Interface. This includes the ability to input and display geometry from cards compatible with APT geometry format and CL tape output for input to APT system for postprocessing.

Model. Geometric elements will be stacked from the front of the model toward the rear. N/C Cutter location data will be stacked from the back of model toward the front. A suitable test will be performed to ensure that interference is flagged to the operator and/or core restacked automatically to take up space left by erasures.

Miscellaneous Functions (SPINDLE, COOLANT, TOOL NO., Etc.). Input – keyboard or pick-APT compatible.

Hard Copy Interface. Output the contents of the display in drafting-machine-compatible language, including alphanumeric characters.



## Section 2

### ELECTRICAL NETWORK ANALYSIS

A system that will enable a circuit designer to use computer graphics to perform electrical network analysis is discussed in this section, and an outline specification of the system is presented.

The system is based on the principle that circuit design is an iterative process of concept, analysis, and modification. In response to a specification, a designer will conceive a circuit that he believes will meet the requirements on a qualitative basis. The concept, normally expressed in the form of a sketched schematic, will be subjected to analysis through a series of time consuming hand calculations and/or bread-board tests, and data will be collected, collated, and evaluated. Then the obtained results will usually be used to modify the original concept, and the modified concept will be subjected to similar analysis. This process will continue until a design is realized.

With the computer graphics tool, the process is pursued in a similar manner. Given the specification, the designer will sketch his schematic with a light pen on a CRT face, enter circuit parameter data with the use of light registers, and command the type of analysis desired with the use of light buttons. The system will respond quickly and provide him pertinent circuit performance data in tabular and/or plot form on a portion of the CRT adjacent to his schematic. After the designer evaluates the displayed data, he may alter the schematic and/or parameter values at once, then he will reexecute the analysis and display functions with the altered concept.

Although the two processes are virtually identical, the computer approach compresses the analysis process from several hours or days to a matter of seconds or minutes. By entering a schematic as input and receiving tables and/or plots as output, the designer interfaces with the system in terms of his own language. (This latter feature is unique to computer graphics.)



As a net result, computer graphics provides the designer with a better job, completed in significantly less time.

## 2.1 SUMMARY

This section consists of a network analysis program study and evaluation, a description of the procedures the user would follow in performing network analysis using graphics, a brief outline of network analysis graphics applications within Lockheed Aircraft Corporation and elsewhere, and a description of general system requirements for network analysis.

In the program study, the criteria for program selection are delineated and each major available program is discussed. It is concluded that the ECAP, POTTLE, and SCEPTRE programs together best meet these criteria.

The procedures portion of the report defines the steps the user follows in entering the network into the graphics system and describes the types of output available to him. Finally, general programming requirements, as based on the (LMSC) Graphics Configuration are presented.

## 2.2 NETWORK ANALYSIS PROGRAM REQUIREMENTS FOR THE GRAPHICS SYSTEM

The basic requirements of (a) network analysis program(s) for graphics computation are described in the following paragraphs.

### 2.2.1 Functional Capability

The program(s) should be capable of performing both steady-state (DC and AC) and transient time domain analyses on linear and non-linear, passive, and active networks. The program(s) should also be capable of performing frequency domain analysis on linear passive and active networks.

### 2.2.2 Circuit Size

The program(s) should be capable of handling reasonably large networks. (Reasonably large in this context would be 30-plus nodes and 100-plus elements.)

### 2.2.3 Model Flexibility

It is important to point out that the modeling of circuit components is the key to success (or failure) in the performance of a network analysis program. Accordingly, the program(s) should handle a wide variety of devices, each having models ranging from simple to complex (see Para's. 2.2.5 and 2.2.6). In terms of structure, the program(s) should accommodate "stored" models of commonly used devices and have the capability for the user to generate and store his own model where required.

### 2.2.4 Parameter Variation

The program(s) should permit the user to alter element values and/or circuit topology and obtain a rapid display of the effects of such changes.

The program(s), moreover, should have the allied capability to perform worst-case and Monte Carlo parameter variations.

### 2.2.5 Accuracy

The accuracy of a network analysis program is determined by both program-oriented and model accuracy.

In the former case, the program(s) should minimize (or detect) errors due to round-off, improper integration step size, numerical instability, etc.

Given the minimization of these sources of error, the model accuracy will dominantly govern the accuracy of the network analysis program results. Since a tradeoff generally exists between model accuracy and speed (see Para. 2.2.6), specific model

accuracy should be tailored to the problem at hand. (For example, design concept verification might use relatively coarse models whereas detailed, "breadboard"-quality simulations might require elaborate models.) As a result, the model accuracy is closely tied into the model flexibility.

#### 2.2.6 Speed

As noted, model simplification enhances speed of computation. Additionally, the program(s) should provide the capability of easily varying sample size, e.g., number of frequencies in frequency response, step size in transient analysis, etc.

#### 2.2.7 Ease of Program Conversion

The program(s) should be convertible to graphics with a minimal amount of program modification and ancillary program development.

### 2.3 DESCRIPTION OF AVAILABLE NETWORK ANALYSIS PROGRAMS

It should be noted at this point that the majority of programs available at the present have been written for use on batch processors. As a result, none of these incorporate all the features that are required in the graphics environment. Moreover, many of the programs are recent developments and consequently have had limited use.

Therefore, detailed discussion is confined to those programs that best meet the graphics requirements and which also have been applied to a wide variety of industrial and government problems. Programs considered in the survey but not discussed in detail are listed with appropriate comments in Table 2-1.

#### 2.3.1 Detailed Program Discussion

For the reasons previously noted, the following detailed discussions are confined to the CIRCUS, ECAP, GNAS, NASAP, NET-1, POTTLE and SCEPTRE programs.

Table 2-1

## NETWORK ANALYSIS PROGRAMS

Program Name	Origin	Description
ARINC	Arinc Research Corporation Santa Ana, California	Solves simultaneous linear circuit equations which the user derives and inputs into program. Performs sensitivity, worst case, and Monte Carlo Analyses.
ASAP (Automated Statistical Analysis Program)	IBM Corporation	Performs DC Monte Carlo Analysis of electronic circuits.
CALAHAN	University of Michigan Dept. of Electrical Engineering	Performs DC and AC linear frequency domain analysis using topological (K-tree method)
CIRCUS	Boeing	See text.
DC AENEAS (DC Automatic Electric Network Analysis)	Automatic Electric Laboratories, Incorporated, Illinois	Performs DC linear analysis using topological formulas (K-tree method).
ECAP	IBM Corporation	See text.
GNAS	Lockheed Missiles & Space Company	See text.
KNCAC (Kaman Nuclear Circuit Analysis Code)	Kaman Nuclear Colorado	Basically a version of ECAP modified for use in radiation effects studies.

Table 2-1 (Cont.)

Program Name	Origin	Description
MISSAP (Michigan State System Analysis Program)	Michigan State University	Basically a transient circuit analysis program. Includes capability for analysis of distributed parameters elements (e.g., transmission lines).
NASAP	NASA/EREC	See text.
NET-1	Los Alamos Scientific Laboratory, Los Alamos, New Mexico	See text.
Oklahoma State Analysis Program	Oklahoma State University Stillwater, Oklahoma	Basically a transient circuit analysis program.
PREDICT	IBM Corporation	Forerunner to SCEPTRE.
POTTLE	Cornell University, Dept. of Electrical Engineering Ithaca, New York	See text.
SCEPTRE	IBM Corporation	See text.
STRAP (Simplified Transient Radiation Analysis Program)	Douglas Aircraft Company Santa Monica, California	Transient circuit analysis program written for use in radiation effects studies.
TAG (Transient Analysis Generator)	Jet Propulsion Laboratory Pasadena, California	Transient circuit analysis.
TRAC (Transient Radiation Analysis by Computer)	Autonetics Anaheim, California	Transient circuit analysis program written for use in radiation effects studies.

CIRCUS (Circuit Simulator). CIRCUS was developed by the Boeing Company for the purpose of studying nuclear radiation effects on electronic circuits. Functionally, CIRCUS performs DC and transient analysis on linear and non-linear networks. The program can handle a circuit of 100 nodes and 300 branches (Univac 1108).

CIRCUS employs a stored (Beaufoy-Sparks) transistor model. (Zener diode and FED models are currently under development.) The model cannot be readily changed by the user.

There is no provision within CIRCUS for automatic parameter modification. As is the case with all existing network analysis programs, CIRCUS is susceptible to errors arising from roundoff, improper time step size and (for the case of non-linear programs) non-convergence. Where these errors can be avoided, CIRCUS, by virtue of its use of non-linear models, provides a high degree of solution accuracy.

In terms of input and output, CIRCUS is adaptable to graphics. The program is written primarily (98 percent) in Fortran IV.

ECAP (Electronic Circuit Analysis Program). ECAP was developed through the joint efforts of IBM and the Norden Division of United Aircraft. The program performs both steady-state (DC and AC) and transient analysis with the additional feature that DC worst-case and standard deviation analyses can be performed.

The program is capable of handling a reasonably large circuit (50 nodes, 200 branches) in the Univac 1108 and IBM 7094 versions.

ECAP employs a user-derived model (piecewise linear) in which the designer develops an equivalent circuit for devices used in the circuit he wishes to analyze. A unique feature of the program is the incorporation of switches which permit component values to be altered when selected currents reach predetermined values. This feature provides the capability to construct piecewise linear models of non-linear devices. The program does not provide the capability to store device models.

Parameter variations are effected in the program through incorporation of a MODIFY command which replaces the value of one or more components and recomputes the solution. Due to the specialized matrix structure of the program, only that part of the circuit containing modified parameters is recomputed, resulting in a rapid recalculation of the circuit solution. Full recomputation, however, is required where circuit topology is changed.

Solution accuracy, where roundoff errors, improper step size, etc. do not dominate, is dependent upon the choice of mode. Through the use of the switches, highly accurate models can be constructed but are done so at the expense of allotted circuit size.

By merit of its simplified input language and comprehensive array of output variables, ECAP is readily amenable to graphics. The program is written in Fortran IV.

GNAS (General Network Analysis System). GNAS, which is based on NET-1 and an assortment of ancillary mathematical and reliability subroutines, was developed by LMSC, beginning in 1963. The program performs both steady-state (DC and AC) and transient analyses. In addition, the steady-state (DC and AC) portion provides worst case, Monte Carlo, and Fourier analysis (AC) solutions. The program is limited to 30 nodes on the Univac 1108.

GNAS employs stored (Ebers-Moll) models which the user calls out by device type and number. The number and variety of models is extensive, including diodes, transistors, FET's, zener diodes, and tunnel diodes. The program has no provision for user (active) models.

Although the program does not explicitly provide for parameter modification, component tolerances can be specified and used in the worst-case and Monte Carlo analyses.

In accommodating non-linear device models, the program affords a high degree of accuracy in solutions of the network.

As is the case with ECAP, GNAS is amenable to graphics use. The program is written in Fortran IV.

NASAP (Network Analysis for Systems Application Program). NASAP is currently under development by NASA/ERC at Cambridge, Massachusetts. The program is designed to perform steady state (AC and DC) and transient (piecewise linear) analyses.

As is the case with ECAP, the user provides an equivalent circuit to represent circuit devices. Conversely, the program does not presently provide a switching function for the non-linear capability.

Being based on the flowgraph approach, NASAP includes a unique and efficient technique for (symbolically) calculating parameter sensitivities (Bode sensitivity) which in turn can provide effects of parameter variation.

Like ECAP, program accuracy is limited to the accuracy of the equivalent circuit model.

NET-1. NET-1 was originally developed for the Maniac II computer for the Atomic Energy Commission at the Los Alamos Scientific Laboratory of the University of California at Los Alamos.

Functionally the program is the same as the DC and transient portions of GNAS (which, as noted, was derived in part from NET-1). NET-1 can handle 100 nodes and 300 elements on the IBM 7094. As with GNAS, the program employs stored (Ebers-Moll) models. The user can, however, input parameter values of his own.

Where roundoff, non-convergence, etc. effects are not present, the program results are accurate by virtue of use of the non-linear device models.

In a test analyzing several circuits on NET-1, PREDICT, and CIRCUS, Net-1 was shown to be substantially slower than CIRCUS.



Like GNAS, NET-1 input and output would present no graphics interface difficulties. The program is written in FAP.

POTTLE. The POTTLE program was developed by Prof. C. Pottle at Cornell University, Ithaca, New York. It functionally derives user-specified transfer functions of an arbitrary active linear network along with (where desired) poles and zeros and frequency response. The program is available as a package of five programs:

- ANALYSIS: A state vector analysis program
- ZROMUL: Rootfinder employing Muller's method
- LADDER: A ladder network analysis program
- TRANSFER: A program for determining frequency response
- A set of polynomial manipulating subroutines

The user incorporates his model into the circuit description as is done in case of the ECAP program. The program affords no provision for automatic parameter variation.

In terms of input and output, POTTLE is amenable to graphics computation. The subprograms ANALYSIS and ZROMUL are available in Fortran IV.

SCEPTRE (System for Circuit Evaluation and Prediction of Transient Radiation Effects). SCEPTRE is an outgrowth of PREDICT, a program designed by IBM for the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, for the purpose of studying nuclear radiation effects on electronic circuits.

The program performs steady-state DC and transient analysis and can handle a circuit of 300 nodes and 300 elements (on the IBM 7094).

The unique feature of SCEPTRE is that it can accommodate both users and stored linear and non-linear device models. The stored model may have up to 25 terminals. The program, moreover, has the capability to employ user generated (Fortran IV) subroutines.

The program will accept parameter variations on a rerun basis. There is, however, no worst-case of Monte Carlo parameter variation capability.

Due to its unique modeling flexibility, SCEPTRE is capable of providing a wide range of solution accuracies.

In terms of speed, SCEPTRE is reportedly competitive with CIRCUS and is amenable to graphics operation. The program is written in Fortran IV.

## 2.4 PROGRAM EVALUATION

The following paragraphs summarize the features of the previously described programs in terms of graphics requirements.

### 2.4.1 Functional Capability

Of the available programs, none presently perform all the required functional analyses. CIRCUS, NET-1, and SCEPTRE lack steady-state AC analysis capability. NASAP and POTTLE are at present limited frequency analysis of linear circuits. ECAP and GNAS perform only time domain analysis.

### 2.4.2 Circuit Size

In their present form, all the programs fulfill the requirements for reasonably large circuits. Where memory is limited, ECAP and GNAS could create problems.

### 2.4.3 Model Flexibility

Of the available programs, SCEPTRE stands out as having a broad flexibility in accepting and storing device models over a full range of device complexity. The program capability to interface with user subroutines is a unique and valuable feature. To a lesser degree, this capability exists in ECAP and NASAP.

#### 2.4.4 Parameter Variation

The MODIFY feature of ECAP is unique among the programs in updating solutions due to parameter variations. Where network topology is altered, however, all the programs require full recomputation of the solution. Only GNAS provides worst-case and Monte Carlo capability. ECAP provides worst-case and standard DEV for the steady-state (DC-only) analysis.

#### 2.4.5 Accuracy

Errors due to roundoffs, inadequate step size, and/or integration instability, are inherent to all of the programs.

In terms of model accuracy, CIRCUS, GNAS, NET-1, and SCEPTRE provide superior accuracy where the network contains non-linear elements.

#### 2.4.6 Speed

There have been no studies on which all the available analysis programs have been compared in terms of speed of solution.

Speed problems arise chiefly in performing transient analysis. The non-linear transient programs (CIRCUS, GNAS, NET-1, and SCEPTRE) are inherently slow since the solution of non-linear equations is an iterative process. This problem is circumvented in the ECAP and NASAP piecewise-linear approaches but at the expense of accuracy. In all the transient programs, time step size also determines the speed of solution. At present, the user is required to define time step size so that an improper choice can lead either to inaccurate results and/or excessive computation time.

Only ECAP, NASAP, POTTLE, and SCEPTRE provide speed advantages in terms of model flexibility.

#### 2.4.7 Ease of Program Conversion

All the programs are readily amenable to graphics.

#### 2.4.8 Summary Table

The foregoing discussion is summarized in Table 2-2.

### 2.5 PROGRAM SELECTION

As can be seen from the preceding paragraphs and Table 2-2, none of the programs fulfill all the graphics requirements. Considering a mix of programs, ECAP, POTTLE, and SCEPTRE together would appear to best satisfy these requirements.

In terms of circuit size and non-linear accuracy, CIRCUS and NET-1 are not considered since they show no outward advantages over SCEPTRE. GNAS is not considered, principally due to model inflexibility. Finally, NASAP is not considered, principally due to the fact that it is still in the developmental stage.

### 2.6 PROGRAM MODIFICATIONS

Since hardware is not yet specified (Subtask 4.2.5), the following modifications are confined to those required for network analysis. Except where stated, the following modifications apply to the three programs.

#### 2.6.1 Input Interface

Subroutines for interpreting the field free inputs in ECAP and SCEPTRE can be removed and the corresponding input variables can be interfaced directly with the input graphics program.

Table 2-2

## NETWORK ANALYSIS PROGRAM EVALUATION FOR GRAPHICS

Program	Complete Functional Capability	Circuit Size	Model Flexibility	Parameter Variation	Non-Linear Accuracy	Speed	Ease of Program Conversion
CIRCUS	Poor	Excellent	Poor	Poor	Excellent	Good	Good
ECAP	Poor	Good	Good	Excellent	Poor	Excellent	Good
GNAS	Poor	Good	Poor	Good	Excellent	Poor	Good
NASAP	Poor	Good	Good	Good	Poor	Good	Poor
NET-1	Poor	Excellent	Poor	Poor	Excellent	Poor	Good
POTTLE	Poor	Good	Good	Good	NA	Good	Good
SCEPTRE	Poor	Excellent	Excellent	Poor	Excellent	Good	Good

### 2.6.2 Output Interface

Output plot (ECAP) and print plot (SCEPTRE) subroutines can be eliminated and output variables can be interfaced with existing graphic display routines.

### 2.6.3 Subnetwork/Network Storage

A subprogram for storage of ECAP and POTTLE models must be developed. It is not presently known to what degree the SCEPTRE model storage subprogram can provide this capability to ECAP.

### 2.6.4 Error Checks

Error checking subroutines in ECAP must be interfaced with graphics output. Similar subroutines must be developed for SCEPTRE for graphics output.

### 2.6.5 Input Functions

Sine, square, and pulse input function generator routines must be constructed and interfaced with graphics input and output.

## 2.7 USER GRAPHIC PROCEDURES

Procedures which the electrical network analysis graphic system user will follow are described in the following paragraphs.

### 2.7.1 Characteristics of the Electrical Network Analysis Graphic System

The graphic system consists of a general purpose computer, cathode-ray tube (CRT) with attached light pen and keyboard, and associated computer programs. Upon initialization, the Electrical Network Analysis Graphic System will define and provide a control surface and working surface on the display surface of the CRT. Templates of circuit components will be constructed on a portion of the working surface, each component

consisting of a body and one or more leads. The components include (see Fig. 2-1) a ground (having one lead), a resistor, an inductor, a capacitor, and diodes (all having two leads), a transistor (having three leads), and a generalized n-terminal (n leads). The n-terminal component represents a subnetwork that the user defines either through card input or light register. It may be used in the problem at hand and/or stored under a unique label for future use.

A schematic storage capability exists. The schematic can be stored during any phase in its construction and retrieved for modification and/or continuation. Typical templates are shown in Fig. 2-1.

### 2.7.2 Schematic Construction

Basic Method. Basically, a circuit schematic is constructed by selecting one or more of the available templates, locating these on the working surface, and interconnecting them with conductor lines (see Fig. 2-2).

Guide Lines. Upon completion of systems initialization, the user may proceed to construct his schematic. As an aid in schematic construction, he can establish parameters within the working area by drawing vertical and horizontal guide lines.

Component Selection. The location and positioning of a component must be determined prior to selecting it. The location points are picked using the light pen. The points may be on guide lines, connecting lines other component leads, or on any void area on the working surface. The rotation of the component is accomplished by the light pen picking the desired rotation angle (90, 180, or 270 deg). The operator selects the desired component by picking the appropriately designated light button which activates the component select function.

Connections. Conductor connections between preselected component leads or points are accomplished by picking desired connection points and activating the connect function:





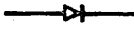
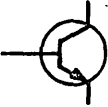
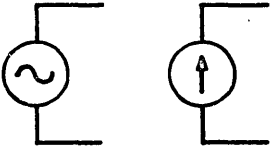
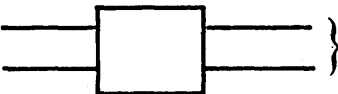
TYPE	SYMBOL	ANNOTATION
RESISTOR		R
BATTERY		E
CAPACITOR		C E(O+)
INDUCTOR		L I(O+)
DIODE		1N XXXX
TRANSISTOR		2N XXXX
GENERATOR		TYPE FREQUENCY DUTY CYCLE PEAK AMPLITUDE FREQUENCY SWEEP
BLACK BOX		USER DEFINED

Fig. 2-1 Typical Component Templates



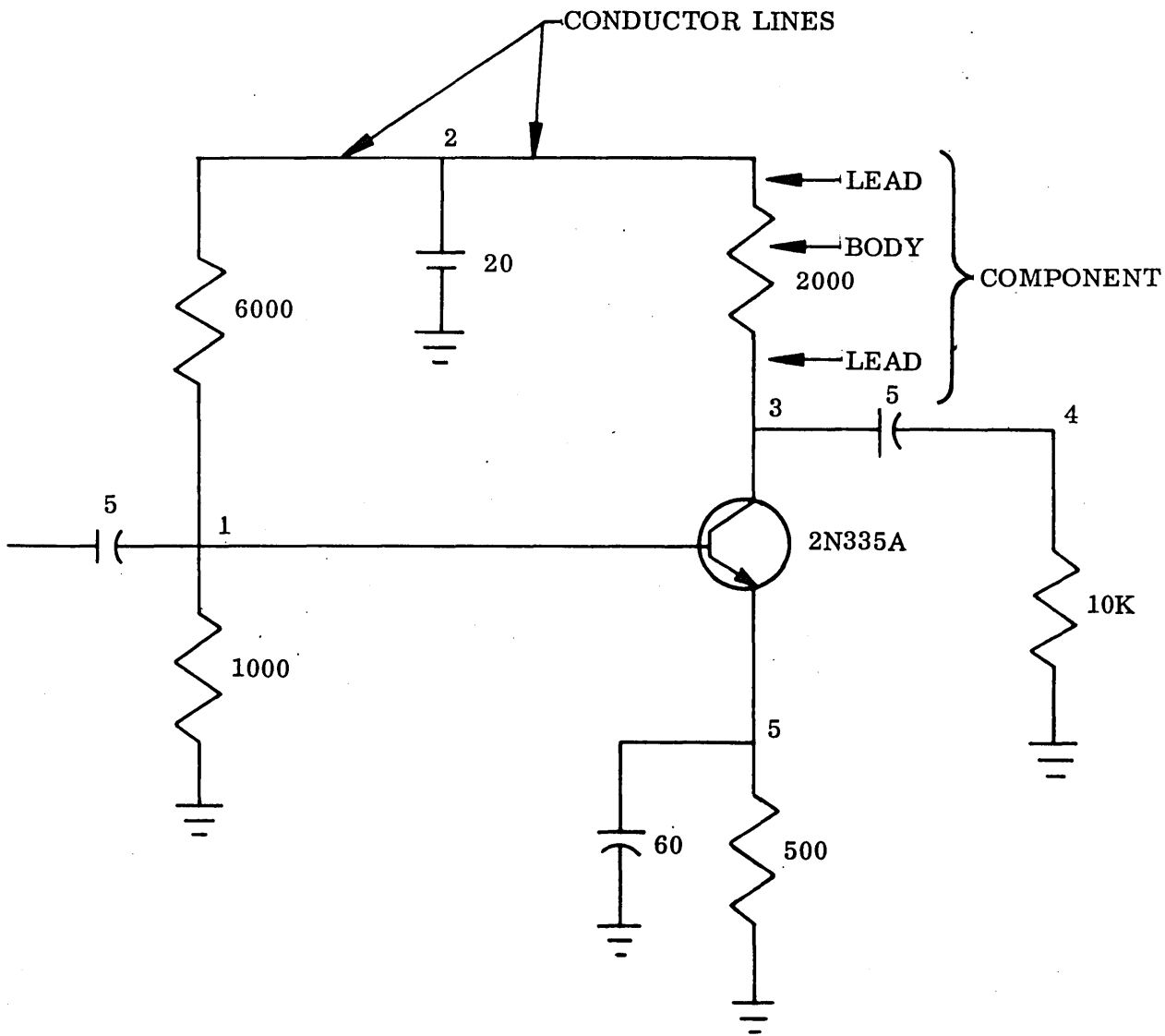


Fig. 2-2 Completed Schematic

- **Point/Point Connect:** By picking two points on the working surface and activating the connect function, an automatic connection will occur. The picked points may be located on conductor lines, component leads, void areas. A point pick on any component lead will be reset to the end of the lead opposite the component body. Additionally, picks on conductor lines other than endpoints will cause the conductor lines to be segmented (and introduce a connectivity circle at the intersection point on the wire).
- **Offset Connection:** Only horizontal and vertical conductor lines are generated. Therefore, if the point picks are offset, a vertical line is drawn from the first picked point to the horizontal plane of the second picked point and extended horizontally to the second point.
- **Point/Line Connect:** The connection of a point and a picked position on a line is accomplished by a normal being drawn from the point to the line. The intersection of the normal and the line will be segmented.
- **Line/Line:** The picking of two lines which are normal to each other will cause both to be segmented and a connectivity circle will be created at the intersection.

Erase. Components, conductor lines, and guide lines can be selectively erased by the light pen picking a point on the desired entity and activating the erase function. The templates are protected from the erase function.

Zooming. This function provides for the magnification of schematic presentation and is activated by requesting the ZOOM function. This magnification can be increased or decreased in integral increments of powers of two.

Duplicate Function. This function allows the operator to produce a duplicate copy of a collection of conductor lines and components at another position on the work area. The duplicate copy has the same orientation as the original. The operator may make as many copies as needed.

Annotation. The circuit schematic must be annotated. Annotation includes:

- Parameter data
- Command data
- Node numbers
- Wave forms

Node numbers are assigned automatically, a node number appearing by each component lead. Parameter data and command data are entered by the user employing general purpose light registers in conjunction with a label function and displayable alphanumeric font. Parameter data for a resistor would be a numeric value denoting its resistance, whereas parameter data for a generator would include the type, frequency, duty cycle, etc. Command data includes:

- Transient time interval
- Time step selection
- Frequency values

Wave forms are also input by the user. A minimum and a maximum value are input, both for the X- and Y-axes. The independent and dependent variable(s) are also entered. Grids for these are then generated and displayed. The user picks points on the grid(s) which are then connected by a best curve fit.

List. When the user is satisfied with a schematic, he calls on the LIST function. This function verifies that parameter data exists for each component and informs the user of unconnected component leads, dangling conductor lines, etc. If the annotation is complete, the nodes are numbered automatically. If the user desires certain node numbers to remain fixed, a manual override capability exists. The user picks those nodes to remain fixed and the other nodes are renumbered accordingly. Input for the analysis routines is then extracted from the schematic and stored for later use. A completed schematic is shown in Fig. 2-2.

### 2.7.3 Analysis/Graphic Displays of Output

Analysis. After making any necessary corrections to the schematic, the user calls on an analysis routine by picking a light button for the appropriate program. The user then selects from the types of analyses; AC, DC, or transient. A log of activities is output on the printer, as well as the topology of the schematic. Printer output of final results is optional.

Graphic Displays. Graphic displays of output are provided, both in table form and in plot form. Table displays are given by selecting the table function and the type of data desired, such as node voltages, transfer functions, plot data, and branch currents. Linear or semi-logarithmic plots are also available. Prior to performing the analysis, plot variables are selected from a list of variables. The schematic and the selected plots are displayed at the same time. Plots are generated in real time and plot points appear in a random order as the analysis proceeds. The user thus sees the trend and may terminate analysis before completion if results appear unsatisfactory. Plots may also be selected after analysis is completed in the same manner. In this case, plots are generated immediately in their totality.

Hard Copy Plots. The same displays mentioned previously, including the circuit schematic, may also be obtained as hard copy output from the SC 4020 or Gerber plotters.

Review. The user interprets the results at hand. He may then return to any point above and continue in the design phase. For example, he could redesign part of the schematic, modify the input data, or call a different analysis routine.

## 2.8 NETWORK ANALYSIS GRAPHICS APPLICATIONS AT LOCKHEED

Graphics status at two Lockheed facilities is described in the following paragraphs.

### 2.8.1 Status of Graphic Electrical Network Analysis at Calac

The graphic electrical network analysis program at Calac is being run on an IBM 360 computer with two 2250 display consoles. It has been demonstrated to several potential users. A new version of this program is being developed and is scheduled for release by November 2, 1967. The approximate cost per scope is \$37.00 per hour.

### 2.8.2 Status of Graphic Electrical Network Analysis at Gelac

Gelac does not have a Graphic Electrical Network Analysis program. Their future plans are to participate with CAD/LMSC for program application and development.

## 2.9 NETWORK ANALYSIS GRAPHICS APPLICATIONS AT OTHER COMPANIES

MIT is using an on-line capability to implement AEDNET for circuit design modification. Circuit diagrams are fed into the computer with a light pen and oscilloscope, and circuit parameter data entered with a typewriter. The operator views the oscilloscope and, using the light pen, locates and moves circuit elements. Component parameter values are assigned using the typewriter. The output displayed is selected by the operator, and eliminates all unwanted information.

Bell Telephone Laboratories is using a console system for problems requiring "dynamic scratchpad" capabilities. Included in such problems have been:

- Printed circuit component and wiring placement
- Schematic circuit design
- Block and flow diagram design
- Text composing and editing
- Placement of cards on a chassis to achieve connecting wire length

IBM uses a small digital computer connected to a large-screen buffered display equipped with a light pen to aid in the design of mask artwork for hybrid integrated circuit modules. The operator uses the light pen to construct a circuit schematic

on the display and the schematic is stored in the computer memory. The operator then uses the light pen to generate the artwork for fabrication of the circuit mask. While the artwork is being generated, it is automatically checked against the stored schematic. When the artwork is complete, the computer is used to drive a plotter that generates the final mask artwork.

Norden Division of United Aircraft Corporation utilizes CADIC (Computer Aided Design of Integrated Circuits) to produce mask artwork. Circuit definition is entered by the use of punched cards. These input data are then translated into size and shape of every component, recalled from computer memory. From this input a layout is displayed on the CRT. The operator then uses the light pen to manipulate the components and make desired changes immediately. CADIC displays the initial layout, but the detailed mask geometries can be called up by the operator by means of the keyboard. Hard copy of mask geometries or detailed layout can be obtained, but a direct interface with mask-plotting equipment is not presently utilized. Future work will include this capability.

## 2.10 PROGRAMMING REQUIREMENTS

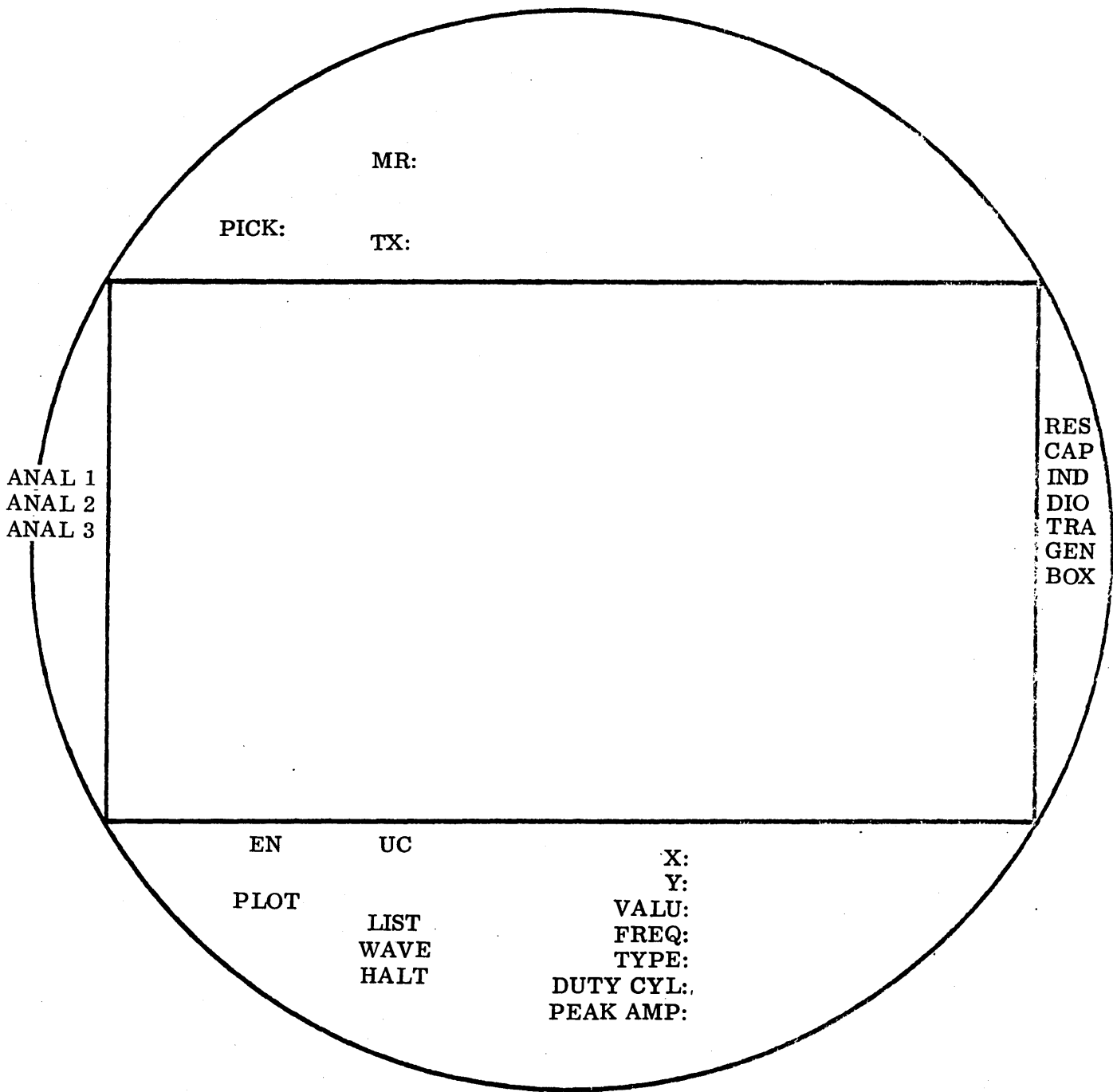
### 2.10.1 System Ground Rules

System ground rules are as follows:

- a. At completion of each program (function) a message indicating completion appears in the message (MR) register located on the control surface. Requests to the operator during execution of functions are displayed in the MR or text (TX) registers (see Fig. 2-3).
- b. The operator will be locked out (cannot interrupt function execution) until a message appears in the MR or TX registers with one exception - during analysis function the operator has the option of stopping execution.

### 2.10.2 System Programs

Initialize. This program is called by picking the primary light button (LB) UC and its secondary LB AP. First, the program searches the computer graphic list for the



NOTE: WORKING SURFACE IS 11 BY 17 IN. ON 22-IN. CRT

Fig. 2-3 Proposed Control Surface Layout

frame entity. If this is not found the computer graphic list (CGL) is assumed to be empty. The frame and control surface (CS) controls are created, the keyboard keys are defined, the template library is entered in the CGL from card images on disc, and the model library is input on cards and written on the disc. If the frame was found, initialization was performed previously.

MR: ENA SYSTEM READY

Component Select. The user picks the LB acronym for the desired component on the right-hand side of the CS (see Fig. 2-3). The program examines the pick table for destination picks and rotation picks. (If rotation is desired, the rotation pick must directly follow the destination pick.) Translations are made using the top right-hand lead of the template as it appears in the library as the origin point. As translation occurs, the category fields are changed to differentiate between template and component and the component is entered in the CGL.

MR: END COMPONENT SELECT

Connect. This program is called by pushing keyboard key 5 (see Fig. 2-4). The pick table is examined and picks are taken in pairs for connection purposes. Depending on the types of picks in each pair, one of three connections is made: point-to-point, point-to-line, or line-to-line. The connection lines are entered in the CGL.

MR: END CONNECT

Erase. The user makes picks on those items to be erased and then depresses the erase keyboard key. The entities selected for erasure are removed from the CGL. If a pick is on any part of a component, the entire component is erased. Only templates and the application frame are protected from erasure.

MR: END OF ERASE

Zoom. The primary LB FM (frame manipulate) is picked. Three secondaries appear: MF (move frame), DP (double viewed area), and HP (half viewed area). The secondaries can be picked singly or in combinations. The operator pushes the accept key after picking secondaries. If the MF is picked, the program requires a point pick in the pick table and the frame is centered about that pick.



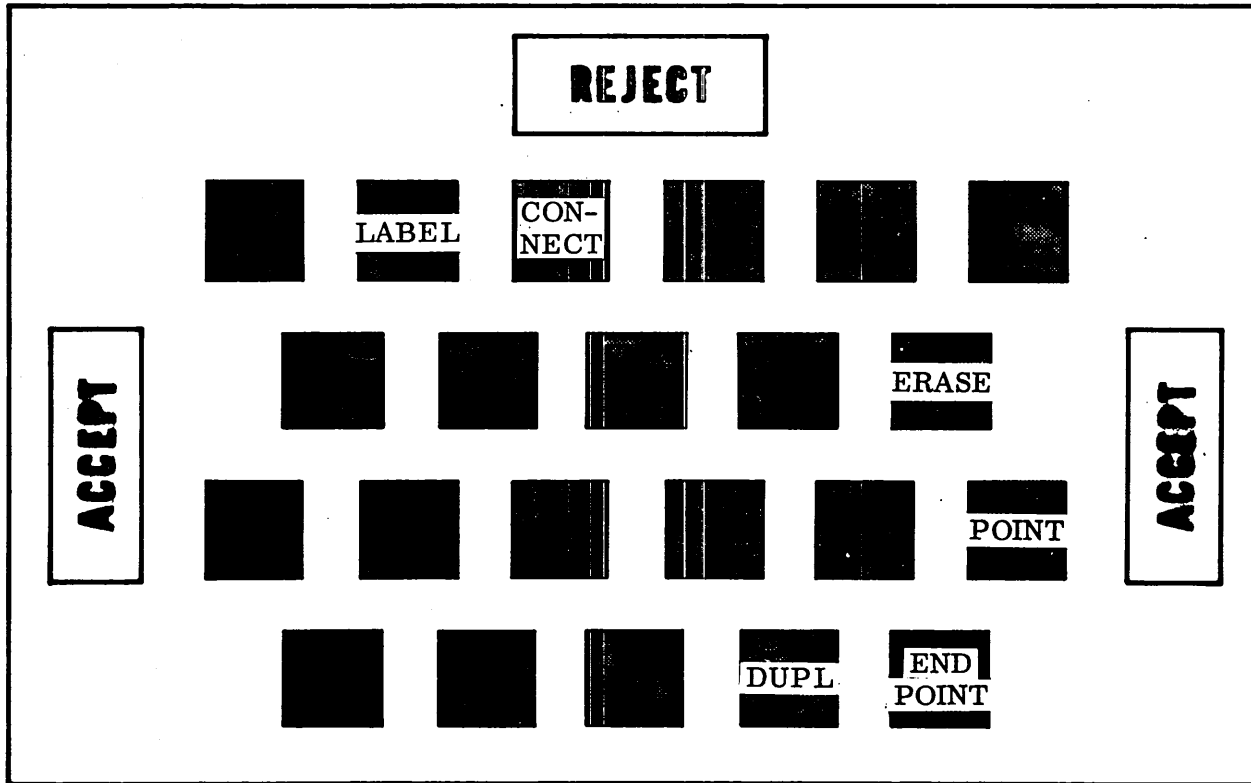


Fig. 2-4 Proposed Keyboard Layout

Duplicate. This function is called by pressing the DUPL key. The program interrogates the pick table for items picked. These items are the ones to be duplicated. The user is then requested to make an origin pick on a component lead (this pick will be translated to that terminal pad) and then will ask for a destination point for that terminal. After the copy is made (entered in the CGL), the user terminates the function by pressing REJECT, or makes another copy by pushing ACCEPT.

MR: END OF DUPLICATE

Label. This program is called by keyboard action. It examines the pick table for a component pick and register picks. The value is extracted from the picked register(s) and the component pick is converted to that component's annotation group. The value is entered in this group and displayed beside the component, the position of the annotation being determined by the orientation of the circuit component.

Wave forms are input by selecting the wave LB. The user is requested to enter the variable names in some register and the minimum/maximum values for the X and Y axes in the X and Y registers. This information is extracted by the wave program and a corresponding grid is constructed and displayed. The user is now requested to pick points on the resulting grid and to push ACCEPT. The program connects the points with a best curve fit and extracts sufficient data points on the curve for future use.

MR: LABELING COMPLETE

List. The primary LB LIST and its secondary LIST are picked to call the LIST function. This program searches the CGL for components and checks these for proper and necessary annotation. The user is informed of any problems via the MR or TX registers or printer output. The connectivity of components is then traced and the operator is informed, as above, of dangling wires, unconnected leads, etc. Any missing annotation or connectivity problem results in termination of this function and control is returned to the operator for corrections. If no problems arise, node numbering can begin. The operator is requested to pick those nodes he wishes to remain fixed and

push ACCEPT. Picks, if any, are examined and the node numbering is accomplished accordingly. The topology of the schematic is stored for use by analysis routines. The connectivity algorithm follows.

MR: END LIST

Connectivity Algorithm. Definition: A pad is the end of a lead opposite the component body. The following steps are performed by the list function:

1. Search the CGL for circuit components. Check each for necessary annotation. If a component is properly annotated, continue the search. If not annotated properly, assign a unique name to that component; for example, R1 for the first resistor encountered which is not properly annotated. This name is displayed to the component and the operator is told that the component of that name needs annotation.

At the finish of the CGL search execute Step 2 if all components have the necessary annotation. If not, return control to the operator so that he may add the required annotation to the schematic. The operator makes changes or additions to the schematic and selects the LIST function again. On subsequent passes through Step 1 names entered during previous searches are removed from the display.

2. Search the CGL for component leads. For each lead encountered the following data are added to a table:
  - X, Y coordinates for the lead's pad
  - A count value
  - A group number

where the count is, with one exception, 0 and the group number is, with one exception, incremented by 1, beginning with 1, for each lead found in the CGL. If the X, Y coordinates of a pad match the X, Y coordinates of a pad already in the table, then the count of both is set to 1 and the new pad entry is given the same group number as its match.

When the entire CGL has been searched, continue to Step 3.

3. Search the CGL for conductor lines. Let the endpoints of a conductor line be  $(X_0, Y_0)$   $(X_1, Y_1)$ .

Is the pair  $(X_0, Y_0)$  in the table described above? If the answer is yes, retain the group number,  $N$ , of the match, set the count of the match equal to 1, and determine if  $(X_0, Y_1)$  is in the table. If  $(X_1, Y_1)$  is not in the table, add the coordinate  $X_1, Y_1$  to the table with count 0, and group number  $N$ . If  $(X_1, Y_1)$  also matches some entry in the table, retain its group number  $M$ , set all group numbers equal to  $M$  or  $N$  equal to the smaller of  $M$  and  $N$ , and set the counts of those entries equal to 1.

If  $(X_0, Y_0)$  is not in the table, check if  $(X_1, Y_1)$  has a match in the table. If  $(X_1, Y_1)$  is also not in the table, enter  $(X_0, Y_0)$  and  $(X_1, Y_1)$  in the table with counts 0 and group numbers  $J + 1$ , where  $J$  was the last assigned group number. If  $(X_1, Y_1)$  is in the table, retain the group number of its match,  $N$ , set the count of its match equal to 1, and enter  $(X_0, Y_0)$  in the table with group number  $N$ .

When all conductor lines have been found and interpreted the table is complete and its contents are examined. If any pad entry has a count equal to 0 it is an unconnected lead; if any conductor line end point has a count equal to 0, the conductor line is dangling and, in either case, the operator is notified of the problem. The group number of each pad with non-zero count, with one exception, will be identical with the group number of one or more additional entries in the table; these pads (leads) being connected. A lead may be connected to a dangling conductor line, and therefore have a non-zero count, but no pad with matching group number in the table. The analysis and the technology of the network is node to node. Using the above table, nodes are now assigned sequentially, with the ground always node zero. Leads with the same group number are of the same node and are numbered as such. If leads A and B are of the same group number and the A pad coordinates and the B pad coordinates are identical, then only one node number appears on the schematic; whereas, if the pad coordinates are not equal, the node number appears alongside each lead.

Analysis Driver. Select an analysis LB on the left side of the control surface and then pick one of the secondaries which appear, denoting the types of analysis available (AC, DC, transient, etc.). This calls the analysis driver program. The analysis options selected and the schematic topology are output on the printer. The selected analysis routine is then called and execution continues. The operator may halt analysis at any time by picking the HALT LB. Final output of analysis is optional. Analysis results are written on disc.

MR: ANALYSIS COMPLETE

Graphic Displays. To call a table display, select the PLOT LB and the TABLE secondary LB. The program displays a list of available tables on the working surface and requests the user to pick the table(s) desired. The data are retrieved from the disc and displayed in table form.

MR: TABLE COMPLETE

Plots generated during analysis are in real time. Before calling the analysis routine the operator selects plot variables from a list of those available. The analysis driver constructs grids for plots if real time plot option was indicated. Points are extracted as analysis proceeds and appear as X's on the respective grids.

Hard Copy. Hard copy plots are available by picking the PLOT LB and either secondary 4020 (SC 4020) or Gerber. In either case the operator is requested to hang magnetic tape and put the identification card in the read hopper and push ACCEPT. The program searches the CGL for all displayed items, performs and outputs plot data only for graphic entities currently displayed on the 11- by 17-in. display area. Tape is then submitted to the appropriate plotter to obtain the hard copy.

MR: END HARD COPY

### 2.10.3 Component Group Structure

The component group structure contains specific subgroups: the body subgroup, the annotation subgroup, and one or more lead subgroups. Each subgroup is made up of

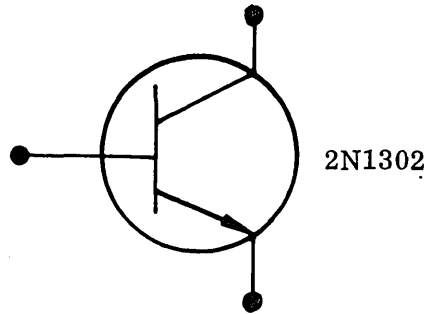
one or more graphic or alphanumeric entities grouped together by a linkage entity. For instance, a lead subgroup consists of a line entity and a dot entity whereas an annotation subgroup consists of one or more alphanumeric entities.

A capability of identifying the type of component represented has been established. The reference word of a component or template top group contains two 12-bit binary fields: (23-13), not defined and (12-0), unique element type no., which is the same for all interchangeable symbols.

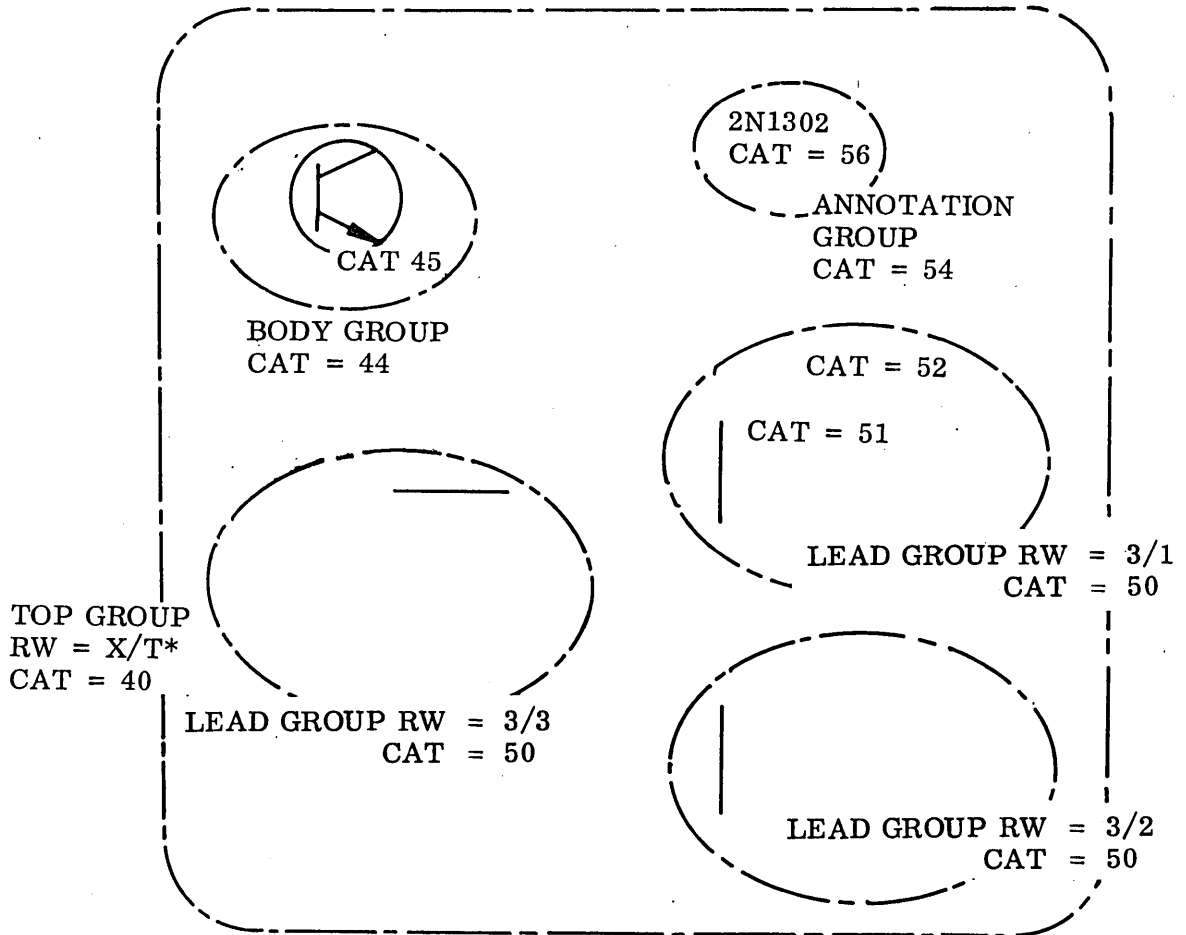
The reference word of the top group of a lead contains two 12-bit binary fields also: (23-13), number of leads on the component and (12-0), lead series number.

The component group structure is depicted in Fig. 2-5 as both a schematic structure and a group structure. The schematic symbol is shown with its value designation. The group structure categories are given in octal (RW, reference word; CAT, category).

TRANSISTOR



GROUP STRUCTURE



\*WHERE T IS THE CODE FOR THE TRANSISTOR

Fig. 2-5 Sample Component Group Structure

Section 3  
FLIGHT MECHANICS

3.1 MODE OF APPLICATION OF GRAPHICS

Computer-graphics can be an extremely helpful computing tool and adjunct. To get the most from it the user must capitalize on its pictorial and interactive aspects. He must incorporate these special aspects into his problem-solving plans at an early stage in order to integrate the graphics capabilities and controls into his computational solution and to bring graphics to bear as effectively as possible on this solution. The following paragraphs discuss ways to most profitably employ graphics, in solving analytical flight mechanics problems, and cautions against certain pitfalls based on this task group's experience with graphics programs.

3.1.1 Recommended Modes

First the operating mode should be kept interactive, that is, it should provide for communication between the computer and the user during the run; he should attempt improvements and note their effects. Furthermore, the user should depend upon graphical indicators from the console scope to show the progress of the calculations, to point out error sources, and to indicate reactions to changes made during the run. If the system is not pictorial and interactive, preparation of a graphics program is probably not required or justified; results could just as well be obtained with a QUICKTRAN or TYMSHARE terminal or a batch-processed CRT output device, possibly at less expense.

Full use should be made of the user's sensory powers, deductive powers, and memory. When the results of a series of events are properly presented to him, he can interpret and effectively monitor their progress and direction. His abilities to draw conclusions and act in a nonlinear manner should be heavily relied upon; linearized computations and decisions should be left to the computer.



The heart of every problem should be a good working picture. It should show the principal working variables, or at least it should clearly present the results of changes. From the information and control options on the console display, the user should be able to control all the significant variables. Just what these displays are, or their best choice, may have to be learned from experience with the graphics system itself, with the system being improved in later modifications. Conceivably, the displays may have to be changed to suit a new and different application.

All displays, messages, and controls put before the user should appear in his language, not that of the computer or the program. General computer or console software functions should not be his concern. All indicators and commands should be phrased in physical and computational language peculiar to his problem so that he can quickly and clearly understand and react to them.

All commands the analyst gives should require only a single action for implementation. His action should set off a series of automatically sequenced, logical and computational steps within the computer-graphics system. His instructions are, in essence, "macro-instructions" because they trigger a large number of other machine-oriented instructions. Single-instruction commands keep him free from routine, distracting decisions.

For each action he takes, the system should display a positive indication that it recognizes and accepts it. For example, the response to selecting a command instruction or to requesting a change in a displayed variable might be an immediate underlining of that display. The response to typing in a new value of a variable might be the display of the numerical data in place of the old.

It is certainly desirable, as suggested in Ref. 3-1, to give the user control over all input variables and system parameters. In this way he is able to perturb them to compute the sensitivities of his solutions, to changes in them, and to possible tradeoffs among them.

The displays should come fast enough to insure that the user can remember them and sense changes. It is even wise to keep the displays driving the user and forcing him

to make comparisons and decisions in order to stay abreast of the progress of the calculations. Pauses for evaluation should be allowed only as absolutely required by the complexity of the final product. Rapid operations will require careful use of data storage devices and careful planning of the transmittal of large blocks of data, and may ultimately call for increased hardware capability for storage and data transmission devices.

Many of the misuses of graphics can be easily inferred by taking the opposite approach from the preceding recommendations. One of the commonest classes of pitfalls is relying too heavily on generalized computer programs and display software. The results are generally exceedingly slow-acting programs and overly cumbersome graphics systems. For example, in developing the PRESTO trajectory optimization program (to be described in Para. 3.4), the Flight Mechanics Task Group saved an order of magnitude in computing time by discarding the general-purpose, variable-step-size integration subroutine commonly in use and developing one specifically tailored to powered-flight problems. The hypersonic aerodynamic formula (FORMAC) computer-graphics system developed earlier by LMSC pointed out the problems associated with multiple-command, sequential, general-purpose, computer-oriented display software. The commands were in some cases so cumbersome that, at best, the user was considerably delayed in working out a problem and, at worst, he could lose his train of thought and purpose in conducting a particular investigation. Many routine data inputs and display changes required him to make a repetitive string of choices and decisions and, consequently, bogged him down in unnecessary detail.

In summary, the features of a good computer-graphics system are that it is particularly keyed to graphical display and interaction, that it is fully controllable by the user, and that it is fast enough and its displays are effective enough to utilize his deductive abilities to a significant degree.

## 3.2 GRAPHICAL-COMPUTER SYSTEM FOR ORBITAL TRANSFER STUDIES

Two specific applications of interactive graphics techniques have been studied in depth during the current study. The first deals with performance calculations for orbital transfers and is discussed in Para's. 3.2 and 3.3; the second is concerned with ascent boost trajectory optimization and is covered in Para's. 3.4 and 3.5. The orbital transfer graphical-computer system itself will be described in Para. 3.2 and a requirements specification, written from the viewpoint not of the programmer but of the user-analyst, will be presented in Para. 3.3.

### 3.2.1 Objectives

The orbital transfer system could be applied to both interplanetary and planetocentric intersatellite transfers. A multitude of direct and flyby interplanetary transfers have already been computed, tabulated, and graphically presented in various Parts of the Planetary Flight Handbook prepared by LMSC for trips to and from Mars and Venus and to Jupiter and the asteroids Ceres and Vesta (Refs. 3-2 and 3-3). Part 6, prepared by the Douglas Aircraft Company (currently being printed), includes Venus swingby missions to and from Mars (Ref. 3-4). All these interplanetary compilations can be kept to a manageable size because the orbital elements of the terminal bodies are definitely known and finite times of interest can be postulated a priori. Extending these handbooks to other bodies of the solar system greatly enlarges the scope of the work; the bounds on the quantity of material could grow "astronomically." Furthermore, preparing similar reports on general interorbital transfers would result in an unmanageable volume of material, most of which would likely turn out to be useless or of value to only a limited number of analysts.

What does appear to be most useful would be a means of readily extending the interplanetary data or the interorbital transfer data to bodies and time periods as needed by mission and system analysts. In order to develop those means, the following objectives were drawn up for the orbital transfer graphics system:

- Develop a system to generate mission contour plots and data as needed, that is, create a "working handbook."

- o Store and recall to view all the contours generated during a computer run.
- o Permanently record the contours for publication and future use.

Specifically, the first objective must encompass the following:

- o Accept orbital elements of any two bodies moving about a specific central body.
- o Enable the user to detect and correct errors in the input data.
- o Compute mission velocity requirements and trajectory geometric data for rendezvous or intercepting transfers between the two bodies, and calculate these data in useful coordinate systems.
- o Compute and plot contours of specified mission/geometric data.
- o Adjust the ranges of the independent variables and the contour levels to show the most interesting regions.

### 3.2.2 Computer Program Description

At this point it is appropriate to describe a computer program already in existence that can perform several of the above objectives in a batch processing, non-interactive mode. This program (MACON) was derived from the Medium Accuracy Orbital Transfer (MAOT) program already in wide use in NASA (Ref. 3-2) and includes contour plotting features executable on the SC-4020 cathode-ray tube plotter. Although versions of MAOT have been developed which treat planetocentric transfers, and a similar graphics system could, in a straightforward way, be developed for them, the current discussion will, without loss in generality, be restricted to interplanetary travel. The interplanetary program could be useful in studying missions among the planets, the asteroids, comets, and hypothetical solar system spacecraft and space stations.

Inputs to MACON consist of planetary constants (such as orbital elements, obliquity, and location of the vernal equinox), designations of departure and arrival planets, departure and arrival date ranges and incremental step sizes, contour specifications (such as line legends, names of variables to be contoured, number of contour levels, increments between them, minimum levels, or a list of their values) and plotting

specifications (such as x- and y-axis labels, minimum trip time, and grid-squaring option indicator). For each pair of departure and arrival dates within the specified ranges, the heliocentric conic trajectory is computed which connects the two terminal planets on these dates. The method used to find this trajectory employs a numerical iteration of Lambert's theorem. The characteristics of this trajectory which are calculated include the semi-major axis and eccentricity, inclinations relative to the terminal planets' orbital planes, the true anomalies of the terminal points, and their heliocentric speeds and directions; of primary importance to the mission analyst, the magnitudes and planetocentric directions of the hyperbolic excess velocity of the departure and arrival hyperbolas at the two terminal planets are also figured.

The subset of all these computed values which the analyst wishes to contour are stored as functions of the two terminal dates. At the option of the user up to four variables may be selected for contouring and up to fifty steps may be taken along each data axis; these maxima are set by the amount of core storage available in the computer. Thus, the analyst can control the spacing between contours and the fineness of the date grid as well as the number of variables to be plotted and the number to appear on each plot. The contour subroutines examine one stored variable at a time and linearly interpolate in each grid square of the departure date-arrival date plane for the contour line segments. Each contouring level is marked with a unique alphanumeric code, and a table of the contour level associated with each code is printed alongside the contour plot. If more than one variable is plotted on the same date plane, a different type of line is assigned to each one and a different table of contour level codes is printed. Finally, all the contour plots and decoding tables are transmitted to the SC-4020 plotter and permanently recorded on microfilm or full-size photographic prints.

### 3.2.3 Interactive Aspects

To improve the turnaround time of a MACON-aided study, the objectives listed in Para. 3.2.1 for a graphics system were drawn up. To accomplish these objectives, graphics interactive features must be added to MACON. The following cycle of operations has been devised to describe a typical graphics study:

1. Input data are prepared. The analyst must estimate the likely range of interesting dates and limiting values of the contours. He may wish to specify limiting constraints on velocities, radii, declination angles, etc., the number of such constraints being dependent on the capacity and coding features of the particular graphics-computer system being used. He should be prepared to check the computer run for input or execution errors and to jot down frame numbers for recall during the run.
2. The computer program is read in and initialized. The graphics scope background is initialized and displayed; this background material consists of data descriptors, control words, graph axes, and a list of the variables which are available for contouring (see Fig. 3-1).
3. The data for the first case are read in. Certain quantities are displayed on the scope for monitoring by the analyst. He may correct errors by means of his light pen.
4. The mission data and contour lines are computed.
5. The contour lines and all the data specifying the number and spacing of them are displayed.
6. There is a pause in the computer's operation while the analyst monitors the display; he may then choose to:
  - a. Correct an error and recycle the operation to 4.
  - b. Change the axis limits or contour specifications and recycle to 4.
  - c. Store the present display for future recall and recycle to 6. The store operation saves the present display in a large capacity memory device and causes a permanent record to be made of it on a peripheral CRT plotter.
7. The next case is read in and the cycle returns to 3.
8. The analyst may recall one or more of the previously stored displays in sequence for a rapid overview, evaluation, and comparison of the entire run. This step is like flipping the pages of a handbook to obtain a rapid comparison of mission opportunities; detailed study of the plots should be postponed

until the permanent copies are available. However, this step may indicate that more detailed or extended calculations in certain areas are needed during the current run.

What the analyst sees to guide and control the run is the scope display shown in Fig. 3-1. This figure is a typical illustration of the layout and contents of the scope. The heavy border encloses a "working area" that is 11- x 17-in. in size. Around the working area is the "control surface area." The printed alphanumeric information composes the unchanging background display; fields for the variable data which will be displayed during operation of the system are indicated by rows of x's preceded, as appropriate, by a space for a + or a - sign. The date plane for the contour plot occupies the left side of the working area; the contour, axis, grid and constraint specifications are on the right. Phase indicators, control commands, and a numerical keyboard appear in the control surface area below the working area. The "light buttons" (that is, regions of the scope face that are sensitive to light pen signals) are marked by individual dashed rectangles and circles on this figure.

A typical sequence of displays occurring during a cycle of operations might be as follows, the numbered steps being those previously defined:

1. No display.
2. The background material is displayed.
3. The DATA indicator, which appears in the control surface area, is underlined. The axis labels and the limiting departure and arrival dates are printed along the axes of the contour plot. The title is printed at the top of the working area. Under Contour Specifications will appear the line legends (LEGD) and scaling factors (SCALE) for the variables (PARAM) to be contoured. This information will show whether the proper variables have been selected. Parameters recommended for use with MACON are indicated in this specification table; the symbols in this table refer to the following variables:

VH1	Departure hyperbolic excess speed
VH2	Arrival hyperbolic excess speed





DV1	Impulsive velocity increment required to transfer from a specified circular orbit about the departure planet to a coplanar tangential hyperbola having an excess speed VH1
DV2	Impulsive velocity increment required to decelerate from a hyperbola approaching the arrival planet and having excess speed VH2 to a specified speed (e.g., orbital or entry) at a specified altitude in a tangential maneuver
DV1 + 2	Sum of DV1 and DV2
DECL1	Declination of the departure hyperbolic asymptote
DECL2	Declination of the arrival hyperbolic asymptote

In Fig. 3-1, the line legends appearing alongside the parameters VH1 and VH2 indicate that these two parameters have been selected for contouring. Under the Axis and Grid Specifications, the axis increments, number of increments, and grid-line spacing for the contour graph will be displayed. Finally, the constraints selected by the analyst in step (1) may be compared with those read into the computer.



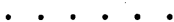

4. The underline of DATA will disappear. Next, the COMPUTE and then the CONTOUR indicators will be underlined, one at a time. These indicators inform the analyst of the progress of the nearly one-minute-long calculations occurring during this step.
5. The underline of CONTOUR will disappear. The minimum (MIN) and the maximum (MAX) values of the scaled contour levels and the number of contour levels (NC) will be displayed for each variable being contoured. The contours will be plotted and the scaled value of each printed at a regular frequency on each contour line.
6. The question mark to the right of the NEXT indicator will appear to signal the analyst that the system awaits his light-pen command. The discussion of his actions and subsequent display reactions will be postponed for a few paragraphs. In general, each command will elicit a positive response and a display of the changed quantities (if any) from the graphics system. Each new picture will be assigned a unique sequential frame number for identification.

7. The cycle of displays is repeated.
8. One or a series of displays will be recalled from storage and redisplayed. Each will include axis labels and values, the contour plot, contour level labels, title, frame number, and all the data and legends in the Contour Specification.

Program operation and console displays having been described, the final portion of this subsection will be devoted to an explanation of the system controls implemented through the light buttons assigned to the display scope face. This explanation will dwell on the meanings and reasons for the controls rather than their software implementation; the latter aspect will be part of the Requirements Specification (Para. 3.3.3). The controls are triggered by the user touching any light button with his light pen during the run. At the next pause, the system will indicate a positive response to this triggering and will wait for the analyst to enter in any changes by means of the keyboard in the control surface area. He may choose to make several changes, each new one signalling the completion of its predecessor. Repeating a change indicates that values just typed in were not acceptable. Changes are completed and the operating cycle is resumed when the analyst triggers the continuation button NEXT or the repeat button REDO.

In the Contour Specifications section of the display, touching a line legend button under LEGD eliminates the associated parameter from the list of those being contoured. This command allows the user to prepare plots of different contours on succeeding frames or to reassign line legends.

Touching a light button under PARAM adds that parameter to the list of those being contoured and assigns the next available line legend to it. An order should be assigned to the various line legends; the following is suggested to insure easily distinguishable contours:

<u>Order</u>	<u>Line</u>	
1st		(heavy solid line)
2nd		
3rd		
4th		(light solid line)

Of course, computer storage, scope resolution, and/or data transmission rates may limit the number of variables that can be simultaneously stored and contoured. This PARAM command is obviously needed to change to a new set of variables during a run and may be employed to change the line legend assigned to a particular variable in case of a data input error. Failure of the user to supply a scaling factor and contour range and/or incrementing specifications for the newly chosen variable will cause the question mark after CONTOUR SPECIFICATIONS to appear when the next control command button is selected and will prevent the resumption of operation.

Touching a light button under SCALE allows the user to change an existing scale factor or to add a new one.

Touching a light button under MIN or MAX permits the extreme contour levels to be modified from the numerical keyboard. Normally, these extremes are input in the data or computed by one of the contour subroutines from the values of the variable being stored. If the user dislikes the input data or has a reason for distrusting the computed values, he may change them with these commands.

Touching a light button under NC leads to a change in the number of contour levels. The user may wish to do this for a variety of reasons. Perhaps an input error needs to be corrected or a coarse interval between contour levels needs refining. One of the contouring subroutines may have computed too many contours or too fine a step size between levels and, consequently, one end of the range of contours was drastically truncated because storage capabilities were exceeded. In any event, after these light button signals, a change in the number of contours can be typed in.

Under the Axis and Grid Specifications the first two rows of light buttons allow the user to shift the portion of the date plane being calculated and contoured left or right, up or down, or to look at a larger part of it or to zoom in on a smaller area of it. Touching the CHANGE FIRST light button under LEAVE (or ARRIVE) allows the user to increase or decrease the first, that is, the smallest, limiting value of the departure (arrival) date axis by a numerical keyboard input. Similarly, the CHANGE LAST button permits the last, or highest, limiting value to be altered. Grid limits and plotter scaling

factors will be adjusted so that the new portion of the date plane fills the 9-in. -square plotting area on the scope.

Touching the light button to the right of AXIS INCREMENT allows the user to change the intervals between successive values of the departure and/or arrival dates and thereby to refine the results produced by the linearized contour line computations. Furthermore, these intervals should be changed when the date ranges along each axis are varied. Incidentally, the contours plotted in Fig. 3-1 were obtained with an axis increment 1/18 of the date ranges. Associated with this light button is the display of the number of increments (NO. OF INCRMTS) along each axis. These two displays are not light buttons, but they are quite important to the analyst because they show whether the maximum storage capacity for date and contoured variable values is being exceeded. Excesses can be corrected by use of the six previously described light buttons.

Touching the light buttons alongside GRID SPACING enables the analyst to adjust the spacing between background grid lines; the data appearing in these fields are the increments between these lines measured along the two data axes. To eliminate all grid lines, spacings greater than the range of dates along the axes should be input.

Touching any one of the CONSTRAINTS light buttons allows the user to type in new values. This command is needed to correct input errors or to modify initial estimates of regions of interest.

The control commands appearing in the control surface area have the following effects:

- NEXT causes the system to proceed to the following normal operation such as commencing calculations after the input data have been verified, executing a pre-specified series of recalls of previously stored displays, or reading in the data cards for the next case. NEXT is the analyst's signal to the computer graphics system that all changes have been made, all data are acceptable, and a normal continuation is in order.
- REDO is a command to repeat the present case with the changed specifications and constraints. Depending upon what has been changed, this command will be interpreted by the system in a variety of ways.

- o STORE saves the present display in a large capacity memory device for subsequent recall and causes a permanent record to be made of it on a peripheral CRT plotter.
- o RECALL enables the analyst to use the numerical keyboard to specify the frame number of one or a series of previously computed displays for re-display. Each touch of this light button is a signal that another frame number is about to be entered. NEXT triggers the actual recall or series of recalls. The frame number being specified appears to right of RECALL.
- o STOP ends the run.

### 3.3 REQUIREMENTS SPECIFICATION FOR ORBITAL TRANSFER STUDIES

#### 3.3.1 General Guidelines

General guidelines are outlined in the following paragraphs.

3.3.1.1 Objectives. The objectives listed in Para 3.2.1 are the objectives of the Orbital Transfer Graphical-Computer System.

3.3.1.2 Nature of This Specification. This specification sets forth requirements on the performance and capabilities of the system from the viewpoint of the user-analyst. As far as possible it is independent of any particular computer or display equipment.

3.3.1.3 System Operation. The system shall be so designed that its operating displays and instructions shall be expressed in the language of the user in that it shall use the vocabulary of flight mechanics and engineering plotting procedures.

#### 3.3.2 Performance Requirements

Performance requirements are outlined in the following paragraphs.

3.3.2.1 Sequence of Operation. This system shall operate in the following sequence of steps as described in Para 3.2.3:

1. Data preparation by the analyst
2. Program read-in and initialization (E)\*

---

\*Agents for the various steps in the operation of the system are indicated as follows: (E) main computer executive program, (M) MAOT routines, (C) contouring subroutines, (G) graphic display routines, and (L) light-button-directed control program.

3. Data display and monitoring (M, G, L)
4. Computation of the mission variables and contour data (M, C)
5. Display of the contour data (G)
6. Review of the contour data by the analyst
  - 6.1 Error correction and recycle (L)
  - 6.2 Specification change and recycle (L)
  - 6.3 Display storage (L, G)
7. Continuation to next case
8. Recall of stored displays (L)

3.3.2.2 Inputs. Inputs are as follows:

1. Program: The system shall read in from magnetic tape or cards an object deck of the MACON program suitably modified for graphics (E).
2. Data: The system shall read in from magnetic tape or cards the following input data (M):
  - 2.1 Number of cases
  - 2.2 First, last, and increment in departure dates
  - 2.3 First, last, and increment in arrival dates
  - 2.4 Departure and arrival planet numbers
  - 2.5 Mission speeds, which include the following:
    - Orbital speed at departure
    - Escape speed at departure
    - Escape speed at arrival
    - Post-propulsion speed at arrival

These are needed to compute the impulsive velocity increments at departure and arrival which were defined in Para. 3.2.3.

- 2.6 Case title, 72 characters
- 2.7 Departure and arrival axis labels, 36 characters each
- 2.8 Departure and arrival grid line intervals
- 2.9 Parameter number(s) of no more than four\* mission variables to be contoured, the parameters being represented by the following symbols:

<u>Parameter No.</u>	<u>Symbol</u>	<u>Mission Variable</u>
1	VH1	Departure hyperbolic excess speed
2	VH2	Arrival hyperbolic excess speed
3	DV1	Departure velocity increment
4	DV2	Arrival velocity increment
5	DV1+2	Sum of DV1 and DV2
6	DECL1	Declination of departure hyperbolic asymptote
7	DECL2	Declination of arrival hyperbolic asymptote

2.9.1 The order in which a parameter is designated in the input shall determine the line legend used to draw its contours.

- 2.10 Contour specifications for each parameter; the following options being available for each parameter:

<u>Option No.</u>	<u>Specifications</u>
1	Number of contours Minimum contour level Increment between contour levels
2	Number of contours Increment between contour levels**

---

\*The upper limit on the number of variables that may be contoured shall be set by the storage and data transmission speeds of the computer-graphics hardware.

\*\*The contouring subroutines shall determine the minimum level from the computed values of the particular parameter.

<u>Option No.</u>	<u>Specifications</u>
3	Number of contours Resolution of the increments between contour levels*
4	Number of parameter already input whose table of contour levels shall also be used for the current variable.

3.3.2.3 Outputs. Outputs are as follows:

1. Printed output: The system shall compute and tabulate the following mission data in the same format utilized in Ref. 3-2 (a sample listing is shown in Table 3-1) (M):

Explanation of Symbols. The symbols used in Table 3-1, along with their column numbers, are explained in the following:

<u>Column</u>	<u>Symbol</u>	<u>Explanation</u>
1	DEPART	Julian date of departure, reckoned from JD 240-0000.
2	ARRIVE	Julian date of arrival, reckoned from JD 240-0000.
3	SPEED (VHI)**	Magnitude (speed of hyperbolic excess departure velocity vector, normalized to Earth's mean orbital speed.
4	RA	Right ascension of hyperbolic excess departure velocity vector (hyperbola asymptote); measured in degrees along the local planetary equator eastward from the "vernal equinox," i. e. , where the Sun's path moves north across the departure planet's equator. For Venus, since the orientation of the planetary North Pole is unknown, the equator has been

---

\*The contouring subroutines will first find the minimum and maximum computed values of the current parameter and will then assign an increment between contour levels which will be the minimum multiple of the resolution that can span the range between these extremes.

\*\*Parameter names used on the graphics display scope for four of these variables are enclosed in parentheses.



Table 3-1

## SAMPLE OUTPUT OF ORBITAL TRANSFER PROGRAM

DEPARTURE PLANET= EARTH ARRIVAL PLANET= ICARUS														DEPARTURE PLANET= EARTH ARRIVAL PLANET= ICARUS						
DEPART	ARRIVE	SPEED	R A	DECL	I 1	V 1	PSI 1	ECCEN	SMA	THET1	THET2	PERI	APHEL	I 2	V 2	PSI 2	R A	DECL	SPEED	
3-9900	4-	5	.205	74.9	56.8	7.49	.473	84.4	.265	.792	163.8	270.4	---	1.001	19.73	1.208	75.2	298.0	15.5	1.473
3-9900	4-	10	.142	75.2	63.8	5.77	.924	80.2	.170	.853	160.9	271.9	---	.998	20.28	1.120	80.4	299.7	17.3	1.287
3-9900	4-	15	.094	76.2	73.8	4.31	.065	87.6	.091	.913	154.7	269.3	---	.997	20.85	1.054	84.8	300.8	19.1	1.141
3-9900	4-	20	.057	217.1	88.0	3.01	.098	86.6	.030	.971	124.8	242.4	---	1.000	21.42	1.001	88.5	301.5	20.9	1.023
3-9900	4-	25	.035	249.0	49.5	1.84	1.025	89.3	.040	1.026	18.3	138.5	---	---	21.99	.959	91.6	301.8	22.8	.927
3-9900	4-	30	.039	249.0	-1.5	.77	1.048	89.9	.084	1.078	1.8	124.2	---	---	22.54	.923	94.2	302.0	24.6	.847
3-9900	4-	35	.057	247.9	-20.3	-.25	1.067	90.3	.124	1.127	357.6	482.0	.987	---	23.10	.893	96.4	301.9	26.5	.779
3-9900	4-	40	.076	240.4	-33.8	-1.16	1.083	90.6	.158	1.172	355.9	482.1	.987	---	23.64	.866	98.3	301.6	28.4	.722
3-9900	4-	45	.095	244.4	-40.4	-2.07	1.096	90.8	.187	1.213	355.2	483.0	.987	---	24.18	.843	99.9	301.1	30.3	.673
3-9900	4-	50	.113	242.1	-51.8	-2.95	1.107	90.9	.211	1.251	355.0	484.3	.987	---	24.72	.821	101.2	300.5	32.3	.631
3-9900	4-	55	.129	239.5	-50.0	-3.77	1.117	90.9	.233	1.286	355.1	485.7	.987	---	25.25	.802	102.3	299.8	34.2	.595
3-9900	4-	60	.144	230.3	-59.5	-4.57	1.125	90.9	.251	1.317	355.3	487.2	.987	---	25.78	.785	103.2	298.9	36.1	.564
3-9900	4-	65	.158	232.7	-62.5	-5.36	1.132	90.9	.266	1.344	355.7	488.8	.987	---	26.31	.768	104.0	298.0	38.0	.537
3-9900	4-	70	.172	220.5	-65.1	-6.13	1.138	90.8	.279	1.369	356.2	490.4	.987	---	26.85	.753	104.5	296.9	39.8	.514
3-9900	4-	75	.185	223.7	-67.3	-6.89	1.143	90.7	.290	1.390	356.8	492.1	.987	---	27.38	.739	105.0	295.8	41.7	.494
3-9900	4-	80	.198	210.3	-69.2	-7.64	1.147	90.6	.306	1.409	357.5	493.8	.987	---	27.92	.725	105.3	294.5	43.5	.476
3-9905	4-	5	.201	76.1	50.2	7.33	.866	85.0	.272	.785	166.4	267.9	---	.990	20.40	1.203	74.7	298.5	15.8	1.482
3-9905	4-	10	.140	70.6	63.1	5.61	.619	86.6	.174	.849	163.6	269.5	---	.997	20.81	1.117	80.1	300.1	17.6	1.293
3-9905	4-	15	.091	78.4	73.0	4.16	.362	87.7	.094	.911	157.5	267.0	---	.996	21.25	1.052	84.6	301.1	19.4	1.145
3-9905	4-	20	.055	191.5	87.4	2.89	.096	88.6	.032	.970	128.3	240.9	---	1.001	21.71	1.001	88.4	301.6	21.2	1.025
3-9905	4-	25	.034	240.2	40.6	1.70	1.024	89.1	.040	1.026	22.9	138.0	---	---	22.16	.959	91.6	301.9	23.0	.927
3-9905	4-	30	.030	247.1	-1.2	.75	1.048	89.6	.085	1.080	5.6	122.9	---	---	22.62	.924	94.3	301.9	24.8	.846
3-9905	4-	35	.050	240.2	-25.9	-.22	1.067	89.9	.125	1.130	1.2	120.6	---	---	23.07	.894	96.6	301.7	26.6	.778
3-9905	4-	40	.075	244.6	-37.8	-1.11	1.083	90.1	.159	1.176	359.6	480.7	.988	---	23.52	.868	98.5	301.3	28.4	.720
3-9905	4-	45	.095	242.5	-45.0	-1.95	1.097	90.2	.189	1.218	358.9	481.7	.988	---	23.96	.844	100.1	300.7	30.2	.671
3-9905	4-	50	.110	240.1	-50.1	-2.75	1.108	90.2	.213	1.256	358.8	483.0	.988	---	24.41	.823	101.4	300.0	32.1	.628
3-9905	4-	55	.125	237.3	-54.0	-3.51	1.118	90.2	.235	1.291	358.9	484.5	.988	---	24.84	.804	102.6	299.2	33.9	.592
3-9905	4-	60	.140	234.0	-57.2	-4.25	1.126	90.2	.253	1.322	359.2	486.1	.988	---	25.28	.787	103.5	299.2	35.7	.560
3-9905	4-	65	.155	230.4	-60.0	-4.97	1.133	90.1	.269	1.350	359.7	487.7	.988	---	25.72	.770	104.2	297.2	37.4	.533
3-9905	4-	70	.160	220.2	-62.3	-5.67	1.138	89.9	.281	1.375	.3	129.5	---	---	26.15	.755	104.8	296.0	39.2	.510
3-9905	4-	75	.178	221.7	-64.2	-6.35	1.143	89.8	.292	1.396	.9	131.2	---	---	26.59	.741	105.2	294.8	40.9	.489
3-9905	4-	80	.189	210.7	-65.9	-7.02	1.147	89.6	.301	1.415	1.7	132.9	---	---	27.03	.727	105.5	293.4	42.5	.471
3-9910	4-	5	.199	70.5	55.7	7.23	.859	85.8	.278	.779	169.0	265.5	---	.996	21.05	1.199	74.2	299.0	16.1	1.492
3-9910	4-	10	.138	77.0	62.5	5.49	.915	87.1	.179	.844	166.4	267.3	---	.996	21.31	1.115	79.8	300.5	17.9	1.299
3-9910	4-	15	.090	79.7	72.3	4.05	.659	87.9	.097	.908	160.3	264.8	---	.996	21.63	1.050	84.4	301.3	19.6	1.148
3-9910	4-	20	.055	179.0	80.7	2.80	.094	86.6	.033	.969	131.9	239.4	---	1.001	21.97	1.000	88.3	301.8	21.4	1.027
3-9910	4-	25	.033	243.2	47.5	1.70	1.025	89.0	.040	1.027	27.4	137.5	---	---	22.33	.959	91.6	302.0	23.1	.927
3-9910	4-	30	.039	244.7	-1.7	.70	1.047	89.3	.086	1.081	9.4	121.8	---	---	22.69	.924	94.4	301.9	24.9	.845
3-9910	4-	35	.050	243.8	-25.3	-.21	1.067	89.4	.126	1.132	5.1	119.4	---	---	23.05	.895	96.7	301.5	26.6	.776

<u>Column</u>	<u>Symbol</u>	<u>Explanation</u>
		assumed coplanar with the orbit, and the "vernal equinox" of Venus is taken in the direction of its orbital perihelion.
5	DECL (DECL1)	Declination of hyperbolic excess departure velocity vector, measured in degrees positively northward and negatively southward from the planet's equator.
6	I1	Inclination of transfer orbit to departure planet's orbit, in degrees; zero is forward, $-90^{\circ}$ is perpendicular southward, $+90^{\circ}$ is perpendicular northward, and $\pm 180^{\circ}$ is backward.
7	V1	Heliocentric speed of departure, normalized to Earth's mean orbital speed.
8	PSI1	Heliocentric angle of departure, in degrees; measured counter-clockwise from the outward heliocentric radius vector to the departure velocity vector in the transfer plane.
9	ECCEN	Eccentricity of heliocentric transfer orbit.
10	SMA	Semi-major axis of the transfer conic, in AU.
11	THET1	True anomaly of departure point, measured in the transfer orbit in degrees, reduced to $0^{\circ} \leq \theta_1 \leq 360^{\circ}$ .
12	THET2	True anomaly of arrival point measured in the transfer orbit, in degrees; THET2 minus THET1 equals the heliocentric transfer angle.
13	PERIH	Perihelion distance of the transfer conic, in AU. This value is listed only if the vehicle traverses the perihelion during the transfer.
14	APHEL	Aphelion distance of the transfer conic, in AU. This value is listed only if the vehicle traverses the aphelion during the transfer.

<u>Column</u>	<u>Symbol</u>	<u>Explanation</u>
15	I2	Inclination of transfer orbit to arrival planet's orbit, in degrees; if approach is directly from the rear (i. e., the arrival vector points in the same direction as the planet's motion), then $I2 = 0^{\circ}$ , etc.
16	V2	Heliocentric arrival speed, normalized to Earth's mean orbital speed.
17	PSI2	Heliocentric angle of arrival, in degrees, measured clockwise from the inward heliocentric radius vector to the arrival velocity vector in the transfer plane. Note that this angle is measured opposite in sense to PSI1.
18	RA	Analogous to departure velocity case; note, however, that this is the right ascension of the position at which the arrival velocity vector would emerge again from the planet's celestial sphere. This convention for reckoning asymptote directions is useful when calculating flyby trips. When the trajectory is unperturbed by the target planet, the approach and departure asymptotes have exactly the same right ascensions and declinations. By noting the discrepancies between the two asymptotes in other cases, the analyst may obtain a rough but rapid estimate of the bend angle without performing trigonometric calculations.
19	DECL (DECL2)	Declination of the position at which the arrival velocity vector would emerge again from the planet's celestial sphere.
20	SPEED (VH2)	Magnitude (speed) of hyperbolic excess arrival velocity vector, normalized to Earth's mean orbital speed.

2. Graphics output (G): The system shall display the data, contours, alphanumeric information and control indicators shown in Fig. 3-1 of Para. 3.2.3.
3. Photographic output (G): The system shall prepare permanent copies of the graphical displays which the analyst requests to be stored during the operation of the system. These copies may be prepared directly from the graphical display by a suitable camera or from a peripheral cathode-ray tube plotter.

3.3.2.4 Timing and Storage. Timing and storage is described below:

1. Detailed Requirements: The code storage, data storage, and operating time requirements for the system have been specified in detail in Table 3-2. These numbers are estimates and are furnished only as a guide to and standard of desired performance. They are not intended to be indicative of any particular computing equipment. Moreover, because of variations in such equipment, the number of words of storage for data and code could easily vary by  $\pm 20$  percent from the values estimated in Table 3-2.
  - 1.1 The code size includes the analytical program, the contouring subroutines, a typical graphics executive routine, a typical graphics display package and a typical control program directed by light button signals. Code size does not include the resident monitor(s) for the computer(s) in the system. The code sizes were based on the MACON program as executed on the Univac 1108 computer and on the graphics software employed by the CDC 3300 system at LMSC.
  - 1.2 The data words were estimated on the basis of one word to each floating point number and each group of six alphanumeric characters.

Table 3-2

## TIMING AND STORAGE REQUIREMENTS FOR ORBITAL TRANSFER SYSTEM

SEQUENCE NUMBER	STEP	OPERATION	CODE SIZE	DATA WORDS	TIMING (SEC)	DATA RATE (WORDS/SEC)	GRAPHICS DISPLAYS
1	DATA PREPARATION	ANALYST ESTIMATES & PREPARES INPUTS	-	-	-	-	NONE
2	PROGRAM READ-IN	2.1 PROGRAM IS READ IN AND STORED	4,200	2,200	M-D(A)	M-D	NONE
		2.2 GRAPHICS BACKGROUND IS INITIALIZED	400	100	M-D	-	NONE
		2.3 GRAPHICS BACKGROUND IS DISPLAYED	10,400	2,100	2	100	STORAGE HEADINGS: HEADINGS FOR SPECIFICATION TABLES; CONTROL INDICATORS. NUMERICAL KEYBOARD
3	DATA DISPLAY	3.1 DATA ARE READ IN	500	150	M-D	M-D	"DATA" INDICATOR IS UNDERLINED
		3.2 VARIABLE DISPLAY DATA ARE SELECTED	500	150	1	-	NO CHANGE
		3.3 DATA ARE DISPLAYED	10,500	2,070	1	100	FRAME TITLE AND NUMBER; AXES LABELS AND LIMITS; CONSTRAINTS; LINE LEGENDS AND SCALING FACTORS OF PARAMETERS TO BE CONTOURED; AXIS INCREMENTS & NUMBER; GRID LINE SPACING:
		3.4 ANALYST MONITORS & CORRECTS ERRORS	12,500	2,150	10/ERROR	100	POSITIVE RESPONSE; CORRECTED DATA
4	COMPUTATIONS	4.1 MISSION DATA ARE COMPUTED & STORED	12,000	3,000(B) 4,500(C)	9(B) 19(C)	-	"COMPUTE" INDICATOR IS UNDERLINED
		4.2 CONTOUR SPECIFICATIONS ARE INTERPRETED AND TABLES OF CONTOUR LEVELS ARE COMPUTED FOR EACH PARAMETER	12,100	2,000 3,500	1 1	-	"CONTOUR" INDICATOR IS UNDERLINED
		4.3 CONTOUR COORDINATES ARE COMPUTED(D)	12,100	1,900 4,200	10 20	-	
		4.4 CONTOUR COORDINATES ARE TRANSFERRED TO GRAPHICS EQUIPMENT(D)	M-D	1,700 4,000		-	200
5	CONTOUR DISPLAY	CONTOUR COORDINATES ARE SCALED AND DISPLAYED ON GRAPHICS SCOPE (D)	10,000	3,700 6,000	10/ERROR	-	MINIMUM & MAXIMUM LEVELS; NUMBER OF CONTOURS; CONTOUR LINES AND VALUES; ALL AXIS VALUES; THE "?" BY "NEXT" APPEARS
6	CONTOUR REVIEW	ANALYST MAKES LIGHT BUTTON CHANGES; ANALYST TOUCHES "NEXT" OR "REDO" LIGHT BUTTON; CONTROL PROGRAM ANALYZES CHANGES AND RECYCLES TO PROPER STEP	12,000	2,150		100	100
	6.1 RECOMPUTE	RECYCLE TO 4.1					THE "?" BY "NEXT" IS TURNED OFF
	6.2 CHANGE CONTOURS	IF NEW PARAMETERS HAVE BEEN SELECTED, RECYCLE TO 4.1					DISPLAYS ARE RECYCLED
	6.3 STORE	IF ONLY CONTOUR LEVELS CHANGED, RECYCLE TO 4.2 ANALYST TOUCHES "STORE" LIGHT BUTTON; GRAPHICS DISPLAYS ARE SAVED; NUMBER OF FRAMES LEFT FOR STORAGE ARE DECREMENTED	12,000	3,700(B) 6,000(C)	2	4,000	"STORE" INDICATOR IS UNDERLINED. FRAMES REMAINING & THE "?" BY "NEXT" APPEARS
		CONTOUR COORDINATES ARE RESCALED FOR CRT EQUIPMENT AND OUTPUT ON MAGNETIC TAPE	2,100	2,000 4,300	9(B) 19(C)	200	
7	NEXT CASE	RECYCLE TO 3.1	-	-	-	-	RECYCLE
8	RECALL	GRAPHICS DISPLAYS RETURNED FROM STORAGE	12,000	3,700 6,000	1	4,000	CONTOUR LINES & VALUES; AXIS LABELS & VALUES; FRAME TITLE & NUMBER; HEADING & DATA IN CONTOUR SPECIFICATIONS
		REMAIN ON DISPLAY UNTIL REPLACED BY NEXT RECALLED DISPLAY	2,000	-	16	-	

NOTES: (A) M-D: MACHINE-DEPENDENT (B) 20 \* 20 AXIAL GRID INCREMENTS (C) 30 \* 30 AXIAL GRID INCREMENTS (D) OCCUR SIMULTANEOUSLY

- 1.3 The data words of core storage and the timing (where it could be estimated) are postulated on two assumed contour specifications. One specification considers the departure date-arrival date plane to be divided into 400 grid rectangles, that is, 20 steps along each axis; the other considers it divided into 900 rectangles, that is, 30 steps along each axis. Both specifications further assumed that two parameters are being contoured with six contour levels for each.
- 1.4 Rapid responses to STORE and RECALL shall be required. High data transmission rates shall be needed to satisfy the timing requirements for these two operations.

2. Overall Requirements:

- 2.1 Storage: Exclusive of the resident monitor(s) for the computer(s) used in the system, the core storage for operation without overlays or chain links is estimated to be as follows:

	<u>20 x 20 Grid</u>	<u>30 x 30 Grid</u>
Code words	19,000	19,000
Data words	9,000	12,000

- 2.2 Timing: The time required to read in the data for each error-free case, compute the mission data and contour lines, and display the contour plot is approximately 24 sec for the 20 x 20 grid and 44 sec for the 30 x 30 grid. These times probably will vary with the particular hardware and software on which the Orbital Transfer System is executed.

The time required to save a particular display and prepare a permanent copy by means of a CRT plotter is estimated to be 11 (or 21) sec for the 20 x 20 (or 30 x 30) grid.

The time required to recall a previously stored display should be only one sec, and only about ten sec more should be allowed for scanning it.

- 2.3 Data Transmission Rate: The time specified for recalling a previously stored display places the most stringent requirements on data transmission rates between hardware components of the system. The rate of 4000 words per sec specified in Table 3-2 is about one to two orders

of magnitude faster than that presently being contemplated for telephone cable transmissions. If the store-recall feature of this specification is to be retained, significant improvements must be made in data storage at the graphics console or in data transmission rates.

### 3.3.3 Light Button Controls

3.3.3.1 Guidelines. Light button control guidelines are outlined below:

1. Each light button action shall be initiated by only a single action of the analyst. It shall automatically set off as long a chain of logical decisions and lower level actions as possible within the graphics-computer system (L).
2. Each light button action shall elicit a positive response from the graphics scope (L, G).
3. Sufficient separation distance shall be provided between neighboring light buttons in an attempt to prevent the analyst from accidentally touching the wrong button; providing a positive response to each light button command will, of course, warn the analyst of any such error. Spacing between the infrequently used CONSTRAINTS light buttons shall be at least one-third of an; between all others, one-half (L, G).

3.3.3.2 Control Actions, Responses and Consequences. The light buttons indicated in Fig. 3-1 of Para. 3.2.3 shall be implemented. Their effects, the display responses and subsequent system effects shall be as specified in Table 3-3. The specified recomputations of mission variables and contour lines shall occur only after an execution command (such as NEXT or REDO) is given (L).

Two explanatory notes follow.

1. PARAM Light Button: Contour line legends shall be assigned in the following order (C):

<u>Order</u>	<u>Line</u>	<u>Description</u>
1st	—————	heavy solid line
2nd	- - - - -	light dashed line
3rd	. . . . .	dotted line
4th	—————	light solid line

2. System Execution Following a Specification Change (L):

- 2.1 The normal continuation shall be indicated by the REDO command. However, after the input data have been reviewed and possibly changed, the normal continuation shall be either REDO or NEXT.
- 2.2 The light button control program shall make a logical analysis of the series of commands given since the previous execution. It shall then cause the system to recycle only through those steps needed to change the contour plot and displays according to that series of commands.



Table 3-3

LIGHT BUTTON ACTIONS FOR ORBITAL TRANSFER SYSTEM

LIGHT BUTTON	POSITIVE RESPONSE	ANALYST'S INPUT	POSITIVE RESPONSE	COMMAND COMPLETION	SYSTEM ACTION	NEXT EXPECTED DISPLAY RESPONSES	ERROR SIGNAL	ANALYST'S REACTION TO ERROR
CONTOUR SPECIFICATIONS - LIGHT BUTTONS								
LEGD	LEGEND DISAPPEARS	NONE REQUIRED	NONE	NONE REQUIRED	THE ASSOCIATED PARAMETER IS REMOVED FROM THE TABLE OF VARIABLES TO BE CONTOURED. THE LINE LEGEND IS AVAILABLE FOR REASSIGNMENT TO ANOTHER PARAMETER. THE CONTOURING SPECIFICATIONS (MIN., MAX., SCALING FACTOR, INCREMENT, NUMBER OF CONTOURS) SHALL BE CLEARED.	NONE REQUIRED	IF THE WRONG PARAMETER IS TOUCHED, ITS LINE LEGEND WILL DISAPPEAR.	RESTORE THE PARAMETER & ALL ITS CONTOUR SPECS.
PARAM	NEXT AVAILABLE LINE LEGEND APPEARS.	PROVIDE MISSING CONTOUR SPECIFICATIONS OR PROCEED.	AS APPROPRIATE TO NEXT COMMAND.	NEXT COMMAND	THIS PARAMETER IS ADDED TO THE TABLE OF VARIABLES TO BE CONTOURED. IF ALREADY IN THAT TABLE, ONLY THE LINE LEGEND IS CHANGED & THE CONTOURS WILL BE REDRAWN. IF THIS IS A NEW PARAMETER, THE MISSION CALCULATIONS MUST BE REPEATED. THE NEW VALUES STORED & NEW CONTOURS DRAWN.	"COMPUTE" UNDERLINED IF A NEW VARIABLE WAS ADDED. "CONTOUR" UNDERLINED. NEW CONTOUR PLOT. "NEXT?" APPEARS.	IF SPECS. ARE MISSING, THE "?" NEXT TO "CONTOUR SPECIFICATIONS" APPEARS & THE SYSTEM PAUSES.  IF MORE PARAMETERS ARE SELECTED THAN ALLOWED, NO LINE LEGEND SHALL APPEAR.	PROVIDE MISSING SPECIFICATIONS.  NONE REQUIRED
SCALE MIN. MAX. NC	THE TOUCHED DISPLAY AREA IS UNDERLINED.	NUMERICAL VALUE OF THE SPECIFICATION.	VALUE APPEARS IN THE DISPLAY AREA.	NEXT COMMAND	THE INPUT VALUE IS STORED FOR THE ASSOCIATED CONTOURING VARIABLE. THE TABLE OF CONTOUR LEVELS IS RECOMPUTED, AND NEW CONTOUR LINES WILL BE DRAWN.	"CONTOUR" UNDERLINED. NEW CONTOUR PLOT. "NEXT?" APPEARS.	UNDERLINE APPEARS AT WRONG PARAMETER.	RESELECT THE PROPER ONE
REDO	THE "?" BY "NEXT" DISAPPEARS.	NONE REQUIRED	NONE	NONE REQUIRED	THE CONTROL PROGRAM ANALYZES THE PREVIOUS COMMANDS, & THE SYSTEM ACTS AS DESCRIBED ABOVE.	AS APPROPRIATE	WRONG VALUE INPUT  NONE	REPEAT THE COMMANDS  WAIT FOR "NEXT?"
AXIS & GRID SPECIFICATIONS - LIGHT BUTTONS								
CHANGE FIRST CHANGE LAST	THE TOUCHED DISPLAY AREA IS UNDERLINED.	SIGNED NUMERICAL CHANGE IN THE INDICATED AXIS LIMIT.	SIGNED VALUE APPEARS IN THE DISPLAY AREA.	NEXT COMMAND	THE DATE LIMIT(S) WILL BE CHANGED. NUMBER OF AXIS INCREMENTS WILL BE RECOMPUTED. THE MISSION CALCULATIONS & VARIABLE STORAGE WILL BE REPEATED & NEW CONTOURS DRAWN. NEW AXIS VALUES WILL BE DISPLAYED.	NEW AXIS LIMITS WILL APPEAR ON THE PLOT AXES. NUMBER OF INCREMENTS WILL BE DISPLAYED. "COMPUTE" "CONTOUR" ARE UNDERLINED. NEW CONTOURS ARE PLOTTED. "NEXT?" APPEARS.	SAME AS "SCALE"	SAME AS "SCALE"
AXIS INCREMENT	THE TOUCHED DISPLAY AREA IS UNDERLINED.	NUMERICAL VALUE	VALUE APPEARS IN THE DISPLAY AREA.	NEXT COMMAND	SAME AS "CHANGE FIRST" EXCEPT THAT DATE LIMITS ARE NOT CHANGED.	SAME AS "CHANGE FIRST" EXCEPT THAT AXIS LIMITS ARE NOT CHANGED.	SAME AS "SCALE"	SAME AS "SCALE"
GRID SPACING	THE TOUCHED DISPLAY AREA IS UNDERLINED.	NUMERICAL VALUE	VALUE APPEARS IN THE DISPLAY AREA.	NEXT COMMAND	GRID LINES ON THE CONTOUR PLOT ARE RESPACED. AXIS VALUES ARE RESPACED. MISSION VALUES & CONTOUR LINES ARE NOT AFFECTED.	"CONTOUR" UNDERLINED. NEW GRID LINES AND AXES VALUE APPEAR. "NEXT?" APPEARS.	SAME AS "SCALE"	SAME AS "SCALE"
REDO	THE "?" BY "NEXT" DISAPPEARS.	NONE REQUIRED	NONE	NONE REQUIRED	SAME AS FIRST "REDO."	AS APPROPRIATE	NONE	WAIT FOR "NEXT?"
CONSTRAINTS - LIGHT BUTTONS								
VH1 THROUGH DV1 + 2	THE TOUCHED DISPLAY AREA IS UNDERLINED.	NUMERICAL VALUE	VALUE APPEARS IN THE DISPLAY AREA.	NEXT COMMAND	THE INPUT VALUE IS STORED AS THE CONSTRAINT LIMIT. MISSION CALCULATIONS WILL BE REPEATED BUT TRIPS WHOSE VELOCITIES EXCEED ANY OF THE CONSTRAINTS WILL NOT BE STORED FOR CONTOURS; THEY WILL APPEAR IN THE PRINTED OUTPUT.	SAME AS "CHANGE FIRST" EXCEPT AXIS LIMITS, VALUES AND INCREMENTS WILL NOT BE CHANGED.	SAME AS "SCALE"	SAME AS "SCALE"
DECL1 DECL2	THE TOUCHED DISPLAY AREA IS UNDERLINED.	NUMERICAL VALUE OF LOWER LIMIT OR UPPER LIMIT.	VALUE APPEARS IN THE DISPLAY AREA.	NEXT COMMAND	SAME AS VH1 EXCEPT NO TRAJECTORIES WILL BE OMITTED FROM CONTOUR STORAGE. INSTEAD, THE LETTER D WILL BE PRINTED ON THE CONTOUR FOR EACH TRIP IN WHICH ONE OF THE DECLINATION CONSTRAINTS IS EXCEEDED.	SAME AS "VH1"	SAME AS "SCALE"	SAME AS "SCALE"
REDO	THE "?" BY "NEXT" DISAPPEARS.	NONE REQUIRED	NONE	NONE REQUIRED	MISSION & CONTOUR CALCULATIONS ARE NOW REPEATED WITH NEW CONSTRAINT VALUES.	AS APPROPRIATE	NONE	WAIT FOR "NEXT?"
CONTROL COMMANDS								
NEXT	THE "?" DISAPPEARS.	NONE REQUIRED	NONE	NONE REQUIRED	PROCEED TO THE FOLLOWING NORMAL OPERATION.	AS APPROPRIATE	NONE	WAIT FOR THE "?"
REDO	THE "?" BY "NEXT" DISAPPEARS.	NONE REQUIRED	NONE	NONE REQUIRED	EXECUTE WITH CHANGES.	AS APPROPRIATE	IF NO CHANGES, THE "?" BY "NEXT" APPEARS IMMEDIATELY.	MAKE CHANGES OR TOUCH "NEXT"
STORE	"STORE" IS UNDERLINED.	NONE REQUIRED	NONE	NONE REQUIRED	LARGE-CAPACITY STORAGE DEVICE SAVES THE CONTOUR LINES & VALUES, AXIS LABELS & VALUES, FRAME TITLE & NUMBER AND HEADINGS & DATA OF THE "CONTOUR SPECIFICATIONS." THE GRAPHICS DISPLAY IS TRANSMITTED TO A CRT PLOTTER.	"STORE" UNDERLINE DISAPPEARS. THE "?" BY "NEXT" APPEARS.	NONE	NONE REQUIRED
RECALL	"RECALL" IS UNDERLINED.	FRAME NUMBER(S)	INPUT NUMBER APPEARS TO THE RIGHT OF "RECALL."	"RECALL" TO ENTER FRAME NO. OR "NEXT."	A TABLE OF FRAME NUMBERS FOR RECALLING IS FORMED. THE SPECIFIED DISPLAYS WILL BE RECALLED FROM STORAGE.	RECALLED CONTOUR PLOTS AND SPECIFICATIONS APPEAR. THE "?" BY "NEXT" APPEARS.	IF THE FRAME NUMBER HAS NOT BEEN STORED, IT IS IGNORED.	NONE REQUIRED
STOP	"STOP" IS UNDERLINED.	NONE REQUIRED	NONE	NONE REQUIRED	JOB IS TERMINATED.	NONE	"STOP" WAS UNDERLINED.	RESTART THE JOB.

### 3.4 GRAPHICAL-COMPUTER SYSTEM FOR ASCENT BOOST OPTIMIZATION

#### 3.4.1 Objectives

3.4.1.1 Introduction to the Problem. The ability to rapidly and accurately evaluate the performance capability of multistage rocket booster systems is a primary requirement for mission planning and preliminary design studies. Achieving maximum payload capability from a given booster system requires utilization of an optimal "tilt" or thrust vector orientation program during powered flight. The sensitivity of performance to changes in hardware parameters cannot be properly evaluated unless the effect of tilt program changes is eliminated by optimizing the tilt program for each set of parameters.

In addition to TILT program complications, the mission generally requires the satisfaction of a number of constraints on the trajectory flown, such as terminal altitude or velocity. Finally, a number of parameters defining such quantities as the initial launch vectors, the coast periods between powered stages, and the upper stage thrust cycles may be treated as variables for certain boost hardware combinations. These "adjustable parameters" should then be optimized along with the tilt programs and the terminal constraint combination to produce maximum performance.

The digital computer program for Rapid Earth-to-Space Trajectory Optimization (PRESTO), Ref. 3-6, has been developed to solve the performance problem outlined above for orbital, lunar, and interplanetary missions. Optimization of pitch and yaw tilt programs and a variety of adjustable parameters and the satisfaction of constraint combinations is accomplished simultaneously with a closed loop steepest descent optimization routine that maximizes payload .

Unfortunately, difficulties and delays often arise in evaluating booster systems that are radically different from those with which experience has been gained and data accumulated. Similarly, synthesis of new missions or trajectory profiles that require unusual pitch and/or yaw tilt programs can present difficulties. The problems

fall into three general categories: (1) errors in data input, (2) poor initial estimates for the control programs and launch angle, and (3) slow optimization convergence because of the relative importance assigned to each of the adjustable parameters and the sensitivity to tolerances on the constraints.

When using the batch processing mode, the analyst is forced to solve these problems by reviewing the data, making the changes deemed appropriate, and resubmitting the computer program. When this data returns, further examination shows whether those changes were sufficient or even appropriate. Thus, solution of the ascent boost optimization process may require repetitive passes at the computer with the associated calendar time delays and increased costs.

3.4.1.2 Objective of the System. The objective of the graphical-computer system for ascent-boost optimization is to furnish the analyst with a on-line monitoring and redirection capability.

The graphical display of trajectory and booster system parameters will enable the analyst to monitor each trajectory profile as the computation proceeds, detect irregularities and delays in convergence towards the optimum solution, and redefine those parameters necessary to speed up or improve the convergence procedure such that the booster system payload capabilities may be quoted with high confidence levels. Specifically, the graphical display system will provide the mechanism to do the following:

- Review and correct the most error-prone portions of input data from both system parameters and control histories.
- Evaluate the desirability of the resultant initial trajectory for the particular mission objectives prior to initiating the optimization cycle of the program.
- During the optimization cycle, monitor the trajectory profile and key optimization indicators to evaluate the success of the convergence scheme as it proceeds towards the optimum solution.

- Interrupt the optimization cycle to change tolerances on the terminal and intermediate constraints, redefine the weighting factors associated with the adjustable parameters, and/or reshape the control program.
- Perform a rapid evaluation of payload sensitivity to system parameter and initial condition changes by allowing the analyst to redefine these parameters of interest and reinitiate the optimization cycle.

### 3.4.2 Computer Program Description\*

The existing batch process digital computer program that solves the payload optimization problem will be described in this section.

3.4.2.1 PRESTO. The digital computer program for Rapid Earth-to-Space Trajectory Optimization (PRESTO) uses a closed-loop, steepest-descent optimization procedure to derive flight trajectories that produce maximum booster payloads for a variety of space missions. Trajectories can be computed in three degrees of freedom about a spherical rotating earth. Four powered stages and three upper-stage thrust cycles are accommodated. Coast periods are permitted between each stage. Aerodynamic lift and drag forces are included in the computations. Univac 1108 computing times under one minute are currently being realized for complete three-stage boost trajectory optimizations from Earth surface to Earth orbit.

The optimization routine simultaneously considers the launch direction and time, interstage coast durations, upper-stage thrust sequencing, complete pitch and yaw attitude histories, and the terminal constraints. Intermediate constraints may be introduced on angle of attack, coast orbit perigee altitude or on the product of angle of attack and dynamic pressure. The closed-loop procedure greatly facilitates the satisfaction of terminal constraints and reduces the number of iterations required to achieve convergence.

---

\*This subsection has been extracted from Ref. 3-6 and is presented here for clarity in understanding the graphic-analytic program interfaces.

Four basic missions are accommodated by PRESTO – Earth launch to Earth orbit, Earth launch to lunar transfer, lunar landing from lunar orbit, and Earth launch to interplanetary transfer. The lunar and interplanetary transfer missions can also be initiated from Earth orbit. From one to six terminal constraints are permitted on Earth orbit and lunar landing missions, two or three constraints on lunar transfers, and two constraints on interplanetary missions. Lunar, Mars, and Venus ephemeris routines are included in the program.

For orbit missions, constraints are imposed either directly on the injection trajectory variables or on the orbit elements. When the orbit elements are specified, functional constraint relationships are used. For lunar missions the constraint input is transfer time, day of launch and, if desired, transfer orbit inclination. The constraint routine defines the functional relationships between the injection trajectory variables that produce the required lunar intercept using special closed-form expressions to represent the transfer orbit. For interplanetary missions the constraint inputs are transfer time and launch date. The departure asymptote direction and excess speed are computed from a matched conics routine and the functional relationships between the constraints and the injection trajectory variables are defined in terms of the excess speed and the departure asymptote direction.

3.4.2.2 Sequence of Trajectory Iterations. The several types of trajectory computations used in this program may be categorized in three ways. First, they are either "forward" or "backward." The equations of motion are integrated on a forward trajectory only; on a backward trajectory, the adjoint differential equations are solved. The next major categories are "guidance" or "optimization." On a guidance trajectory, one is concerned only with meeting terminal constraints on the trajectory variables; in optimization, mass improvement is attempted as well. Since solutions of the adjoint equations are used differently for guidance and optimization runs, a backward guidance run must precede a forward guidance run. Similarly, a forward optimization run is preceded by a backward optimization run. The remaining major categories are "successful" or "unsuccessful," as judged at the end of each forward trajectory. A successful forward guidance trajectory satisfies terminal constraints within acceptable limits, and a successful forward optimization achieves some increase in terminal mass as well.

The sequence of trajectory iterations starts with the initial nominal run which, of necessity, is simply an "open-loop" integration of the equations of motion using an input thrust-orientation history. This trajectory is then used as the basis for a backward guidance trajectory, which is followed by a forward guidance run. It is always assumed that a forward guidance trajectory represents an improvement over the previous nominal. Thus each forward guidance trajectory becomes the new nominal and the basis for a new solution of the adjoint equations. Backward and forward guidance runs are continued until one is judged "successful." At this point a backward optimization run is made and the magnitude of the initial mass improvement request is computed. A forward optimization run is then made. If it is successful, it is used as the new nominal, and backward and forward optimization trajectories are computed. If it is unsuccessful, that trajectory is discarded and another forward optimization is computed with half of the previous attempt at mass improvement specified. Thus, a succession of successful and unsuccessful optimization runs are computed either until the attempted mass improvement is smaller than a specified minimum or until the count of forward trajectories is within one of a specified limit. In either case, the sequence is then ended with a backward and a forward guidance trajectory.

### 3.4.3 Interactive Aspects

3.4.3.1 Cycle of Operations. The following description of the cycle of operations for the ascent boost optimization problem uses the terminology PRESTO to represent the optimization program, GRAPHICS to denote the display logic program, and EXEC to denote the decision link between PRESTO and GRAPHICS. The interactions between these three links and the two graphic display formats to be used, designated REVIEW and MONITOR, are as follows:

- The problem is initiated with computer control transferred to the EXEC program. All punch card data has been read into storage, and this represents all the basic data necessary to process all cases known at this time. GRAPHICS presents the "decision panel" on the display screen.
- EXEC extracts from storage the data required for the first case.

- Using this data, PRESTO computes the initial nominal trajectory for the case, and determines the terminal constraint errors.
- If the analyst is confident that the input data and initial nominal trajectory are satisfactory without reviewing these results, a NO REVIEW selection may be made on the decision panel, this step is bypassed. Otherwise, selection of REVIEW on the decision panel activates that display from GRAPHICS for examination by the analyst. This review of both the initial nominal trajectory and input data that produced it provides a unique opportunity not only to isolate and correct input errors but to evaluate the desirability of the initial trajectory. If the trajectory profile does not appear conducive to mission success, the analyst may change a combination of initial conditions and control parameters and regenerate the initial nominal trajectory. This process becomes particularly desirable for missions that require unusual maneuvers (i.e., dogleg turns) or when a new booster system is being evaluated. When the analyst is satisfied with both data and trajectory, he initiates the optimization cycle with a light button.
- The ascent boost optimization cycle uses both PRESTO and GRAPHICS. PRESTO computes trajectories in a prescribed sequence, and, after each forward trajectory attempted, certain parameters are displayed by GRAPHICS on the scope working area in the MONITOR format. With no interruption by the analyst, PRESTO continues the optimization procedure and the GRAPHICS routines update the scope after each forward trajectory. This dynamic display allows the analyst to monitor key criteria of the optimization process and to interrupt a slow or poorly converging case in order to redirect it.
- In addition to the information of the MONITOR format, the characteristics of the last available successful trajectory are being stored in order to display them in the REVIEW format, if desired. This display would be used as backup information when analyzing a case with poor convergence.
- Optimization continues to completion or until the maximum allowable number of iterations have been computed. Prior to computing the final guided trajectory, the optimization program will halt and give the analyst the opportunity to extend the optimization cycle or, if satisfied with the present results,

to complete the case. The final guidance trajectory characteristics will then be displayed in the MONITOR format, and the REVIEW format would be available upon light button command.

- Satisfied with the results, the analyst would trigger the return to the EXEC program, which would then extract the data necessary for the following case from the stored (punch card) input data, and reinitiate the cycle.
- If all of the input cases have been processed, the EXEC would initiate re-display of the last REVIEW format but with all characteristics resulting from trajectory integration eliminated, and await input from the analyst. This would permit the synthesis of characteristics for additional trajectory optimizations.
- When the study is completed, triggering the OFF light button would provide a printed listing of the final trajectories of the cases evaluated using the standard PRESTO output format. The program would then terminate.

3.4.3.2 Displays for Ascent Boost Optimization. Two graphical displays have been developed for the ascent boost optimization problem. They serve two different functions. The REVIEW display provides the means for a more leisurely and thorough perusal of numerical data, whereas the MONITOR display presents the rapidly changing information in a format where errors, slow convergence, and oscillations in key parameters should be immediately apparent.

Both displays contain light buttons within the scope working area. In keeping with the premise of minimizing analyst decision chains, the computer reaction to a single analyst light button action should be as complete a logical chain of commands as is possible. Only when absolutely necessary should the analyst be required to input a chain of light button commands to achieve the desired result. For example, if the analyst wishes to change the input data, touching the light button surrounding the desired data word should initiate machine responses that would include the following:

- Underline the data word so the analyst can confirm his choice.
- Accept new input data from the scope keyboard in the correct format, and display the new value.



- Recognize that when control histories are input via light pen a single light button command should allow the analyst to construct the curve and then automatically manipulate it into the data format necessary for PRESTO.
- Cancel the data input capability for the parameter just selected if the analyst touched another light button (such as GO) prior to inputting data and terminate that data acceptance sequence with no further machine action. Touching another light button after data input would signal acceptance of those data.
- Determine whether those data require a special restart or reinitialization in the optimization program. (For example, inputting an allowable number of iterations increases that counter and sets a flag to indicate the end has not yet been reached, whereas, changing weighting factors requires a backward integration.) React to all data changes such that all necessary changes are made, but do not cause duplication or false restarts.

When developing the two display layouts, a working area of 11 x 17 in. was assumed available. (Illustrations are shown 3/4 size.) An alphanumeric font of seven characters per inch, six lines per inch was used. However, experience has shown that a smaller font, approximately half size, can be used for the titles while retaining full size for numbers or single letters. This is possible since titles can be easily recognized as a group, whereas numbers have to be individually interpreted. The advantage of this dual size font layout is that it reduces the apparent amount of information displayed (scope clutter) and differentiates between guides (titles) and information (data).

The decision panel light buttons and a standard alphanumeric keyboard for data input changes will remain on the scope control surface area outside these displays during the entire time of program operation. The decision panel layout would be

STOP

NO REVIEW MONITOR  
RESTART GO?

OFF

Some of these decision buttons have already been referred to in the cycle of operations section; their functions are:

STOP

An immediately identifiable panic button that stops program operation and lets the analyst make a decision on what action to take next.

REVIEW

Calls the REVIEW display to the scope, interrupting the optimization cycle if necessary.

NO

Bypasses the REVIEW format pause for the initial nominal and permits computation to proceed into the optimization mode.

MONITOR

Calls that display to the scope, but does not initiate the optimization process.

RESTART

Initiates the computation of the initial nominal trajectory and subsequent display in the REVIEW format (this could be used repeatedly to refine initial estimates of control history profiles, for example).

GO?

The question mark appears when the program is waiting for input of any type by the analyst. GO after STOP would continue operation from the interruption point. GO when the REVIEW display of the initial nominal is on the scope initiates the optimization computation and calls the MONITOR display. GO with a REVIEW display of any other trajectory continues computation from the point of interruption or as determined by other light buttons and calls the MONITOR display. GO with the MONITOR display on the scope continues computation from the point of interruption or as determined by other light buttons.

OFF

Triggers hard copy printout and terminates the program

Finally, any error messages generated during program operation will be displayed in the control surface area above the display layouts for quick analyst response. Provisions must be made such that analytic program errors don't cause total program failure, i.e., termination.

3.4.3.3 REVIEW Display (Fig. 3-2). Details of the working areas of the displays correspond to the functional layout on the left of Fig. 3-2. Light buttons are indicated by the dashed boxes on the illustration.

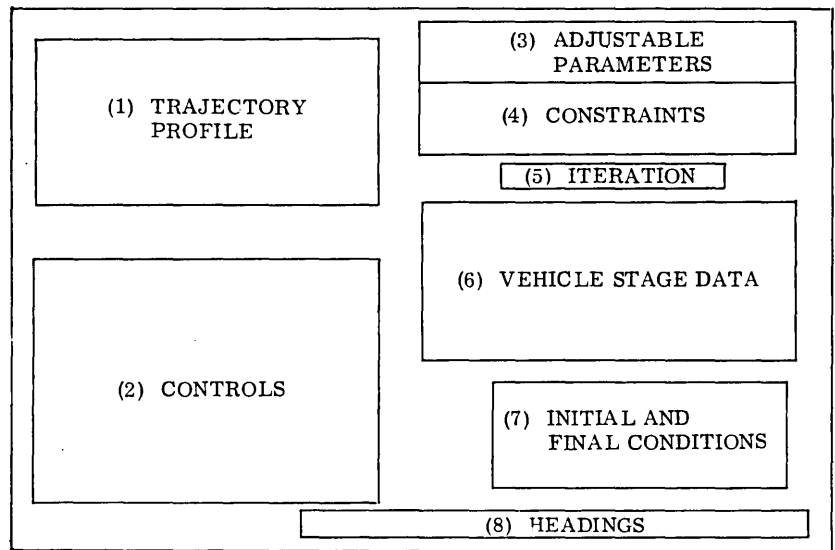
Trajectory Profile. The combination of altitude and aerodynamic velocity provides an excellent indication of the trajectory profile. For example: Is the terminal altitude anywhere near the desired target condition? Did any stage impact the ground? Is each stage contributing its expected velocity increment? Note that both altitude and velocity can legitimately be decreasing; for example, descent to lunar surface from lunar orbit.

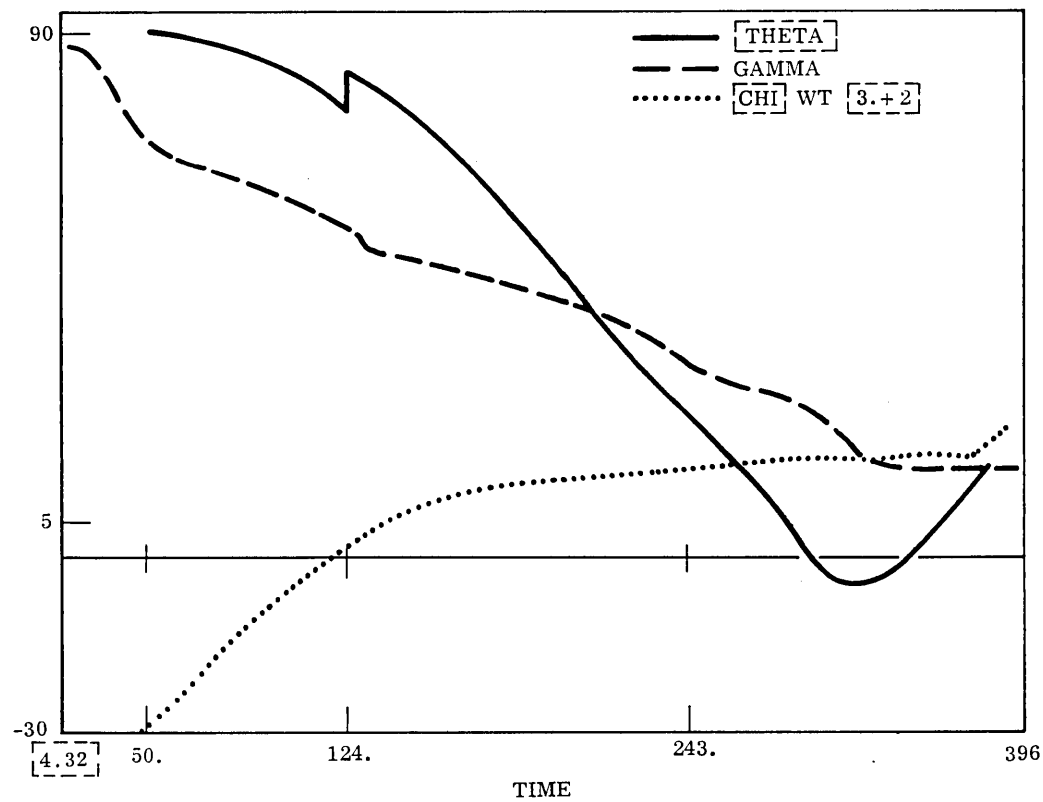
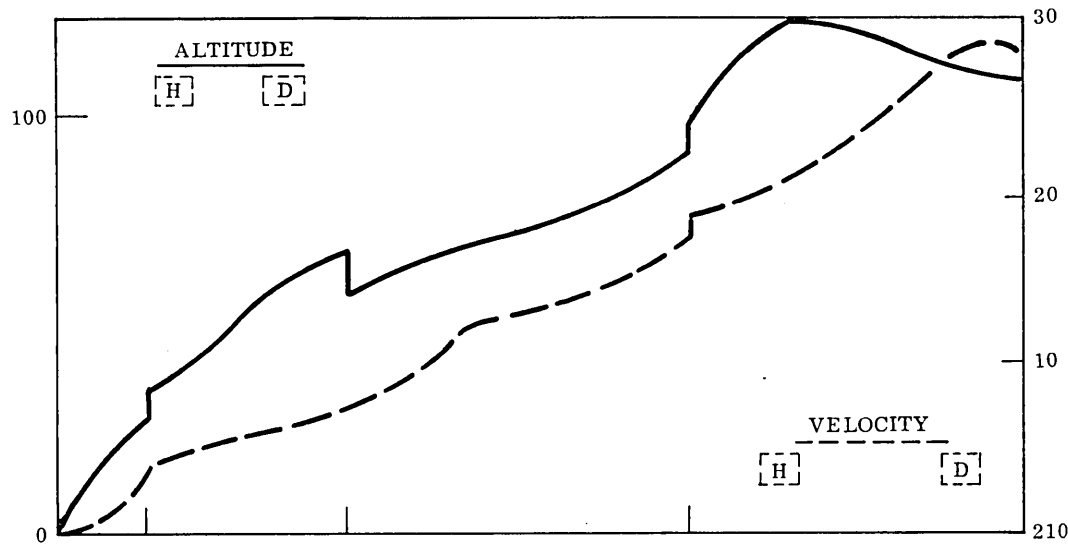
Simplicity is the keynote; the altitude scale runs from zero to 100 nm and velocity to 30,000 ft/sec in increments of 10,000 ft/sec, where both these scales can be halved or doubled by single light button commands (H,D); repetitive commands change scales once for each command. Curves are plotted only for the time under power; verticals connect any "breaks" that might occur at staging because coast stage conditions are not being shown. The time scale unit length is set by the total amount of time under power and automatically scaled by the GRAPHICS program. Numerical values are not shown on the time scale since it is identical to the control history time scale directly below. However, stage ignition points are indicated with short vertical line segments. The vehicle velocity at the end of the closed-form liftoff is output in ft/sec at the bottom of the velocity scale and unaffected by scale changes.

Controls. Pitch (theta) and yaw (chi) control histories are plotted starting at the time the zero angle of attack constraint is removed (i.e., only when meaningful). The control history to be used on the next forward trajectory may be input by touching the appropriate light button and then using the light pen to indicate the new data points that the control history curve should fit. The weighting factor on chi is changed by touching its light button and entering a numerical value via the keyboard.

The angular scale nominally spans +90 to -30 deg, and may be repetitively doubled or halved by touching the appropriate light button the desired number of times. The 5-deg tick mark is always shown to ease rapid estimates of scale height changes, especially for redrawing. If the upper scale limit becomes 10 deg or less, this mark is eliminated.

FUNCTIONAL LAYOUT





ADJUST	6	AZM	GAMMA0	CST 4, 2	BURN 4, 2	CST 4, 1	BURN 4, 1
MAGNITUDE		1.43 + 2	3.27 + 2	4.31 + 0	2.73 + 1	3.21 - 1	3.24 + 0
ADJUSTMT		1.12 - 3	-4.32 - 1	3.21 - 3	-1.11 + 1	6.23 - 3	1.32 - 1
WEIGHT		2. + 3	3. + 1	1. - 5	1. - 2	1. + 0	1. - 2
CONSTRN	6	ALTIT	GAMMAI	INCLN	NODE	LONG	LAT
MAGNITUDE		6.02 + 5	0. + 0	1.57 - 0	1.24 + 0	-1.12 + 0	-2.21 - 1
OK ERROR		2.32 + 2	1.20 - 3	1.20 - 3	3.21 - 2	4.63 - 1	3.62 - 4
CORRECT'N		6.21 + 1	3.62 - 7	-3.24 - 4	6.21 - 3	-3.21 - 4	6.21 - 3

ITERATION 1 CONSTRAINTS

STG	ISP	S	T <sub>v</sub> , T	WEIGHT	T/W <sub>0</sub>	STG SEQ
1	DUR'N	AE				
	267.43	236.45	7000000.	2430000.	1.24	
2	50.73	398.12	6213642.	1672000.		1
	268.42	18.88	3000000.	1430000.	2.13	2
3	72.86	13.72	2742295.	666000.		5
	282.73	18.92	745000.	587000.	1.37	3
4	123.24	10.00	744672.	274000.		6
	287.71	4.62	303000.	245432.	1.62	4
5	154.83	10.00	303000.	85692.		7
	.02	13.27		668000.		4
						8

TIME	VEL	GAMMA	PSI	ALTITUDE	LAM	TAU
0.	0.	72.48	181.50	378.	34.72	-120.62
390.2	25630.	1.3	183.45	712345.	21.18	-121.42

STOP	IN VEL	EYE	BETAP	OMEGAE
	25581.	90.15	80.30	159.83

QBAR MAX 32482. AT 172.8

72 CHARACTER CASE HEADING LAUNCH SOUTH, 100 NM PERIGEE 6 CONSTRAINT

72 CHARACTER MAIN HEADING SMALL SOLID BOOSTER ZEUS UPPER STAGE WTR

Fig. 3-2 Ascent Boost Optimization REVIEW Display



The flight path angle ( $\gamma$ ) is shown from the end of closed-form liftoff, unless it is off scale, in which event the curve does not appear until it returns within scale. Similarly, when the controls exceed scale height, they are not shown until within scale. However, when redefining the shape of a control history, the analyst may work outside the scales then shown, assuming scale extensions at the current unit rate.

The time scale begins at the end of closed-form liftoff, and this value may be changed by light pen input. Coast period times are not included, and breaks in the curves across coasts are fitted with vertical segments. Time scales are identical for the MONITOR layout.

Adjustable Parameters. Names, input magnitudes, current adjustment, and input weighting factors are shown only for the adjustable parameters being used. Provision is made for up to six of the ten possible parameters, since experience shows this number represents an upper limit requirement. Inputs may be changed by a light button-keyboard sequence. The number of parameters and the parameters themselves (numeric codes) may be changed by light button.

Constraints. Names, input magnitudes, error tolerances, and current corrections are shown only for the constraints being used. Inputs, the number of constraints, and the constraints themselves (numeric codes) may be changed via light button-keyboard sequence.

Current Iteration Number and Trajectory Status. Status nomenclature includes:

CONSTRAINTS	The trajectory failed to meet constraints.
SATURATE	The control parameter changes necessary to satisfy constraints exceeded limits (saturated controls).
PAYOFF	The trajectory failed to achieve a payoff improvement.
SUCCESS	The constraints were met, and, when optimizing, a payoff improvement was achieved.
FINAL	Final guidance run (end of case).

Vehicle Stage Data. A combination of raw input data and program computed values is shown that ensures a rapid verification of vehicle parameters. The input parameters that may be changed for each of the five main stages via light button-keyboard are:

DUR'N	Stage burn time duration (sec)
ISP	Stage vacuum specific impulse (sec)
AE	Stage aerodynamic reference area (ft <sup>2</sup> )
S	Stage total nozzle exit area (ft <sup>2</sup> )
T <sub>v</sub>	Stage ignition vacuum thrust (lb) (if this value is changed, the program constructs a rectangular thrust curve at the new level for the entire burn time)
WEIGHT	Stage initial weight (lb)
STG SEQ	Stage sequence, a numerical code indicating the order of computation of powered and coasting stages

Other parameters that are output for checkout only and cannot be altered are:

T	Net thrust at stage ignition (lb)
WEIGHT	The second value of stage weight is the program computed value of stage burnout weight (lb)
T/W <sub>0</sub>	Ignition thrust-to-weight ratio (g)

This data block is sufficient to allow rapid variation of stage parameters to determine payoff sensitivities, as well as verify inputs.

Initial and Final Conditions. Initial conditions may be changed via light button-keyboard; final conditions are output for information on terminal trajectory conditions. In addition to the six state variables, the stopping parameter and three orbital parameters are output.

Headings - Identification. When a character is touched by the light pen, the six-character alphanumeric word of which it is a part is underlined. Keyboard changes



must then start with the first character of that word, but may continue beyond that one word until the end of the title or until another light button is activated.

3.4.3.4 MONITOR Display (Fig. 3-3). Characteristics of the current trajectory are shown on the MONITOR display; this display changes as fast as a new forward trajectory is computed. Refer to the left side of the illustration for area definition. Light buttons are indicated by dashed rectangles.

Trajectory Profile. Only the altitude profile is displayed, and to the same scale as the REVIEW display. The scale may be halved or doubled, and changes here carry over to the next display of the REVIEW format. Placing the indicated upper altitude value at 80 percent of the physical scale height allows for reasonable overboost trajectories to a given terminal altitude.

Both the trajectory profile and controls are in the same area and with the same format as the REVIEW display; this should facilitate rapid transition between the two displays and be less confusing to the analyst.

Controls. With the flight path angle eliminated, changes in the two controls between iterations should be more apparent. The angular scale may be doubled or halved, and any change will be reflected on the REVIEW format. The time scale is identical to that established for the REVIEW format. If the REVIEW format has not been used, the initial values are as shown for altitude and control magnitudes. The chi weighting factor is manipulated in the same manner as the weighting factors on adjustable parameters, discussed below.

Adjustable Parameters. The adjustable parameters, the indicators, and the constraints are all represented by "thermometers" or linear indicators. The horizontal segment of the arrow head indicates the magnitude of the parameters, the direction of the point shows whether the parameter is increasing or decreasing in absolute magnitude, and the  $\pm$  sign grouping is a flashing signal that appears only when the sign of the parameter changes from one run to the next. These concepts should enhance the analyst's ability

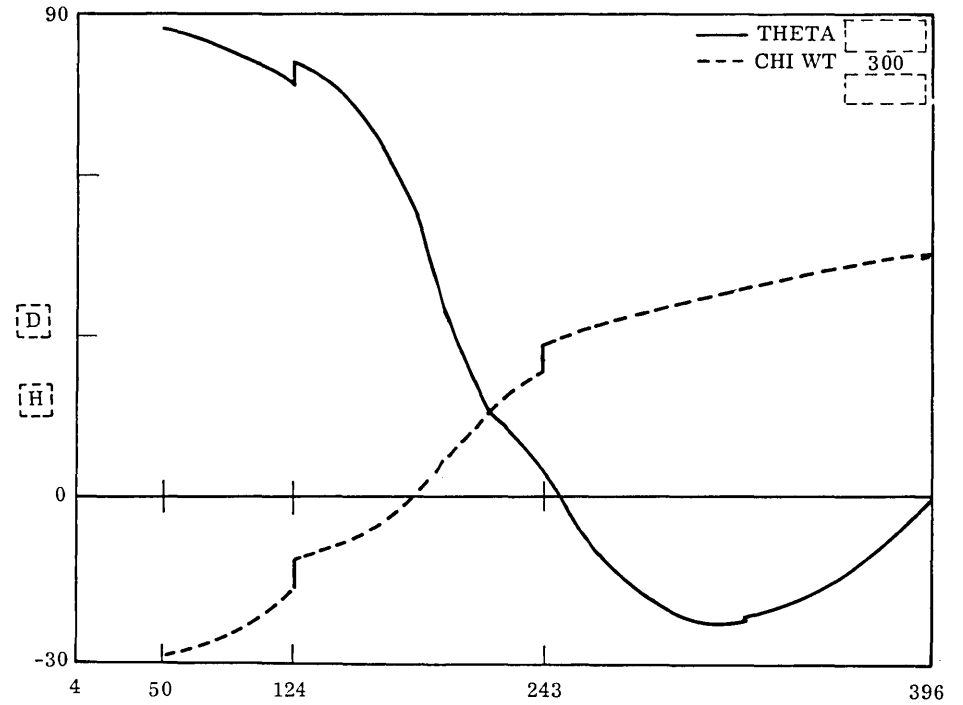
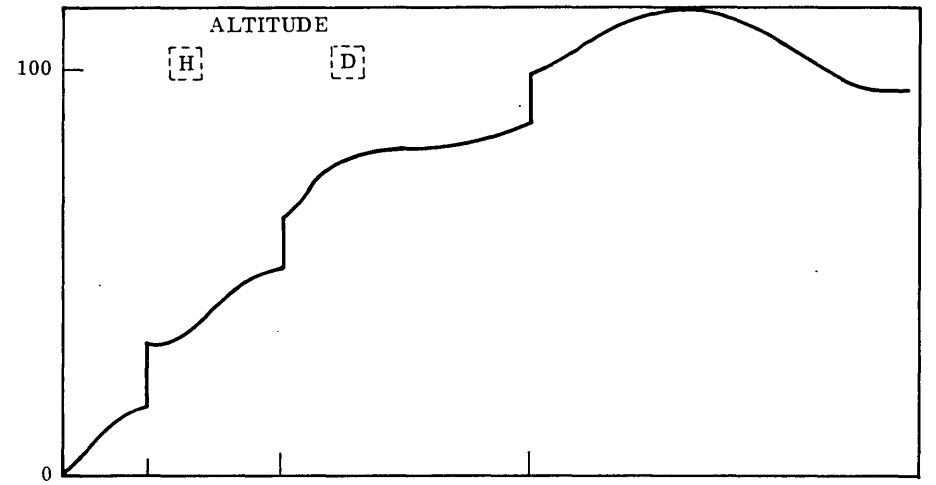
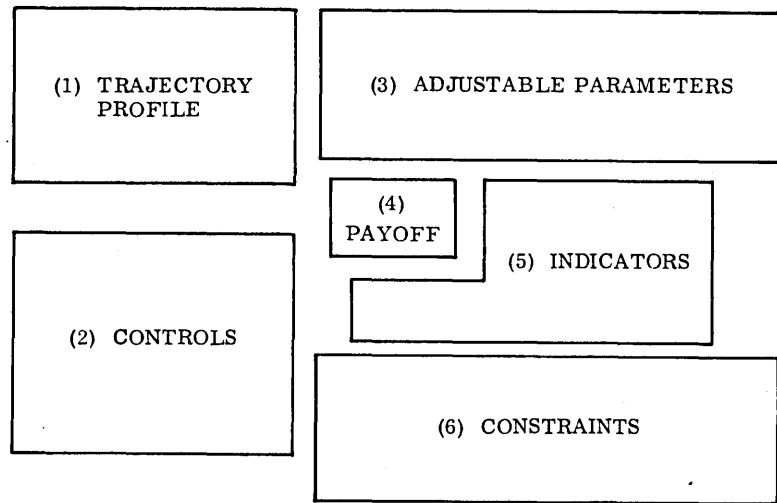
to detect changes that require a reaction. Since more information is being presented that can be monitored on each iteration, the display pattern has been designed to capitalize on the dynamic motion of the error signals.

Thermometer scales are shown only for those adjustable parameters actually being used, and have a one-to-one correspondence to those shown on the REVIEW display. The left side of the thermometer shows the magnitude of the current adjustment. During guidance iterations, the scale is renormalized after each trajectory, and thus represents a step-by-step progression. After optimization starts, the normalizing value is fixed at that on the last successful forward guidance. Thus the arrows should represent changes in adjustable parameters caused by mass improvement requests. However, the scale may be renormalized to the current value during optimization by touching the light button on the top left side of the respective thermometer scale.

The constraint name appears directly below each scale, and the magnitude of the weighting factor appears below the name. The numerical value of the gain, or weighting factor, is output as "a fixed-point number" since exact values are unimportant; only relative magnitudes between gains are of interest. Fractional values should be output with the correct number of preceding zeros, but no decimal point. The magnitude of the gains may be doubled or halved by touching light buttons directly above or below, respectively, the current numerical value. Each click of the light pen causes an additional doubling or halving operation, i.e., three clicks above a gain would increase the value by a factor of eight (3 doubles). Weighting factors, when changed, are introduced into the program without altering the normal trajectory sequencing. In the lower left corner of this block, the D and H serve as reminders of this multiplying capability. The number above the word ADJUST indicates the number of adjustable parameters being computed.

Arrows on the right side of the thermometer scales represent changes in the payoff quantity per unit change in the adjustable parameter, where the computation also includes the effects of changes in control parameters necessary to satisfy terminal constraints. As such, the computation represents a "total derivative" type of sensitivity. The scale runs from 0 to 2 showing only absolute magnitude. It is normalized with

FUNCTIONAL LAYOUT



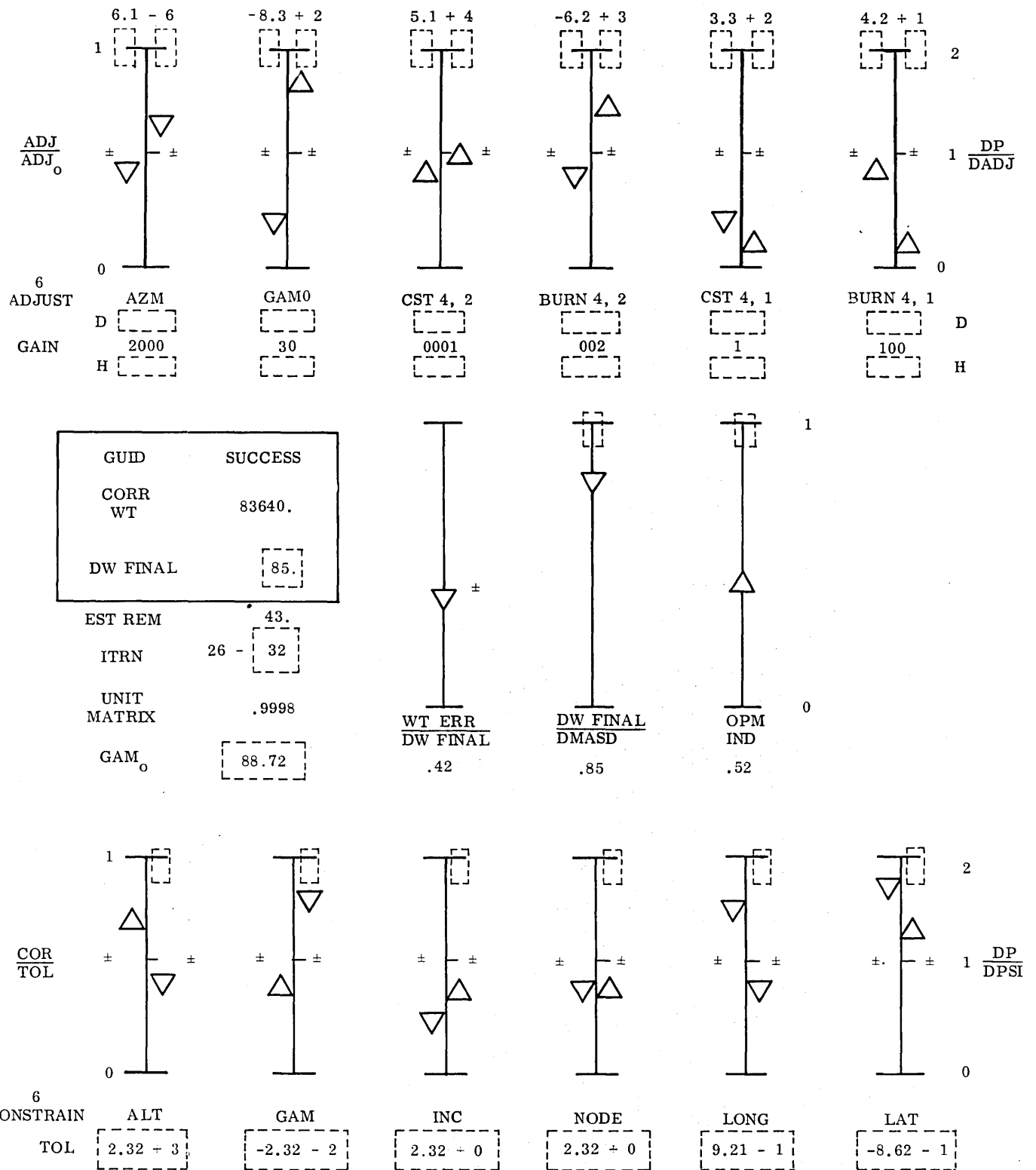
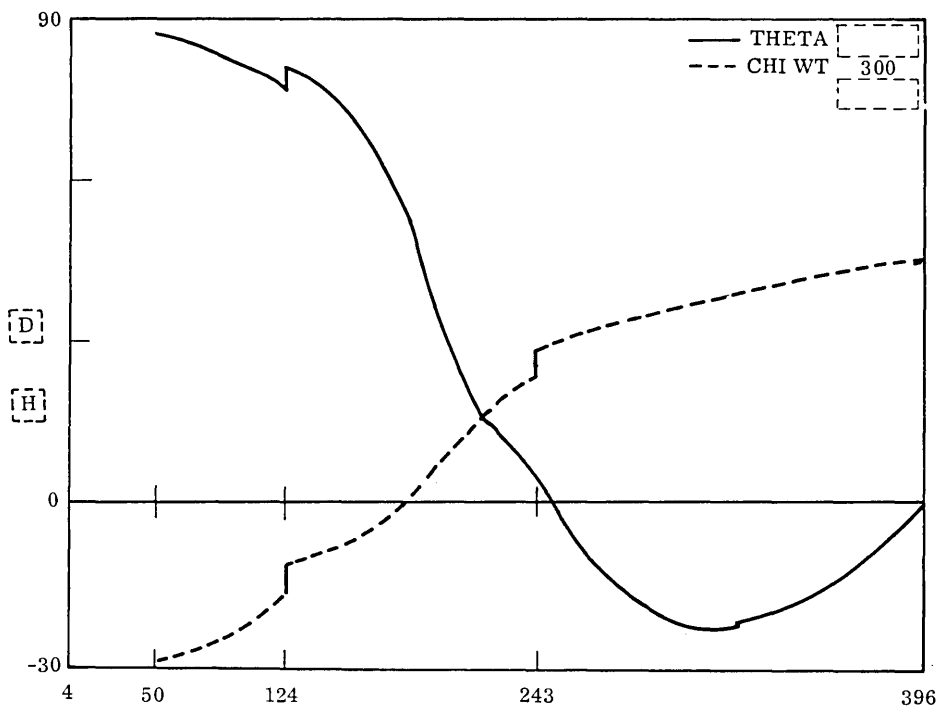
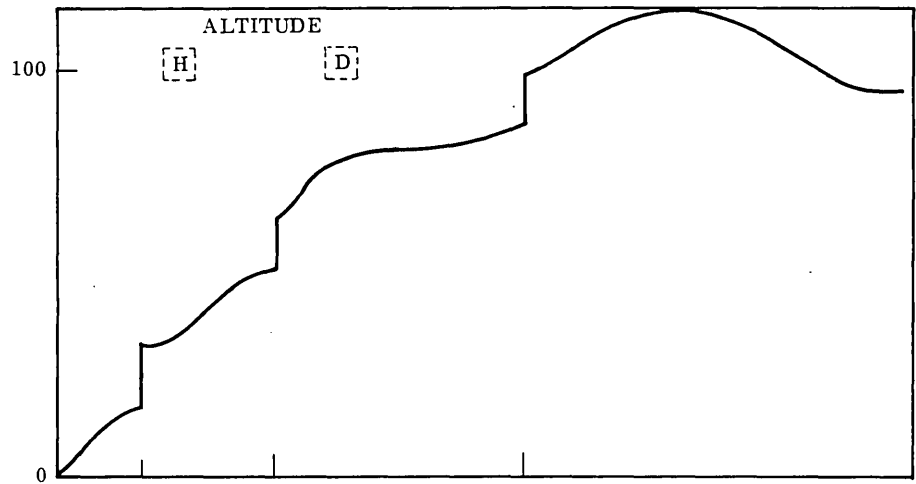


Fig. 3-3 Ascent Boost Optimization MONITOR Display

respect to the value of the first backward optimization and may be renormalized to the value of the last successful trajectory via the light button on the top righthand side of the respective thermometer scale. The current actual value appears above the scale, and carries the correct sign. Sign changes are also indicated by the on-off behavior of the  $\pm$  symbol. Note that the current sign is not shown with either the + or - sign, but rather the combination is used to indicate a changing condition.

If the optimization were proceeding successfully, the behavior of the adjustable parameter thermometer scales would be a sinking of the left side towards zero and movement of the right side to the center of the scale. Since all scales would exhibit the same characteristic behavior, deviations should be quite apparent.

Payoff parameters are boxed for rapid visual access. The current trajectory mode is indicated by GUID for guidance or OPTM for optimization attempts, and the success or failure indicated by the trajectory status, identical to the REVIEW display (SUCCESS, PAYOFF, CONSTRAINTS, SATURATE, FINAL). CORR WT is the vehicle burnout weight in pounds corrected for terminal constraint misses and represents the optimization payoff quantity. DW FINAL is the nominal value of the burnout weight improvement requested, and diminishes as the optimization nears completion. Analyst changes in DW FINAL would not require changes in the normal trajectory sequence.

Indicators. Four numerical values and three thermometers are output as key indicators of the degree of success the optimization procedure is having.

EST REM	Estimated remaining weight improvement that can be realized for this case (lb).
ITRN	Current iteration number and the total that will be allowed. Total may be changed via light button-keyboard and does not interrupt the normal trajectory sequencing operation.
UNIT MATRIX	Deviations from a value of 1.0 indicate failure in matrix inversion process.

$GAM_0$  Initial flight path angle (deg). Oscillations signify failure of convergence for this key adjustable parameter. If the analyst changes  $GAM_0$ , the trajectory sequencing should be restarted from the initial nominal logic since changes in this parameter alter the trajectory and adjoint equation integrations. Control histories, gains, and vehicle weights from the last successful trajectory should be retained.

Each of the three thermometer scales are normalized (zero to one range). They are so designed that with a properly proceeding optimization, each of the three arrowheads will sink to the bottom of the scale. The actual numerical values are shown directly beneath each scale for reference. Only the first scale requires the  $\pm$  symbol to indicate sign changes.

Scale 1 shows the weight error due to missing terminal constraints divided by the requested weight improvement. A trajectory that exactly satisfies the terminal constraints produces a value of zero. On the other hand, a value of one means the terminal constraint errors may cancel any achieved weight improvement, and the constraint tolerances may have to be reduced.

Scale 2 shows the ratio between the current requested weight improvement  $DW_{FINAL}$  and its initial value  $DMASD$ , and indicates how far the optimization has progressed in the sense of requiring smaller step sizes in the requested payoff.

Scale 3 is the optimization indicator, actually the inverse of the optimization indicator currently computed on PRESTO. The inversion causes the displayed value to start at 1.0 and decrease towards zero, rather than increasing without bound.

Since the values of both scales 2 and 3 can change by several orders of magnitude, these scales may be renormalized to the value on the last successful trajectory by merely touching the light button on the top of the scale.

Constraints. The thermometer indicator scales for the constraints operate similarly to the adjustable parameter indicators. Scales are shown only for the constraints being used, and under normal operation the left arrowheads will sink to the bottom of the scale and the right arrows move to the center of the scale. Direction of motion is indicated by the arrow point. Sign changes would be signalled via the  $\pm$  symbols, which are displayed only when the sign actually changes between iterations. Thus, any erratic behavior patterns should be immediately evident as the optimization progresses.

Beneath each scale appears the constraint name and the allowable constraint magnitude. If the analyst changes the constraint tolerance during optimization, the trajectory sequencing will return to the guidance mode until all constraints are again satisfied within the specified tolerances. The total number of constraints currently imposed appears over the word CONSTRAIN in the lower left corner of this functional area.

The left side of each scale represents the correction required to null the constraint error normalized to the allowable tolerance; therefore, a trajectory completely satisfying the constraints shows a value of zero. The right side shows the sensitivity indicators DP/DPSI, which represent the change in payoff per unit change in the constraint. This scale runs from 0 to 2, and during guidance, the arrowhead represents the current value normalized to the value on the previous successful trajectory. During optimization the current value is normalized to that of the first successful optimization trajectory, unless the analyst rescales to the last successful optimization by touching the light button at the top of the scale. If the computed values on either side of the scale exceed the limits shown, the arrow is located at the top of the scale.

## 3.5 REQUIREMENTS SPECIFICATION FOR ASCENT BOOST OPTIMIZATION

### 3.5.1 General Guidelines

3.5.1.1 Objectives. The five objectives listed in Section 3.4.1 will be the objectives of the Ascent Boost Optimization Graphical-Computer System.

3.5.1.2 Nature of This Specification. This Specification sets forth requirements on the performance and capabilities of the system from the viewpoint of the user-analyst. As far as possible, it is independent of any particular piece of computer or display equipment.

3.5.1.3 System Operation. The system will be so designed that its operating displays and instructions will be expressed in the language of the user in that it will use the vocabulary of flight mechanics and engineering plotting procedures.

### 3.5.2 Performance Requirements

3.5.2.1 Sequence of Operations. This system will operate in the following sequence of steps as described in Section 3.4.3:

- Data preparation by the analyst
- Program read-in and initialization
- Computation of the initial nominal trajectory
- Data display in the REVIEW format:
  - Data correction
  - Recycle to initial nominal trajectory
- Computation of trajectories in the optimization cycle
- Concurrent display in the MONITOR format:
  - Interrupt and recycle
  - Data change and recycle
- Continuation to next case from stored input data
- Generation of new case from graphic input data



3.5.2.2 Inputs. Entries are as follows:

- Program. The system will read in from magnetic tape or cards either source or object decks of the PRESTO program suitably modified for graphics.
- Data. The system will read in from magnetic tape or cards all the input data necessary for program operation using the standard PRESTO data read-in subroutine. Data blocks, definitions, and formats are described in detail in the PRESTO manual (Ref. 3-6).

3.5.2.3 Outputs. Output is as follows:

- Printed output. The system will compute and tabulate all the trajectory and intermediate output from each trajectory iteration using the standard PRESTO output subroutine. Data, definitions, and formats are described in Ref. 3-6. In addition, the system will tabulate a separate output of only the final guidance trajectories that were computed. This can be done by transferring the standard final output as it is generated onto a second storage device.
- Graphics output. The system shall display the data, plots, alphanumeric information and optimization control indicators shown in Figs. 3-2 and 3-3 of Section 3.4.3.
- Photographic output. No photographic output is required.

3.5.2.4 Storage and Timing Requirements for Ascent Boost Optimization.

- Overall requirements.

(1) Storage – Exclusive of the resident monitor(s) for the computer(s) used in the system, the core storage requirement for operation without overlays or chain links is estimated to be:

Code words	33,000
Data words	<u>24,000</u>
Total	57,000 (CDC System words)

- (2) Timing – During the optimization cycle, the MONITOR display should be updated every 2 secs with the new trajectory information in less than 1 sec. The complete REVIEW display should appear in 6 secs.
  - (3) Data transmission rate – The time specified for updating the MONITOR display places the most stringent requirements on data transmission rates between the system hardware components where the trajectory is computed and the display information prepared. It is estimated a data transmission rate of 3,000 words/sec will be required.
- Detailed requirements. The computer storage requirements, reaction times, and data transmission rates have been estimated for the ascent boost optimization graphical-computer system, and are summarized in Table 3-4. They are furnished only as an indication of desired performance, and not to indicate the actual capability of any particular hardware. The sequence numbers in the table correspond to the steps described in Para. 3.5.2.1.
- (1) Storage requirements – The PRESTO computer program, including systems subroutines, currently requires 31,400 storage cells on the LMSC Univac 1108 Exec II System, using Fortran IV. This comprises 17,670 code, 2,200 data, and 8,600 common words, the balance is system routines. These figures were converted to CDC System requirements by assuming a one-to-one code correspondence, 30 percent larger system routines, and, since the CDC System requires two words to represent one Univac floating point word, the data and common requirements were doubled. Thus the estimated CDC core requirements become 17,600 code, 4,400 data, 17,200 common, and 3,800 system words for a total requirement of 43,000 cells.

Core storage requirements may be reduced by approximately 10,000 cells if an overlay technique is used. PRESTO was developed by LMSC for the NASA Langley Research Center under Contract NAS 1-2678. When delivered to Langley RC, the program had to be segmented into a chain link or overlay program in order to fit on their IBM 7094 computer, and is still being used in that configuration. The program may be divided into three overlays without seriously impairing computational speed. Column 4 of

Table 3-4

## STORAGE AND TIMING REQUIREMENTS FOR ASCENT BOOST OPTIMIZATION

Sequence Number	Step	Operation	Required Computer Core Size <sup>(a)</sup> (1,000 words)		Desired Reaction Times (sec)	Estimated Data Rate Requirements (words/sec)
			Code	Data		
1	Preparation	Analyst assembles and prepares inputs	-	-	-	-
2	Initiate	PRESTO program loaded and stored <sup>(b)</sup>	21.0	22.0	45-60	9,000 card source deck at card reader speed
		Punch data read in and stored	2.4	5.0	15	100 to 1,000 cards at card reader speed
		GRAPHICS initialized and decision panel displayed	10.0	2.0	3	-
3	Compute Initial	Compute initial nominal trajectory				
		• Data initialization	6.0	3.0	2	-
		• Trajectory integration	13.0	20.5		
4	REVIEW	REVIEW background displayed	10.0	2.0	3	1,000
		REVIEW data displayed	12.0	2.0	3	1,800
		Analyst edits and corrects errors	12.0	2.0	As necessary	100
		Regenerate initial nominal trajectory				
		• Data initialization	6.0	3.0	2	-
		• Trajectory integration	13.0	20.5		
		Update REVIEW display	12.0	2.0	3	1,800
5	Compute Sequence	Compute one backward and one forward trajectory in iteration cycle	13.0	20.5	1-2	-
6	MONITOR	Initial display of MONITOR background	10.0	2.0	1	2,500
		Update scope with data for current trajectory at end of compute sequence (every 1-2 sec)	12.0	2.0	1	3,000
7	Next Case	Retrieve next case data from stored input data	2.4	5.0	3	100
8	New Case	REVIEW display			As described above	
		Analyst generates data	12.0	2.0	As necessary	100

(a) CDC Computer System assumed. Data include common requirements.  
(b) Total program size shown for this operation.

Table 3-4 assumes utilization of this overlay mode, and the three links and their estimated core sizes in a CDC System are:

<u>Overlay Function</u>	<u>Code</u>	<u>Data</u>	<u>Total</u>
Data read-in/retrieval	2,400	5,000	7,400
Initialize case	6,000	3,000	9,000
Trajectory integration	3,000	20,500	33,500

Storage requirements for a typical graphics executive routine, a graphics display package and a control program directed by light button signals were estimated using currently operational LMSC CDC 3300 graphics packages.

- (2) Timing requirements – The pacing item in the ascent boost optimization graphics system is the computational speed of PRESTO. Currently, one trajectory iteration (backward and forward trajectories) requires less than 2 sec. When a forward trajectory fails, the next iteration does not compute another backward trajectory so the computing time is even less. In order to capitalize on the PRESTO capabilities, this speed should not be compromised.

However, it is apparent that the GRAPHICS system requires a finite time interval to display the trajectory information and the analyst needs time for recognition and reaction. Thus, the following sequence of events and desirable timing was developed:

Step 1 – Initial nominal trajectory computed	1-2 sec
Step 2 – Data transferred and displayed in REVIEW format	6 sec or less
Step 3 – Backward trajectory integration	} Computation in 1-2 sec
Step 4 – Forward trajectory integration	
Step 5 – Data transferred and displayed in MONITOR format	1 sec or less

The cycle from Step 3 to Step 5 is then repeated, resulting in an overall cycle time of 3 sec. This timing sequence assumes that the data transfer

and calculations necessary for the next display will not interrupt the current display on the console, such that the time interval with no data on the screen is minimized.

- (3) Data transmission rates – The estimated data transmission rate requirements shown in column 6 in Table 3-4 were generated by assuming that the data required for a particular display should be transferred from one hardware unit to another in one-tenth of the time allotted for the overall transfer-display operation.

### 3.6 CONCLUSIONS AND RECOMMENDATIONS

Digital computer-graphics systems have been devised to treat two analytical flight mechanics problems. These two problems are representative of a number of dynamics studies in the fields of flight, structural and orbit mechanics. Related applications would include, in part, such topics as recoverable booster flight profiles, sea-going tracking ship placement, planet surface coverage, interplanetary mission analyses, locating and designing spent booster vent ports, and analyses of frequency responses using s and z transforms and generating root locus and Bode plots. The features that relate the two problems which have been studied to all these applications are operation with graphical cues in an interactive man-machine mode, the ability to store console displays and to recall them rapidly for subsequent comparison and reevaluation, and the need for effective control of the system by means of light button instructions easily operated and understood by the user. Furthermore, both the interplanetary transfer computer program MAOT and the ascent boost optimization program PRESTO are widely used in NASA.

Recommendation. Therefore, it is recommended that both computer-graphics systems based on these programs be developed and put into operation as soon as possible.

Working with an actual graphics program is an invaluable experience. It provides virtually the only means of assessing the effectiveness of the human operator; it

furnishes data on his capabilities, his capacities for pattern recognition and data handling, and his speeds of response. It is the acid test of the features incorporated in the system; it quickly points up the inadequacies and superfluties in the software (such as choices of variables, displays, controls, and programming methods) and hardware (such as storage capacities, data transmission rates, operating speeds, and display size and clarity) and can thus guide more future program developments and equipment acquisitions. In a broad sense, it probes the program philosophy of graphics operations and shows where major changes in approaches and standards are needed.

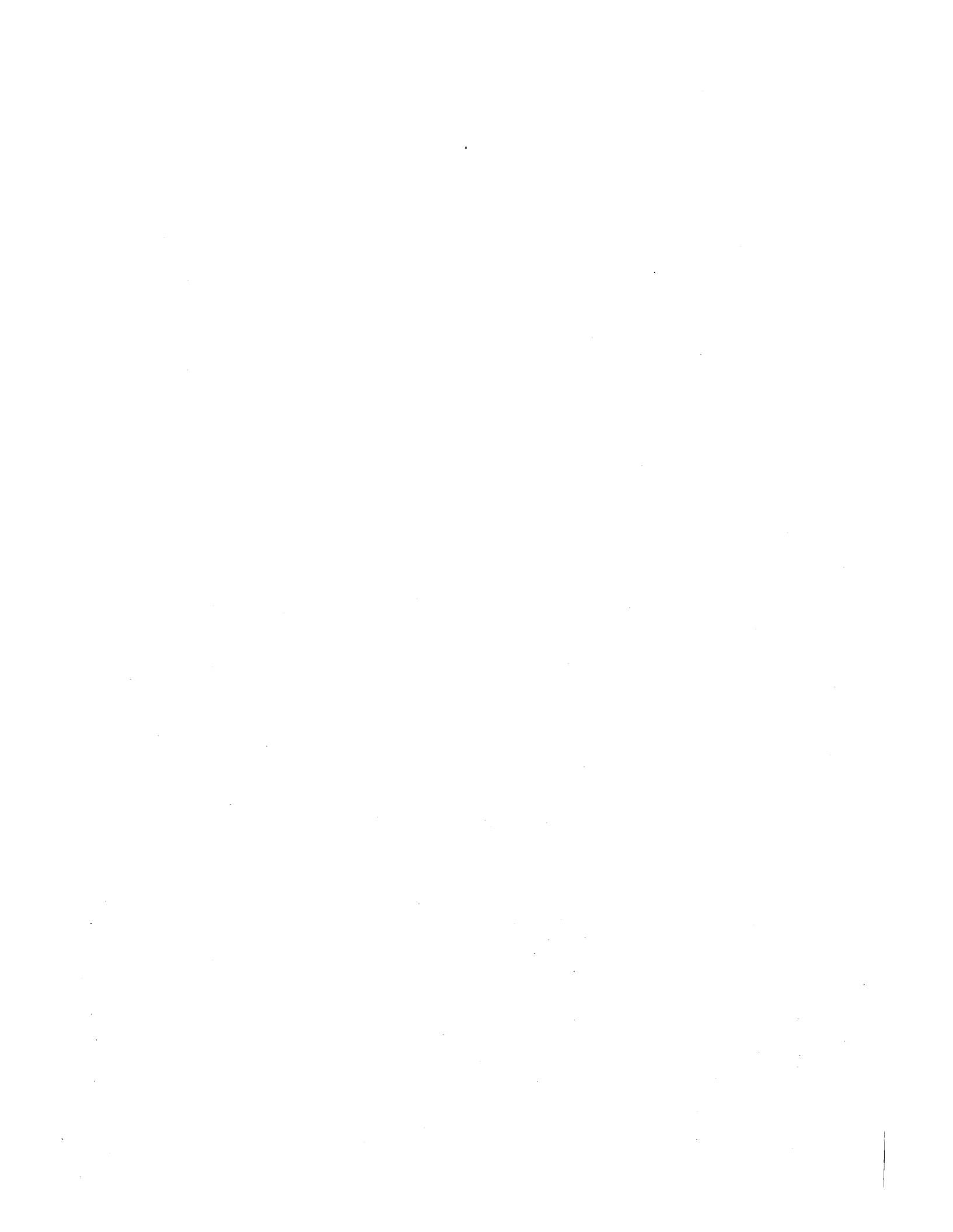
Detailed Requirements Specifications have been drawn up to define the sizes and operating speeds needed for the operation of the two studied systems. Code and data storage requirements do not appear to be so large as to exceed the capabilities of the Univac 1108 computer. However, the storage and recall requirements for high-speed data transmission between the graphics console and a remote large-capacity data storage device do seem to be one to two orders of magnitude larger than those presently being contemplated for the MSFC graphics installation.

Recommendation. Therefore, it is recommended that two methods be studied and compared for meeting these requirements, namely, that currently planned data transmission rates be increased to about 3000 words per sec or that a high-speed, large-capacity data storage device be added to the graphics satellite terminal, and that one of these methods be included in the MSFC graphics system.

### 3.7 REFERENCES

- 3-1 B. W. Boehm and J. E. Rieber, "Graphical Aids to Aerospace Vehicle Mission Analysis," Paper No. 67-897 presented at the AIAA 4th Annual Meeting and Technical Display, Anaheim, Calif., 23-27 Oct 1967
- 3-2 National Aeronautics & Space Administration, Space Flight Handbooks, Vol. 3, "Planetary Flight Handbook," Parts 1-3, NASA SP-35, Washington, D. C., 1963 (Contract No. NAS 8-5031)

- 3-3 National Aeronautics & Space Administration, Space Flight Handbooks, Trajectories to Jupiter, Ceres, and Vesta," Part 5, NASA SP-35, Washington, D. C. , 1966
- 3-4 -----, Space Flight Handbooks, "Mars Stopover Missions Using Venus Swingbys," Part 6, in press (Contract No. NAS 2-4175)
- 3-5 -----, PRESTO: Program for Rapid Earth-to-Space Trajectory Optimization, R. E. Willwerth, Jr. , R. C. Rosenbaum, and W. Chuck, NASA Cr-158, Feb 1965, also LMSC 4-36-65-1, 18 May 1964 (Contract No. NAS 1-2678)





## Section 4 STRUCTURAL DESIGN ANALYSIS

### 4.1 BACKGROUND

While the scope of structural analysis may be simply defined as the assurance of structural integrity of a structure immersed in a hostile environment, the actual ways and means of providing and assessing this assurance are many. The field of structural analysis abounds with "methods" (a blend of theory and magic). There exists no unified structural theory of practical value. Each structural problem does, in general, possess its own unique solution, based on a unique set of assumptions. Some idea about the variety and, sometimes, multiplicity, of solutions to structural problems may be obtained from Table 4-1, where a representative selection (not a universal listing) of computer programs used by the structural analysis department of one aerospace company is shown.

The list comprises programs which are used by "line" personnel in the solution of more or less routine assignments. It is this group of programs, and this group of persons who would benefit most from the Computer Aided Design (CAD) capability. It is questionable that the engineer engaged in scientific research exploration of structural problems would get any real benefit from CAD.

Notice, scanning the list of computer programs (Table 4-1), that with few exceptions they compute the behavior of a specified structure in a specified environment. They do not define a structure that would be adequate in a given environment, much less a best structure.

It appears that the main justification for implementing any of these programs with a computer graphic capability would be to change this condition, and let the computer system be a direct design tool, rather than an analysis machine. The ramifications

Table 4-1

REPRESENTATIVE LISTING OF STRUCTURAL ANALYSIS COMPUTER PROGRAMS

<b>1. STATIC STRUCTURAL ANALYSIS PROGRAMS</b>	
<b>1.1</b>	<b>Frame Analysis Programs</b>
1.1.1	WFRAME – WHETSTONE Program For Linear Analysis of Three-Dimensional Space Frames Subject to Concentrated External Loads
1.1.2	STRESS – MIT Program For Linear Analysis of Three-Dimensional Space Frames Subject to Concentrated or Distributed Loads, Support Motions and Temperature Effects
1.1.3	FRAN – IBM Program For Linear Analysis of Three-Dimensional Space Frames Subject to Concentrated or Distributed Loads, Support Motions and Temperature Effects
1.1.4	PFRAME – Program For Linear Analysis of Two-Dimensional Plane Frames Subject to Concentrated or Distributed Loads
1.1.5	GFRAME – Program For Linear Analysis of Two-Dimensional Grid Frames Subject to Concentrated or Distributed Loads Normal to Frame Plane
1.1.6	SFRAME – Program For Linear Analysis of Three-Dimensional Space Frames Subject to Concentrated Loads at Joints
1.1.7	GFLAC FRAME – Program For Linear Analysis of Two-Dimensional Plane Frames Subject to Concentrated or Distributed Loads
1.1.8	UBEAM – Program For Determining Internal Loads in Semi-Monocoque Structures by Unit Beam Method
1.1.9	BATMAN – Program For Linear Analysis of Three-Dimensional Space Frame/Panel/Plate Structures Subject to Discrete Applied Loads
1.1.10	CARR – Program For Linear Analysis of Three-Dimensional Space Frame and Panel Structures Subject to Concentrated and Distributed Loads

Table 4-1 (Cont.)

1.2 Shell Analysis Programs	
1.2.1	SADISTIC-I – Program For Elastic Analysis of a Bi-layered Orthotropic Conical Shell With a Non-Deformable Bond Layer Subject to Axisymmetric Surface Loads and Thermal Gradients
1.2.2	SADISTIC-II – Program For Elastic Analysis of a Bi-layer Orthotropic Conical Shell With a Shear Deformable Bond Layer Subject to Axisymmetric Surface Loads and Thermal Gradients
1.2.3	SADISTIC-III – Program For Elastic-Perfectly Plastic Analysis of a Multi-layered Orthotropic Conical Shell Subject to Axisymmetric Temperature and Pressure Loading
1.2.4	SADISTIC-IV – Program For Elastic Analysis of Multi-layered Orthotropic Shells of Revolution Subject to Axisymmetric Thermal and Surface Loads with Geometric and Load Discontinuities
1.2.5	SADISTIC-V – Program For Elastic Analysis of Multi-layered Orthotropic Shells of Revolution Subject to Asymmetric Loading
1.2.6	PEANUTS – Program For Elastic Analysis of Non-Uniform Thickness to Axisymmetric Loads and Thermal Gradients
1.2.7	EPSAD – Program For Elastic/Plastic Analysis of Isotropic Thin Shells of Revolution Subject to Axisymmetric Loads and Thermal Gradients
1.2.8	PLOCS – Program For Computing Elastic Stresses in Thin Cylindrical Shells Due to Localized Lateral Loads
1.2.9	NCCS – Program For Computing Forces and Displacements in Simply Supported, Non-Circular Orthotropic Shells Under Pressure Loading
1.2.10	SABOR III – Program For the Linear Elastic Analysis of Thin Isotropic Shells of Revolution Under Axisymmetric or Asymmetric Loading By Using The Matrix Displacement Method
1.2.11	KALNINS – Program For The Linear Elastic Analysis of Multi-Layer Orthotropic Thin Shells Subject to Symmetrical and Non-Symmetrical Loads
1.2.12	TSLD-2 – Finite Difference Program For Elastic-Plastic Analysis of Isotropic Shells of Revolution Subject to Axisymmetric Loads and Thermal Gradients

Table 4-1 (Cont.)

1.2.13	RMONIC – Program For Analysis of Axisymmetric Creep of Circular Cylindrical Shells in Axial Compression
1.2.14	THORNE – Program For Elastic Plastic Creep Analysis of Shells of Revolution Under Axisymmetric Loading
1.2.15	SHAZAM – Program For Computing Stresses and Displacements In Ring Stiffened Cylinders Subject to Non-Symmetrical Thermal Gradients
1.2.16	JOSE – Program For Computing Discontinuity Stresses at The Juncture of Shell Elements
1.2.17	COIC/P – Program For Computing Influence Coefficients and Fixed End Forces in Shell Elements For Use With JOSE
1.2.18	BERK – General Shell Analysis – Finite Element Method
<b>1.3 Shell Instability Programs</b>	
1.3.1	VERSION II – Program to Predict the Elastic Buckling Load of Cylinders Under Axial Compression Stiffened with Rings and Stringers of Rectangular Cross Section
1.3.2	GSAP – Program to Predict the Elastic Buckling Load of Cylinders Under Axial Compression Stiffened With Rings and/or Stringers of Arbitrary Cross Section
1.3.3	BOCONE – Program to Predict the Elastic Buckling Load of Truncated Ring Supported Cones Subject to External Pressure
1.3.4	SCAR – Program to Predict the General Elastic Instability Loads of Ring/Stringer Stiffened Cylinders Subject to Axial Compression and Lateral Pressure
1.3.5	BOC – Program to Predict the Buckling Load of Isotropic Cylinders Under Axially Varying Radial Pressure
1.3.6	BOOC – Program to Predict the Buckling Load of Orthotropic Cylinders Under Axially Varying Radial Pressures
1.3.7	BOCSIC – Program to Predict the Buckling Loads of a Laminated Cylindrical Shell Under a Combination of Axial Load, Bending Moment and Pressure

Table 4-1 (Cont.)

1.3.8	BOSOR – Program to Predict Buckling of Shells of Revolution With Various Wall Constructions Subjected to Axisymmetric Edge Loading, Axial Loading and Pressure
1.3.9	SALTS – Program to Predict the Creep Buckling of a Circular Cylindrical Shell Under Axial Compression and Internal Pressure
1.3.10	BARSIN – Program to Predict Classic Elastic Buckling Loads of Simply Supported Cylinders with Various Wall Constructions Subject to Internal or External Pressure and Axial Load
1.3.11	BENCYL – Program to Predict Elastic Buckling of Simply Supported Cylinders with Various Wall Constructions Subject to Radial Pressure, Bending Moments and Axial Loads Using Knockdown Factors For Buckling
1.3.12	INTACT – Program to Provide Interaction Elastic Buckling Curves for Simply Supported Cylinders of Various Wall Construction Subject to Radial Pressure Bending and Axial Loads Using Method of BENCYL Program
<b>1.4 Structural Optimization Programs</b>	
1.4.1	MARK III – Program for Structural Optimization of Axially Compressed Cylinders Stiffened With Rings/Stringers of Rectangular Cross Section
1.4.2	OPUS I – Program for Structural Optimization of Axially Compressed Cylinders Stiffened With Longitudinal Trapezoidal Corrugations and Rings of Angle Cross Section
1.4.3	MARK IV – Program for Structural Optimization of Cylinders Stiffened With Rings/Stringers of Rectangular Cross Section Subject to Axial Compression and Radial Lateral Pressure
1.4.4	OPVWHH – Program for Structural Optimization of Thin Walled Isotropic Cylindrical Pressure Vessels With Hemispherical Heads
<b>1.5 Solids of Revolution and Thick Shells</b>	
1.5.1	WILSON – Finite Element Program For Non-Linear Elastic Analysis of Orthotropic Solids or Thick Shells Subject to Arbitrary Thermal, Mechanical and Acceleration Loads

Table 4-1 (Cont.)

1.5.2	RAHFEP – Finite Element Program for Linear Elastic Analysis of Isotropic Solids or Thick Shells Subject to Applied Loads and Thermal Gradients
1.5.3	SINC – Program to Compute Elastic Stresses in Bodies of Revolution Subject to Axisymmetric Forces and Thermal Gradients
1.5.4	TKCYL – Program to Compute the Plane Strain Elastic Stresses in a Thick Walled Cylinder Subject to a Steady State Temperature Gradient Across the Wall
<b>1.6 Miscellaneous Programs</b>	
1.6.1	SSRING – Program For the Analysis of Circular Shell Supported Rings Loaded Normal to its Plane of Curvature
1.6.2	CBEAM – Program to Compute the Deflected Shape and Stresses In A Cantilever Beam with Varying Moment of Inertia
1.6.3	HERRMAN – Finite Element Program for Bending Analysis of a Plate of Arbitrary Shape and Thickness Subject to Non-Uniform Transverse Loading and Thermal Gradient
1.6.4	COMP2 – Program to Predict Mechanical Properties of Layered Continuous Filamentary Composites with Three Method Options
1.6.5	COMP3 – Program to Predict Mechanical Properties of Filament Wound Composites
1.6.6	TSISS – Program to Compute Elastic Thermal Stresses in a Series of Simple Structural Shapes Subject to a One-Dimensional Thermal Gradient
1.6.7	TSOC – Program to Compute the Plane Strain Elastic Thermal Stress in an Orthotropic Thick Cylinder Subject to a Radial Thermal Gradient
1.6.8	TSOLC - Program to Compute the Elastic Thermal Stress in Layered Orthotropic Thick Cylinder Subject to a Radial Thermal Gradient and Internal Pressure Loading
<b>1.7 Programs for Special Problems in Vehicle Re-Entry</b>	
1.7.1	SINC-STRESSES IN NOSE CONES – Program Analyzes a Finite Length Anisotropic Elastic Solid of Revolution Subjected to Axisymmetric Surface and Thermal Loads

Table 4-1 (Cont.)

1.7.2	INTERFACE STRESS – Program Computes the Stresses Developed at the Interfaces of Any Three-Layered System Due to an Applied Shock on the Surface
1.7.3	SWAP-8 – Program Describes Shock Interactions in a Multi-layered (up to 100), One-Dimensional System – Materials are Strain-Rate Dependent and Obey Elastic-Plastic Theory
1.7.4	TWIN LEAP – Program Computes by Finite Differences the Stresses in a Three-Layered, Semi-Infinite Slab Whose End Has Been Subjected to an In-Plane Velocity Step
1.7.5	LEAP CODE II – Program Utilizes the Finite Difference Method for Solving One-Dimensional Shock Wave Propagation Problems With Prescribed Initial and Boundary Conditions
1.7.6	MATED LEAP CODE – Program Utilizes the LEAP CODE II to Analyze a Nose Section as a One-Dimensional, Conical Bar and is Mated to the TWIN LEAP CODE to Determine the Effects of Coupling
1.7.7	SHAZAM – Program Determines the Elastic Stresses Which Occur in a Ring-Stiffened, Circular, Cylindrical Shell Which is Exposed to an Arbitrarily Varying Circumferential Temperature Field
1.7.8	REBEL I – Program Determines the RMS Values and Power Spectral Density of Acceleration Response of a Multilayered, Orthotropic, Conical Shell Frustum and an Attached Equipment Package to an External Random Pressure Field due to Boundary Layer Turbulance
<b>2. DYNAMIC STRUCTURAL ANALYSIS PROGRAMS</b>	
<b>2.1 Ring Programs</b>	
2.1.1	RING BOND – Program for Inelastic Dynamic Analysis of Impulsively Loaded Layered Rings With Finite Bond Strength
2.1.2	GIRLS I – Program for Inelastic Dynamic Analysis of a Dynamically Loaded Three-Layered Ring
<b>2.2 Shell Programs</b>	
2.2.1	GIRLS II – Program for Inelastic Dynamic Analysis of a Three-Layered Shell Loaded by an Axisymmetric Time Dependent Force

Table 4-1 (Cont.)

2.2.2	NESCO – Program to Compute Interface Stresses in a Three-Layered System due to an Applied Surface Shock
<b>3. MASS PROPERTIES ANALYSIS PROGRAMS</b>	
<b>3.1 Mass Property Programs</b>	
3.1.1	SUBMD – A Variant of the MIMKS Program for Submarine Mass Distribution
3.1.2	MMASS I – Program to Compute Flight Mass Properties of Multi-stage Missiles
3.1.3	MMASS II – Program to Compute Mass Properties of Multi-stage Solid Propellant Missiles as Function of Flight Time
3.1.4	STDDEV – Program to Compute Standard Deviation in Mass Properties Using the Monte Carlo Technique
3.1.5	SHAPE-MASS – Program to Compute Mass Properties of Various Shell Elements
<b>3.2 Miscellaneous Programs</b>	
3.2.1	AXIS TRANSFORM – Program to Compute Axis Transformation of Mass Properties
3.2.2	ORBIT – Program to Compute Classical Mechanics Orbital Parameters for any Attitude Range and Planetary Mass

of this on the economic and organizational aspects of any operation utilizing such a system is not part of the present investigation, but a few areas of interest in this respect will be mentioned:

- With the increasing complexity of structural analysis the functions of design and analysis have become more and more separable and specialized. Computer graphics may change this trend and a designer-analyst may evolve to replace the present designer and analyst.



- The direct man-machine link established by the computer-graphics system will allow the engineer to refine his analysis and/or design to a high degree. How is it established that this refinement is really worthwhile, or whether it merely means a waste of time? Or, in other words, how is the required product quality related to the quality of analysis and design?
- What are the psychological effects of the graphics program to structural design analysis on persons who are established in a noninteractive mode of operation?

A second justification for the use of computer graphics in structural analysis would be the control of input data. Some programs require a large amount of input data for description of loading and structure geometry. An example of this is the finite-element program BERK (Item 1.2.18 in the table) where the surface of the shell to be analyzed is broken down into a large number of triangular or rectangular surface elements, each of which has to be identified to the program. In this case a substantial reduction in time spent in checking and punching input data can be realized by letting the computing system take over this task.

While it is difficult to objectively assess the benefits of computer graphic techniques in structural analysis the following subjective list should suffice to give an ad justification of its use in the present situation:

- Rapid turnaround time
- Instantaneous display of results in an easy-to-digest form
- Control of the flow of the program
- Control of input data
- The ability of the man-machine team to let intuition play a role in the design-analysis process

The following paragraphs will attempt to visualize the actual steps or action that an engineer would take when solving a structural problem with the aid of a computer program with graphics capability. Three examples, all based on operating programs, have been chosen. They range from the fairly simple section properties program to the very sophisticated BERK finite-element program.

#### 4.1.1 Section Properties Program

The Section Properties program is used to calculate the area, moment of inertia and center of gravity location for an arbitrary cross section of a beam or bar. These calculations use up a lot of a structures analysts time, and the problem is particularly well suited for computer graphics application. There are two ways in which the program may be used:

- To calculate section properties for a given cross section
- To find the geometry of a cross section with a given set of section properties

Being a one-shot calculation, the first of the two alternatives would not benefit from the use of the computer graphics capability. The second alternative, however, will readily benefit from the direct communication between the engineer and the machine via the cathode-ray tube. This problem has an infinite number of solutions, and a "best" one can evolve gradually on the screen. The engineer would sketch the cross section on the screen and the corresponding section properties would appear in tabular form. By making changes to the displayed figure, a table of satisfactory section properties would finally ensue.

Thus, for this quite simple problem the engineer would require the following of the graphics system:

- Input Display: Two-dimensional, arbitrary geometrical configuration
- CRT Manipulations: Change geometry of displayed cross section (light pen) and rotation of figure
- Output: Table of section properties
- Other: Recall of previously made calculations for comparison with present calculations and (possibly) plotting of selected output values versus selected input variables

#### 4.1.2 Shell Program

Item number 1.2.6 in Table 4-1, PEANUTS (Program for the Elastic Analysis of Non-Uniform Thickness Shells), exemplifies the present state-of-the-art in structural analysis, as performed by the non-researcher in the field. This program, which analyzes a rotationally symmetric shell of non-uniform thickness with arbitrary boundary conditions, acted upon by arbitrary, rotationally symmetric loads, is being used by experienced engineers in fairly non-routine assignments. This is not a "cook-book" program; its use requires a thorough familiarity with the theory and practice of shell analysis.

The input to this program consists of:

- Description of geometry – meridional shape and thickness variation
- Material data
- Description of boundary conditions
- Description of loads

The input is either in the form of coefficients in analytical expression for the variation of thickness, etc. as a function of shell length, or in the form of point-by-point values for thickness, load, etc. There are some 90 pieces of information available for input (using the analysis-function option). A typical problem, however, requires only some 20-30 values to be input.

The output consists of about 30 quantities relating to the state of stress and deformation of the shell. These quantities are all a function of the location along the shell. Of primary interest to the engineer would be the six stresses (inner and outer meridional, inner and outer circumferential, and effective stresses) the deformations (rotation, and displacements), and the meridional variation of these quantities. Other output variables, such as stress resultants and strains, would be more infrequently regarded as of primary interest.

A typical problem for which PEANUTS is used is illustrated in Fig. 4-1. Here the problem was to design the transition region between a thick and a thin region on a

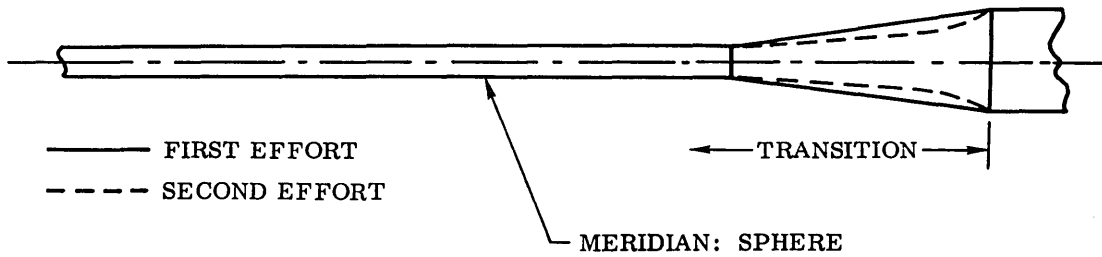
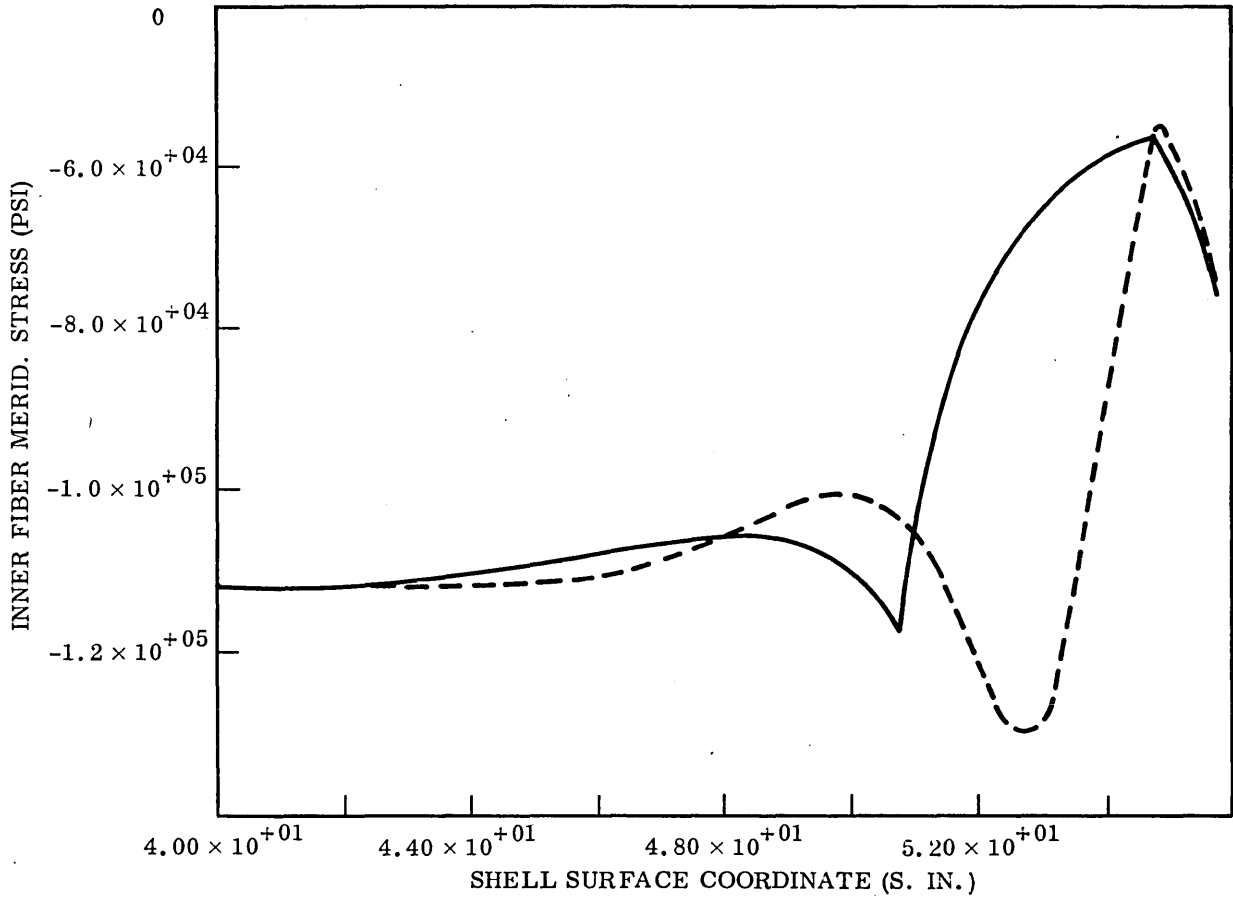


Fig. 4-1 PEANUTS - Structural Design Evolution

shell subjected to hydrostatic pressure. The illustration shows at the bottom the thickness variation of the shell (the curvature of the meridian is not shown) and at the top the corresponding meridional stress on the inside surface of the shell.

Having defined the necessary geometrical information (in this case, sphere) boundary conditions, material data, and load data, using numerical values (keyboard input) the engineer makes an initial "stab" at the transition region geometry; he assumes a linear variation in thickness between the thin and the thick region. This information could be communicated via a light pen or by specifying linear variation. After about one minute, the computer is through with the solution to the problem. The engineer now displays, one after the other, or perhaps two or three at a time, the various stresses which have been computed. He finds, by inspection, that the critical one is the inner fiber meridional stress. The display of this stress shows a sharp discontinuity at the beginning of the transition region, and a rather low stress in the transition region itself, indicating inefficient design. By use of the light pen (or by specifying coefficients in an algebraic expression) the engineer now attempts to improve on his design by changing the linear variation to a smooth curve (dashed lines in the illustration). This second effort did away with the sharp discontinuity and also utilizes the material better in the transition region. However, a stress peak has appeared in the neighborhood of meridional station 51 in., and a new modification of the thickness variation is necessary.

Thus, by making small changes on the displayed geometry and scanning the results of the ensuing computations the engineer gradually evolves a design that will satisfy his particular demands. The computing time required per case varies according to the computing systems. During this time (about one minute for 1108) the engineer will be occupied with studying the results of the previous calculation, making plans for the direction in which his effort will take, etc. (A waiting time of more than 10 minutes per case seems, subjectively, excessive for a graphics mode of operation, probably resulting in waste of time. However, the nature of the problem should decide how long waiting periods one can accept.)

To summarize, the requirements on a graphics systems for operating a relatively advanced structural program such as the PEANUTS would be:

- Input Display:
  - a. Two dimensional graphs for geometry and loading
  - b. Symbols for boundary conditions
  - c. Numerical values for loads, boundary conditions, and geometry (alterative to graphs)
- CRT Manipulations:
  - a. Change geometry of displayed figure by use of light pen, or by changing coefficients in algebraic formula
  - b. Same for loads and boundary conditions
- Output: Graphs of selected pertinent variables plotted versus meridional shell coordinates. Numerical values (bars) for weight of structure
- Other: Recall of previously made calculations and superimposing of plots, current and previous

A closer examination of the PEANUTS program is provided in Para. 4.6.

#### 4.1.3 Finite-Element Program

The application of finite-element analysis has lately been the subject of intensive research and can now be said to have reached a position where it can be applied successfully to analysis of a large variety of structural problems. One exponent of a finite-element structural program is BERK, number 1.2.18 in Table 4-1. This is a large capacity program, developed by professor R. Clough and Mr. P. Johnson at the University of California, Berkeley. The program is a completely general shell program, which at present is operational for static problems, but which by the addition of an eigensolution routine will also be capable of performing a modal response analysis for a very general class of shells. At present this program is not used for routine advanced engineering computations. One may regard it as representative for the next generation of structural analysis programs.

The basic feature of the finite-element method is that the actual structure is seen as an assemblage of a finite number of individual standard elements connected at a finite number of nodal points. The elements are constrained in such a way as to maintain continuity along the border lines of adjacent elements. The finite element solution will converge toward the true solution as the mesh (element) size is made smaller and smaller. We have here the main drawback and obstacle to the use of finite-element programs: the preparation of input data, which at best is tedious and time-consuming, e.g., each nodal point has to be located and its coordinates calculated. It appears that the use of graphic control offers an attractive solution to this problem; in combination with automatic grid generators for stereotyped grid patterns the CRT could be used for accurate control and generation of input data.

The requirements on the graphic system for the finite-element program, as seen by the engineer-user, would be:

#### Geometric Display.

- Three-view display of structure
- Blow-up capability of selected areas
- Rotation of structure about selected axis

(It is hard to visualize a genuine engineering requirement for three-dimensional display in this case.)

Geometry Description. Descriptive shell data, such as shape of meridians, thickness variations, etc., should be described by combinations of the following:

- Light pen
- Numerical values (in analytical functions)
- Selection of preprogrammed geometries (cones, ellipsoids, etc.)

Non-shell structural components (stringers, rings, beams, etc.) should be distinguishable from the shell and each other by clear symbolism. Descriptive data for these components should be input as numerical values (requirement for rotation; the structure about selected axis enables the designer-analyst to precisely locate these and other structural features, such as cutouts).

Loads. Various types of loads must be distinguishable from each other, e.g., a "dead" weight should be interpreted as such, and not as a normal pressure. Variation of forces over the structure should be described in a way similar to the geometric display. Loads to be considered are:

- Surface pressure
- Line loads, meridional direction
- Line loads, hoop direction
- Line loads, arbitrary orientation between nodes
- Point loads at nodes, any direction
- Moments: applied in manner similar to line and point loads

Stacking of Cases. Recall capability of data is mandatory.

Manipulation.

- Remove and add surface elements to design cutouts and adjust mesh size in selected areas
- Remove and add beam and bar elements (rings, stringers, reinforcements)
- Adjust meridian geometry
- Adjust shell thickness

Techniques of Output Display. The success of the graphic-aided design analysis stands or falls with the clarity and completeness of the displayed output data. In general, the output data will not exhibit the symmetry or regularity of the input data, making it perhaps necessary to employ some form of three-dimensional plotting. It is suggested that conventional 3D plotting is of dubious value to the engineer, but that the technique of carpet plotting may be useful (carpet plotting is quasi-3D plotting made by displacing overlaid plots horizontally). For time-dependent variables an animated display could be worthwhile, but not mandatory.

Based on the above discussion, the program specifications outlined in the following section have been developed. It is thought that they are satisfactory for application of computer graphics techniques to the solution of most structural programs in the next decade.



## 4.2 INTRODUCTION TO SPECIFICATIONS

In the following, general program requirements and specifications are developed for a Computer-Aided Structural/Design system (CASCADES) in which a wide variety of structural programs can be imbedded easily and operated efficiently. The program specifications are written so that program requirements can be made on a selective basis. The requirements are within the capabilities of available computer graphic systems.

Program operation is based on a COMMAND structure, i. e., the user initiates an operation by typing in a command, pushing a particular command button, or pointing to a command displayed on the CRT. This action will send coded signals to the computer which are decoded by the computer and the appropriate command is carried out. When a command has been completed, the computer idles until another command is given. If, in the process of performing the command, the computer program being executed requires a yes or no answer from the user, the computer will continually test the yes and no commands until a reply is sensed, at which time appropriate action is taken. While the program is waiting for the yes or no command it is operating and not in a wait or idle state.

For illustrative purposes, all commands are assumed to be displayed on the cathode-ray tube (CRT) and are selected by some type of pointing mechanism. Many of the commands could be labeled buttons or simply typed in. However, the pointing method (light pen) is preferred.

Program requirement specifications have been divided into eight subsections:

- Application Data Base
- Sign-On
- Program Selection
- Input
- Data Modification
- Computation
- Output
- CRT Layout

A brief section on hardware and software requirements is provided. Requirements for each section will be discussed separately and justified where necessary. All requirements are strictly program requirements but these do imply certain hardware and software requirements, consequently those hardware and software requirements are included.

When utilizing small computers with available memory smaller than that required for a particular application, it becomes necessary to modularize the application program. Structural analysis programs utilizing computer graphics as an I/O device can be separated into two basic functions: graphic operation and numerical computation. Numerical computations vary in type and complexity, dependent upon the particular problem, many of which are easily modularized.

The graphic operations (commands) required are dependent on the particular application program being utilized; however, a list of operations applicable to a wide variety of structural programs will be specified and justifications given where necessary. These operations can be treated as individual entities and, consequently, are easily modularized. In general, each graphic operation (command) is a separate program.

A successful computer-aided structural design system requires an application data base structure broad enough so each application program can access the data with minimum interface problems. Adequate tags (flags) must accompany the data to categorize it as to program, case, type of data, etc. (Can it be plotted, rescaled, etc?)

### 4.3 PROGRAM REQUIREMENT SPECIFICATIONS

The requirement specifications have been divided into eight units: Application Data Base, Sign-on, Program Selection, Input, Data Modification, Computation, Output and CRT Layout. This division of sections has been made to accomplish a logical chain of building blocks for accomplishing the tasks sketched out in the background section.

Some of the terms used in the specifications have specific meaning and may not be easily understood. However, the glossary in the appendix should help alleviate any problems in this respect.

A planned-for comparison between a structural design program operating in a graphic environment could not be performed due to unexpected problems which arose in converting the structural design program (Ref. 4-1) called SGOP to other computer hardware. Several of the key operations that allowed the user to go from one graphic operation to the next would not function properly and this limited its uses to single operations.

Even in its limited capacity, SGOP proved to be a valuable aid in developing these specifications. Many of the requirements that appear in the following sections already existed in SGOP to some degree. In some case SGOP capabilities were more extensive than needed for these specification.

#### 4.3.1 Application Data-Base Requirement Specification

Two types of data base are required for a general computer-aided structural design system. The first one, the graphic data base, is necessary to support each graphic console for display purposes. The second, which is discussed in this section, is needed to support the application programs.

The application data base must be flexible and easily modified to incorporate new data items into the data base. Also, it should be easy to change values once they are in the data base. To achieve this flexibility requires some sort of string processing

technique. Generally, the graphic data base possess these same requirements and could possibly be used for the application data base in limited problems. However, it was determined in a previous study (Ref. 4-1) that relying on the graphic data base only severely restricts the number of cases the user could retain.

It is often necessary to retain all cases computed, but it is unlikely that all cases will fit into the graphic data base. Also, for complex design studies, many application programs could be required in the design process and the output from several programs used as input to another program. Consequently, a large amount of data storage is required.

Since output from one program could be input to another program there should be no large distinction between input data and output data. Each should have a unique item name, program ID, case ID, date, and time, with date and time to be used in cases of ambiguity. A program name is included with the data so that two different programs can be used to generate different values for the same variable. This also allows for unique referencing.

With the above requirements, a rather large data base could be generated which would require auxiliary storage. Rather than repeating data for each case, a pointer should point to the data required from the requesting programs and also a pointer in the data set should point back to the programs requesting it so it is not erased accidentally from the data base. With this linkage one can thread one's way back from the final design and gather all data used in the design or computation process.

#### 4.3.2 Sign-On Requirement Specification

After all intrinsic operations have been performed to initialize the graphic console, the first things that should appear on the CRT is a library of application programs and a set of operating commands. There could be as many programs as will fit into the graphic system in the library. Each program should have a program ID and a short description. The programs appearing in the library can be interconnected; e.g., a

program that computes mode shapes can be related to a program that computes forced response. The program library should be accessible on demand, therefore a library command (LIBRARY) is required.

Various operating commands will be discussed in the following paragraphs. All of those will appear at sign-on or when the library is selected.

To enhance the utility of computer graphics it is necessary to be able to stop the graphic process at various steps and be able to resume at a later time, i. e., one hour, one day, or one month later. Consequently a STOP command capable of generating a restart tape is required. This restart requirement differs from a system restart when an unexpected system crash occurs. To achieve a general application restart capability it is necessary to have an application data base that can be copied to a permanent-type storage device and retrieved for later use.

#### 4.3.3 Program Selection Requirement Specification

Program selection should be made from the library of programs displayed on the CRT at SIGN-ON or when the Library is selected. Any one of the program commands should be selectable by the user. Obviously, if they are interrelated, the proper sequence of program selection must be maintained. When a program is selected, its ID should be displayed on the CRT.

Multiple executions of the same program are required during any session which depicts some type of case identification. The user should supply a case ID for each case which is displayed near the program ID for complete identification. The case ID is used in conjunction with the program ID to reference data items in the application data base.

#### 4.3.4 Input Requirement Specifications

For versatility, each item of data will be input on a selective basis with an arbitrary order of selection.

Since input requirements vary from one application program to another, a set of input requirements applicable to a wide variety of structural programs is tabulated in Table 4-2. Included in Table 4-2 is a set of data types to be associated with each input mode. The association is designated by an X.

Each input data item will be displayed on a predetermined area of the CRT (see Fig. 4-2). If all data cannot be displayed simultaneously, multiple displays will be required, each composed of predetermined areas. The actual arrangement will depend on the particular application.

#### 4.3.5 Data Modification Requirement Specification

One of the more powerful features of computer graphics is the fact that the user has the ability to change data rather simply via the light pen or whatever device is used (Ref. 4-2).

All input data should be easily modifiable and the new value or function displayed. If data from another case or program is to be modified as input to the current program that data should first be requested via input commands and then modified.

Table 4-2  
LIST OF INPUT REQUIREMENTS

	Single Item	1-Dimensional Arrays	2-Dimensional Arrays	3-D Arrays	Time Dependent
1. Card	X	X	X	X	X
2. Tape	X	X	X	X	X
3. CRT & Light Pen <sup>†</sup>	X	X	X		X
4. Data Base	X	X	X	X	X
5. Key Board	X	X	X	X	X
6. Trace		*	*		*

<sup>†</sup>Devices other than a light pen can be used. For a comprehensive list see Ref. 4-2.  
\*TRACE is limited to certain types of devices (see Ref. 4-2).

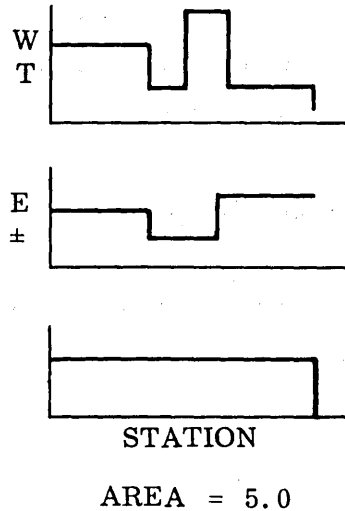


Fig. 4-2 Data Input Display

In order to modify data it must appear on the CRT. If the item to be modified is visible, it can be modified immediately. If the item is not visible it must be displayed using a display command (DISPLAY). This command will have options that allow the user to display many of the internal tables like the resident case table (RCT), etc. One option allows the user to display the input data for the case which appears in the Case ID Register. If a window concept is not employed by the system and the data item to be modified is not visible, a plot routine will have to be called using a plot command (PLOT).

Normal system software does not handle engineering scaling factors. Consequently, if scaled data points are to be entered onto a graph, a program must be developed to allow scaled data points to be entered. It would be acquired via a (XY-INPUT) command. Scaled data point input is not a requirement at this time.

A similar situation exists regarding rescaling of graphical data to extend or compress graph scales. Normal system software does not handle rescaling of graphs, only magnification, which is the same in both directions. To perform this function would require a rescale (RESCALE) command which accesses a rescaling program. Rescaling is not a program requirement at this time.

#### 4.3.6 Computation Requirement Specification

This portion deals primarily with precomputation requirements. It does not contain things like required numerical accuracies, core requirements, cycle time requirements, etc.

After all input data requirements have been satisfied, the user should be able to exercise the intended application program by selecting a compute command (COMPUTE). However, if the user wants to check the data for completeness or place the data in the data base without computation, two more commands are required: a Check Input Command (CKINPT) and a Check Input and Save Command (CKNSAV). Program specifications for the two commands will not be required.

When the compute command is selected, the input data to be used in the computations should first be interpreted and checked for completeness. It is then stored in the application data base and finally the computation is performed. If the data is not complete or interpretable, an appropriate message is displayed and the user is requested to make corrections before trying to compute again with the data.

The user should be able to stop computation at logical places, check answers, and elect to continue or stop. This requirement is strictly application-program-dependent and specifications will not be required. If such a capability is built into the numerical program, two more commands will be required: an interrupt command (INRUPT) which is displayed only upon entering the computation program and turned off when computations are complete, and a Stop and Go command (STOPNGO) which allows the user to continue or stop computations. The stop and go command should be displayed only



when the interrupt command has been selected and turned off as soon as the STOPNGO command has been selected. Obviously the STOPNGO command will have the two options: STOP and GO.

If the graphic operating system is a satellite of a large central computer which performs all numerical computations, it becomes necessary to prepare the data to be transmitted to the central computer. The simplest method would be to format the data coming from the graphic program to the numerical program as if the numerical program were reading it from a standard input device. This method would require little or no change to existing application program input. The data could be written onto an auxiliary device which could then be accessed by the central computer as a standard input device. The application program would possibly have to be modified since normal output cannot serve as input unless reformatted. Many application programs have a plot capability and data to be plotted is often written on an intermediate scratch file before it is read by a plot program which in turn generates a SC 4020-type plot. This intermediate scratch file could be on the same device as that read for input and the plot program replaced with a graphic program.

For a multi-console graphic system, each console would have its own I/O connection with the central computer. The central computer could sample the input device of each console and when the device was ready the data could be merged into the central computer input stream. The amount of input data per execution is normally small but the output is much greater. The output is normally written in small amounts at a time during execution of the program.

#### 4.3.7 Output Requirement Specification

Output values should be displayable on a selective basis or in a standard set with the capability of voiding output for the program and case that appear in the program and case ID registers. This requires an output command (OUTPUT) that the user can select.

The program specifications for plotting (displaying) the output is not a requirement since it is too dependent on the system software that is available.

Two dimensional plots of data are required and a capability of making crossplots of three dimensional data is also required. Actual 3-D displays will not be required since their utility is questionable and the amount of work to develop a general 3-D display routine appears too great. Development of a plot package as comprehensive as some of the SC 4020 software might become expensive. Such a plot capability is desirable for application programming but is not a requirement at this time.

Superimposing two or more graphs having a common variable is desirable and should be straightforward once the plotting capability exists.

#### 4.3.8 CRT Layout Requirement Specification

The CRT layout is dependent on the particular application and on the operating system. Since neither of these have been explicitly defined, it is not possible to specify the CRT layout in detail. However, the CRT layout requirements for the particular commands detailed in the previous requirement descriptions are presented. For explanation purposes a circular CRT is assumed, but the discussion applies also to a rectangular tube. The advantages and disadvantages of circular and rectangular tubes are not germane to this section.

The CRT is used for two purposes: display and communication. It is best to try and keep these two functions distinguished by physical separation on the CRT. Consequently, a region of the CRT will be for displaying data and another region for man-machine communication.

The communication region is frequently accessed by the user and should be in a relatively convenient location. When many types of communications are required they should be grouped, and the most frequently used groups should be more accessible than the less frequent ones. A natural place for frequently accessed commands would be on the right-hand side of the CRT (for right-handed people).

To facilitate comparing displayed data it is necessary to be able to align various plotted data. Also, when magnitudes of various functions are to be compared, they normally reference a common variable (or variables) and can be superimposed on the same graph. A rectangular display area seems convenient. A rectangle that is higher than it is wide (Fig. 4-3) makes it possible to align various graphs and also allows ample communication area on the right-hand side of the display area. If more communication area is necessary a more square display area can be used (see Fig. 4-4).

As the display area becomes more square it can be positioned more to the left allowing even more communication area on the right (see Fig. 4-5).

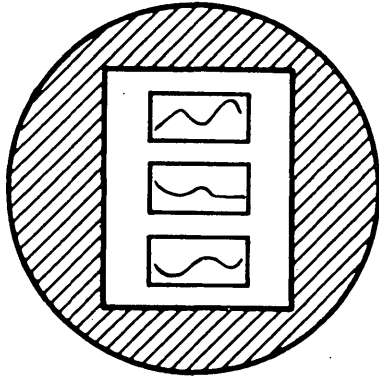
Therefore, depending upon the amount of communication area required and the amount of frequently accessed communicative area needed, a rectangular layout as in Fig. 4-3 with a square as a limiting configuration (Fig. 4-4) is a workable layout for the CRT. The symmetrical-type layouts are more pleasing aesthetically and could possibly be less fatiguing for the user.

Two types of communications are used: programs activated in the form of error messages and operating instructions for the user to read along with user instruction to the program.

Error messages and instructions for the user should be in the same area (grouped) for all applications so a user will develop a habit of glancing at this area frequently while operating the graphic system. Since no physical motion of the hands is required these messages can be located at a relatively distant place that is easy to glance at. A likely place is near the top of the CRT (Fig. 4-6).

If a large number of program-activated messages are required, the display will have to become more square as in Fig. 4-4 or, for exceptionally long messages, they will have to be displayed on the data display area.

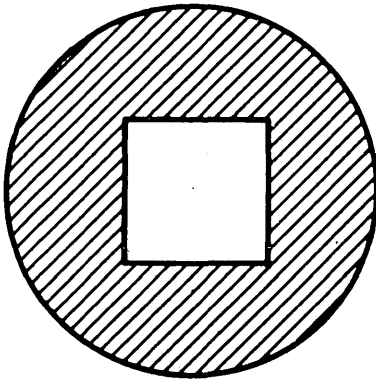
Data displays can be classified into two groups: those that can be modified and those that cannot. A modifiable display includes those quantities or functions that can be



LEGEND

 COMMUNICATION AREA

Fig. 4-3 Rectangular Communication and Data Display Layout



LEGEND

 COMMUNICATION AREA

Fig. 4-4 Square Communication and Data Display Layout

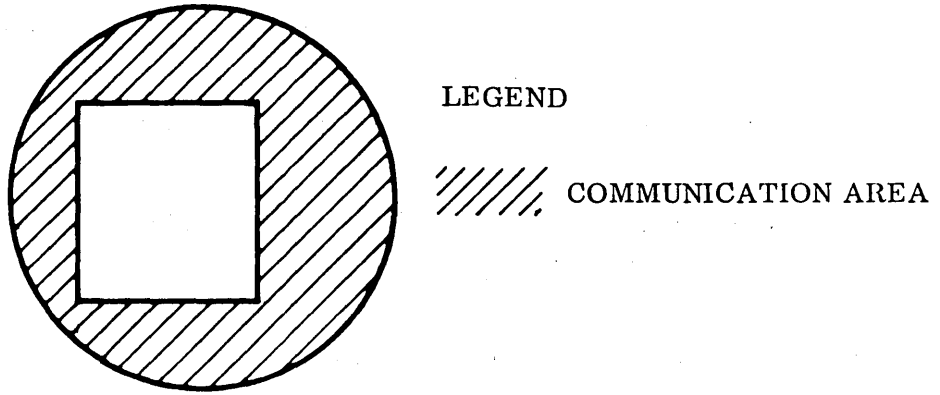


Fig. 4-5 Shifted Square Communication and Data Display Layout

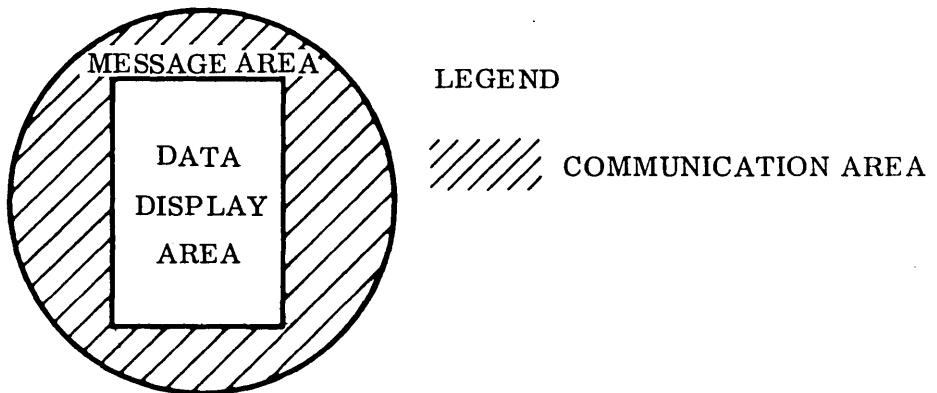


Fig. 4-6 Communication Message Area Layout

changed by the user when such action is desired. If numbers or functions are displayed for viewing purposes only they can be presented in a compact form, whereas data to be modified should be placed further apart for easier selection. The same piece of data can fall into both groups for continuity since as much of the input is displayed at one time as is possible. Also, for completeness and interpretation, particular input quantities which normally are modifiable should be displayed along with a set of computed values. Since the CRT is limited in size, it is frequently not possible to display all the input and output data simultaneously. It then becomes necessary to pick and choose those quantities that should be displayed together.

Four basic frame types are shown in Fig. 4-7: Library, Input/Output Selection, Input Display and Modification, and Output.

When large amounts of input and output are required and all cannot be displayed on their respective frames, multiple displays are necessary. If a window concept is used the frames can be separate with the window positioned to the proper frame. However, if the window concept is not available, each frame will have to be generated using some type of erase and plot routine.

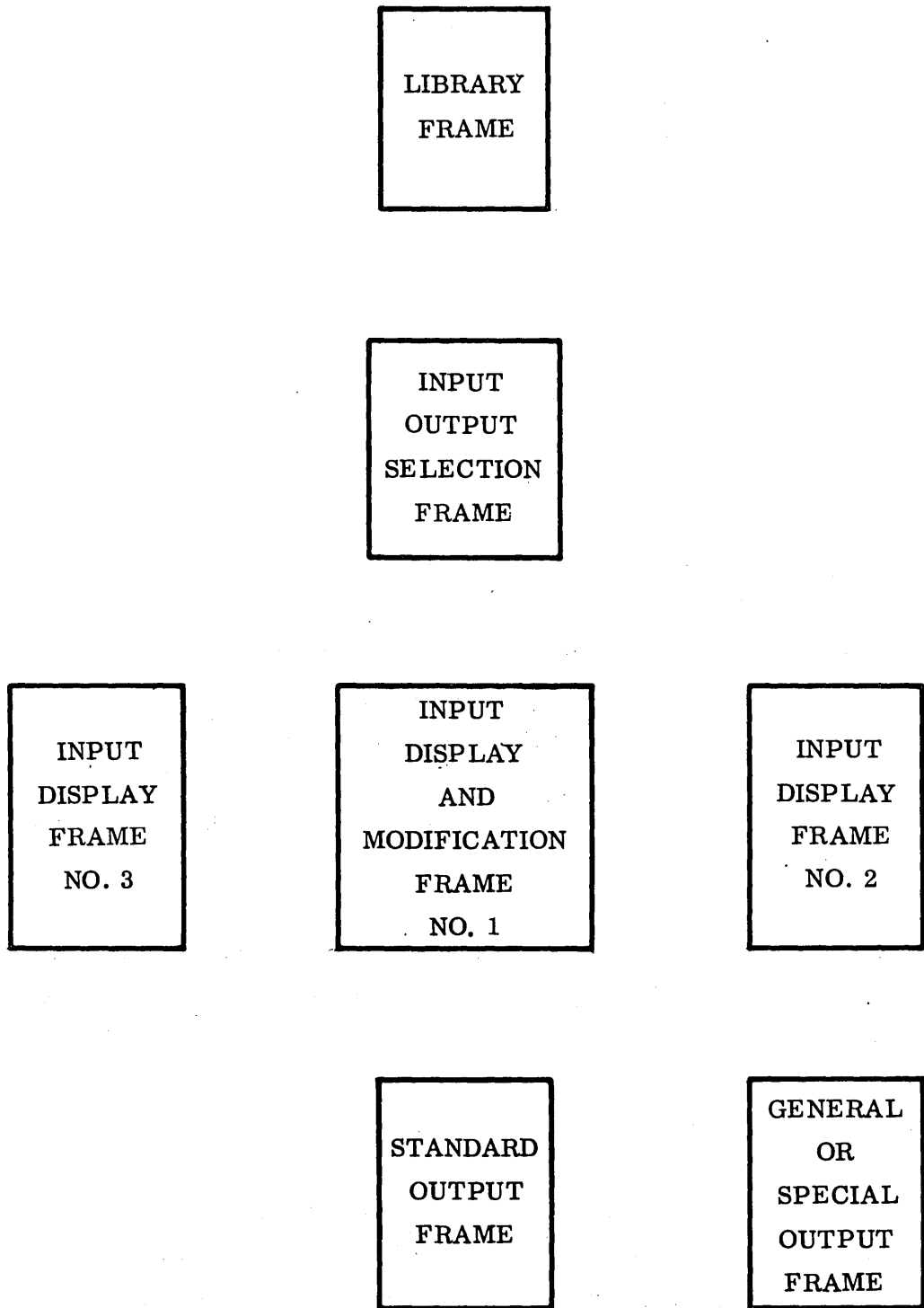


Fig. 4-7 Frame Types

#### 4.4 HARDWARE/SOFTWARE REQUIREMENTS

Some hardware/software requirements are briefly mentioned here to give an idea of the general system configuration. Core size, memory cycle time, etc., are not included.

Clearly, the basic operating system requires a type of digital computer system that has a cathode-ray tube (CRT) as an I/O medium. The CRT requires a device for man-machine communication; a light pen, Joy Stick, Rand Tablet, etc., (see Ref. 4-2). Card I/O as well as some type of auxiliary storage devices are required.

Manipulation of line drawings and BCD text displayed on the CRT is required. The operator must be able to select items that appear on the CRT. When an item is selected by the user, the application program must be able to recognize the item and have access to it. For discussion purposes assume that the item selected is placed in a so-called Pick Table which is assessable by application programs. Using the contents of the Pick Table the application program should be able to search for and retrieve the selected item. (Call this a fetch capability.)

It is necessary that the graphic system be able to store displays for later use or additional application software will be required.

Application restarts require the capability of storing the contents of the CRT onto an auxiliary device for later recall. If this does not exist, additional application software is required.

#### 4.5 PROGRAM SPECIFICATIONS

Program specifications are presented in the same form as the program requirements. There are eight sections: Application Data Base, Sign-on, Program Selection, Input, Data Modification, Computation, Output and CRT Layout. Hardware specifications are not presented in this task. Each section corresponds to a requirement specification section.



These specifications are detailed enough to present program flow and to depict the relative complexity of each section. More detail can be added in many of the sections but it is felt that they would complicate the overall picture.

#### 4.5.1 Application Data Base Specification

The Data Base Structure (DBS) is composed of four elements, a Data Base Directory (DBD), a Data Requirement Table (DRT), Program Data Reference Table (PDRT) and a Data Set (DS). The DBS (Fig. 4-8) has been formulated in a semi-string fashion but could be formulated in terms of fixed length tables for each element. Presently some of the tables are fixed length and the rest have individual string formats.

Each table is separate and has its own access program and table format depending on how frequently it is accessed and the best way to conserve storage. Since the application programs to be incorporated into the system are not specified, the table formats are only suggested and are considered optimum rather than best choice.

The individual access programs will be entered from a Data Base Access Program (Fig. 4-9). This program loads in all the necessary Data Base Tables used in accessing data items in the data set and controls the sequence. It is a type of supervisory program.

Each table access program will perform three functions: read data out of table, place data into table, and remove data from table. An access program which uses relative addresses would allow the table to be loaded anywhere in core and could be easily coded in a Fortran type language using arrays. A dynamic allocation scheme for the tables could be implemented in a Fortran type language by having one large array into which the tables are loaded starting at any convenient index. The starting index would be the base address.

The Data Base Directory consists of three elements: Resident Name Table (RNT), Program ID Table (PIDT) and Case ID Table (CIDT). These elements are used to locate data items in the data string. The tables are constructed in a string format for

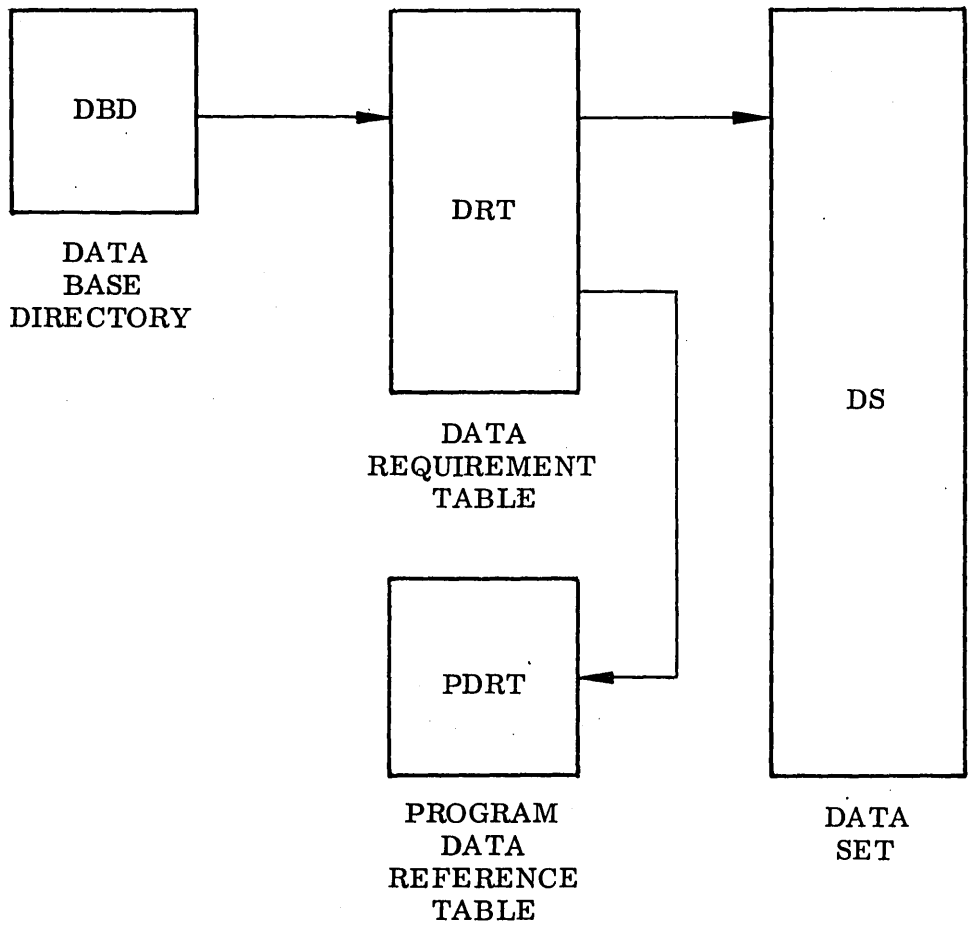


Fig. 4-8 Application Data Base Structure

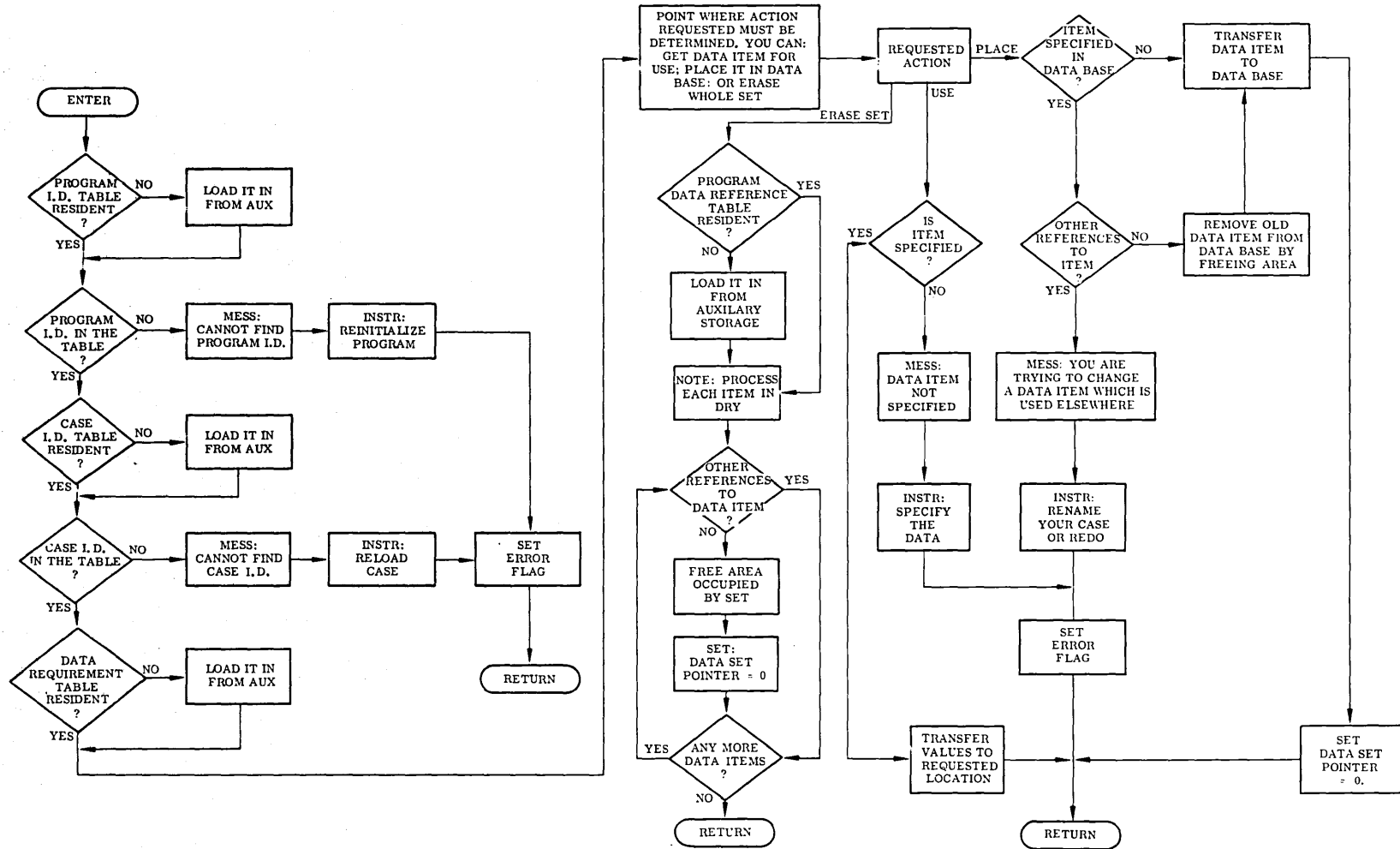


Fig. 4-9 Data Base Access Program

flexibility. The number of computer words used for each name in the string depends on the word size and the number of characters desired. For discussion purposes we will use three words. One word is used for the name (identification), one for the pointer (address to the name), and a third for a pointer (address) to the next item in the table.

The Resident Name Table (RNT), Fig. 4-10, contains the names of all elements in the Data Base and indicates if it is in core or on auxiliary storage. If the element is in core the RNT contains the core address.

The Program ID Table (PIDT), Fig. 4-11, contains the name of each program that has been exercised, a pointer to the Case ID Table (CIDT) for that program, and a pointer to the name of the next program in the table.

The Case ID Table, Fig. 4-12, contains the name of the case, a pointer (address) to the Data Requirement Table (DRT) for the case, and a pointer to the next case in the CIDT.

The three tables, RNT, PIDT, and CIDT, also form a string which can reside in core if space is available. If not, a paging scheme could be employed which would bring in part of the string at a time. This string is traversed in one direction only, starting with a program ID and case ID down to the Data Requirement Table (DRT) pertaining to the particular program and case ID.

The Data Requirement Table (Fig. 4-13) is of fixed length for every case of a particular program. Each data item in the DRT has a name identical to the name used in the application program and a pointer (address) to its value or set of values in the data base. The value or set of values is called the Data Set (DS). Associated with each data item is a pointer to a string of Program ID's that reference the data item along with a word indicating input or output item, variable mode, modifiable, etc. The reference string is called the Program Data Reference Table (PDRT). Other string pointers and indicators can be included in the DRT as required.

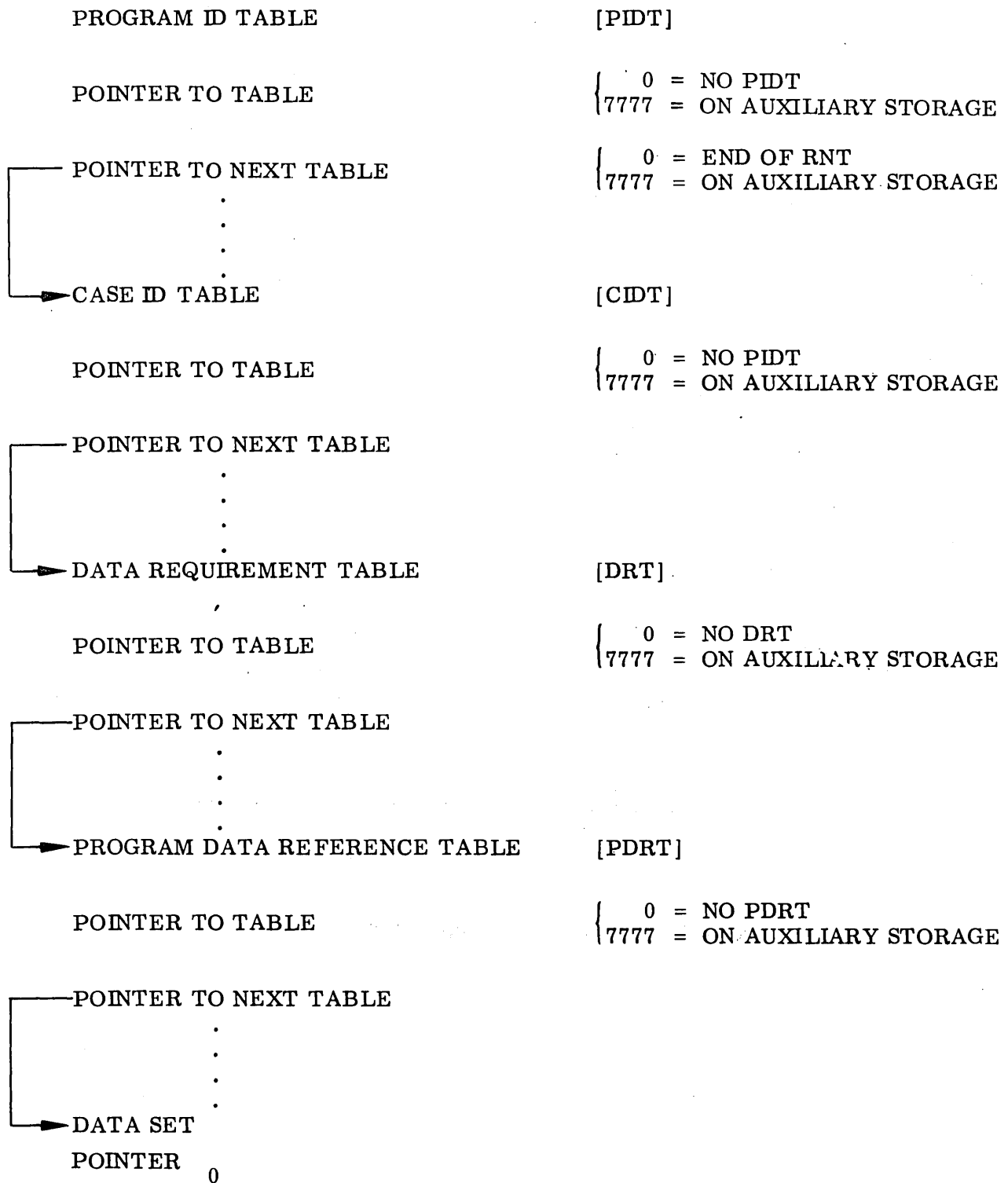
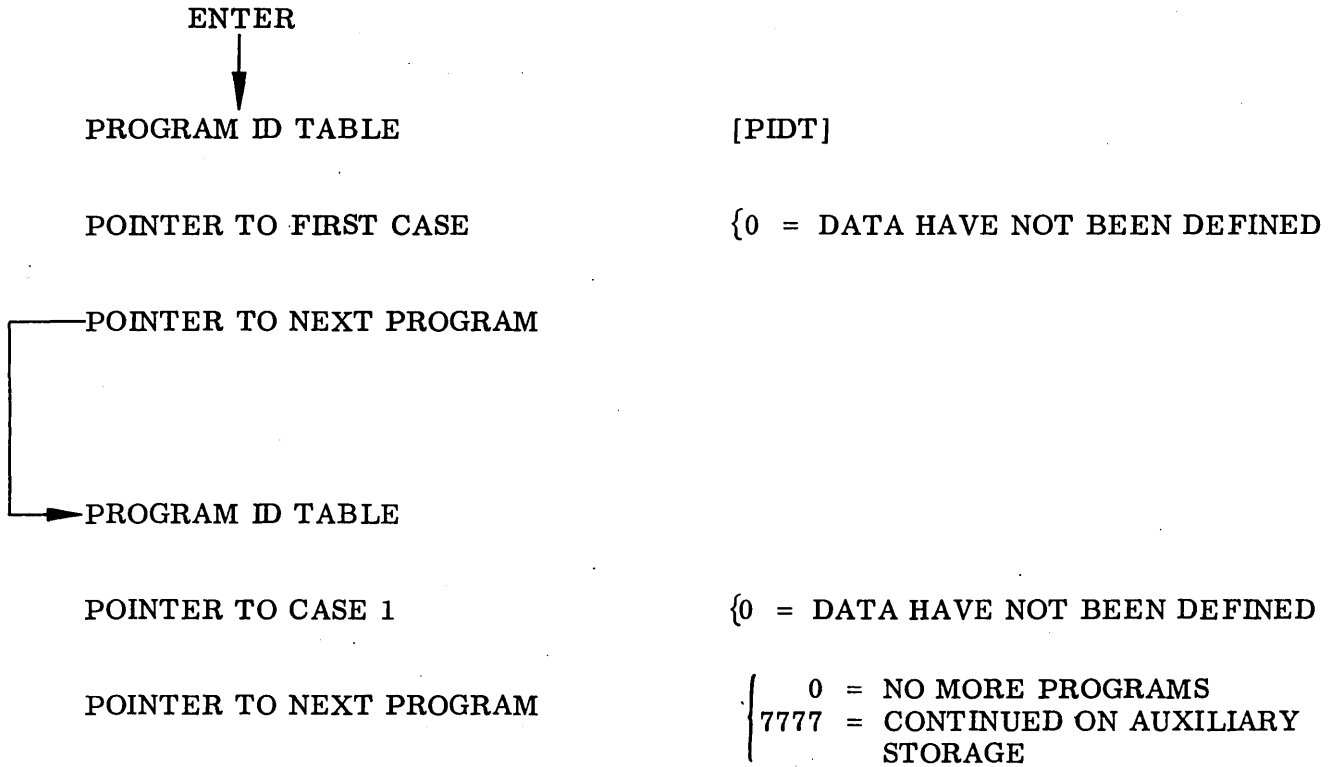


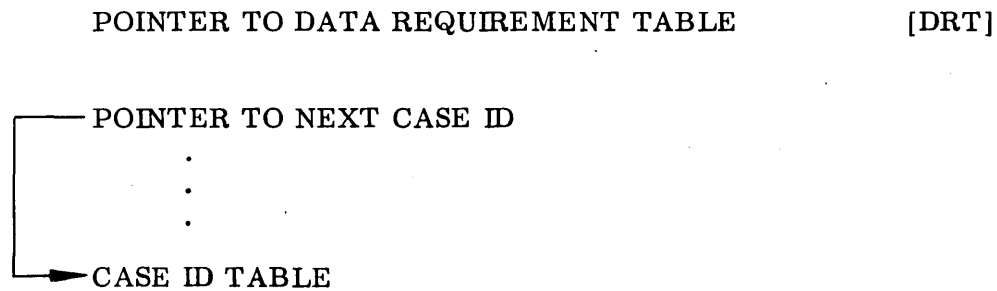
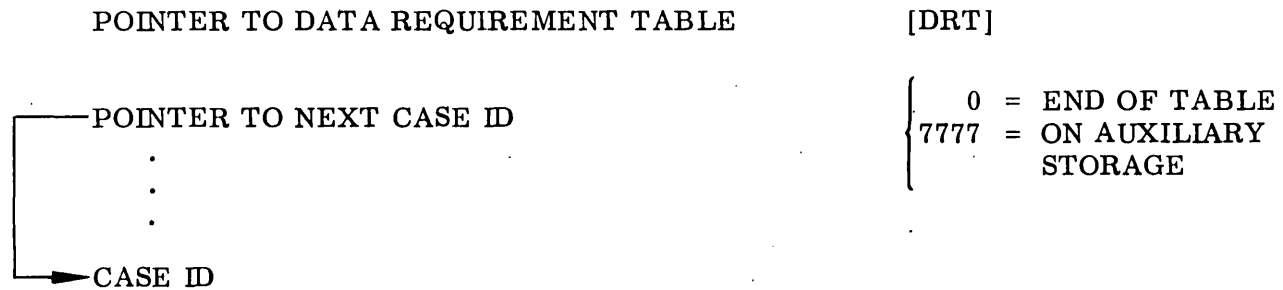
Fig. 4-10 Resident Name Table [RNT]



NOTE: The items shown here reside in Auxiliary Storage until the program is selected if they cannot all fit into core.

Fig. 4-11 Program ID Table [PIDT]

CASE ID TABLE



POINTER TO DRT

0

Fig. 4-12 Case ID Table [CIDT]

NUMBER OF DATA ITEMS

ITEM<sub>1</sub> [VARIABLE NAME USED IN NUMERICAL PACKAGE OR IN INPUT DESCRIPTION IF SO]

POINTER TO TABLE OF PROBLEMS } 0 = USED ONLY BY ITS OWN CASE  
USING ITEM<sub>1</sub>

POINTER TO DATA SET { 0 = NOT SPECIFIED  
7777 = ERASED (NOT REALLY  
NECESSARY)

ITEM TYPE WORD

ITEM<sub>2</sub>

POINTER TO TABLE OF } 0 = USED ONLY BY ITS OWN CASE

POINTER TO DATA SET

ITEM<sub>2</sub> TYPE WORD

ITEM<sub>3</sub>

.

.

.

ITEM<sub>N</sub>

.

.

.

Fig. 4-13 Data Requirement Table [DRT]



Since the Data Requirement Table is fixed in length for each program, the table format is contiguous and not a string. The first word in the table is the table length. It might be necessary to make the table format into a string compatible with the other string forms once the code is developed.

Some of the reasons for making the DRT contiguous are that it saves space, it is not modified once loaded, and it does not have to be removed from the data base frequently.

The Program Data Reference Table (PDRT), Fig. 4-14, contains a Program ID, Case ID, Date Item First Referenced, Time Referenced, and a pointer to the next Program ID in the Program Data Reference Table. The PDRT is in string format but the number of words used can vary depending on computer word size and the number of characters allowed for the Program ID, Case ID, date, and time. Four words will be assumed for discussion purposes.

The data set is the end item in the string. The format should be general enough to include data sets for a wide variety of application problems. However, to develop a general-purpose format has the pitfalls of any general-purpose program, usually ending up too complicated to be practical. A detailed requirement for the type of data to be included in the data base is needed; for instance, "Does it require complex numbers, double precision numbers, time dependent arrays, etc.?" Of course, the type of data depends on the application programs that will access the data base. The data set format can not be specified until the application programs that access the data set are specified. For completeness a sample data set format is shown in Fig. 4-15.

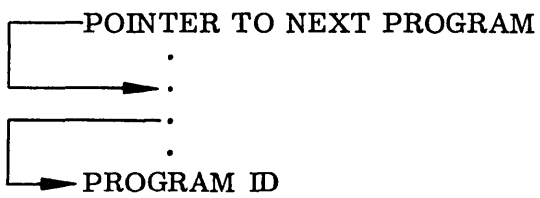
Two more words per table are required. The first word in each table contains the allotted length of the table, i. e., the amount of continuous core available. The second word contains the relative address where the first available location is in the table. The absolute address of the first word in the table is the pointer used to get to the table.

PROGRAM ID

CASE ID

DATE

TIME



{ 0 = END OF STRING  
7777 = CONTINUED ON AUXILIARY STORAGE

CASE ID

DATE

TIME

0

Fig. 4-14 Program Data Reference Table [PDRT]

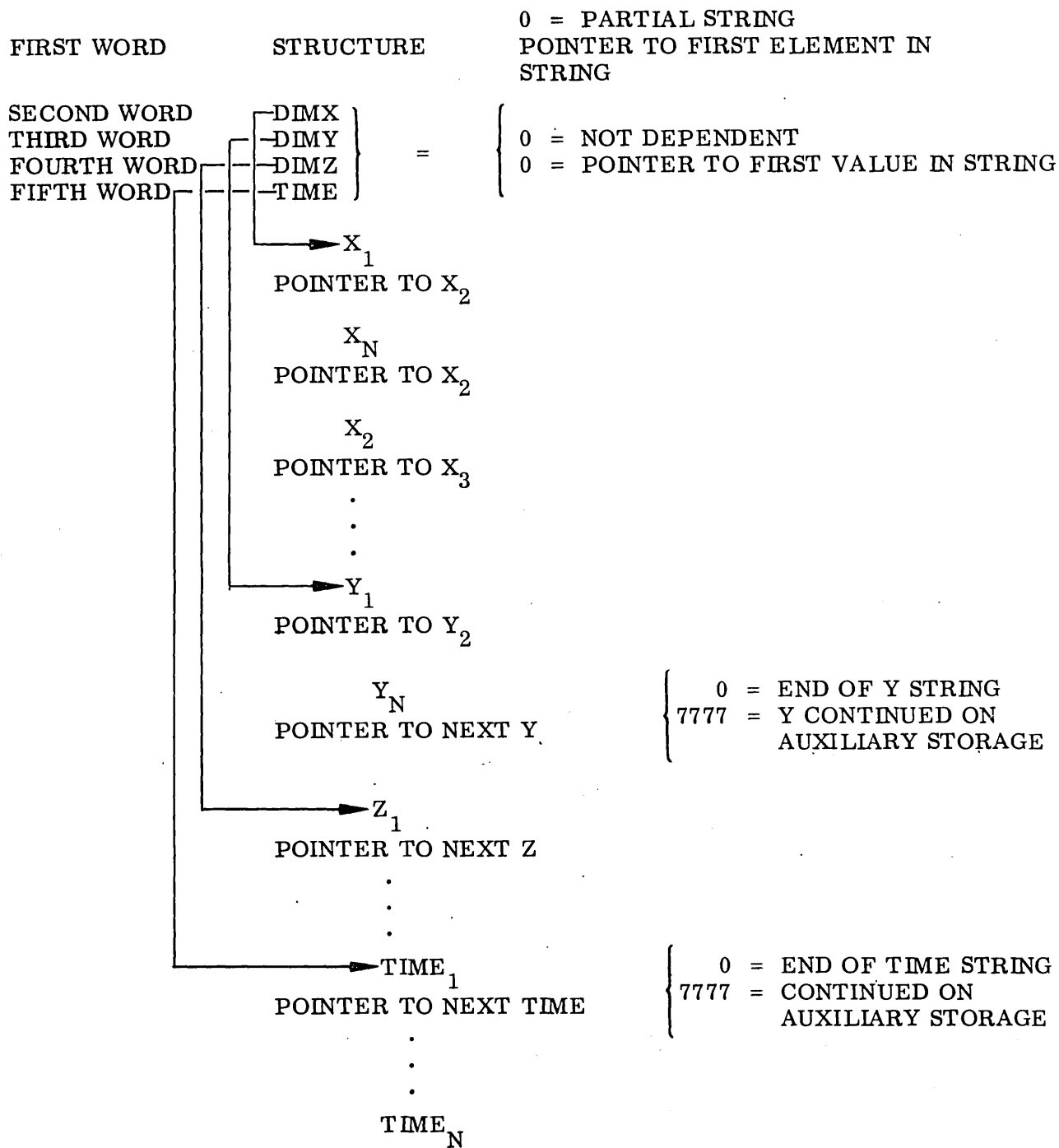


Fig. 4-15 Sample Data Set Format

#### 4.5.2 Sign-On Specification

The way in which a sign-on is implemented depends on the software capabilities of the graphic system being used. Two types of sign-on conditions are possible: a fresh start or restarting from a previous session. For a fresh start, the library must be displayed, all commands displayed, data files opened, and registers cleared. For a restart condition various compatibility checks are required, data files opened, restart device checked, and registers set with proper values. Assume that the display console has been reloaded on a restart condition and that a fetch capability exists in the system. To distinguish a restart from a fresh start the graphic data base will be searched for key items. Therefore, various restart conditions can be accommodated. The primary item to be searched for will be a code word which was set when the stop command with restart option was selected.

Two stop commands are used: one that saves the application data base, called the stop command (STOP), and another which transfers the graphic data base to a permanent-type storage device and then performs a normal stop operation. This command is called STOPNSAV. At present, one program performs both stop operations.

One sign-on per session is all that is necessary, however, the user may want to exercise several programs during one session. Consequently, the library must be available to the user with a library command (LIBRARY). When the library command is selected, a library program is entered (Fig. 4-16) which saves output from the last case and then enters the sign-on program automatically. Program flow for sign-on and stop programs is delineated in Fig. 4-17 and Fig. 4-18, respectively.

#### 4.5.3 Program Selection Specification

Each time a program is selected from the library, the corresponding program ID is obtained and displayed in the program ID register (PROG. ID). The case ID will be input via a LABEL register and checked against existing cases. If the case ID is unique it is displayed in the case ID register and initialization takes place. If it is not, a message is displayed as a warning.

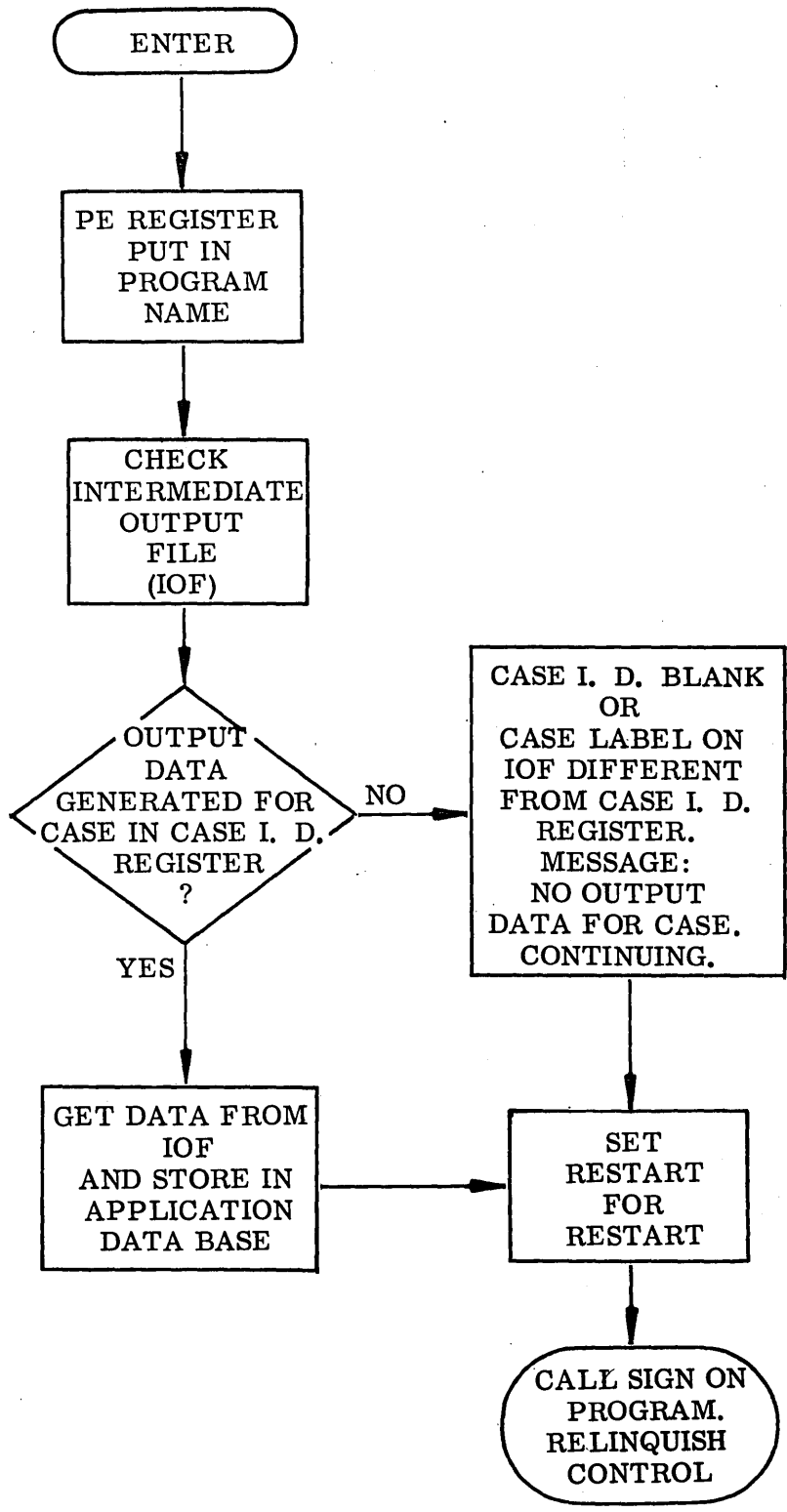


Fig. 4-16 Library Program

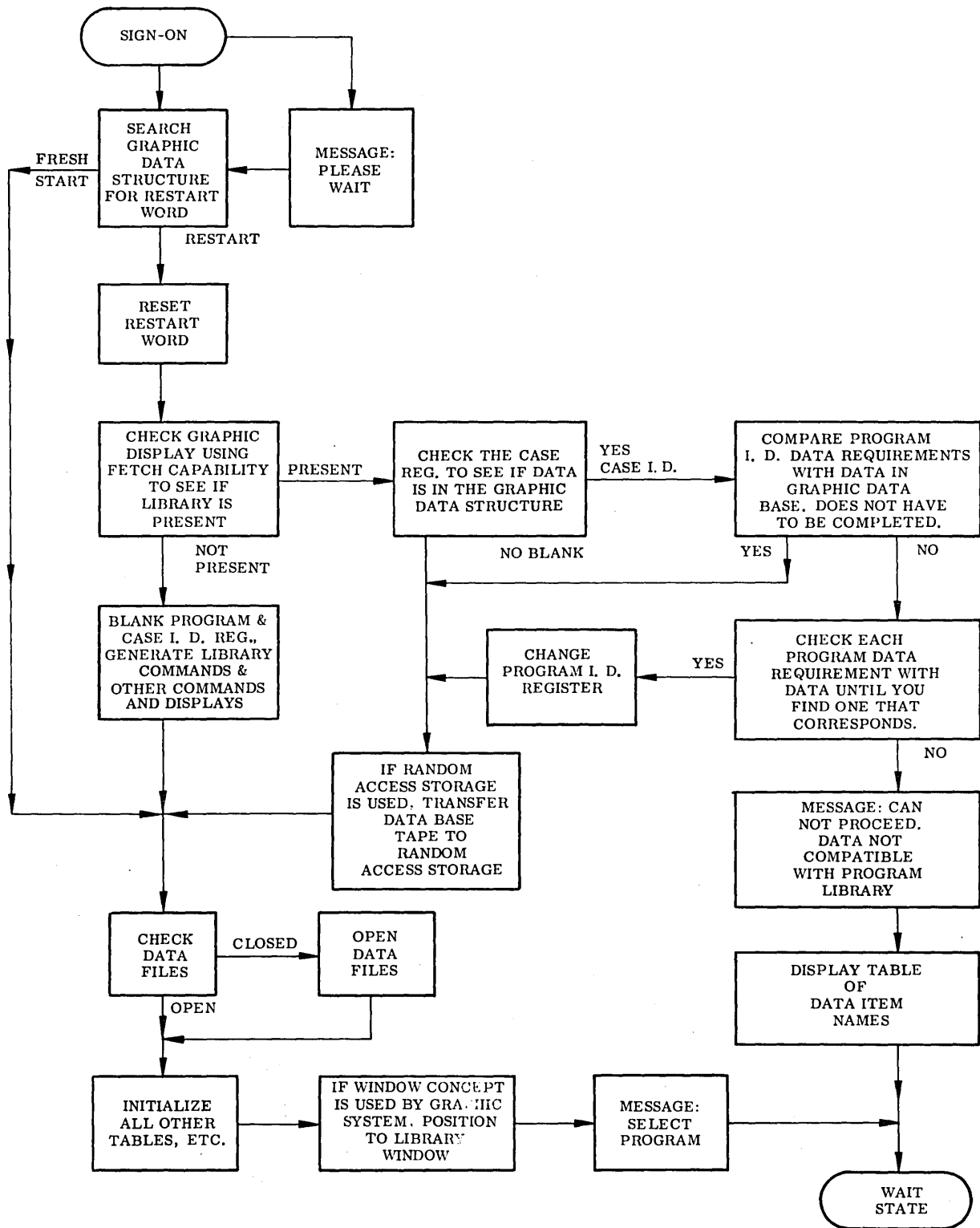


Fig. 4-17 Sign-on Program

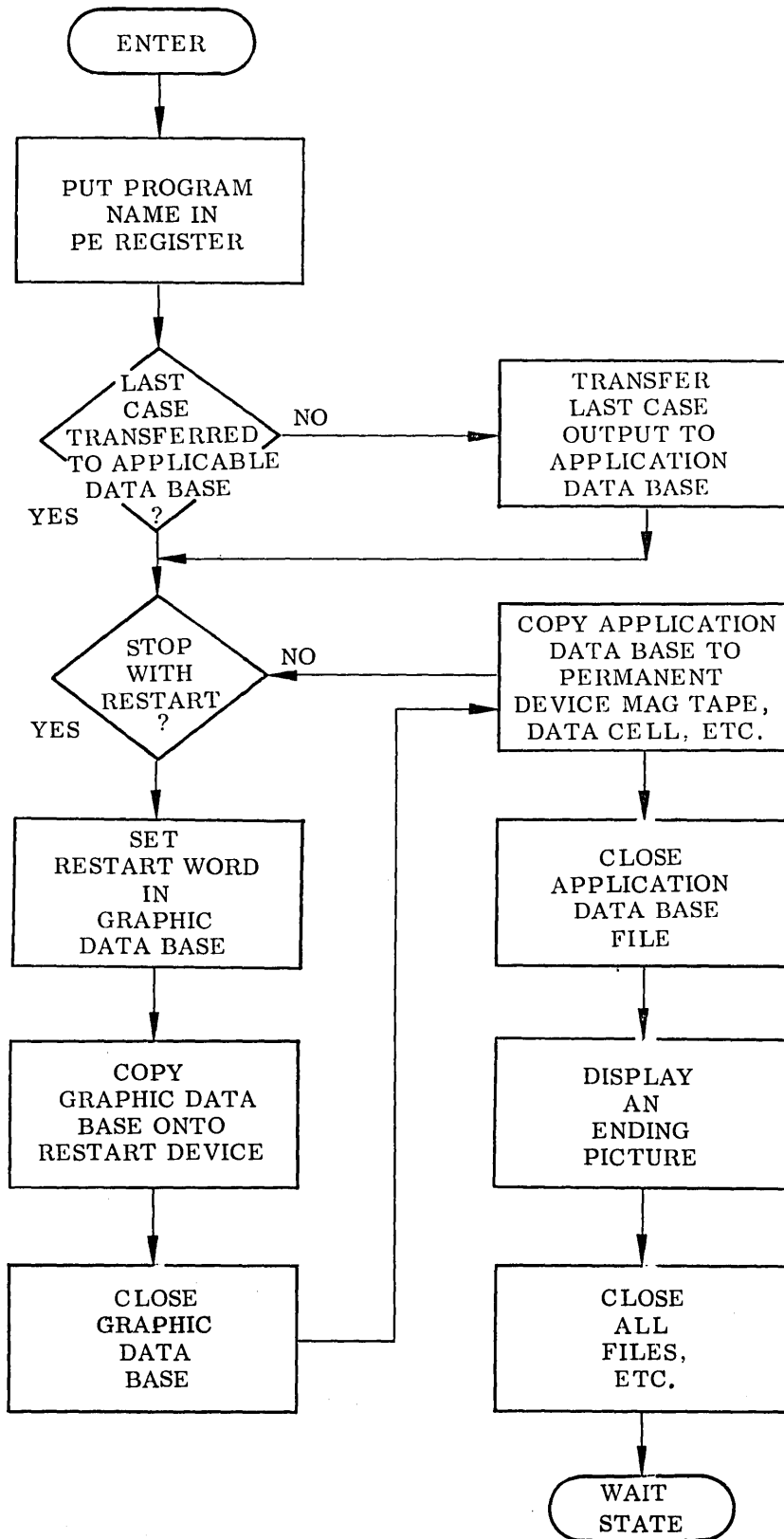


Fig. 4-18 Stop Program

The next case for the same program will be initialized by selecting a next case command (NXCASE). When the NXCASE command is selected, output data from the previous case is placed in the application data base and control is passed to the program selection program. Program logic is shown in Figs. 4-19 and 4-20.

#### 4.5.4 Input Specification

Input data is requested by selecting the input command. When input is selected, a list of input data requirements for the program whose ID appears in the program ID register will be displayed on the CRT in the form of commands. Included in the list will be a command to load all data. The user will select from the input list one item at a time in any order. Of course, if one item depends upon another the independent one must be selected first; if not a message is displayed. After the item is selected the user must designate the input mode: card, tape, etc. The mode can be any of those that appear in Table 4-2 except for command ALL DATA for which CRT input does not apply. Input data flow for each mode of input is shown in Figs. 4-21 and 4-22.

To effect data input, a set of routines will be required. One will be the main input program (Fig. 4-23) that will check status of the input requirements, display all input requirements not yet satisfied, and identify the command selected. This program will be executed each time data input is required.

Each input mode will require a separate routine and can operate independently of the main input program. These programs must load in all tables from auxiliary storage, determine the required formats to be used in reading in the data, remove selected items from the input list after the data has been read in, and place the data in an output buffer with the proper labels to identify the data. When the data has been loaded into the output buffer an output program is automatically entered to display and print the data. Output is discussed in Para. 4-7.

It has been assumed that all application programs will standardize card and tape input formats, consequently only one input program for each input mode is required.



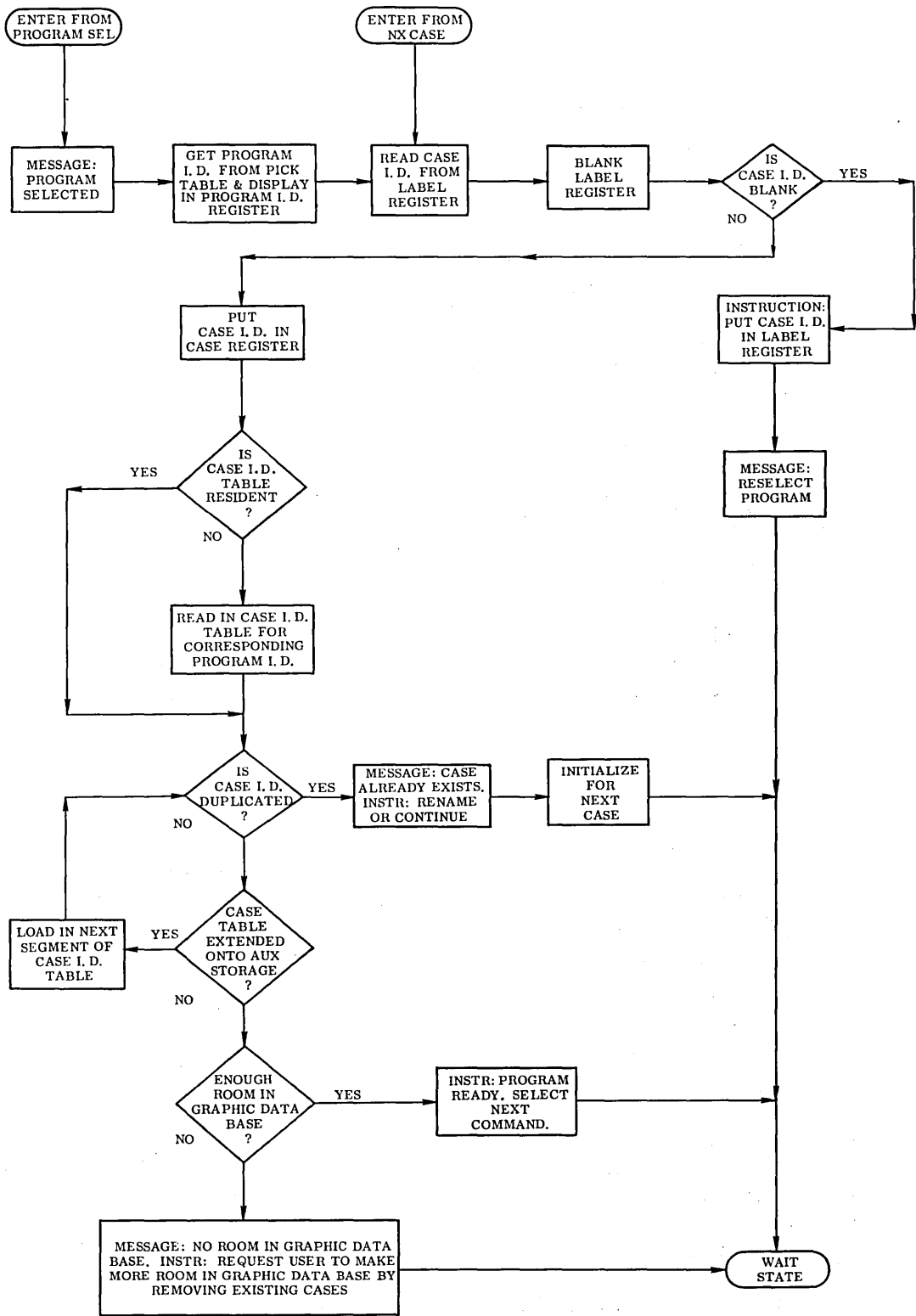
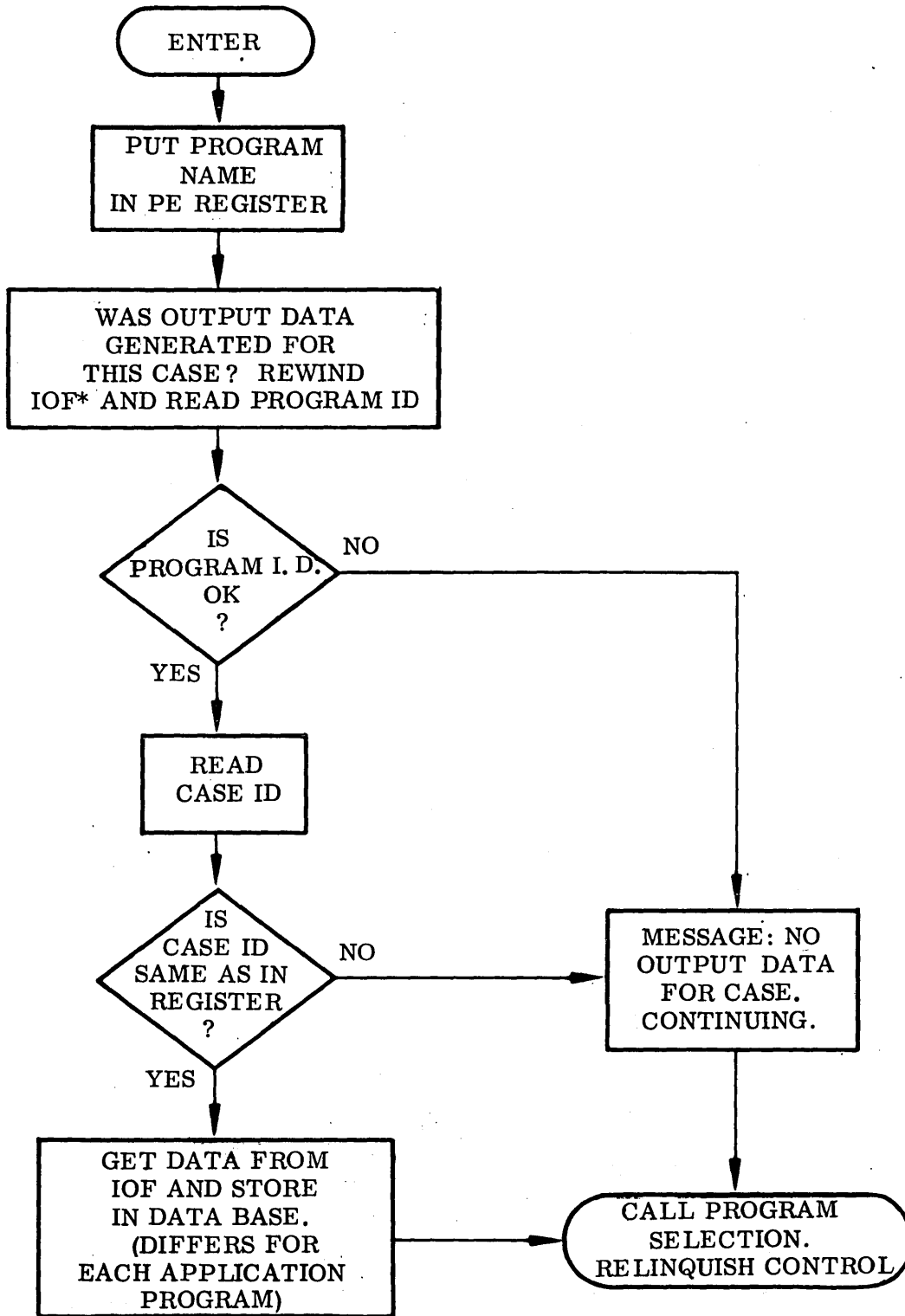


Fig. 4-19 Program Selection Program



NOTE: This program is similar to Program Library.

\*IOF = Intermediate output file

Fig. 4-20 Program Next Case

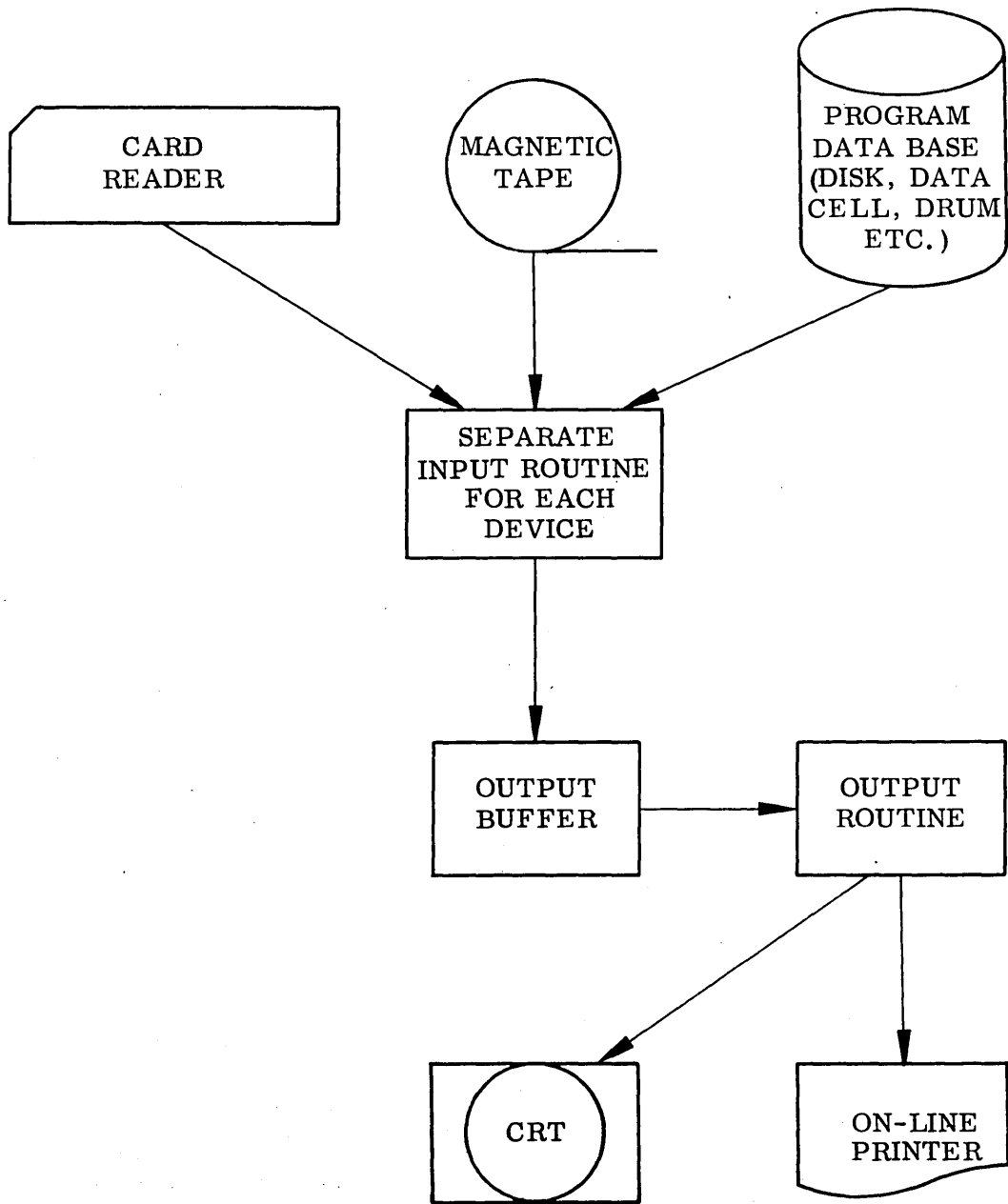


Fig. 4-21 Input-Output Flow for Card, Magnetic Tape and Data Base

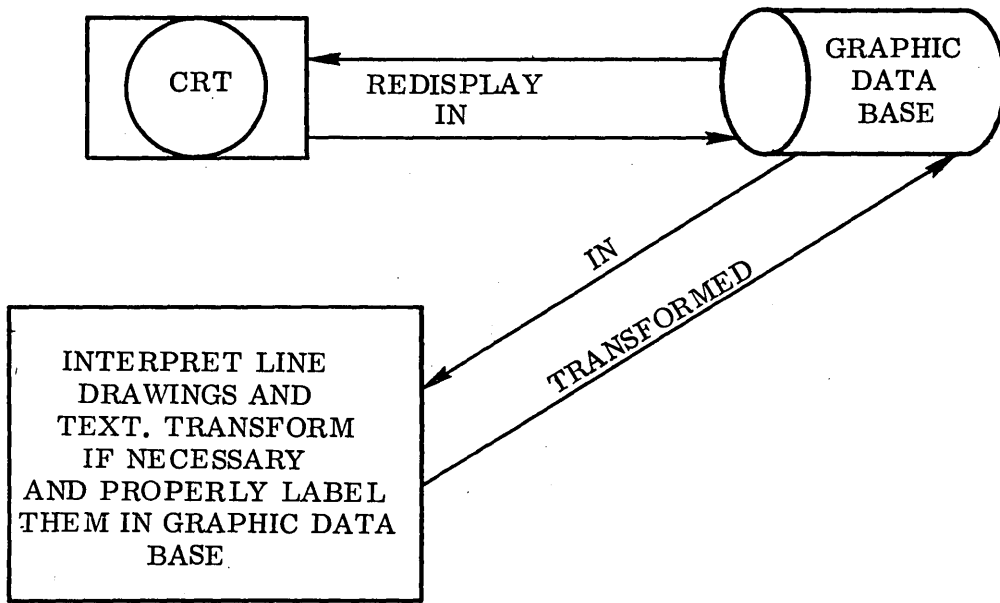


Fig. 4-22 Input Data Flow for Cathode-Ray Tube

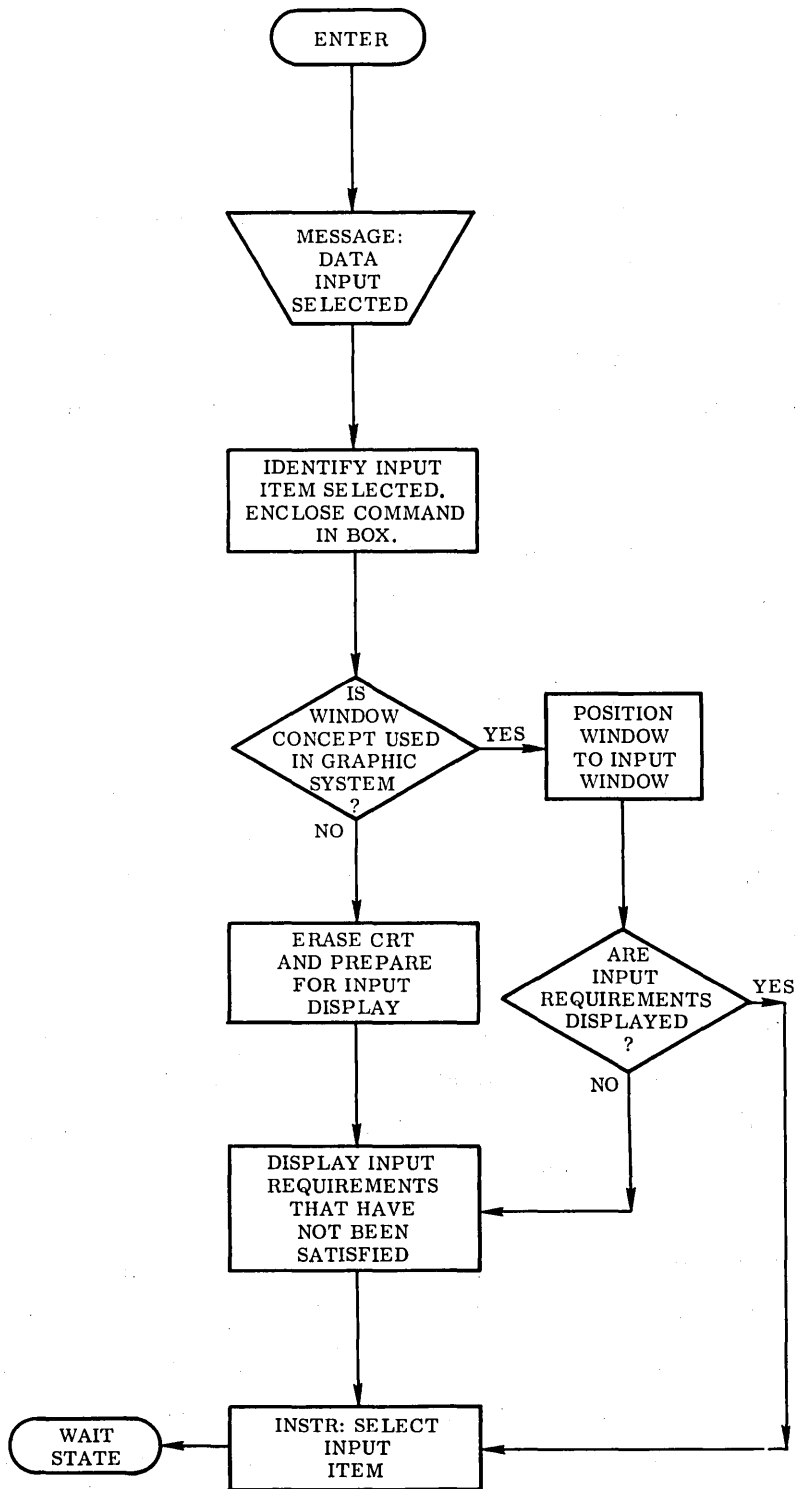


Fig. 4-23 Input Program

Otherwise each application program will require separate input routines. Separate input routines might be preferred so existing application programs can retain their familiar formats for card input.

The following "standard" formats are assumed. Card formats will be 6E12.8 for floating data, 6I12 for integer data, and 12A6(18A4) for Hollerith data. The latter depends on computer word length. Tape records are card images. Data arrays must be preceded by a card containing the number in integer format of single items, pairs, or triplets in the array. The format to be used for any data item will be determined by the input routine.

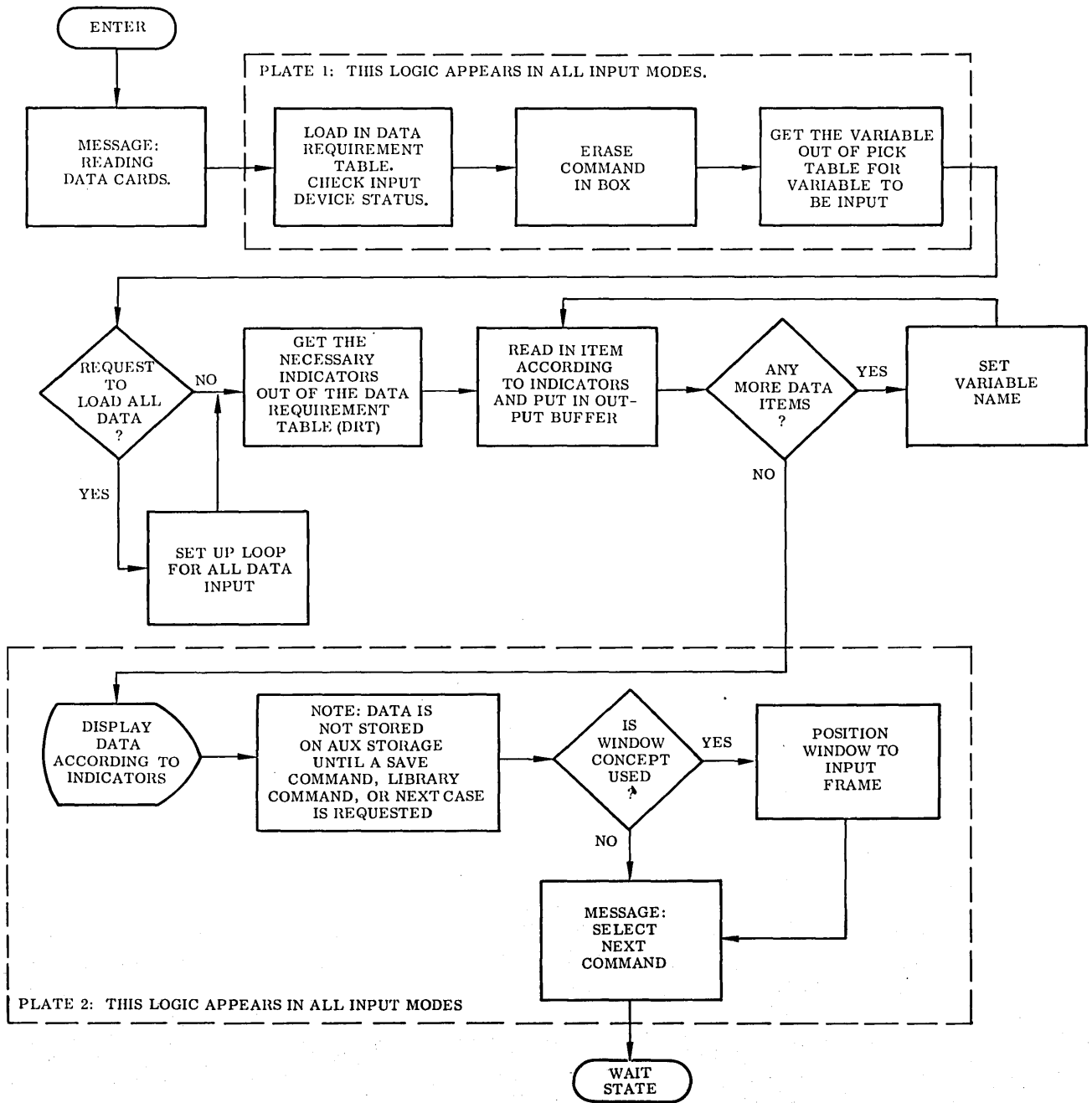
Program flow for card input, tape input, data base input, and CRT input are shown in Figs. 4-24, 4-25, 4-26, and 4-27, respectively.

#### 4.5.5 Data Modification Specification

Data does not assume meaning to the numerical program until computations are requested. Until then, data is nothing but groups of lines and text on the CRT with unique names. Consequently, modification of data is limited primarily by system hardware and software.

Actual manipulation of lines and BCD text will be performed by basic system software and thereby limited by it. It is assumed that the graphic hardware and software system being used will have adequate line drawing capabilities and ability to move lines and text or replace said line and text.

Performing data manipulations using only system hardware and software capabilities could impose limitations. Some of the drawbacks might be the necessity of imposing various restrictions on modifiable data or having a piece of data displayed but not modifiable. Also, unless the system used is highly sophisticated, extra work by the user is often required for simple modifications like changing one value on a function curve. To overcome these drawbacks would require another command (MODATA) for modifying data. This was successfully done in Ref. 4-1 but the program was written



NOTE: THIS PROGRAM READS CARDS FROM THE ON-LINE CARD READER IN A STANDARD FORMAT SUITABLE FOR THE ITEM TO BE READ.

Fig. 4-24 Card Input Program

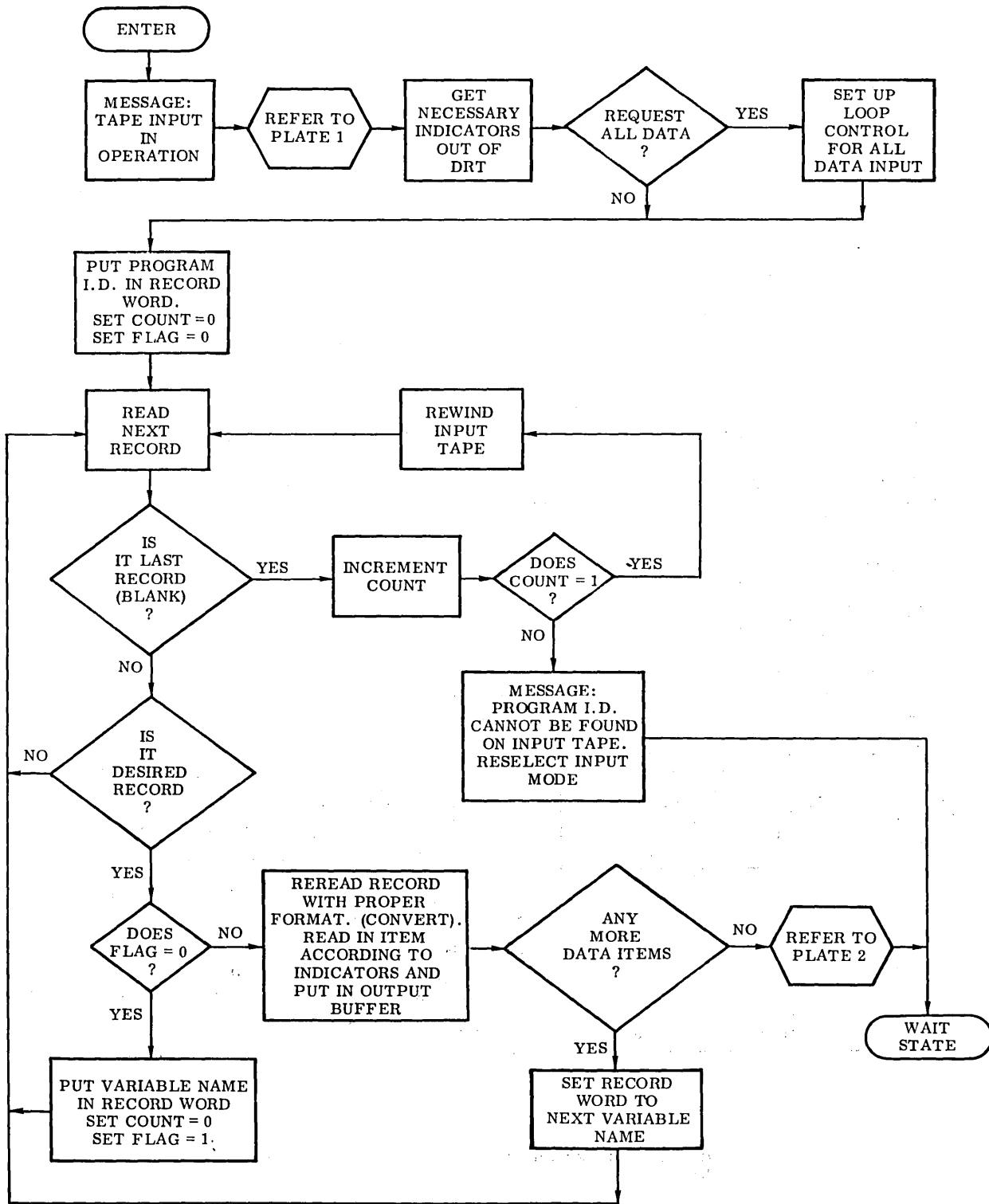


Fig. 4-25 Tape Input Program



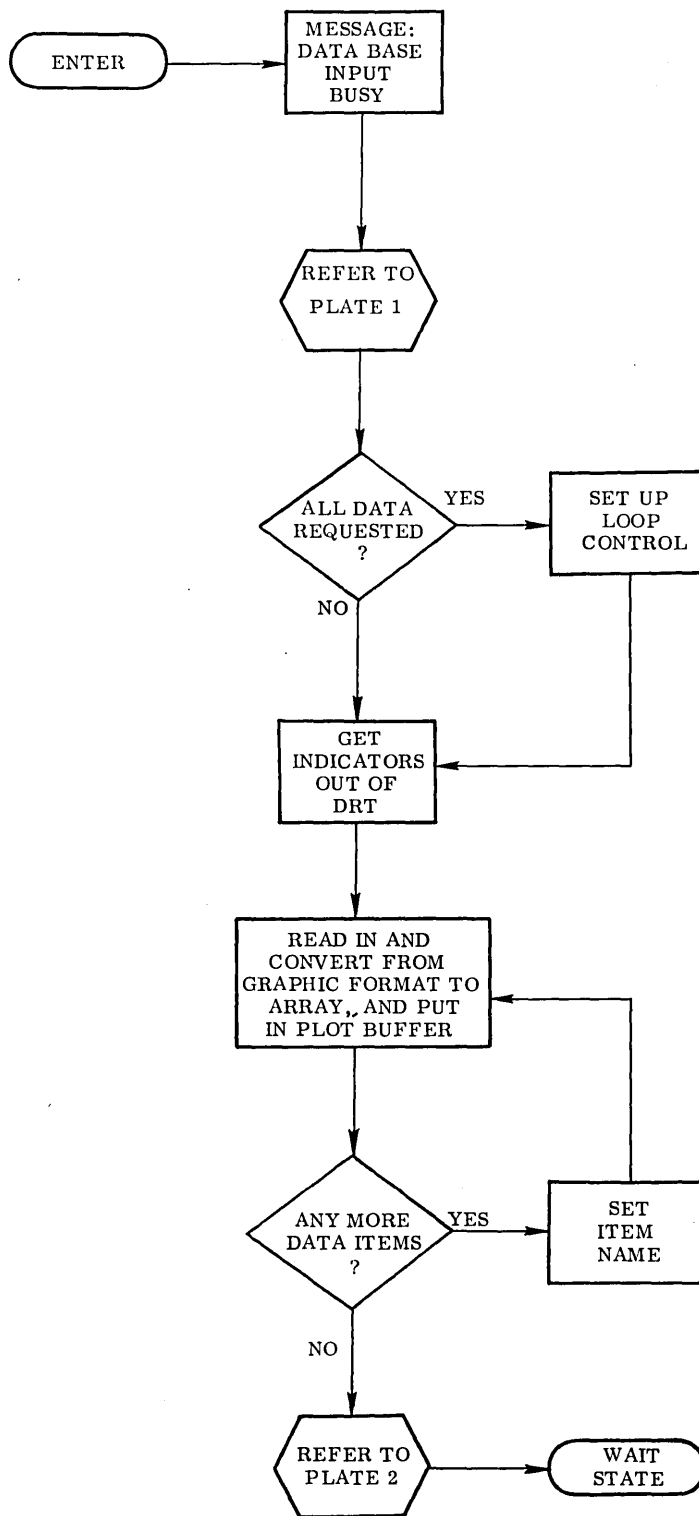


Fig. 4-26 Data Base Input Program

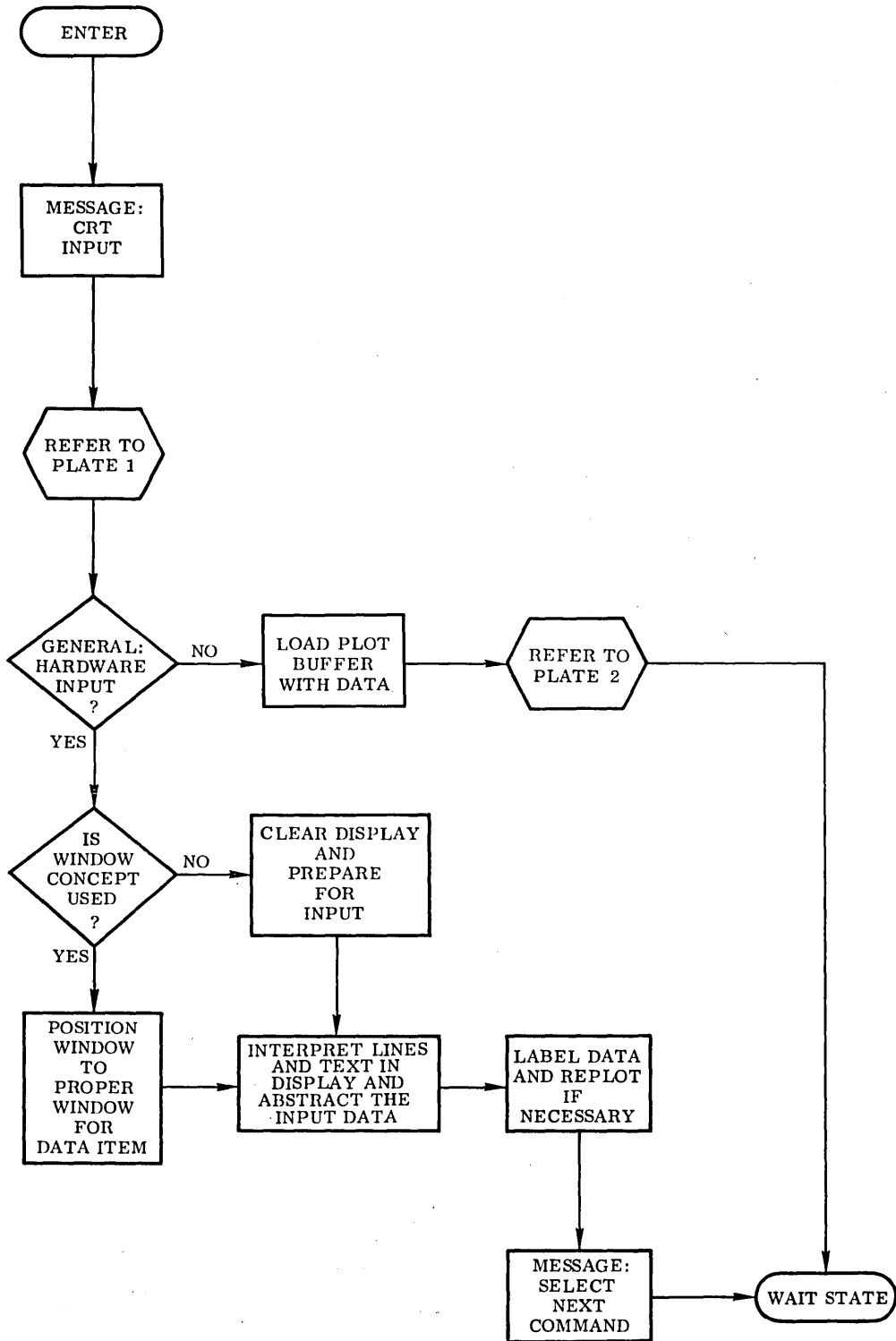


Fig. 4-27 CRT Input Program

for specific data and data formats. Many peculiarities of the data were accounted for and required considerable time to develop. It is felt that a general data manipulation routine could be developed if the data structure were specified.

Precise coordinate input and rescaling of graphs have both been successfully implemented in Ref 4-1. However, in this case, the data structure was designed specifically for the two programs which limited the utility of the coordinate input program and the rescaling program for other data structures. It is felt that a general coordinate program and rescale program could be developed for another data structure if adequate data structure specifications were available.

#### 4.5.6 Computation Specification

The computation program is entered by selecting the compute command (COMPUTE). When selected, the application data base is checked to make sure that the case ID appearing in the Case ID Register has not been computed previously. If not, all input data are obtained from the graphic data base and input requirements are checked for completeness. If input requirements are not satisfied a message is displayed and the input program is automatically entered at which time a list of input requirements that have not been satisfied is displayed.

If the input requirements are satisfied the application data base is checked to see if input data already exists for the case. If data already exists it is checked to see if it is referenced by other programs or other cases. If so, a message is displayed and computation aborted. If no other references are made, the data in the application data base is deleted and the data from the graphic data base is placed in the application data base. An override button can be set by the user which allows him to perform the computations using the data in the application data base rather than that in the graphic data base.

If the graphic system is operating as a satellite computer, the data is formatted for transmission to the central computer and all necessary controls are set. Otherwise, the application program is entered directly and control turned over to it.

Computed items are stored on an intermediate scratch file with each item identified. This file is opened at sign-on and has a program ID and case ID for a file label. The program ID and a blank case ID are written each time a program is selected from the library. The actual case ID is not written until data is ready to be written on the file. The case ID will then indicate if data has been written onto the file or not. This scheme would be compatible with a satellite-type system where the intermediate scratch unit would be an output device for the central computer. This program is shown in Fig. 4-28.

#### 4.5.7 Output Specification

Computed quantities are not automatically displayed when computed. If this were done the flexibility of graphics would be greatly restricted. The chances are that all output quantities would not be displayed at once so a standard set of output would have to be preprogrammed, limiting the users access to computed items.

Output is requested by selecting the output command (OUTPUT). A list of possible items is displayed and the user is given a choice of standard output or selecting a specific item. Included in the list is an option to void output for current case. If the standard set is chosen, an output routine is automatically entered and the data is immediately displayed. One item at a time can be selected which automatically enters a general output program which displays the selected item. If the user elects to void the output another program is entered which removes the data.

Computed quantities are not automatically placed into the application data base. This allows the user to iterate on the input data until satisfactory results are obtained. Once satisfactory results have been achieved, the user has a complete case and can proceed to the next case by selecting the "next case" command or return to the library by selecting the "library" command. If either the next case or library is selected the quantities just computed are transferred from the intermediate file to the application data base with proper identification.

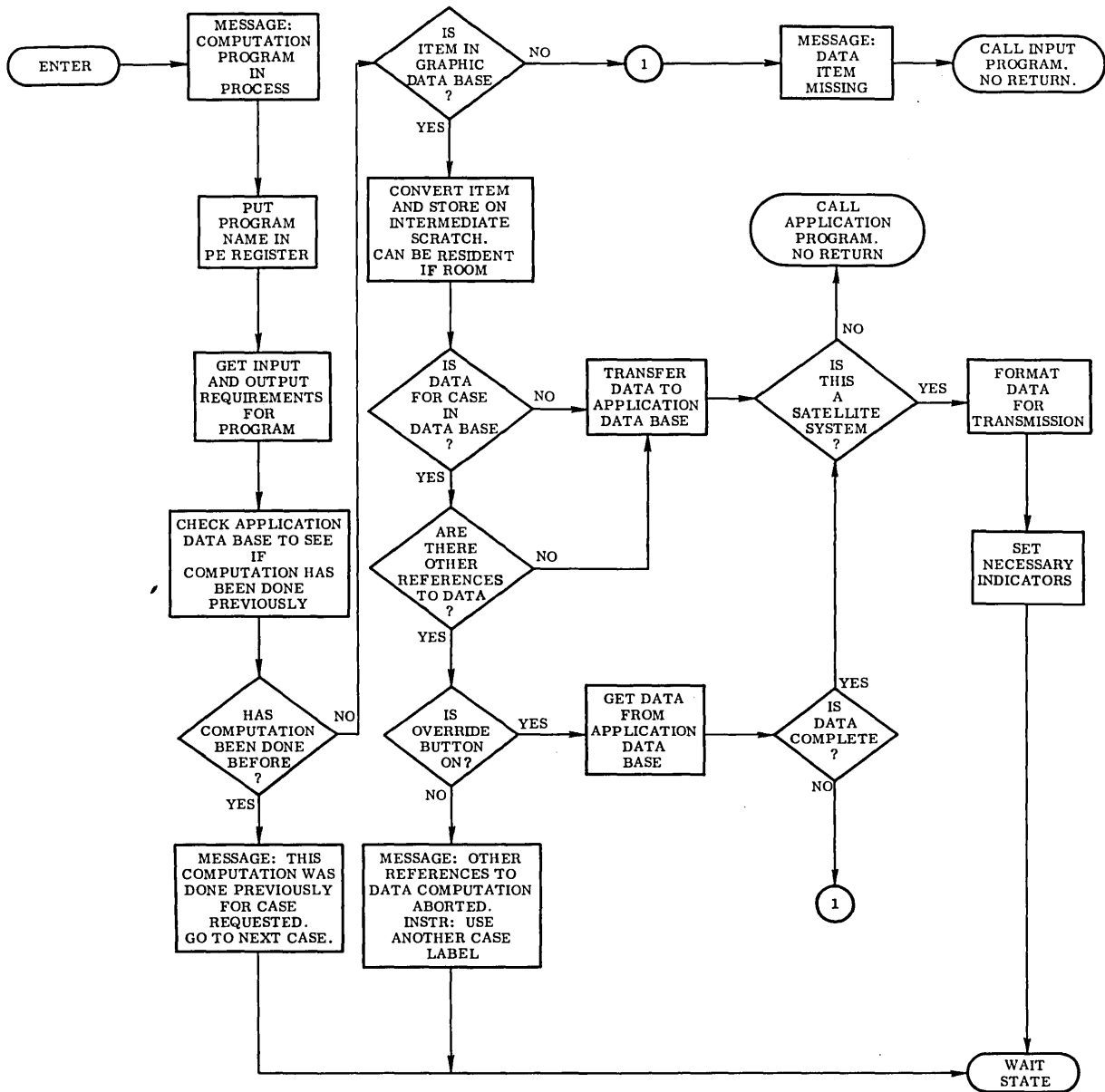


Fig. 4-28 Compute Program

Program flow for standard output, general output, and voiding data is shown in Figs. 4-29, 4-30, and 4-31, respectively. The next case and library command were discussed previously.

#### 4.5.8 CRT Layout Specification

CRT layout specifications will pertain only to the communication area. The data display area is rather arbitrary so specifications are not attempted. A table of all commands required is shown in Table 4-3 with each command cross referenced with the specification section where the command is described. For illustrative purposes eight character mnemonics are used.

Those commands with an asterisk appearing in the subsection column were not required but are briefly mentioned in the specification. They are included in the command requirement table for completeness. The location of each command on the CRT is shown in Fig. 4-32 with only the specified ones shown.

A table of registers required is shown in Table 4-4 with each register referenced to the specification section where it is described. Several registers appear in the table but are not referenced to any section. These contain an asterisk in the subsection column and are included for completeness. Some of these will be used with some of the commands in the command table that also have an asterisk in the subsection field.

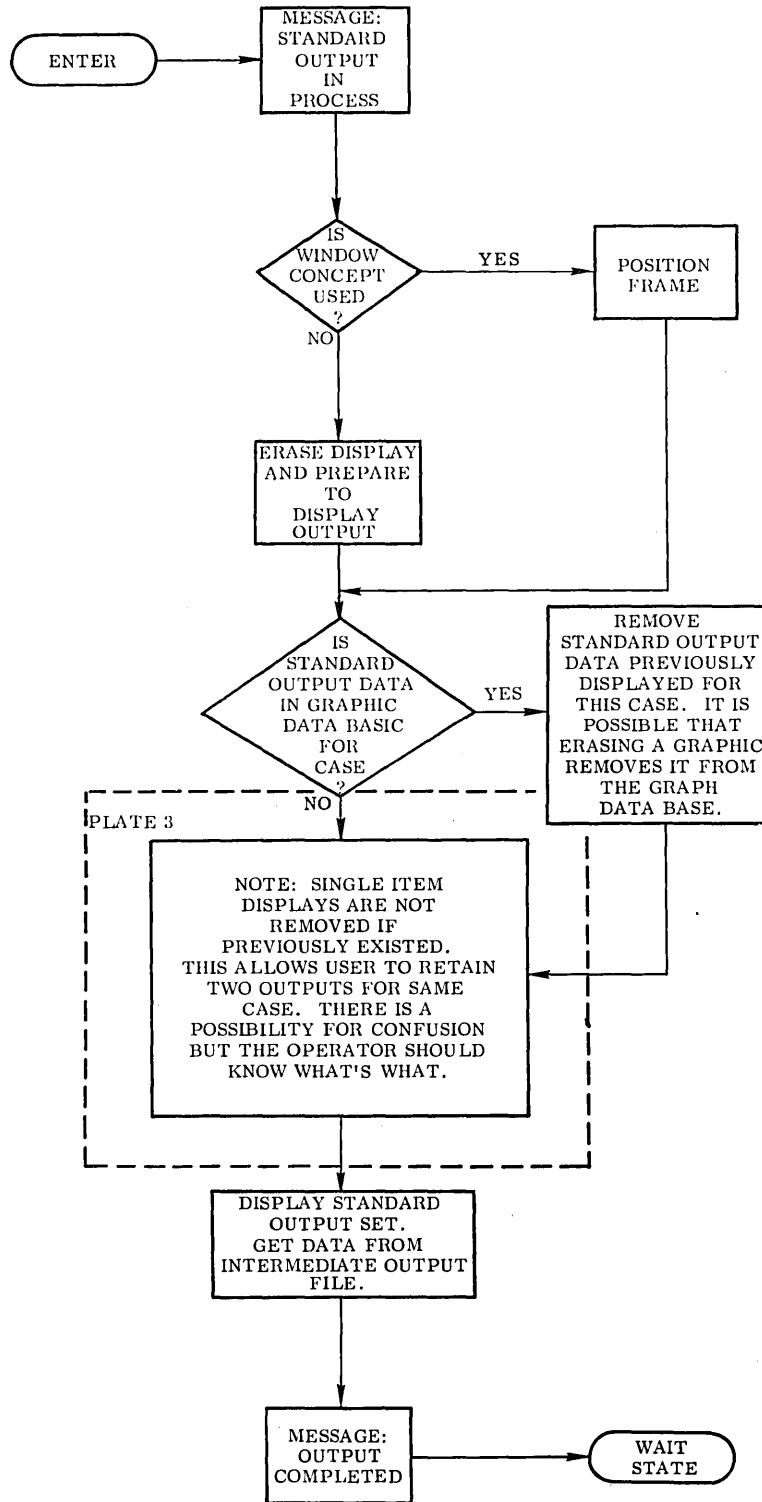
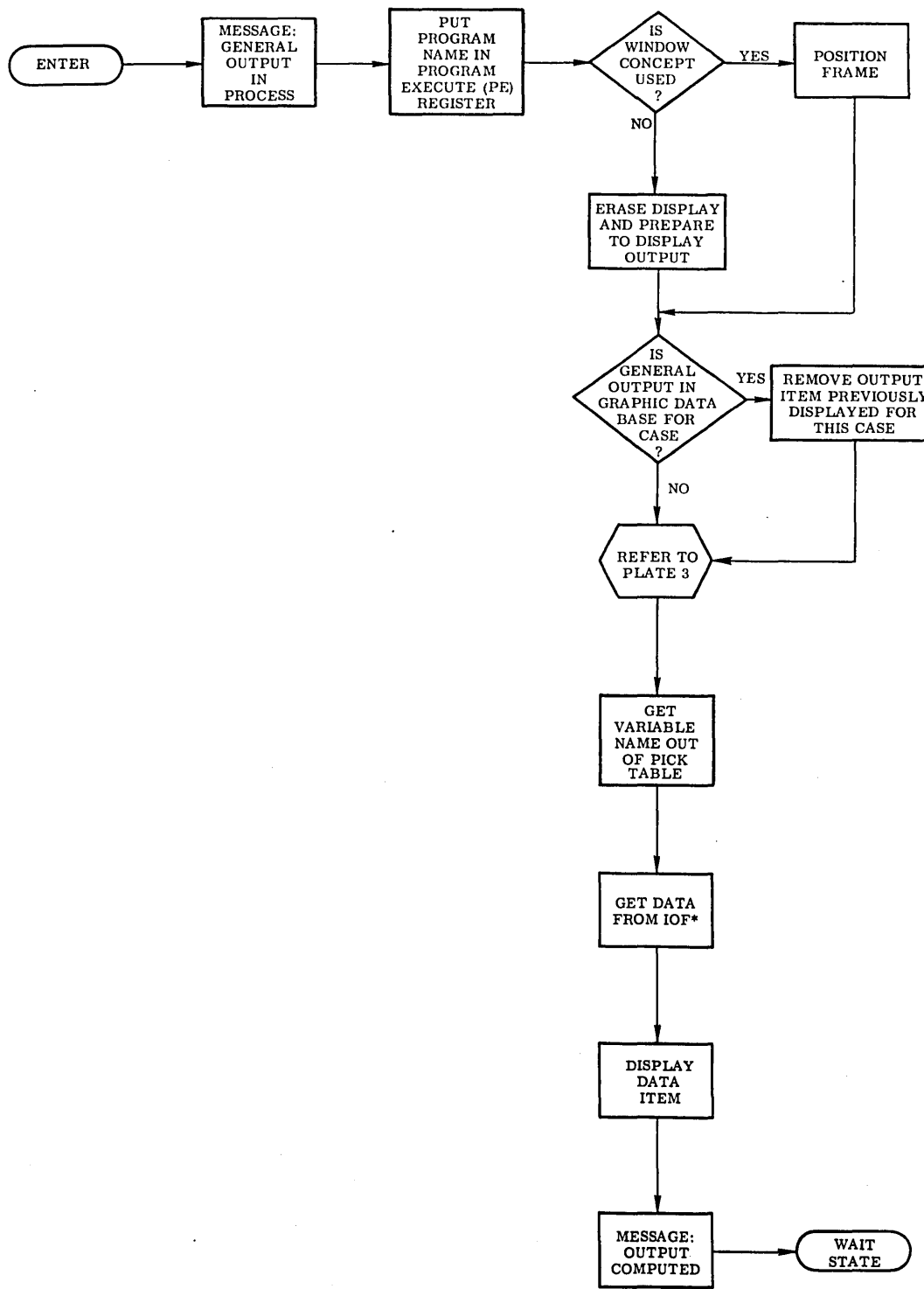


Fig. 4-29 Standard Output Program



\*IOF - INTERMEDIATE OUTPUT

Fig. 4-30 General Output Program



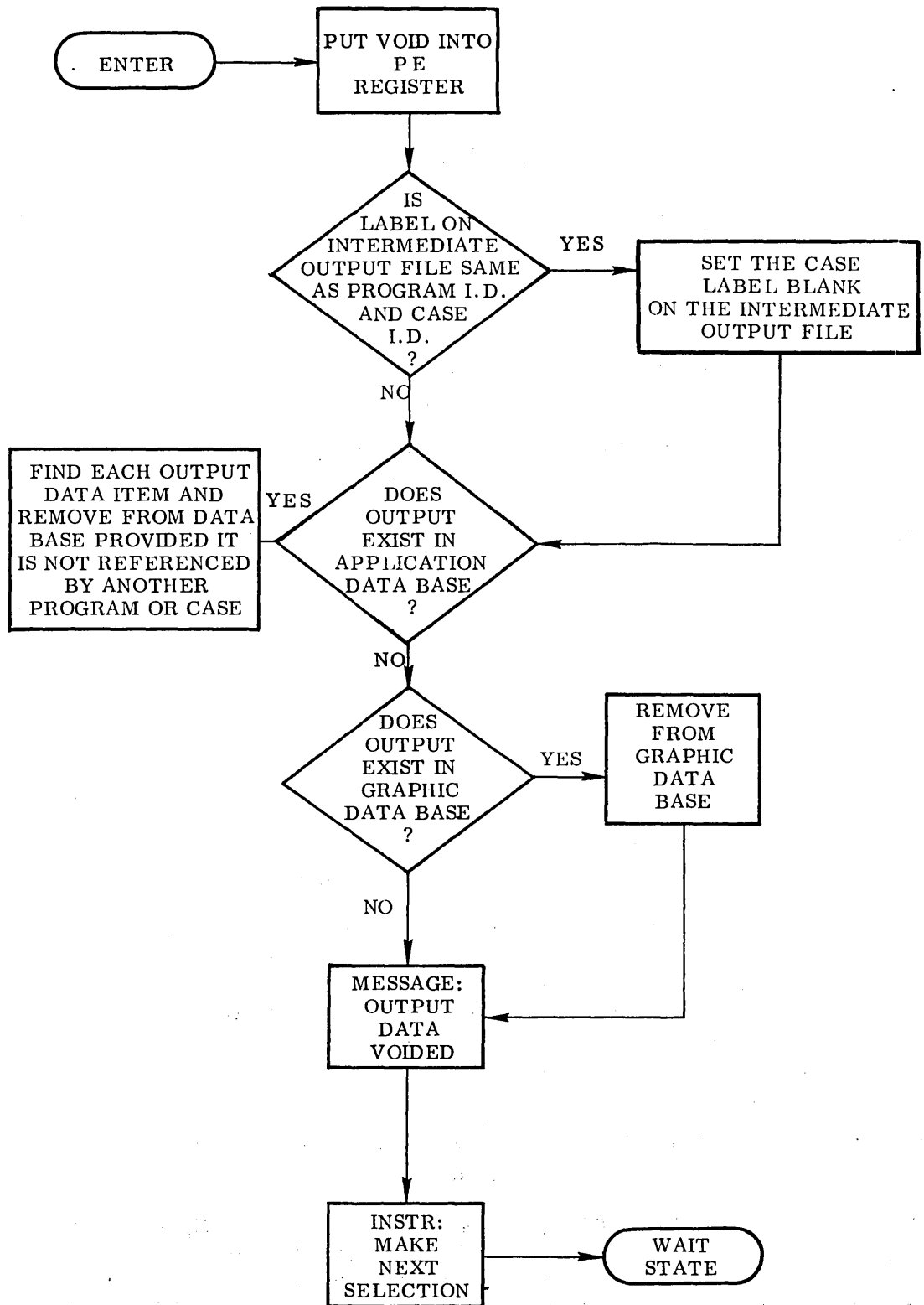


Fig. 4-31 Void Output Data Program

Table 4-3

## COMMAND TABLE

Command	MNEMONIC	Brief Description	Subsection
Library	LIBRARY	Displays a list of executable application programs	2.2
Input	INPUT	Displays a list of input requirements	2.4
Modify Data	MODATA	Activates special data manipulation routine	*2.5
Compute	COMPUTE	Activates the computation program	2.6
Check Input	CKINPT	Checks input requirements for completeness	*2.6
Check Input and Save	CKNSAV	Checks input requirements and puts data into data base	*2.6
Output	OUTPUT	Displays a list of computed output quantities	2.7
Next Case	NXCASE	Saves current case output and prepares for next case	2.3
Stop	STOP	Transfers application data base to permanent storage device	2.2
Stop and Save	STOPNSAV	Same as STOP, plus it saves the graphic data base	2.2
Display			2.5
Plot	PLOT	Used to plot individual data items on request	2.5
X-Y Input	XY-INPUT	Used in conjunction with X, Y, Z, and T registers to enter precise values on a graph	*2.5

Table 4-3 (Cont.)

Command	MNEMONIC	Brief Description	Subsection
Rescale	RESCALE	Used in conjunction with the X and Y registers to rescale graphical displays	*2.5
Interrupt	INRUPT	Used to stop numerical computations at selected points	*2.6
Stop and Go	STOPNGO	Used to terminate computation or to continue after the INRUPT command is selected	*2.6

\* These commands were not included in the requirements but are included in the table for completeness.

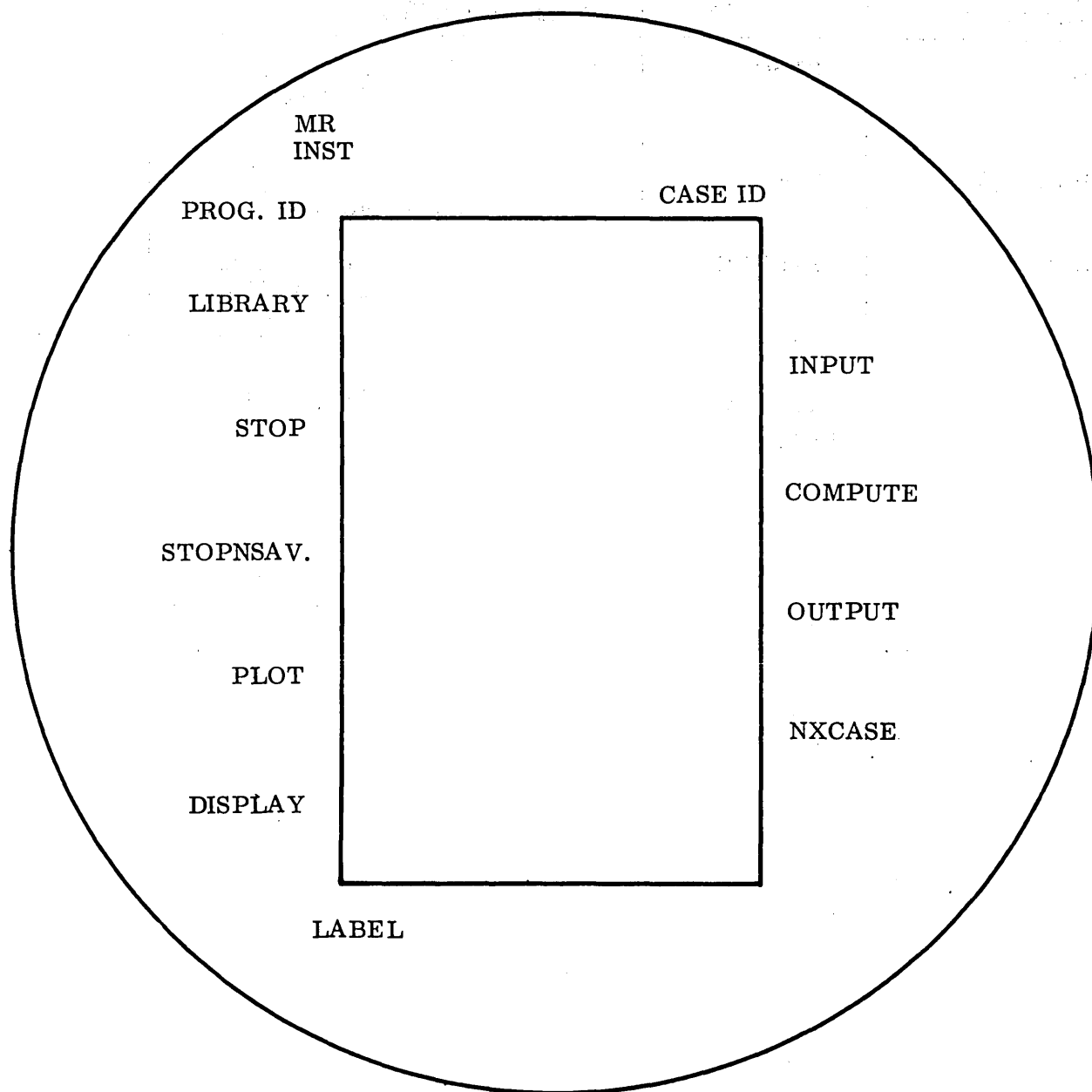


Fig. 4-32 Cathode-Ray Tube Layout

Table 4-4  
REGISTER TABLE

Register	Mnemonic	Brief Description	Section
PROGRAM ID	PROG. <sub>b</sub> ID	Displays name of application program being executed	2.3
CASE ID	CASE <sub>b</sub> ID	Contains Case ID supplied by user	2.3
MESSAGE	MR	Messages to the operator are displayed in the MR	2.8
INSTRUCTION	INST	Instructions for the operator are displayed in the INST	2.8
X-INPUT	X	Enter floating pt number	*
Y-INPUT	Y	Enter floating pt number	*
Z-INPUT	Z	Enter floating pt number	*
TIME	T	Enter floating pt number	*
LABEL	LABEL	Enter any BCD text	2.3
PROGRAM EXECUTION	PE	Contains the name of the subroutine on program being executed	†

† Appears in various program flow diagrams.

#### 4.6 PEANUTS: AN EXAMPLE OF GRAPHIC PROGRAM REQUIREMENTS AND SPECIFICATIONS

A comprehensive list of active structural programs (Table 4-1) was presented in the background section. One of these programs, PEANUTS, is a popular structural program used for performing nonroutine structural design tasks. For this reason PEANUTS was chosen as the structural design program to be conceptually incorporated into a computer-aided design environment in which man and machine communicate via a cathode ray tube utilizing some mechanism like a light pen.

The more tractable structural programs to be incorporated into computer graphics environment are separable into three functions: input, computation and output as shown in Fig. 4-33. A functional block diagram of PEANUTS as it is used in production is shown in Fig. 4-34. A close look at PEANUTS (Fig. 4-34) reveals that it is not easily separable into the ideal function structure. However, it is still well suited from the logical view point for a graphic environment.

Program requirements and specifications for the PEANUTS program will be presented here in three sections: Input, Computation, Output.

##### 4.6.1 PEANUTS Example: Input Requirements and Specifications

The PEANUTS Program input requirements are nominal but possible options are numerous. A complete list is shown in Table 4-5. Two types of data are required: tabular data and single items. Tabular data is representable as two-dimensional curves which are normally input in array form. These arrays define geometry, surface loads, etc. Single items represent coefficients of various kinds and various boundary conditions.

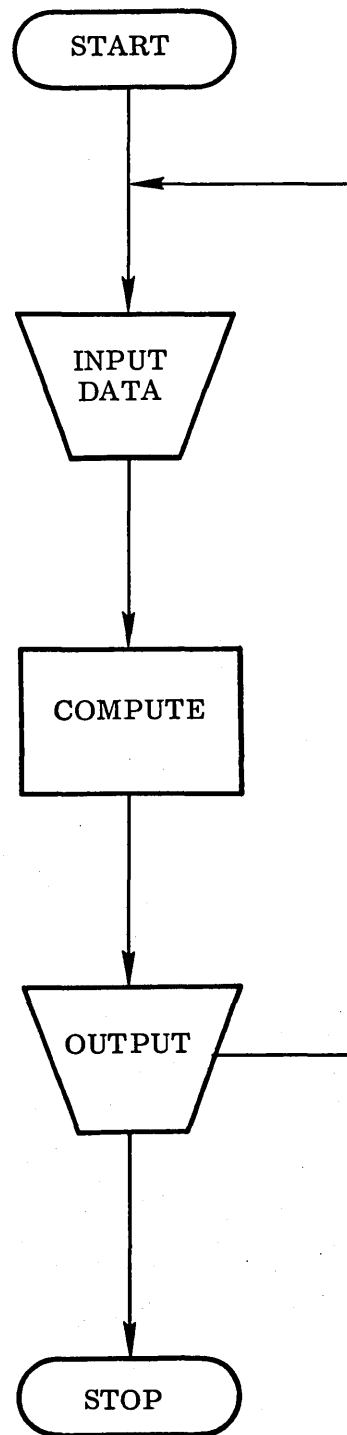


Fig. 4-33 Functional Block Diagram of an Ideal Structural Program

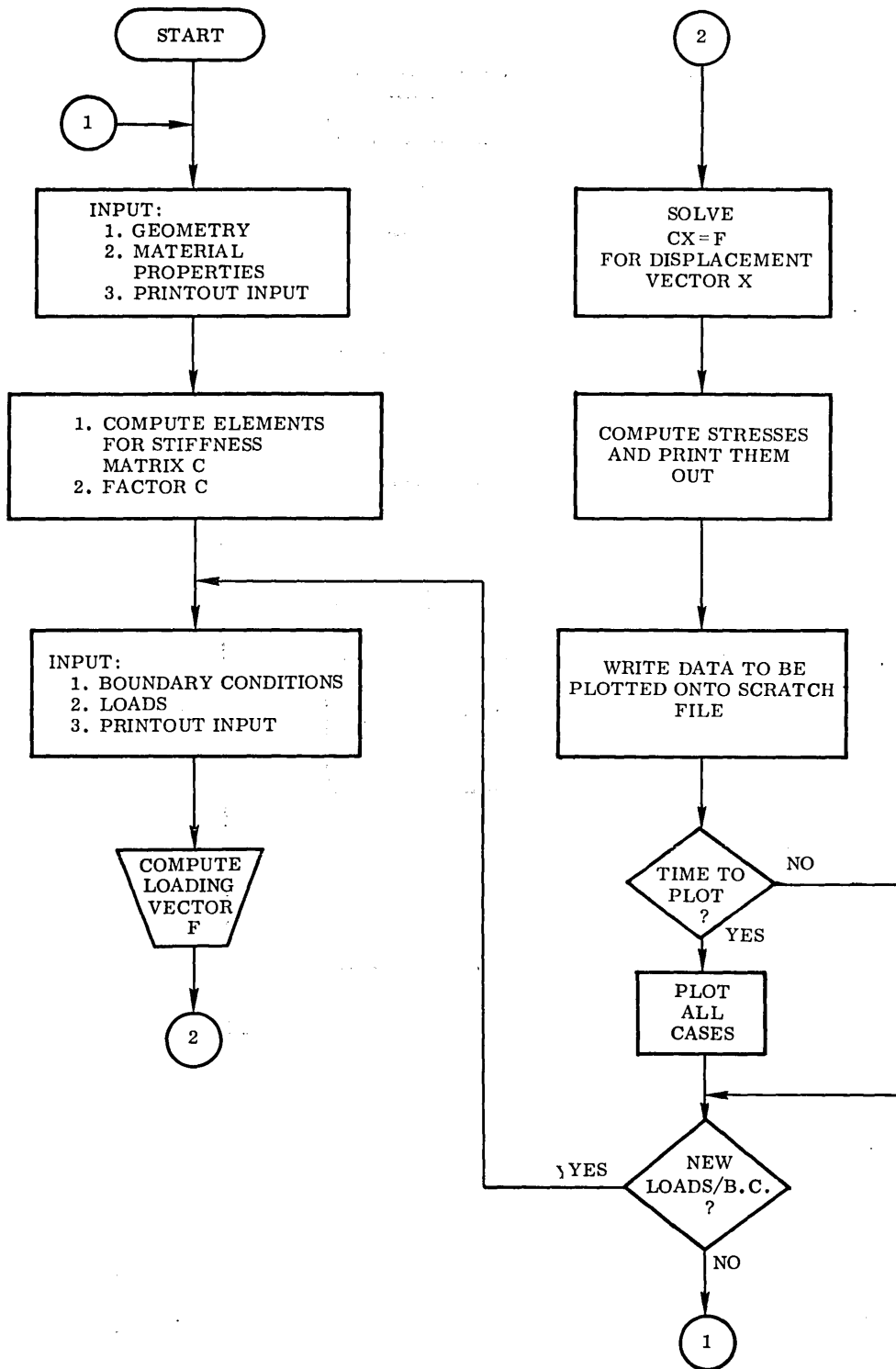


Fig. 4-34 Functional Block Diagram of PEANUTS



Table 4-5

## DESCRIPTION OF INPUT QUANTITIES

Quantity	Description
RECORD	Seventy two characters of descriptive Hollerith information (arbitrary) to be printed in output headings and on plots
M	Problem number (optional, will be printed in title line)
SS1	Value of $s$ , the independent variable (surface coordinate) at the "starting" boundary (ignore if IGEOM = 2, 3, or 4)
SS2	Value of $s$ at the terminal boundary B (ignore if IGEOM = 2, 3, or 4)
NE	Number of "stations" for which the solution is to be obtained (150 maximum). These will be equally spaced between the boundaries defined by SS1 and SS2, and need not coincide with the stations used to define geometry.
IGEOM	An indicator pertaining to geometry input If IGEOM = 0, a subroutine containing analytical expressions applicable to a variable thickness sphere or cylinder is used. If IGEOM = 1, values of geometric quantities are read in point by point in $s$ using a matrix format for fifty points maximum. This permits generation of a general shell of revolution. If IGEOM = 2, specialized geometric data for a toroidal shell of nonuniform thickness and circular section are read. If IGEOM = 3, specialized geometric data for a toroidal shell are read as above. If IGEOM = 4, specialized geometric data for an ellipsoidal shell element are read.
NRITE	The increment in "stations" desired between printout of computed quantities for the shell interior region.
HN	Total number of input stations for which geometric data will be read in matrix form.
E	Value of Young's modulus for the shell material
PR	Poisson's ratio

Table 4-5 (Cont.)

Quantity	Description
ALP	Coefficient of thermal expansion for the shell material
R1 <sup>(a)</sup>	Meridional radius of curvature
R2	Circumferential radius of curvature
NA	Number of input stations for which geometric quantities are to be computed in the region between S1 and S2 below
NB	Ditto for the region between S2 and S3 below
NC	Ditto for the region between S3 and S4 below
S0	Value of the shell surface coordinate $s$ defining boundary A of the geometry
S1	Value of $s$ defining intermediate point 1 at which the equations for thickness variation change
S2	Same for intermediate point 2
S3	Same for intermediate point 3
S4	Same for intermediate point 4
S5	Same for intermediate point 5
S6	Value of $s$ for "terminal" point of geometric input data
B6	Derivative of thickness with respect to $s$ for the region between S5 and S6
HA0	Coefficient in equation for thickness variation
HA1	Same
HA2	Same
HB1	Same
HB2	Same

<sup>(a)</sup>Quantities R1 through EXC2 pertain to the computations for geometry used if IGEOM = 0. If IGEOM = 2 or 4 some of these quantities are used.

Table 4-5 (Cont.)

Quantity	Description
HC1	Coefficient in equation for thickness variation
HC2	Same
EXA1	Exponent in equation for thickness variation
EXA2	Same
EXB1	Same
EXB2	Same
EXC1	Same
EXC2 <sup>(a)</sup>	Same
SI(IIN, 1)	A matrix (column vector) of values of $s$ , the independent variable, for which input geometry is to be read. (50, 1) maximum dimensions
RI(IIN, 1)	Values for $R$ , the shell radius normal to the axis of revolution corresponding to $s$ above
RII(IIN, 1)	Values for $R_1$ , the meridional radius of curvature
R21(IIN, 1)	Values for $R_2$ , the circumferential radius of curvature
HI(IIN, 1)	Values for the shell thickness
RA <sup>(b)</sup>	Radius of revolution of toroidal shell or the equatorial radius of the ellipsoid
RB	Radius of convolution of toroidal shell or the polar radius of the ellipsoid
THETA1	Angle (radians) defining the "starting" boundary A of the shell sector
THETA2 <sup>(b)</sup>	Angle (radians) defining the terminal boundary B of the shell sector

(a) Quantities R1 through EXC2 pertain to the computations for geometry used if IGEOM = 0. If IGEOM = 2 or 4 some of these quantities are used.

(b) Quantities RA through THETA2 pertain to the toroidal or ellipsoidal shell geometry used if IGEOM = 2, 3, or 4.

Table 4-5 (Cont.)

Quantity	Description
H	Thickness of toroidal shell (used only if IGEON = 3)
T1I(IIN, 1)	Values of outer fiber temperature relative to "no thermal stress" or reference temperature. If IGEOM = 1, the values (if not constant) must correspond spatially to geometry input stations specified through the SI matrix. If IGEOM = 0, 2, 3, or 4, the values will be assumed to correspond to stations equally spaced between the two boundaries of the shell element.
T2I(IIN, 1)	Input values for inner fiber temperature (Comments for T1I apply)
PZI (IIN, 1) <sup>(a)</sup>	Input values for normal pressure; internal pressure is positive (Comments for T1I apply)
PSI(IIN, 1)	Input values for surface "shear" positive with increasing s (Comments for T1I apply)
ISOL	<p>An indicator pertaining to the solution desired</p> <p>If ISOL = 0, the program will solve a two-boundary problem using input influence coefficients for the adjacent structure at each boundary.</p> <p>If ISOL = 1, the program uses force input for boundary A, and adjacent structure influence coefficients for boundary B.</p> <p>If ISOL = 2, the program computes influence coefficients and particular solution boundary values only.</p> <p>If ISOL = 3, the program reads force boundary conditions as input data for both boundaries and solves the corresponding problem.</p> <p>If ISOL = 4, the program solves the axially redundant two-boundary problem using influence coefficients for adjacent elements.</p>
MAXP	<p>An indicator pertaining to printout</p> <p>If MAXP = 0, printing is restricted to the minimum consistent with the information sought.</p> <p>If MAXP = 1, complete printout (equation checks, complete influence functions, etc.) will result.</p>

(a)As a convenience, the program will interpret a single value provided as input for PZI(1,1) as a constant pressure acting over the entire shell.

Table 4-5 (Cont.)

Quantity	Description
FVA2	Component of stress resultant parallel to axis of revolution
FHA2	Component of stress resultant normal to axis of revolution
MXA2	Resultant meridional moment
FVB2 FHB2 MXB2	Analogous to FVA2, etc., above. Boundary conditions to be applied to shell element 2 at juncture B.
FVAE	External line load parallel to axis of revolution
HAE	External line load normal to axis of revolution
MAE	External meridional line moment
FVBE HBE MBE	Analogous to FVAE, etc., above. External line loads acting at the juncture of shell elements 2 and 3 (juncture B).
CVV	The "reference" value of axial displacement at boundary A (juncture A) used to compute the "absolute" displacements
NPLOT	An indicator for plotting For NPLOT = 0, no plots are to be made. For $1 \leq \text{NPLOT} \leq 30$ , plots will be made as indicated in Table 4-6.
NFLAG	An indicator pertaining to plotting problems in sequence For NFLAG = 0, the current problem is the last or only one to be plotted. For NFLAG = 1, other problems will be plotted later.
NEWP	An indicator pertaining to usage of geometric data For NEWP = 0, the next problem will use the current geometry with a new force vector as prescribed through ISOL below. For NEWP = 1, new geometric data will be read for the next problem.

Table 4-5 (Cont.)

Quantity	Description
A11 A12 A22 A13 <sup>(a)</sup> A23 <sup>(a)</sup> A33 <sup>(a)</sup>	Influence coefficients for the adjacent structure (shell element 1) at juncture A. Symmetry is used to compute values for elements not specified as input data. For axially determinant cases, ignore starred quantities (leave blank on data card).
C11 C12 C22 C13 <sup>(a)</sup> C23 <sup>(a)</sup> C33 <sup>(a)</sup>	Influence coefficients for the adjacent structure (shell element 3) at juncture B. Comments above apply.
V1AP	Rotation
W1AP	Displacement normal to axis
M1AP	Moment
H1AP	Component of stress resultant normal to axis
DV1AP <sup>(a)</sup>	Axial displacement
FV1AP <sup>(a)</sup>	Axial component of stress resultant
V3BP W3BP M3BP H3BP DV3BP <sup>(a)</sup> FV3BP <sup>(a)</sup>	Analogous to above but for adjacent structure (shell element 3) at juncture B

Corresponding to the "particular" solution characterizing the adjacent structure (shell element 1) at juncture A

<sup>(a)</sup> These quantities are ignored for an axially determinant solution (leave blank on data card).

Table 4-6

## LIST OF POSSIBLE OUTPUT FROM PEANUTS

NPLOT	Function Name	Description
1	H	Shell thickness
2	SIGX1	External meridional stress
3	SIGX2	Internal meridional stress
4	SIGT1	External circumferential stress
5	SIGT2	Internal circumferential stress
6	SIG1	External "effective" stress
7	SIG2	Internal "effective" stress
8	NX	Meridional membrane stress resultant
9	NT	Circumferential membrane stress resultant
10	MX	Meridional moment
11	MT	Circumferential moment
12	Q	Shear
13	FH	Net stress resultant normal to axis of revolution
14	FV	Net stress resultant parallel to axis of revolution
15	DELTA	Deflection normal to axis of revolution
16	V	Rotation (dependent variable)
17	EXZ1	External surface meridional strain
18	EXZ2	Internal surface meridional strain
19	ETZ1	External surface circumferential strain
20	ETZ2	Internal surface circumferential strain
21	DH	dH/ds (s is the independent variable, the meridional distance along the shell reference surface.)
22	R	Radius normal to the axis of revolution
23	DR	dR/ds
24	R1	Meridional radius of curvature
25	R2	Circumferential radius of curvature
26	U	Dependent variable (QR2)

Table 4-6 (Cont.)

NPLOT	Function Name	Description
27	DU	$dU/ds$
28	DDU	$d^2U/ds^2$
29	DV	$dV/ds$
30	DDV	$d^2V/ds^2$

To accommodate numerous input as described above using a CRT requires two different displays as shown in Figs. 4-35 and 4-36. Only the data display area is shown on these illustrations. The input has been separated into function (tabular) and single item input. Each occupies separate displays. Item input is divided into Material Properties, Boundary Conditions, and Influence Coefficients.

Input is achieved by selecting an input command (INPUT) which displays two options: Functions and Single Items. Selecting function or single item results in displaying Fig. 4-35 and Fig. 4-36, respectively.

In function input the geometry should be specified first. The necessary parameters are input via registers. Thickness, pressure, shear and temperature can be input in any order and in any of the desired modes, card, CRT, etc.

Single items are input in groups of material properties, influence coefficients, boundary conditions, and particular solution. Each group is input by selecting the appropriate juncture A or B and a group label, e.g., Boundary Conditions. A juncture specification is not required for material. This will cause the input options to be displayed. Any one of the options can be selected. However, the option selected pertains to the whole group, e.g., if CARD input were selected for juncture A Boundary Condition; FVA, FHA, . . . , CVV would be read from cards and displayed on the CRT. The order in which the groups are selected is arbitrary as long as they are consistent with PEANUTS input for the particular problem to be solved. An example of input requirements necessary to design the transition region between a thick and thin region on a



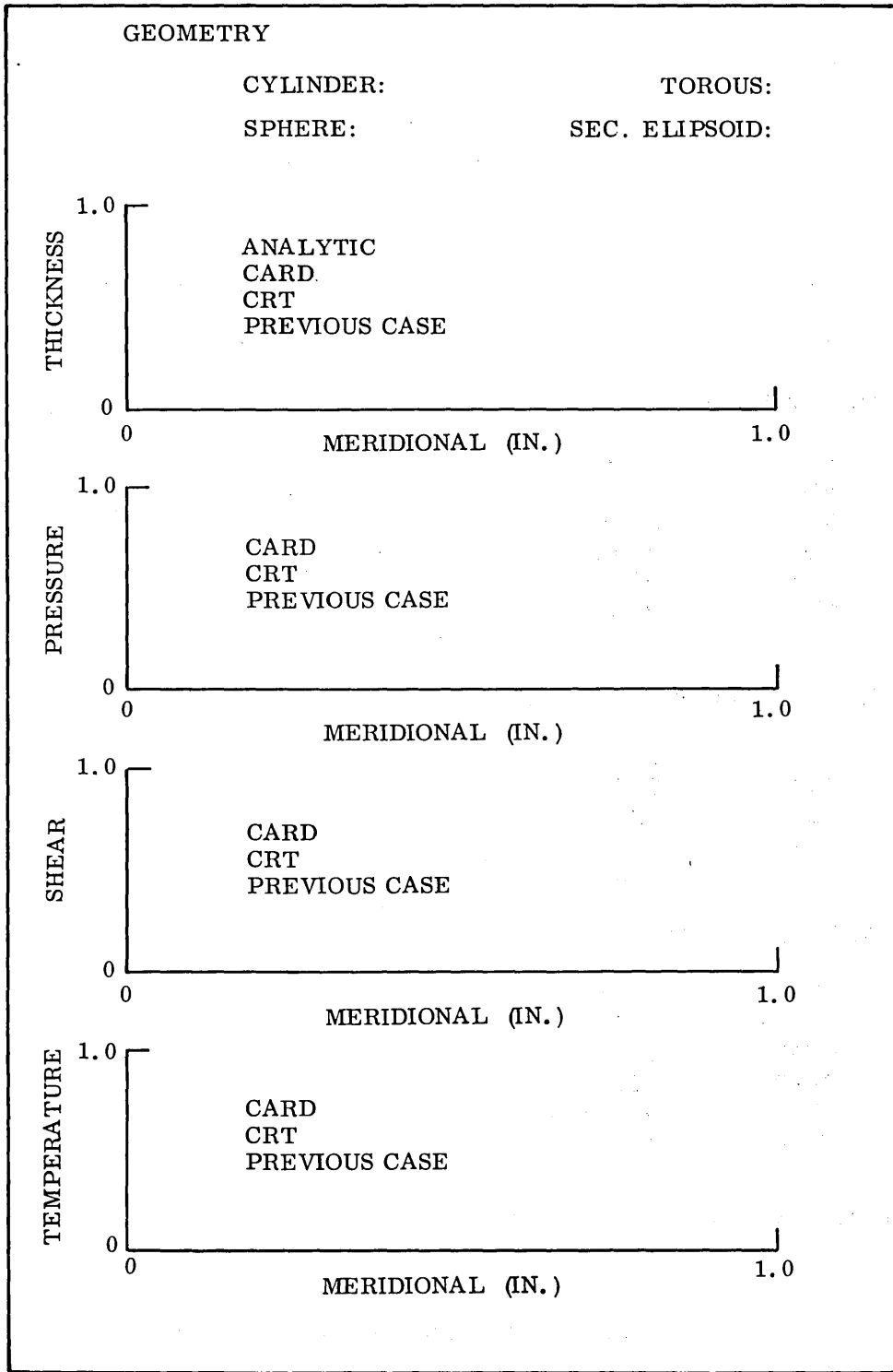


Fig. 4-35 Function Input

**MATERIAL PROP.**

**E:  
PR:  
ALP:**

**INPUT OPTIONS**

**CARD  
CRT  
PREVIOUS CASE**

**JUNCTURE A**

**B**

**BOUNDARY FORCES**

**FVA:        FVB:  
FHA:        FHB:  
MXA:        MXB:  
FVAE:       FVBE:  
HAE:        HBE:  
MAE:        MBE:  
CVV:**

**INFLUENCE COEFFICIENTS**

**A11:        C11:  
A12:        C12:  
A22:        C22:  
A23:        C23:  
A33:        C33:**

**PARTICULAR SOLUTION**

**VIAP:       V3BP:  
WIAP:       W3BP:  
MIAP:       M3BP:  
HIAP:       H3BP:  
DVIA:       DV3B:  
FVIA:       FV3B:**

**NOTE: A signifies a register.**

**Fig. 4-36 Single Item Input**

shell subjected to hydrostatic pressure was presented in Para. 4.1. These will be repeated here for clarity. Geometry is to be a sphere having variable thickness. The thickness variation is to be input via CRT. The surface is subjected to hydrostatic pressure and force boundary conditions are applied. Both are described via CRT. Material properties are also input via CRT. This constitutes all input requirements. An example step-by-step procedure for a computer-aided design environment is shown below:

<u>Step</u>	<u>Procedure</u>
1.	Select INPUT COMMAND.
2.	Select FUNCTION, Input OPTION.
3.	Input Radius of Sphere, Initial Start, and Stop value in appropriate registers.
4.	Select SPHERE.
5.	INPUT graph ordinates for thickness into appropriate registers.
6.	Select CRT input for Thickness function.
7.	Draw in thickness variation.
8.	Repeat steps 5, 6, and 7 for pressure, shear, and temperature. The latter two are set to zero in this example.
9.	Select Input command.
10.	Select Single Item Input option.
11.	Select Material Property.
12.	Select CRT input mode.
13.	Select E and input value.
14.	Select PR and input value.
15.	Select ALP and input value.
16.	Select Boundary Forces.
17.	Select CRT input mode.
18.	Select FVA and input value.
19.	Repeat step 18 for FHA, MXA, ... , CVV and FVB, FHB, ... MBE. The juncture does not have to be designated for CRT input since each individual register is selected.
20.	Select Influence Coefficients.
21.	Repeat steps 17 to 19 but for A11 to C33.

Particular solutions for "adjacent structures" was not required as input for this example. All input has now been defined. The next step is to perform the computation. This will be discussed in the following paragraph.

#### 4.6.2 PEANUTS Example: Computation Requirements and Specifications

PEANUTS is a powerful computation tool which employs a finite difference scheme in its solution procedure. In essence, PEANUTS solves the matrix equation

$$(C) (X) = (F)$$

for X, where C is a banded stiffness matrix, X is a displacement vector, and F a loading vector. Currently the program can handle 150 grid points with a bandwidth of 17 elements. Two variables are specified at each point requiring solution of 300 simultaneous equations.

The number of grid points used in the solution procedure can be specified as input. It is customary to use all 150. However, far fewer could handle the job adequately in most cases. The solution procedure uses real numbers with single precision accuracy.

To incorporate PEANUTS into a man-machine interactive type environment requires minor changes to the PEANUTS program. The production program was depicted in Fig. 4-34 and will retain its present character. However, input data flow will have to be routed through a graphic application program for display and modification purposes and then interpreted by an interface program which converts the data into a format acceptable to PEANUTS.

For simplicity, the interface program will retrieve the data from the Display Console, convert it to formatted data in the form PEANUTS normally expects, and write the data onto an input file which is read by PEANUTS as if it were a standard input file. PEANUTS then operates in a normal fashion. Stacked cases must be handled individually. The capability of iterating on the boundary conditions and loads without modifying

the geometry can be handled by proper positioning of the input file. Each case will be displayed and not stacked as is presently done. Output is discussed in the next paragraph.

Computations will be requested by selecting a compute command (COMPUTE). The compute command will have two options: new problem and change loads.

#### 4.6.3 PEANUTS Example: Output Requirements and Specifications

PEANUTS produces considerable output quantities, all of which the user is occasionally interested in. A complete list of possible program output is shown in Table 4-6. This list is far too extensive to be displayed as a standard set. A reasonable list of expected output quantities were discussed in Para. 4-1. These will be considered as the standard set and are repeated here for clarity. Six stresses: SIGX1, SIGX2, SIGT1, SIGT2, SIG1 and SIG2, and two displacements DELTA and V, are shown in Table 4-6. The shell thickness H will be included for convenience.

Output will be achieved by selecting the output command (OUTPUT). This displays the list of possible output quantities shown in Table 4-5 plus an option for the standard set of output discussed previously. A standard output display is shown in Fig. 4-37. External and internal values are displayed on the same graph, e.g., SIGX1 and SIGX2 would be displayed on the SIGX graph. They can be distinguished by different types of lines, i.e., solid and dashed.

It should be possible to display the results of the previous design on the respective graphs of the current design by selecting a superimpose command.

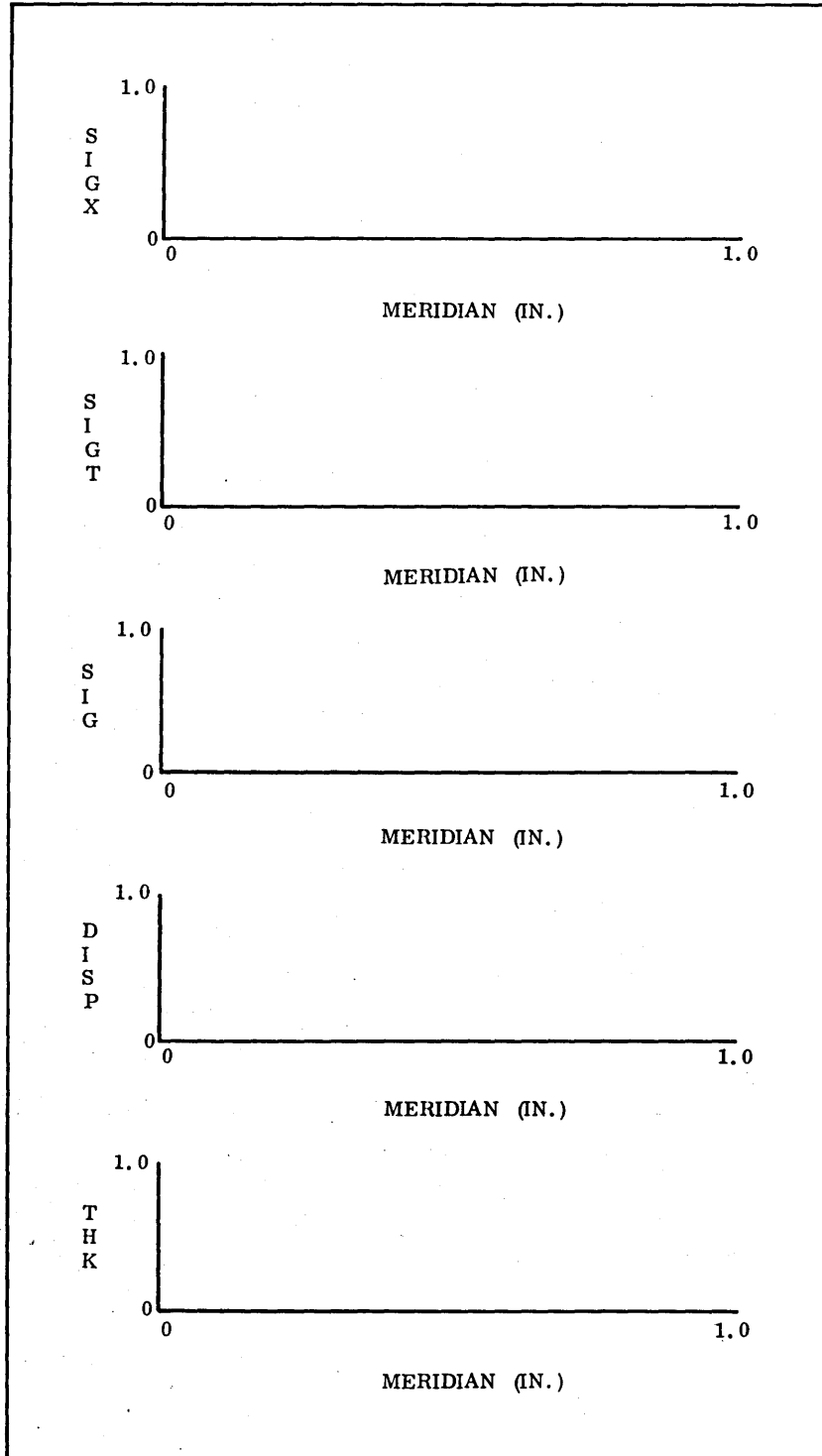


Fig. 4-37 Standard Output Display

#### 4.7 REFERENCES

- 4-1 K. J. Forsberg and S. K. Ferriera, Development of Improved Structural Dynamic Analysis, Volume II, Computer Graphics, Technical Report AFFDL-TR-66-187, Mar 1967
- 4-2 David N. Keast, "Survey of Graphics for Computer-Aided Design," Machine Design, Aug 3, 1967





Section 5  
ENGINEERING DOCUMENT RETRIEVAL

5.1 INTRODUCTION

This section provides the information a computer system analyst needs to perform the detailed design of an on-line system for the retrieval of engineering information.

First, the analyst must know the scope of the problem. Paragraph 5.2 describes the system requirements in terms of current and projected usage of the MSFC Document Repository. The description treats the Repository as a black box and has the form of an input/output specification.

Next, the analyst must know the environment in which the system is to operate. Paragraph 5.3 describes the operations of the Repository and the Engineering Release Center, and the way in which they function.

The analyst must then have a conceptual system design which he can translate into programming specifications. Paragraph 5.4, Recommendations, suggests a retrieval system and outlines the programming of some of the major operations implied by the design. It is worthy of mention that a system similar to the one suggested, using several of the same functions, is in existence at LMSC.

Finally, the analyst should consider existing hardware that might be used in implementing such a design. For this purpose, a brief survey of the hardware state-of-the-art is discussed in Paragraph 5.5.

Nothing in the proposed system prohibits the addition of further retrieval techniques, such as coordinate indexing, as auxiliaries. The proposed system might be the first step in the development of a more powerful facility.

For brevity in description, the system developed here will be called REDI (Retrieval of Engineering Design Information).

## 5.2 SYSTEM REQUIREMENTS

### 5.2.1 Current Usage

Any system, if it is to be useful, must be responsive to the needs or requirements of the user. Responsiveness implies that there is a natural and efficient interface between the system and the man or group using the capabilities of the system. From the user's point of view, the "document retrieval system" can be viewed as a black box—the user inputs information to the function and withdraws information from the function or process taking place within the box. The user should not really concern himself with the inner workings of the process or function as long as it is responsive to his needs.

As a first step in this task of engineering document retrieval analysis, the present usage of the Document Repository material will be examined.

A thorough understanding of the present file structures — who is using the Engineering Document Retrieval system at MSFC, and how they use the files — is essential before any alternate methods of operation may be suggested.

Although this approach is not really directed at the document retrieval requirements per se, it will bring into focus the present methods of operation. An analysis of the file dynamics, however, (input, data manipulation and output) should provide a feeling for the system requirements, regardless of the detailed functioning of the Repository Operation. System requirements for a document retrieval system define the following parameters:

- Input
  1. Originating source of information
  2. Type of information
  3. Rates
  4. Physical form of information

- Processing Function
  1. Response time
  2. Accessibility of information (file structure)
  3. File maintenance
- Output
  1. Recipient of information
  2. Type of information
  3. Rates
  4. Physical form of information

Note that user requirements are in no way tied to physical hardware or detailed processing schemes. The requirements placed on the document retrieval system by the user will, of course, affect hardware selection. Response time requirements, for example, strongly affect the choice of hardware needed for retrieval of stored material. A thorough systems analysis would begin with the user asking "What is the data to be retrieved? How much is there, and how should it be indexed?" The techniques suggested here are submitted to improve rather than revolutionize an existing operation at MSFC in the form of a Document Repository and an automated Engineering Order (EO) System under the direction of the Release Center. This study will therefore build upon the capabilities of the present Repository operation and EO data base and hopefully lead the way to an improved and expanded service in fulfillment of the mission of an engineering document retrieval system.

5.2.1.1 Input. This discussion provides a comprehensive review of the information that is input to the Repository. The following questions will be explored: Who inputs material, what does he input, when, and how much, and what is the physical form of the material?

Originating Source of Information. The following three groups comprise the user population that will enter information into the Central Repository:

- MSFC Personnel
- Contractor Personnel
- NASA Intercenter Personnel

The following paragraphs will identify the users within the groups listed above.

MSFC Submittals. MSFC personnel may input material originally generated by in-house MSFC projects or by contractors. The October 1967 MS-D Input Chart (Central Repository) supplied by the Repository personnel shows the following MSFC groups as having input material to the Repository during this month:

<u>Laboratory</u>	<u>Branch of Laboratory</u>				
ASTR	-BX	-EA -ESE -ESS	-GD	-ND	-ME
P&VE	-SAE -SSV	-VSD -VSL	-XX		
AERO	-AB -AD	-DA			

The input records for the Repository show that the Engineering Release Center (P&VE-VSD) handles a large share of the documents input to the Repository from MSFC activities.

Data supplied by the Engineering Release Center indicates that the following Laboratories/Branches submit documentation through P&VE-VSD:

<u>Laboratory</u>	<u>Branch of Laboratory</u>		
P&VE	-P	-S	-VM -VV
ASTR	-E	-M	
ME	-ME		
AERO	-E		
QUAL	-P		
TEST	-D		

Contractor Submittals. The list of contractors submitting documentation to the Central Repository directly or indirectly through a MSFC laboratory/branch monitoring agency is very large. To give a specific example, reference is made to the October 1967 MS-D Input Chart (Central Repository) which lists the following contractors:

Bendix	Hayes International
Boeing	IBM
Chrysler	North American
Douglas	Pratt & Whitney
Fairchild	Rocketdyne
General Electric	Sanders Associates
General Motors	Thiokol
Greer	Walter Kidde
Grumman	

Different divisions within the above listed companies also contribute documentation independently.

There are many other contractors not listed that have at some time contributed to the data base at the central Repository.

NASA Intercenter Submittals. Intercenter submittals may originate from Kennedy Space Center (KSC) or the Manned Space Center (MSC) in Houston, Texas. The content of these data submittals will be discussed in the following paragraphs.

Type of Information Submitted. There are at least 85 different types of documents handled by the Repository. In order to avoid becoming enveloped in too fine a level of detail within these classifications, only four categories will be discussed here and subsequent discussions of rate and file structure will be limited to these. Slightly more detailed subdivisions are shown for illustrative purposes only. These data were extracted from MS-D Input Chart (Central Repository) for the month of October 1967.

<u>Major Category</u>	<u>Subdivision</u>
1. Drawings and Control Documents	Drawings CEI Packages Engineering Orders (EOs) Documentation Release List (DRL) Engineering Master Parts List (EMPL) Generation Breakdown (GB) Interface Control Document (ICD) Interface Revision Notice (IRN)

<u>Major Category</u>	<u>Subdivision</u>
2. Reports	Test Trajectory documents Status
3. Specifications	Saturn-V Display System
4. Standards	

Rates for Submittals. The statistics gathered on input rates fall into two categories.

1. Input rates for the document Repository from all sources
2. Engineering documentation sent to the Repository by the MSFC Release Center (P&VE-VSD).

Category 2 above is included in the rates of Category 1. The release center activity was analyzed separately since any engineering design/graphic interface would most likely affect the release center operation. The release center is the agency concerned with the checking, formalizing, and controlling of official drawing and control documentation in support of the in-house MSFC design groups.

The information used here is extracted from complete statistics normally maintained by the Central Repository personnel. Figure 5-1 depicts the average monthly input activity for FY 1965 - 1967 and shows the actual monthly rates for FY 1967.

Figure 5-2 shows the release center activity for the period January 1967 through October 1967. Four subcategories of information are shown on this chart, namely:

1. Drawings
2. Engineering Orders (EOs)
3. Documentation Release List/Engineering Parts List (DRL/EPL)
4. Interface Control Document/Interface Revision Notice (ICD/IRN)

The extent of automation (computer programs) in the release center operation will be discussed in Paragraph 5.3 Storage and Retrieval Systems.

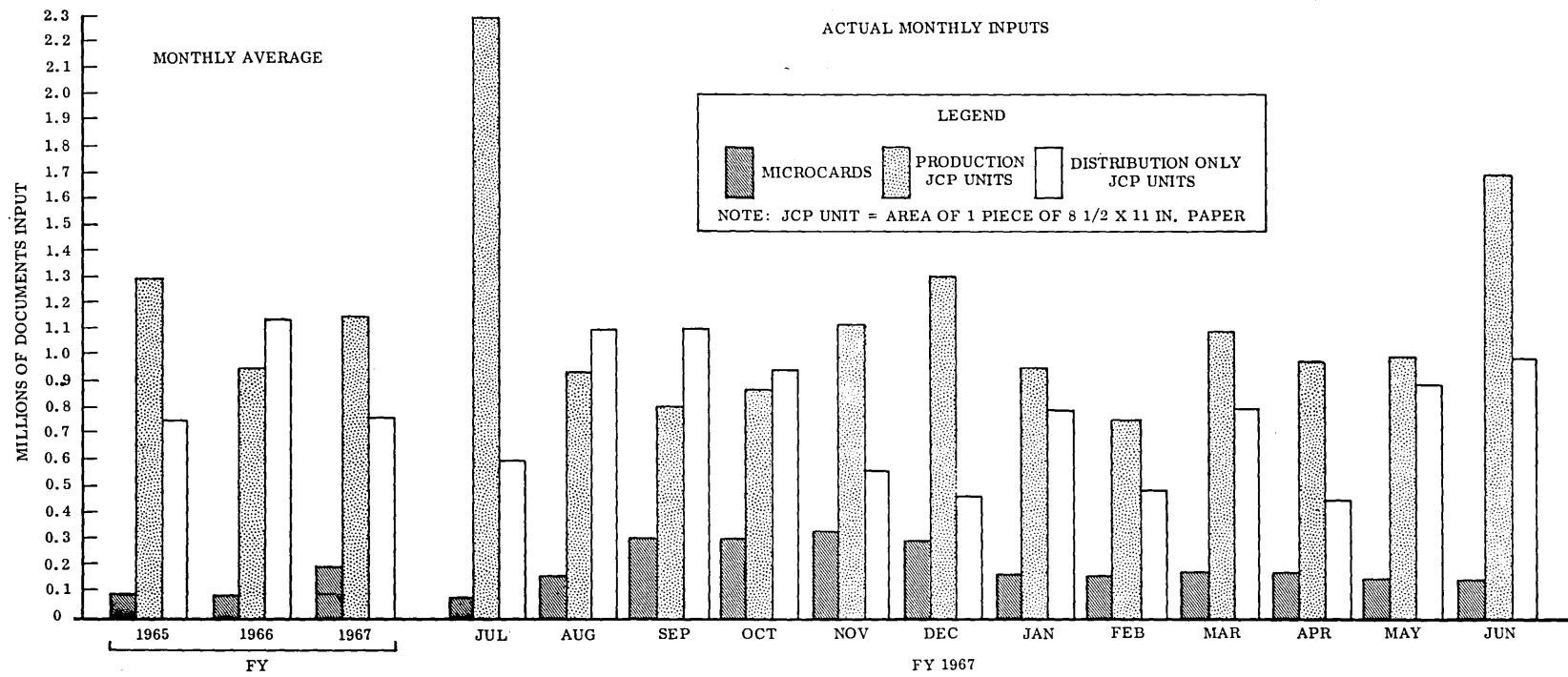


Fig. 5-1 Documentation Repository Input

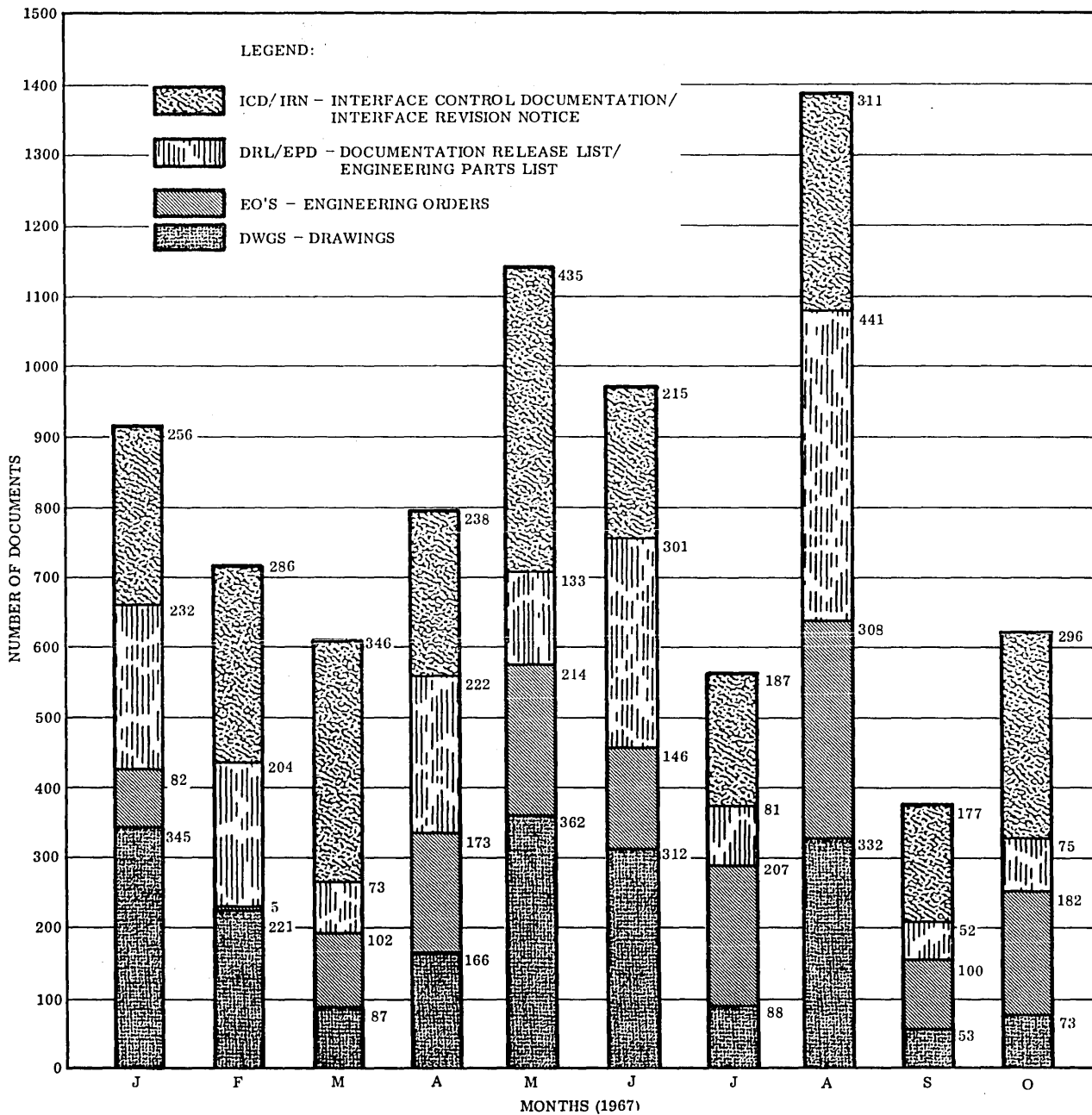


Fig. 5-2 Engineering Documentation Released by MSFC Release Center and Forwarded to Central Repository



Physical Form of Information Submitted. MSFC Administrative Regulation 25-9 outlines the procedures for submission of documentation to the MSFC Central Repository. Acceptable media are:

- Drawings and Associated Documents
  1. Original vellum
  2. Aperture cards
  3. Microfilm roll
  4. Black and white vellum reproducible
  5. Hard copy black line
- Reports, Manuals, Specifications, etc.
  1. Original manuscript
  2. Black and white vellum reproducibles
  3. Printed copy, black line
  4. Microfiche

5.2.1.2 Processing Functions. This portion discusses the user requirements imposed on the processing function or storage/retrieval system operation. The user is concerned with three main attributes of the processing function.

- Response time of "function"
- Accessibility of data/information
- File updating (currency of information)

The above three points are very important. The degree of user confidence that a repository can adequately meet the requirements in the above areas determines the extent to which he uses the service. If users are not satisfied with response times, cannot find data in the file, or are deluged with out-of-date material, then small private file stores will build up and eventually annul the benefits of a central repository.

Response Time Requirements. This is a very subjective area of discussion. Some engineering personnel feel they need repository material immediately while others are willing to accept longer response times. Project schedules and tight deadlines do, of course, sometimes force data requests to have high priorities.

To provide a system that is as flexible as possible and that is responsive to the needs of its users, two modes of requests should be honored: normal response and high priority response.

To match these two desired modes of response, the central repository maintains a 24-hr response for small requests (five documents or less). Requests for large quantities of data (either many single sheet items or one document having many pages) may take 36 to 48 hr or longer depending on the workload of the printing facilities required.

Direct acquisition of copies of documents from satellite microfilm files is possible if the user has an urgent need for the data. The aperture cards may be retrieved by repository personnel and the user may view and/or copy the document if desired.

Mode of document delivery also affects system response time. Various methods of transmitting the document are available such as the following:

- Call when ready
- Will pick up
- Delivery
- Mail

Accessibility of Information. This topic is the heart of a document retrieval system. The user must first know what is in the data base and secondly be able to ask for it in terms of his goal. There is a major difference between a document retrieval system and an information retrieval system. It must be recognized that the repository is primarily a clearing house for documents and therefore is a document retrieval system. Its mission does not include providing information from the documents it handles, but rather providing the documents themselves.

Engineering design activities are functionally oriented: for example, an engineer may be given the task of designing a relay or a motor. The topic of indexing and how to find out what is in the file relevant to his specific task will be discussed at length in the next sections of this report and in the section on retrieval of design data.

User requirements for data accessibility will be divided into three areas.

- Physical accessibility
- Procedural accessibility
- Document content accessibility

To meet the needs of physical accessibility the Repository files should be physically located near the user. To this end the Repository has provided one main work order desk at MSFC and five satellite work order desks – four located at MSFC and one in downtown Huntsville, chiefly for Boeing/Chrysler use.

Procedural accessibility deals with the various methods of requesting services. Two main classes exist here, namely automatic (regular distribution) and individual (request). In the automatic mode of distribution, lists of addresses keyed to document types are predetermined by laboratory, stage, or project management. Entry of these documents into the Repository automatically triggers the reproduction and distribution of the required data.

Individual requests may be entered by phone, mail, or in-person via MSFC Form 433. Document content accessibility is presently limited to document number and revision. There are master index lists (by document number) that are located at each Repository work order desk to enable the user to determine whether a given document number is in the file. This index also contains a description of the document.

File Updating. A file, to be useful, must contain current information. There must be a minimal time lag between entry of data into the Repository and the capability to withdraw that data. The specifications for the variety of hardware used to expose, process, index/code, duplicate, and store the documentation are known but no statistics are currently available on cycle time (entry to storage) for typical packages of data.

Unless superceded data are periodically purged from the file, the data base will continue to expand in size and response time (search time) will become longer. All associated support activities such as the preparation of new index lists and automatic distribution activities will also increase. To keep the overhead support under control,

some scheme of file purge or relegating data to "history" files must be implemented. If the latest revision of a drawing is E, for example, it would not be desirable to maintain Rev. A on immediate access files (assuming, of course, that Rev. A is truly history and not being used in some current model of hardware).

The Central Repository scheme for file purging depends entirely on the contractor. Each contractor is asked periodically if certain documentation has been superseded and if so the data are transferred to the supply warehouse and inactive files.

**5.2.1.3 Output.** This section discusses the characteristics of the material that is output from the Central Repository. The method of discussion covers the same building blocks as the input section, namely: who receives the information, what kind of data is withdrawn from the files, the rates of output, and the physical form of the output material.

**Recipient of Information.** As in the case of input material, there are many groups of personnel receiving data from the files. The Cost Center Usage Report for October 1967 shows that 325 separate groups (cost centers) within the several laboratories at MSFC and other NASA centers or facilities requested material from the Repository. Contractor usage statistics are reflected in MSFC Industrial Operation usage statistics.

**Types of Information.** MSFC Form 433 is used for requesting documentation from the Repository. The different types of information as enumerated on the Form 433 is as follows:

1. Drawings
2. Engineering Orders (EOs)
3. Parts Lists
4. Specifications
5. Reports
6. Interface Control Documentation (ICD)
7. Interface Revision Notice (IRN)
8. Documentation Release Lists (DRL) or Engineering Assembly Parts Lists (EAPLs)

Note that the requests (Form 433) are really for documents, not information per se. Exactly what information is extracted from these documents to accomplish the users' task has not been examined because that information is not available from the Form 433.

5.2.1.4 Rates of Output. MSFC is subdivided into two main groups – Research and Development (in-house government personnel) and Industrial Operations (contractor interface). These groups are commonly referred to as R&D and IO. The cost center usage report for October 1967 shows the usage rates for the R&D and IO classifications. The item "JCP Unit" means the area of one 8-1/2 in. x 11 in. sheet of paper. Drawings, for example, may contain several JCP units. Table 5-1 provides a typical summary of usage.

Table 5-1

SUMMARY OF R&D AND IO USAGE

User	Activity	October (This Month)	September (Last Month)	April – October (Year to Date)
R&D	Documents	46,639	44,139	421,762
	JCP Units	956,774	1,548,390	8,948,331
	Cards Handled	712,147	506,152	3,574,972
IO	Documents	18,893	19,290	180,133
	JCP Units	1,099,577	920,322	6,632,853
	Cards Handled	767,977	353,475	2,097,300
R&D and IO	Documents	65,532	63,429	601,895
	JCP Units	2,056,351	2,468,712	15,581,184
	Cards Handled	1,480,124	859,627	5,672,272

Figure 5-3 shows the total output for the period FY 1961 – FY 1967.

As a final example of output rates, a tabulation of the six most active requestors for the month of October 1967 is shown in Table 5-2.

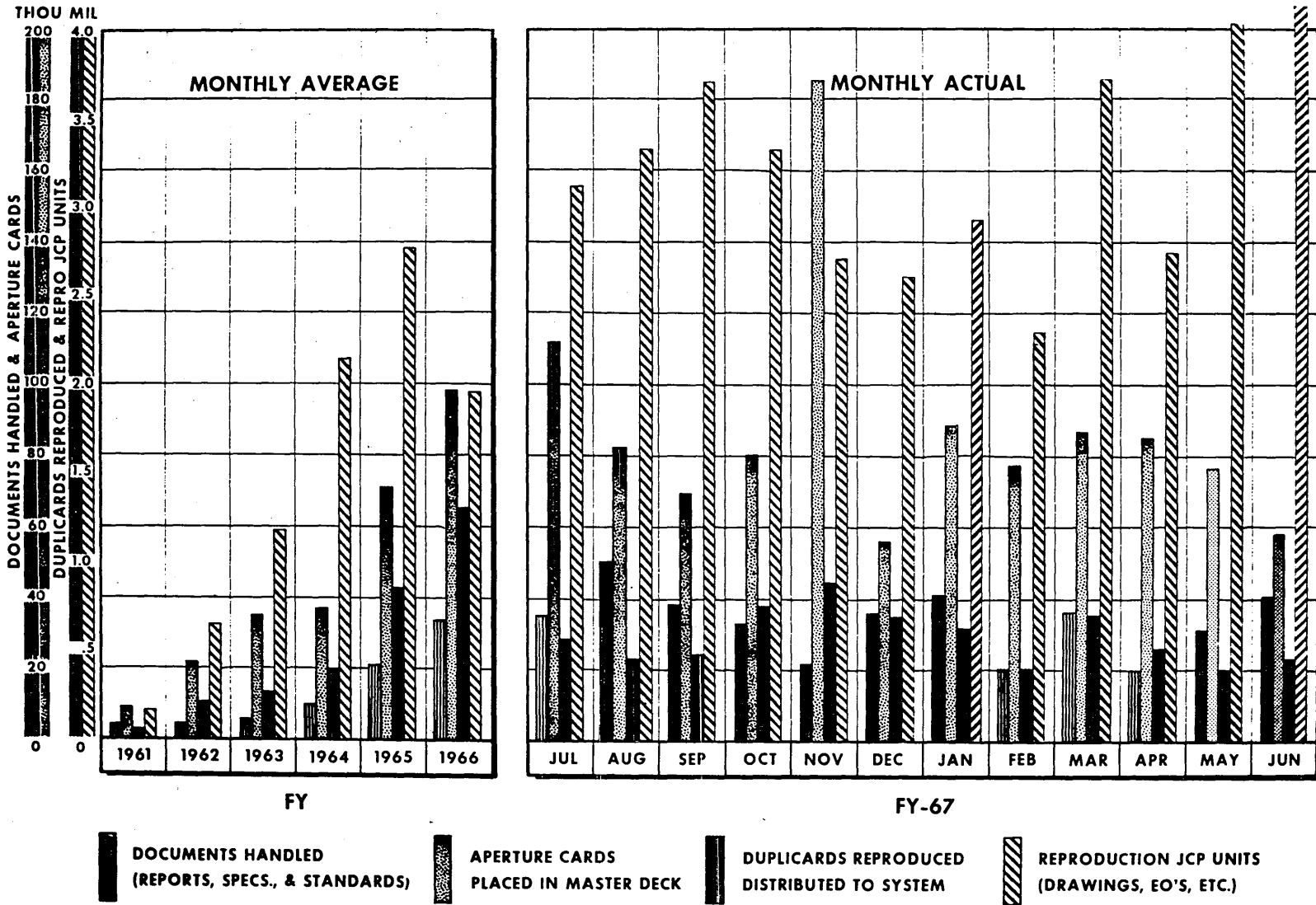


Fig. 5-3 Documentation Repository Output

Table 5-2

## SIX MOST ACTIVE REQUESTORS - OCTOBER 1967

Order of Usage	Documents			JCP Units			Cards Handled		
	Cost Center	Group	Number	Cost Center	Group	Number	Cost Center	Group	Number
1	2000	R&D Operations	24.5k	8501	Industrial Operations	831.7k	8501	Industrial Operations	676.2k
2	8501	Industrial Operations	15.0k	2000	R&D Operations	271.5k	6432	Quality Laboratory Systems Analysis	362.4k
3	5000	P&VE Laboratory	3.2k	5000	P&VE Laboratory	205.7k	5001	P&VE Lab. Director	103.2k
4	3311	Planning and Engineering	2.7k	9520	SAT V Systems Eng.	121.6k	5102	P&VE Vehicle Systems	90.7k
5	1261	Facilities and Design	2.2k	2920	KSC Launch Support	79.6k	9100	Contracts Office	81.8k
6	5134	P&VE Systems Design	1.9k	1261	Facilities and Design	34.2k	8001	Test Laboratory Director	39.7k

5.2.1.5 Physical Forms of Output Material. There are three basic physical media output from the Central Repository:

- Hard copy (paper)
- Microfilm aperture card
- Microfilm roll

All data contained in the file may be reproduced on paper. This is the main output mode for the repository. Some users may request and receive duplicard distribution as an output, i.e., duplicate decks of microfilm aperture cards. A positive microfilm is sent to the Archives for all data input to the Repository.

## 5.2.2 Projected Usage

This section discusses anticipated future usage of the Document Repository/Engineering Release Center Operation. There are two different assumptions that might be made relative to the mode of Repository operation at MSFC.

- Continue the present mode
- Augment the engineering design/document retrieved operation with on-line consoles

5.2.2.1 Present Mode of Operation. Under the first assumption above, the type of data handled, the user population, and physical media of input are expected to remain the same. New subject matter will of course be created; the Apollo Application Program (AAP) is an example. New contractors will begin to use the Repository; the Martin Company, for example, will now be a prime input source for AAP material. Because of random fluctuations, it is difficult to extrapolate past Repository usage to arrive at a meaningful projected rate.

In the effort to project usage of the Repository/Engineering Release activity at MSFC reference was made to "1968 NASA Authorizations," hearings before the Subcommittee of Science and Astronautics, U.S. House of Representatives, March 14 to 16, 20 and 21, 1967. Several statistics from those hearings were used for predicting a projected activity rate. The following illustration (Fig. 5-4) was extracted from the hearings.



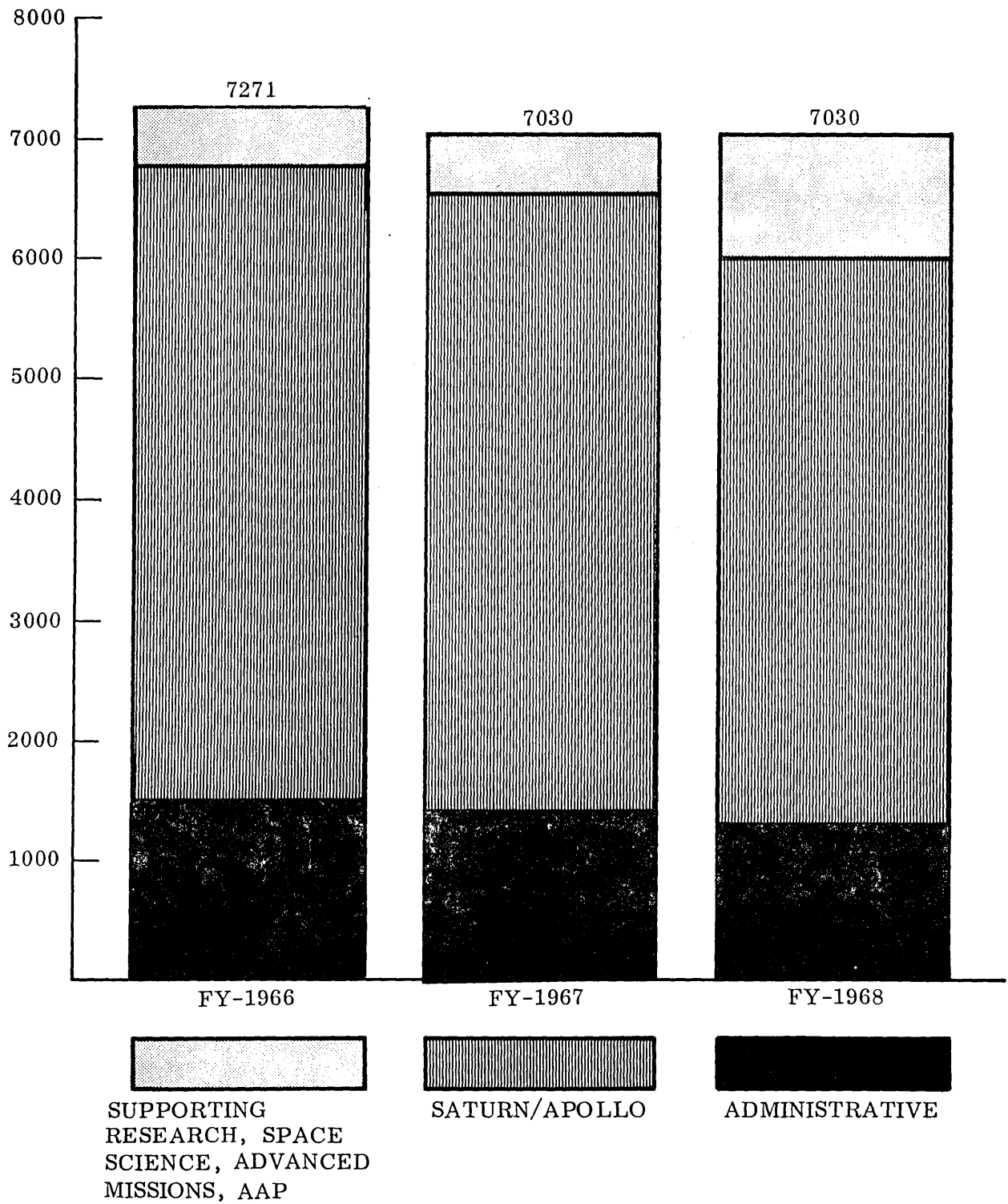


Fig. 5-4 Program Distribution of Manpower at MSFC

The Research and Development appropriation increase for MSFC for FY 1968 is 4.3 percent. This increase should cause a corresponding increase in the documentation activity of MSFC.

Figure 5-5 shows the AAP documentation for the period June 1967 to October 1967 as processed by the Engineering Release Center.

The AAP is one of the major next generation efforts of MSFC and should produce a large volume of engineering documentation. The fluctuating statistics for the AAP-released documentation is an indication of the problem of predicting the rate of activity for the Engineering Release/Repository operation.

5.2.2.2 Graphic System in Operation. For the next few years it is foreseen that the computer graphics design effort will have a small impact on the total volume of activity of the Repository and Release Center. The effect of this innovation will be felt more in the way documents are retrieved.

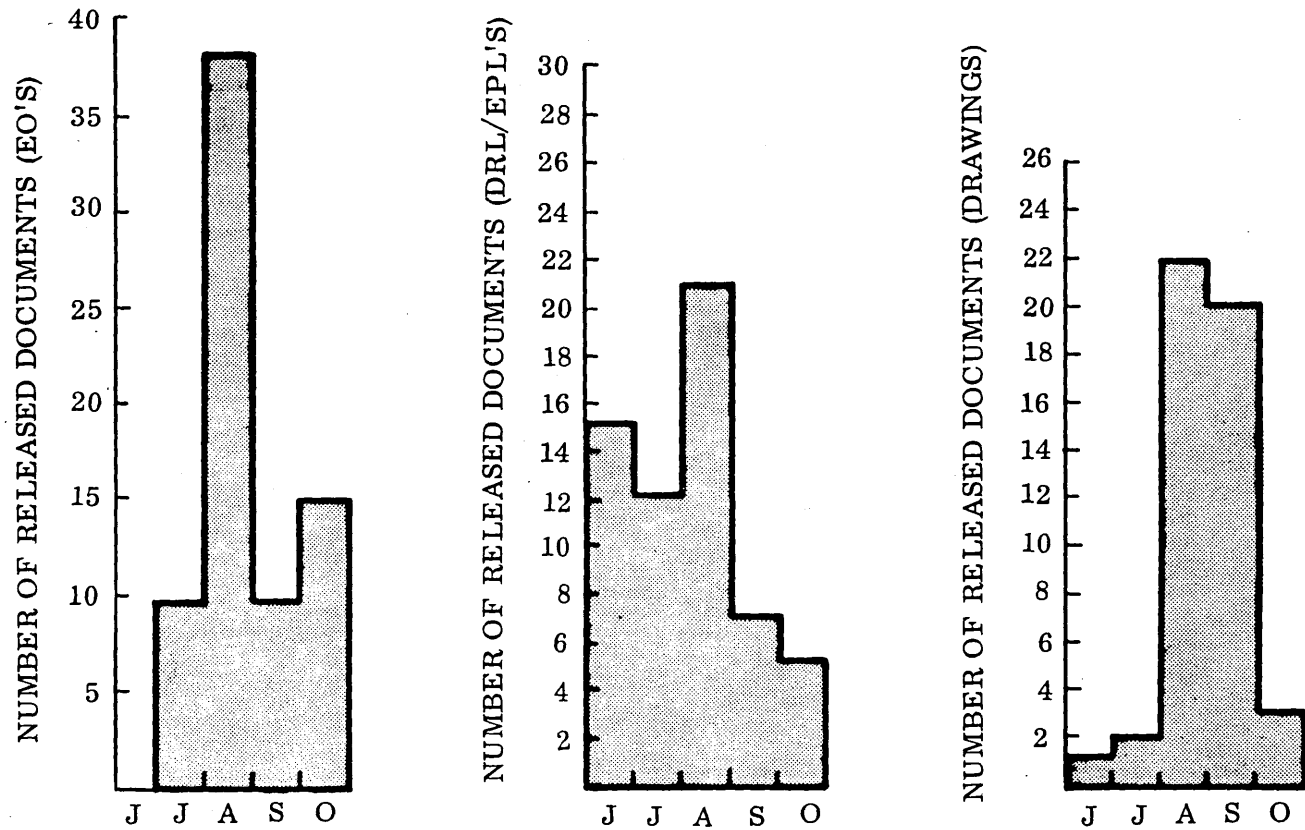


Fig. 5-5 AAP Documentation Processed by MSFC Engineering Release Center Between June and December 1967

### 5.3 STORAGE AND RETRIEVAL SYSTEMS

The previous portion of this study examined the MSFC user requirements with respect to data input to, processing within, and output of material from the Central Repository. This section will discuss the following storage and retrieval systems at MSFC:

- The Central Repository
- Engineering Release Center
- The Saturn V Operational Display System
- The Ampex Videofile System

This investigation is reported because, before system integration or modification can be suggested, an understanding must be attained of what is presently in being at MSFC.

#### 5.3.1 Central Repository Operation

This paragraph describes the function of the "black box" (Central Repository). Paragraph 5.2, User Requirements, defined what data was input to and output from the function and those few constraints imposed by the user on the operations within the box. The specific data flow within and the attributes of the function will be reviewed. The following items will be covered:

- Documentation flow
- File structure
- Configuration of facilities
- Operating procedures/forms
- File storage media

5.3.1.1 Documentation Flow. Figure 5-6 shows the documentation flow within the Central Repository. This flowchart is self-explanatory and shows clearly the internal operations performed by the Repository personnel. In summary, there are six operations performed by the Repository, namely:

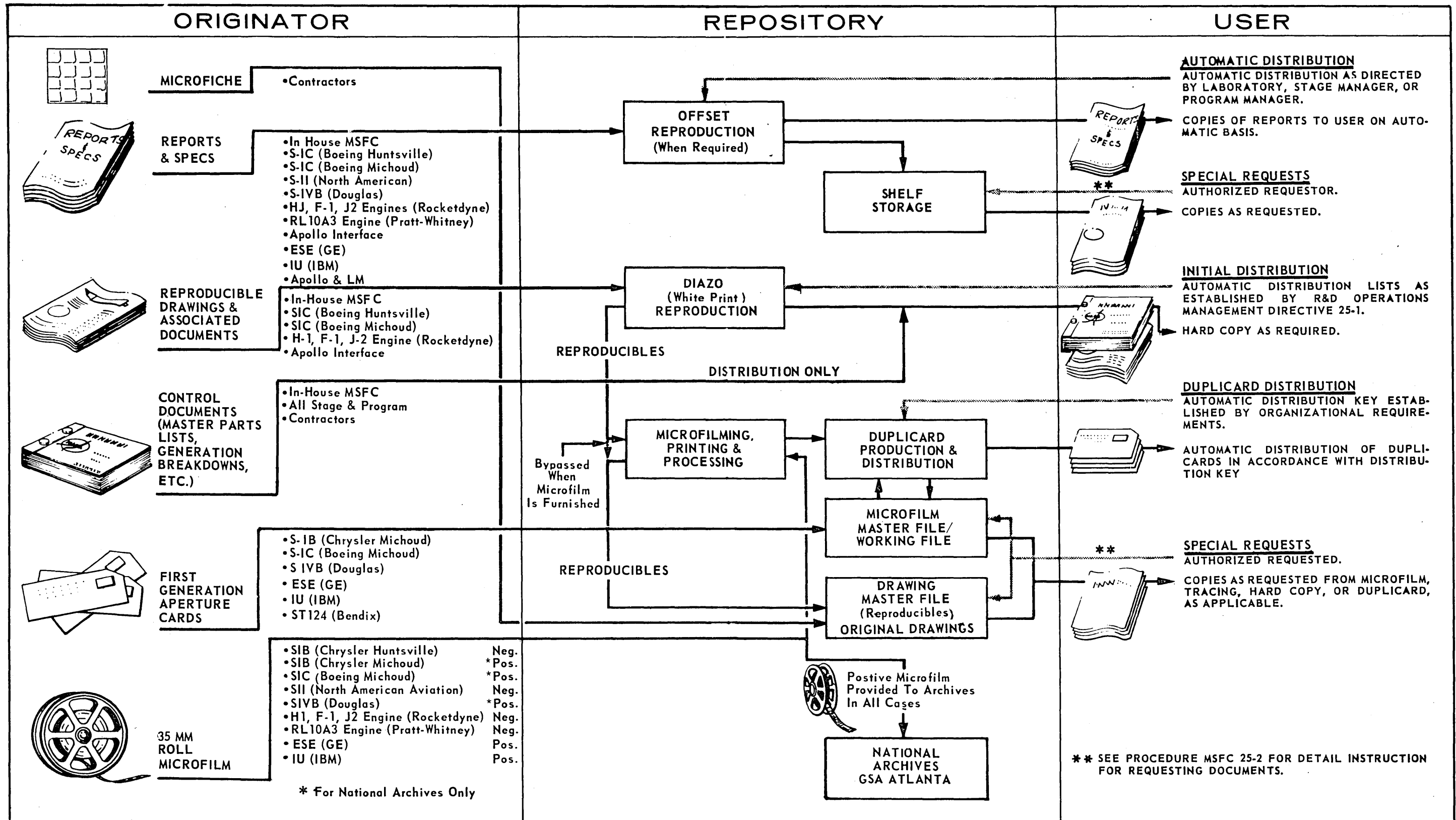


Fig. 5-6 Documentation Flow

- Photographing
- Indexing/Coding
- Duplicating
- Retrieving
- Processing
- Storing

5.3.1.2 File Structure. There are two main classes of documents retained by the document Repository: Class I – those documents originated by the government following standardized formats and bearing numbers controlled by the government, and Class II – those documents originated by participating contractors and having numbering systems managed by the contractor.

The numbering system for Class I documents uses eight characters. The first two specify the Government organization at MSFC that originated the document, the third denotes the NASA center (constant M for MSFC) and the last five are the serial number of the specific document. Contractors are sometimes requested to supply Class I type documentation and in those cases the first three digits are 6XY, where the X can be 0, 5, or 6 and the Y can (for example) be B or C (Boeing or Chrysler).

Following is an example of the numbering scheme for the internal MSFC organizations.

<u>Drawing Prefix</u>	<u>Organization Code</u>
10M	R-P&VE-VV
20M	R-P&VE-P
30M	R-P&VE-S
40M	R-ASTR-E
50M	R-ASTR-M
80M	R-AERO-E
90M	R-TEST-D

The numbering scheme is more detailed than indicated in the above example; the intent of the example is to depict the general indexing scheme used by in-house MSFC organizations.

Technical documentation other than drawings (i.e., procedures, specifications, and standards) originated by MSFC organizations has the following indexing scheme.

<u>Subject</u>	<u>Prefix</u>
Procedures	MSFC-PROC-(000)
Specifications	MSFC-SPEC-(000)
Standards (Book form)	MSFC-STD-(000)
Standards (Sheet form)	MC-(000)

Class II documents have contractor-managed numbering schemes. There has been no attempt to standardize the indexing schemes originated by the several contractors; further, in filing and indexing Class II documents, the Repository does not apply any additional index numbers over and above the contractor index.

The Repository files all drawings and associated documentation (both Class I and II) in numerical sequence by vehicle stage within major categories such as S-IC, S-IVB, etc. Specifications, standards and reports are indexed by numerical sequence and originator such as MSFC, contractor, etc.

The present size of the files are as follows:

<u>File Type</u>	<u>Size</u>
Drawings (Aperture Cards)	2.5 million
Reports	6,000.
Specifications	30,000.
Hard Copy Vellum Drawings	100,000.
Inactive File (Aperture Cards) (Warehouse - Building 7207)	1.5 million

5.3.1.3 Configuration of Facilities. Facilities of the Central Repository activity are located both at MSFC and in downtown Huntsville.

The main repository operation is located at Building 4494 MSFC. Activities at that location are receiving, inspecting, photographing, reproducing, and filing. The main Work Order Desk is also located there.

The Huntsville Industrial Center (HIC building) is located near downtown Huntsville and performs the Repository operations of microfilming, producing duplicard, and filing.

The main reports and specifications file, the Master Microfilm File, and a Microfilm Satellite File (mainly for use by Boeing and Chrysler) are maintained at the HIC Building.

There are four other satellite files located on MSFC in addition to the main file (Bldg 4494) and the HIC Building. Their locations are as follows:

- Test Laboratory – Building 4666
- P&VE Laboratory (2) – Building 4881
- Astrionics Laboratory – Building T-12 (Trailer)

There are two other facilities that contain Repository documentation, e.g., Classified Storage, Bldg 8497 MSFC, and Supply Warehouse and Inactive Files, Bldg 7207 MSFC.

The backup/disaster files are located at the National Archives in Atlanta, Georgia. Positive roll microfilm is provided to the Archives for all material entered into the master Repository files.

There are at least 200 items of hardware used in the internal operations of the Central Repository. A partial list of the major types with some of their characteristics follows

<u>Item</u>	<u>Description</u>
Card punch	Model 026
Card to card printer	Model 041, 2000 cards/hour
Reader/printer	3M and other models
Collator	Model 189, 650 cards/minute
Semiautomatic mounter (microfilm onto aperture cards)	3000 cards per day
Card interpreter	Model 557, 6000 cards/hour
Contact printer	40 x 60 inch size capability



<u>Item</u>	<u>Description</u>
Electrostatic copy machine	
Flat bed copy camera	
Offset master generator	180 masters per hour
Offset press	
Diazo reproducer	42 inches wide capability
Automatic drawing folder	
Microfilm camera	45 x 63 inches maximum drawing size
	Reduction ratio variable 12x to 36x
Rotary aperture card files	Diebold-200 to 400K capacity
	Remington Rand
IBM 1401 Computer	Index list preparation master magnetic tape index generation
Files (general)	Map, letter size, shelf, etc.
Microfiche reader/printer	

5.3.1.4 Operating Procedures/Forms. These items are MSFC administrative regulations and procedures that define the operations of the Central Repository. They are:

MSFC 25-9 – Input of Documentation to the MSFC Central Repository

MSFC 25-2 – Requests for Documentation From the MSFC Central Repository

Figures 5-7 and 5-8 are, respectively, "Document Input Record," MSFC Form 2896, and "Request for Documentation," MSFC Form 433.

Another important item of user control documentation is the "Distribution Data Acquisition List," MSFC Form 2598. (See Fig. 5-9.) This describes the procedure to be followed by project, laboratory, or stage managers to establish automatic distribution lists.

MSFC FORM 2896 AND INSTRUCTIONS  
FOR COMPLETING IT

DOCUMENT-INPUT RECORD

SUBMITTAL DESCRIPTION: ①		CONTROL NUMBER: ②	DATE SUBMITTED: ③
SUBMITTING ORGANIZATION: ④	PERFORMING ACTIVITY AND/OR CONTRACT NUMBER: ⑤	SUBMITTAL AUTHORITY: ⑥	RELEASING AUTHORITY: ⑦
SPECIAL INSTRUCTIONS: ⑧			
SUBMITTAL INCLUDES: ⑨  <b>SAMPLE</b>			
TO BE COMPLETED BY MSFC CENTRAL REPOSITORY AND A COPY RETURNED TO SENDER			
RECEIVED BY: ⑩		DATE RECEIVED: ⑪	WORK ORDER NUMBER: ⑫

MSFC Form 2896 (March 1967)

BLOCK NUMBER	TYPE OF INFORMATION OR EXPLANATION
①	Identification of the material being submitted (example - MSFC Drawing Package, Apollo ICD, or 67-FMP2 AS-501 "Joint Oper. Traj.")
②	Any identification number utilized by the submitting organization (example - Release Package Number, Documents Requirements List, line item number, etc.)
③	Self-explanatory
④	Office of origin or office of control
⑤	Performing activity code as defined in Annex E of Volume II of the Financial Management Handbook, or Contract Number if a contractor submittal
⑥	Reference document that directs or authorizes the submittal of data to the repository, (example - MSFC Drafting Manual, Engineering Bulletin, Program Directive No. 6 from I-V, Appropriate Contractual Data Submittal Document, etc.)
⑦	For Class I Documentation of R & D Operations Organizations only. Enter in this block the official release authority
⑧	Any special handling or distribution requirements
⑨	Complete identification and quantity of data submitted
⑩, ⑪, & ⑫	To be completed by MSFC Central Repository

Fig. 5-7 Sample of MSFC Form 2896 - Document Input Record

CUSTOMER REFERENCE NUMBER: (Used as Required) 11-48		<b>REQUEST FOR DOCUMENTATION</b>			FOR USE OF MS-D ONLY WORK ORDER NUMBER: 00 2773							
REQUESTED BY: H. S. Garrett		PHONE NO.: 876-1673	BLDG. NO.: 4746	DATE REQUESTED: 11/28/66	DATE NEEDED: 11/29/66							
ORGANIZATION SYMBOL OR NAME OF CONTRACTOR: MS-D			It is certified that the documents are for the official use of, and properly chargeable to, the stated activity:									
COST CENTER CODE: 1410		CONTRACT NUMBER:			SIGNATURE OF REQUESTING AUTHORITY							
PROGRAM (3-DIGIT CODE)		MATERIAL REQUESTED			INSTRUCTIONS							
		<input checked="" type="checkbox"/> DWGS <input type="checkbox"/> SPECIFICATIONS <input type="checkbox"/> DRL OR EAPL <input checked="" type="checkbox"/> ED'S <input type="checkbox"/> REPORTS <input type="checkbox"/> CARDS <input type="checkbox"/> PL OR LM <input type="checkbox"/> ICD <input type="checkbox"/> IRN			<input type="checkbox"/> CALL WHEN READY <input type="checkbox"/> MAIL <input checked="" type="checkbox"/> WILL PICK UP <input type="checkbox"/> OTHER <input type="checkbox"/> DELIVER							
LINE ITEM NO.	DOCUMENT NUMBER	REV.	NO. OF COPIES	NO. OF REPRO.	FOR REPOSITORY USE ONLY							
					SIZE	FRWD.	NOT RELEASED TO MS-D	BACKORDER		FRWD.		
1.	40M39513	A	2			✓		✓				
2.	40M39526	NC		1	B			✓				
3.	36-001M-53 Aero Quip	C	1		C	✓						
4.	50M68475	B		10					✓	09-0621		
SAMPLE												
TAKEN BY:  Lillian Locke		PROCESS DIAZO:    OFFSET:    XEROX: SHELS:    PHOTO:    OTHER: CONTRACTOR CODE: 6    REPRODUCTION TYPE: E				DISTRIBUTED BY:  Frances Cobble						
DISPOSITION OF ORIGINALS		REPRODUCTION INSTRUCTIONS										
<input type="checkbox"/> CUSTOMER <input type="checkbox"/> LOGGING <input checked="" type="checkbox"/> FILE <input type="checkbox"/> FILE <input type="checkbox"/> <input type="checkbox"/> CUSTOMER		INPUT TYPE	CARDS	SHEETS	MATS	REPRO	COPIES		UNITS			
			4			1	TOTAL INPUT					
		COPIES REQ'D	2			1	OUTPUT	9	20			

MSFC - Form 433 (Rev February 1967)

REFERENCE ASEP, MSFC 25-2

Fig. 5-8 Sample of MSFC Form 433 - Request for Documentation

1. DATE:	DISTRIBUTION DATA ACQUISITION LIST										2. PAGE ____ OF ____		
3. TITLE: Plan, First Article Configuration Insp. (FACI) (CM-512)						4. IDENT. NUMBER:		5. CONTRACT NO.: NAS7-101	6. DRL NO.: 021	7. DRL LINE NO.: 028	8. CONTRACTOR IDENT. CODE: 1093617		
9. NAME OF RECIPIENT			10. ORGANIZATION SYMBOL	11. BLDG. NO.	12. ROOM NO.	13. STREET AND NUMBER	14. CITY	15. STATE	16. ZIP CODE	17. TYPE COPY	18. COPIES		
LAST NAME	FIRST INITIAL	SECOND INITIAL											
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													

SAMPLE

DATA MANAGEMENT COORDINATION APPROVAL												
PROGRAM OFFICE		19. NAME:			20. DATE:		DATA MANAGER		21. NAME:			22. DATE:

MSFC - Form 2598 (February 1966)

Fig. 5-9 Sample of MSFC Form 2598 - Distribution Data Acquisition List

**5.3.1.5 File Storage Media.** There are two basic forms of storage media employed by the Document Repository:

- Microfilm – 90 percent of storage
- Vellum/hardcopy – 10 percent of storage

Within the class of microfilm storage there are actually three physical forms:

- Aperture cards
- Microfiche
- 35 mm roll microfilm (positive and negative)

The microfilm files located at the main work order desk and the several working files located at the Satellite File Stations are in aperture card form (standard IBM cards with 35 mm microfilm attached). Microfiche storage is used primarily for reports and specifications. Positive 35 mm roll microfilm is provided to the National Archives GSA Atlanta, Georgia, for the permanent historical records.

### **5.3.2 Engineering Release Center**

The Engineering Release Center is an important contributor to the Central Repository data base. All official engineering documentation originated by in-house MSFC personnel must pass through the release center operation cycle. Some contractors also generate Class I documentation that is release center controlled – Boeing and Chrysler are prime examples.

**5.3.2.1 Release Center Operation.** The MSFC documentation release system and its associated, adjacent operational organizations are depicted graphically in Fig. 5-10. The MSFC design groups within the various laboratories prepare engineering drawings, Engineering Orders (EOs), Documentation Release Lists (DRLs), Engineering Parts Lists (EPLs), specifications, standards, procedures, etc., which are approved technically within their respective organizations, and then forwarded to R-P&VE-VSD branch of the Propulsion and Vehicle Engineering Laboratory for official release. Working groups and panels prepare other documentation, such as interface control

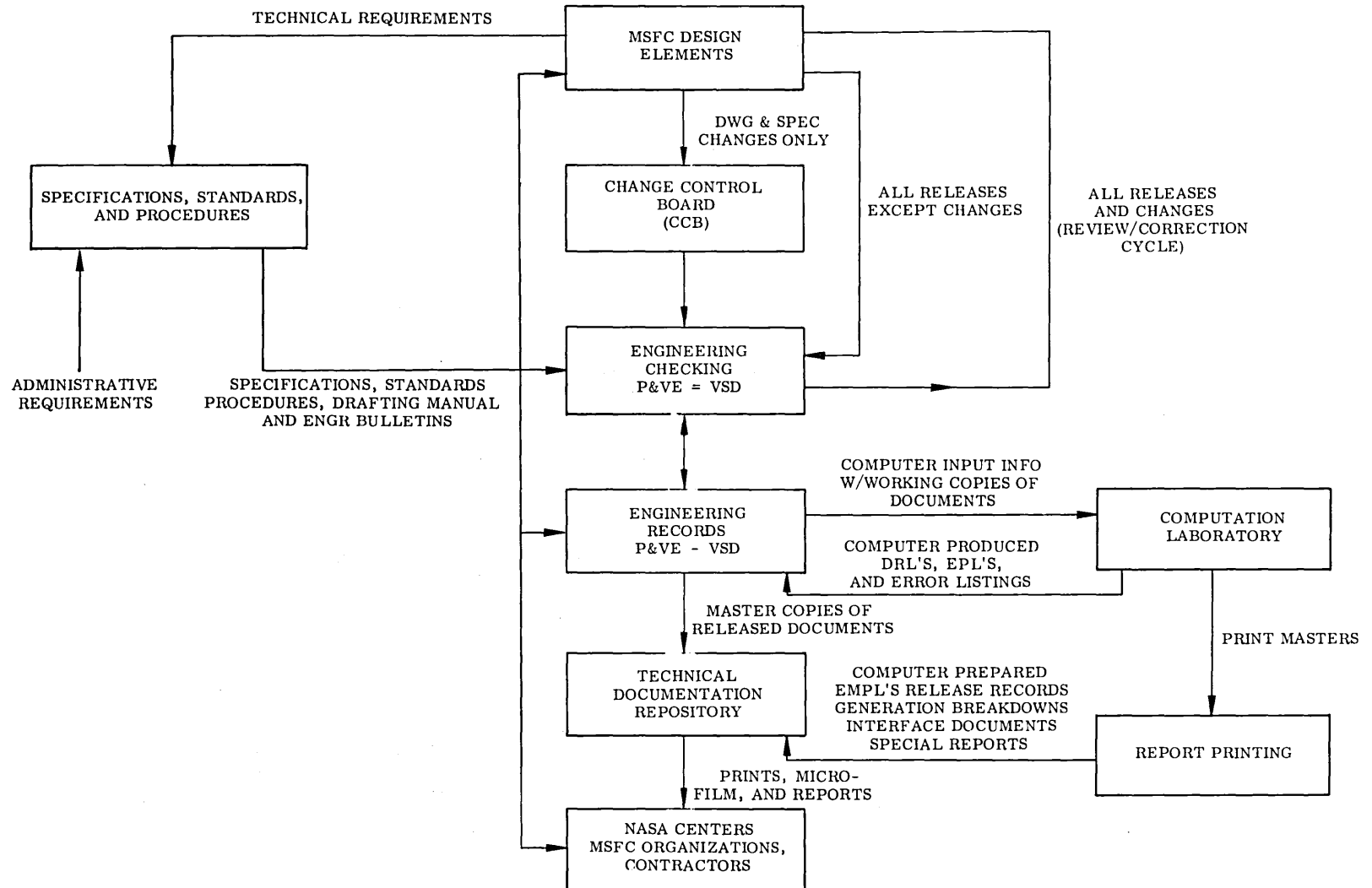


Fig. 5-10 Release and Accounting Procedure Flow

documents (ICDs), ICD input forms, and interface revision notices (IRNs), which, after review and approval by the appropriate Change Control Board (CCB), are also forwarded for release. All such documentation enters the release system at the release clerk's desk in R-P&VE-V, where the documentation record is posted and maintained. Engineering documents – DRLs, drawings, EPLs, and EOs – are then checked for technical accuracy, adequacy, and conformance to all MSFC requirements. Following checking, those documents which lend themselves to a computerized operation – DRLs, EPLs, EOs, ICD input forms, IRNs, etc. – are forwarded to the Computation Laboratory for keypunching and processing by computer programs. Standard edits are programmed to prevent the entry of erroneous data into the computer data base. If an error is detected, the input will be rejected automatically with the reason given for its rejection. If all programmed edits are satisfied, a computer-prepared document (output) is furnished, which is the basis for reports such as Release Records, Engineering Master Parts Lists (EMPL), ICD logs, etc. The computer-prepared output documents are checked against the original input forms for keypunch errors by the release center. The reports are forwarded to Field Printing, where they are reproduced in sufficient copies to meet the requirements of the various user organizations at MSFC, other NASA centers, and contractor locations. Distribution is the responsibility of the Documentation Repository. The repository reproduces as much as it can of the computer-prepared output documents it distributes.

The automated portion of the reporting system, by using the data processing capabilities of the Computation Laboratory, provides many needed services which cannot easily be obtained from an entirely manual system. Some of these services are:

- More rigid and standard edits
- Greater speed of reporting
- Greater variety of reports
- Ease of data search and identification
- Generation breakdowns on main assemblies
- Engineering master parts lists

5.3.2.2 Computer Support to Release Center. Several computer programs are in existence at the Computation Laboratory to support the release center operation. The following list contains a description of the computer-prepared publications within the Engineering Documentation Accounting System.

Advance Release. This publication is a complete listing of Advance Release Documentation Release Lists (DRLs), Documentation Release List and Engineering Parts Lists (DRL/EPLs), and Engineering Orders (EOs) applicable to Saturn I, Block II Vehicles Saturn IB Vehicles, and Saturn V, S-IC Stages. The information listed in the report is obtained directly from advance release documents.

Official Release. This publication is a complete listing of Documentation Release Lists (DRLs), Documentation Release List, and Engineering Parts List (DRL/EPLs), and Engineering Orders (EOs) applicable to the Saturn IB program. The information listed in the report is obtained directly from the officially released documents.

Engineering Report. This publication is a complete listing in alphanumeric sequence, of parts, drawings, specifications, and other engineering documentation applicable to Saturn IB program and Test Instrument Units. The information listed in this report is obtained from officially released Documentation Release Lists (DRLs), Documentation Release List and Engineering Parts Lists (DRL/EPLs), and Engineering Orders (EOs).

Generation Breakdown (GB). This publication lists all parts officially released to define a given assembly for a specific effectivity in generic (Christmas Tree) form. It is published automatically in conjunction with a GB Cross Reference Index. Both are published on request; the specific assembly and effectivity to be generated are supplied by the requestor. The GB Index is an index of all parts called for in a given GB, listed in alphanumeric sequence. The GB Index also supplies the total numbers of each component required for the end assembly generated.



Saturn Interface Control Document Log. This publication is a listing, in alphanumeric sequence by effectivity, of Saturn interface documents and change documents released and scheduled to be released for Saturn I, Saturn IB, and Saturn V vehicles. The information listed in the report is obtained from the Interface Control Document Log Input Form submitted to the Engineering Records Unit by the responsible Interface Working Group.

Apollo Interface Control Document Log. This log is published for NASA Headquarters by MSFC and is a listing, in alphanumeric sequence by effectivity, of Apollo Interface documents and revisions released and scheduled to be released for Saturn IB, and Saturn V vehicles with respect to Apollo. The information listed in the report is obtained from Interface Control Document Log Input Forms (MSDC Form 2053, Rev. January 1966) submitted to MSFC by the responsible Interface Panels.

Configuration Management Accounting System. This system of programs uses as input the Configuration Management Accounting Base Data Input Form (MSFC-Form 2572), the Engineering Change Proposal Input Form (MSFC Form-2573), the Installation Notice Card (MSFC Form 2490) and Configuration Control Board Directives. There are two main output reports, the Configuration Identification Index and the Modification Status Report.

### 5.3.3 Saturn V Operational Display Checkout System

The operation of the Saturn V Operational Display Checkout System is outlined. This system is of interest because of the techniques it uses to retrieve 35 mm film data.

The present system at MSFC Astrionics Laboratory and Quality Laboratory is configured as follows: Remote consoles are connected to a Computer Control Company (CCC) data processor (DDP-224) by means of a Central Logic Unit (built by Sanders Associates). The DDP-224 processor is further connected via a hardware/software data link to an RCA-110A computer which in turn is connected to the Saturn V

function to be checked out. There is a 35 mm slide holder capable of holding 256 slides located in the central logic unit. Provision has been made for expansion to include a second slide bank at a future time.

Each operator console has multiple function buttons and an alphanumeric keyboard. Each operator may select any one of the 256 slides directly from his console. The slide holder is activated by the DDP-224 by software command and positioned to the desired slide. A high resolution video subsystem (945 scan lines to the inch) then transmits the contents of the slide to a cathode ray tube in the operator's console.

In this manual mode of operation, the operator must know the number of the slide of interest. The slide may contain any item of information that the vehicle checkout engineer may need such as; tabulated lists, standards or circuit diagrams. The RCA-110A computer can also select slides automatically. If, for example, a certain checkout test procedure yields results in a range pre-determined as marginal the computer can automatically display the corresponding slide to an operator's console.

The Saturn V Display System is an example of an on-line real-time retrieval scheme in which both manual and automatic selection of slides are available. The file size (256 slides) is admittedly small but, with new, computer-addressable microfilm hardware, an on-line retrieval scheme is quite feasible. Its obvious weakness is in the means of selection of the slide, which assumes that the user knows the relationship between slide contents and location in the slide file.

#### 5.3.4 Mass Graphic Storage System (Ampex Videofile)

Approximately two years ago MSFC obtained a Mass Graphic Storage System, a system more commonly known as a Videofile.

The unit is located at the Brown Engineering facility at Huntsville and was procured to support PRINCE (Parts Reliability Information Center) and APIC (Apollo Parts Information Center).

The unit converts hardcopy into a video tape recording, and adds index coding to the video tape to facilitate retrieval. Available information on parts reliability maintained in the above-mentioned information centers is as follows: inspection reports, failure reports, environmental test reports, material and process reports, etc. These previously mentioned information centers have no interface to the Central Repository or Engineering Release Center operation of MSFC. There is, however, some overlap in the information stored in the two facilities.

The Videofile system is currently a developmental effort and consequently no file (data) conversion, from the existing manual scheme to the video tape recording, has taken place. As a result, no operating statistics are available. The system is rated as capable of driving up to twenty remote consoles, but in this initial configuration only the main input/output console is being used.

Further detail on this system will be found in Para. 5.5.2.3. This unit will be modified with a higher resolution camera.

## 5.4 RECOMMENDATION

### 5.4.1 Engineering Design Data Retrieval – An Overview

If not only the problem of design data but also that of control of the entire cycle of design, procurement, and manufacturing (or assembly) is considered, the MSFC activity lends itself to what is generally called a product-oriented data-base system. In this overall structure, the design data would have a natural place.

Attention is focused on this concept because, as is well known, engineering design does not terminate with the original releases. It has been estimated that, at LMSC, a major portion of the design activity takes the form of changes. These changes in an established design structure must be reflected in activity in procurement and configuration control, changes in testing and inspection procedures, alterations of the drawing structure, etc. and this interdependence clearly calls for a rapid and convenient access to all these related and affected data.

The product-oriented data-base system would contain, for example, all the technical information about the Voyager project as it relates to MSFC; that is, the "product" will be the sequence of Voyager spacecraft. The data-base would include the complete assembly/component structure for the vehicle (each assembly defined in terms of its next-lower-order assemblies and so on down to the part level) complete with tolerances, inspection attributes, pointers to related drawings and to descriptions of components, pointers to related specifications, procedures, and manuals, etc. It would also contain schedule/status data, usage data (the assemblies using and drawings specifying or defining a named component), a map of the drawing structure, etc. From this extensive and constantly updated data bank, specialized programs would provide answers to questions posed by the functions of design, procurement and inventory control, assembly (configuration control, status, etc.), and product assurance (at all levels between part and end-item test and checkout).

This data-base system would be an ambitious project but it is now both technically feasible and economically justifiable. Its design is beyond the scope of this study,

but its potential advantages indicate a strong possibility that such an approach may be realized by MSFC within the next few years. The storage and retrieval system specified in this section would constitute a major part of such an overall data base system.

There are two storage and retrieval problems within the scope of this study. The first, which is the subject of the system design, is the large-scale problem of dealing with engineering information at the stage of completed, formal "packages" of drawings, specifications, test and inspection procedures, etc. This system does not assume that all or even a large part of its end-item data is in computer storage. What is retrievable at the computer console is the identification and location of the data, and a description of how one package of data relates to others. This problem was selected for study at this time because it seems probable that the second problem, dealing with the retrieval of machine-readable information to be used in the on-line creation of a design, is neither well defined nor acute at this time. The second problem will, however, be the subject of discussion.

#### 5.4.2 Concepts for a Program to Retrieve Engineering Design Information

5.4.2.1 Introduction. There are two fundamental modes of retrieval of recorded information. The first might be called document reference retrieval, and consists of identifying a document (a book, drawing, etc.) which contains the needed information. To select a set small enough to be scanned from the set of documents to be searched, search keys such as titles, authors, contract numbers, and drawing numbers are used.

The second mode is the retrieval of appropriate documents by reference to their content. The search keys used here depend on relationships of one kind or another. This might be the relationship of a document content to a standard index/search term vocabulary, the relationship of the document to an arbitrary description of knowledge (e.g., the Dewey Decimal System), the relationship of the document to other documents, etc.

The first and second modes differ in several ways. The first demands more specialized information about the document, the second about the retrieval system itself. The first is oriented toward finding a particular document, the second toward the desired information independent of its representation. They are not, however, mutually exclusive; in fact, the more powerful retrieval systems use both modes of operation.

Traditionally, in engineering, only the first mode has been used; engineering drawings are retrieved by drawing number and further design information is retrieved using the drawing. This is understandable as it can be very difficult to assign index terms, which specifically describe content, to non-textual information. In the following sections a variation on the second mode of retrieval, involving the identification of design information with the structure it represents, will be developed.

5.4.2.2 Discussion of the Problem. In any information retrieval area in which the data base is not prespecified and the potential user population is not clearly defined, two obvious questions arise: What is to be retrieved, and how? Possible answers to the first question include end-item information, or references to sources of end-item information, as well as the definition of information categories. Answers to the second include retrieval by part of drawing number, which implies that the user is technically sophisticated in the contents of the data base; or such techniques as retrieval by tracing a drawing structure tree, which can be achieved by a layman.

To treat, first, the problems of the type and extent of immediately retrievable information, this information will be defined as that subset of engineering design data and references to sources of such data which are in computer-accessible storage. If cost were no object, all of the data which could be useful to the designer would be included; but, in a practical system, much of the data will consist of pointers to sources. These sources are typically drawings, specifications, operating manuals, and other collections of data too voluminous to be stored economically in digital form. The first problem of the retrieval system designer is to decide what data will and what data will not be made "immediate." A second question is that of the extent of the data to be considered: Is the entire engineering design file to be made accessible

(directly in the system or indirectly as a reference) via the retrieval system, or only the high-activity items? What assumptions can be made about the designer's ability to locate relevant ancillary information without the help of such a system? An example of such a question is: Is it worthwhile to include an often-used manual of component standards in the computer storage, or even a reference to the manual, or can it be assumed that the design engineer knows about the manual and can obtain his information as required?

The second kind of problem is that of available means of retrieval. Obviously, the techniques chosen for retrieval depend on what is available; it is not so obvious that they depend on the background of the system user. Most systems for the retrieval of engineering drawings (and most, for example, the Central Repository, are manual) depend entirely on one item: the drawing number. This is usually adequate, at least for the designer working in daily contact with a restricted area and well-defined set of problems. It is very difficult, however, for anyone with broader responsibilities (system integration, test and checkout) to get to the specific data he needs. A basis for an auxiliary tool, to help the latter class of user, is the next subject of discussion. The question of what information is to be made "immediate" will be taken up later, in Paragraph 5.4.2.5.

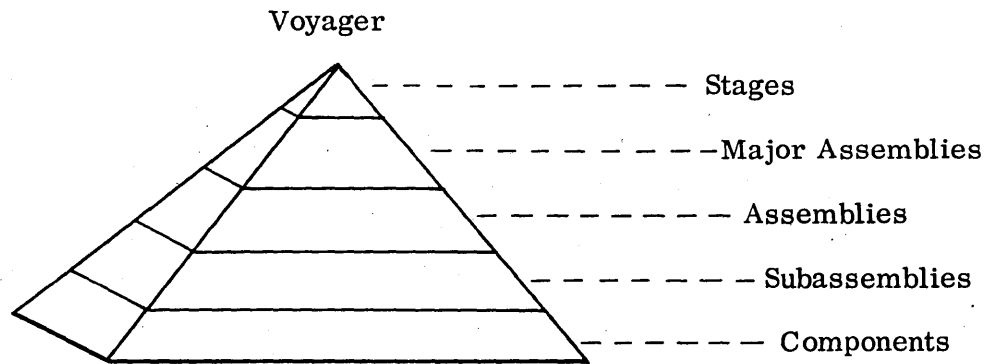
**5.4.2.3 Relationships in Engineering Design Data.** There are several kinds and degrees of relationships among engineering data. Three of these types of relationships will be exploited in this retrieval system design.

The first is that between the elements of a "package," which might be defined as the data set required to make a complete and meaningful junction requirement at a given level of detail. It might, for example, consist of a drawing, a specification, and a parts list, all of which are required for the fabrication of a given component. The basis of this relationship must be defined in terms of the use to which the package is to be put, as, for example, a manufacturing package is usually a different set of data from that implied by the needs of design or those of procurement. In the following discussion will be restricted to engineering design packages.

The second is a relation between packages, usually called an interface. The significance of this relationship is that one component, assembly, etc., is connected directly to another in a more or less complex manner. This interface may be so complex as to demand a "package" of its own to describe it.

The third, and perhaps the most obvious relation which exists between packages, is the component-assembly or "goes-into" pattern. It can be represented by a set of nested boxes, or better, by a tree showing its partial ordering in a hierarchical pattern.

The interface and component relations between packages can be represented simultaneously in a three-dimensional pattern. A convenient representation is a pyramid, the apex of which is the name of the construct being described. Successively lower "layers," defined by planes parallel to the base of the pyramid, contain data packages at levels of increasingly greater detail.





The assembly-component relationship is thus represented by a vertical trace through the structure. For example, from the apex,

Voyager

Stages

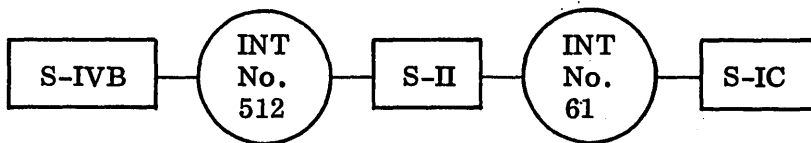
Payload

S-IVB

S-II

S-IC

Each horizontal plane, then, represents a set of data at a given level of detail, represented by a set of data packages or by a set of component names. The interface relation between items at a given level can be graphically represented by component (or package) identifications connected by lines indicating a physical connection or contact. In the example above, the interface diagram for the S-II stage at the "stage" level of detail is simply



where the rectangles identify the components, and the circles the interface specifications. In the case of interfaces which are not described by documentary specifications, the interface circle could contain a list of the types of interfaces shared by the components; for example, "elec./mech/hydra" for electrical, mechanical, hydraulic.

Figures 5-11 and 5-12, extracted from the "Nomenclature and Breakdown Charts for Saturn Vehicle, SA-5" (M-P&VE-EA-62-1, NASA MSFC), depict the type of structure identified above.

5.4.2.4 Retrieval by Structural Clues. The following discussion, which is concerned with the definition of a group of operations to facilitate the retrieval of design data,

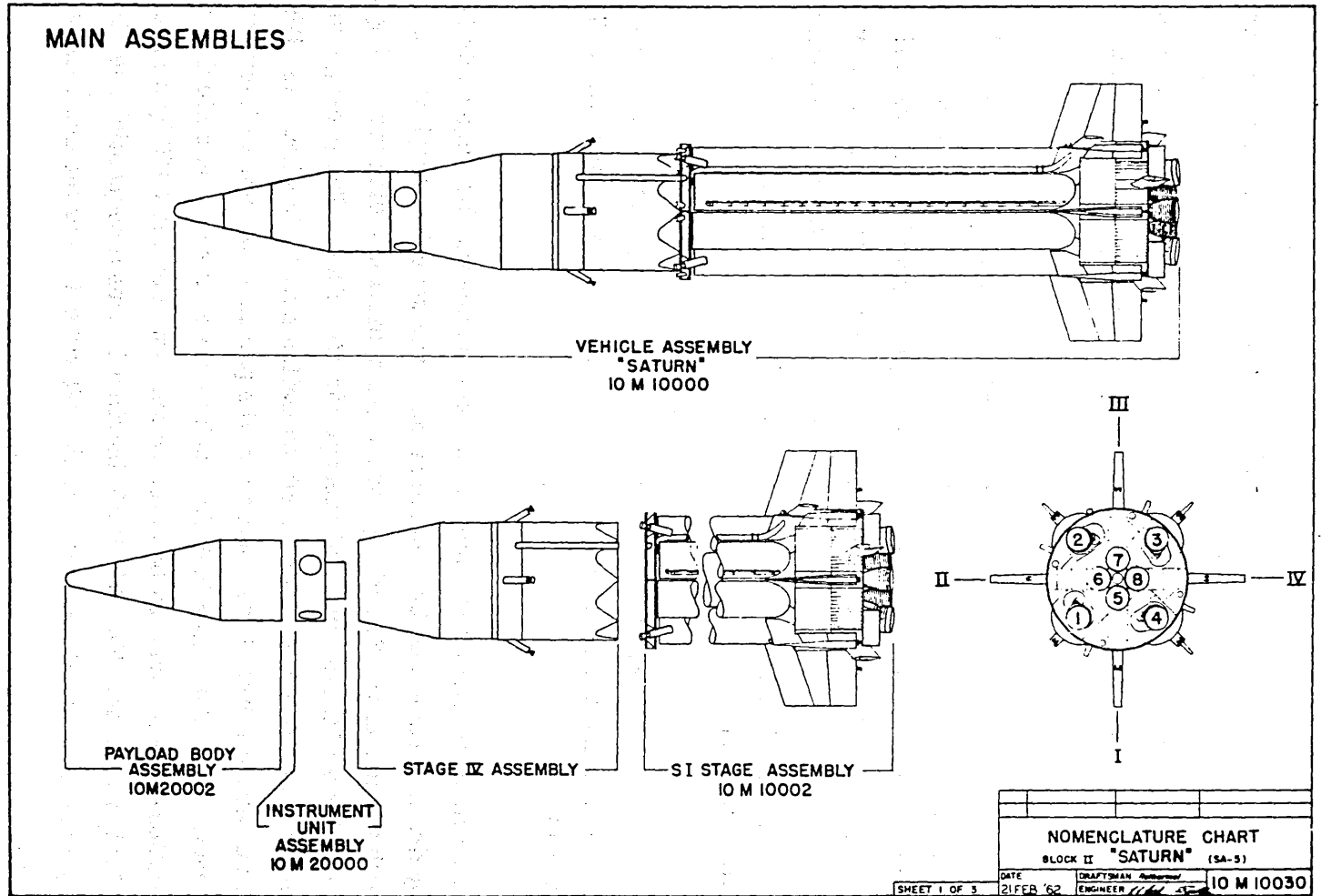


Fig. 5-11 Saturn SA-5 Main Assemblies

presupposes that the system user will be working at a CRT console on-line to a computer. He will have a light pen and an alphanumeric keyboard as input devices. The commands, i.e., EXPLODE, are used to directly address or instruct the computer through the light pen/keyboard device, and they and available answers are displayed in real-time on the CRT surface. The computer will have a large direct-access peripheral storage in which a body of data, organized as suggested in Paragraph 5.4.2.3 is available.

As shown previously, engineering design packages can easily be related to the physical structure they represent, which in turn can be factored into successively greater levels of detail. The operation of going from general to specific will be called EXPLODE. At a given level of detail the relation of interfacing and the identification of the interface specifications will be displayed when the INTERFACE command is given. Both EXPLODE and INTERFACE are used, of course, in reference to specified data packages.

Another function should be available, to permit the user to perform the opposite of the EXPLODE operation and move upward from components to assemblies. This will be called USED IN.

The packages will be identified by number (in some cases the drawing-parts list number), by drawing name and by generic name. The search through the structure will start with the use of the START operation (an operation indicating the beginning of a search) and one of these forms of package identification. The package name or number input will be displayed and a fourth operation may be invoked – the DISPLAY option – which causes a list of the contents of the package, by type of information, to be displayed. If the data base includes such items as inspection criteria, tolerances, and specifications, these can then be selected and listed on the CRT by a second use of the DISPLAY operation. The DISPLAY function is thus a sequential one, the first use detailing package contents by item and the second detailing the contents of a selected item.

There are two cases in which finding the desired starting-point package may not be an immediate result of the input package identification. The first occurs when the

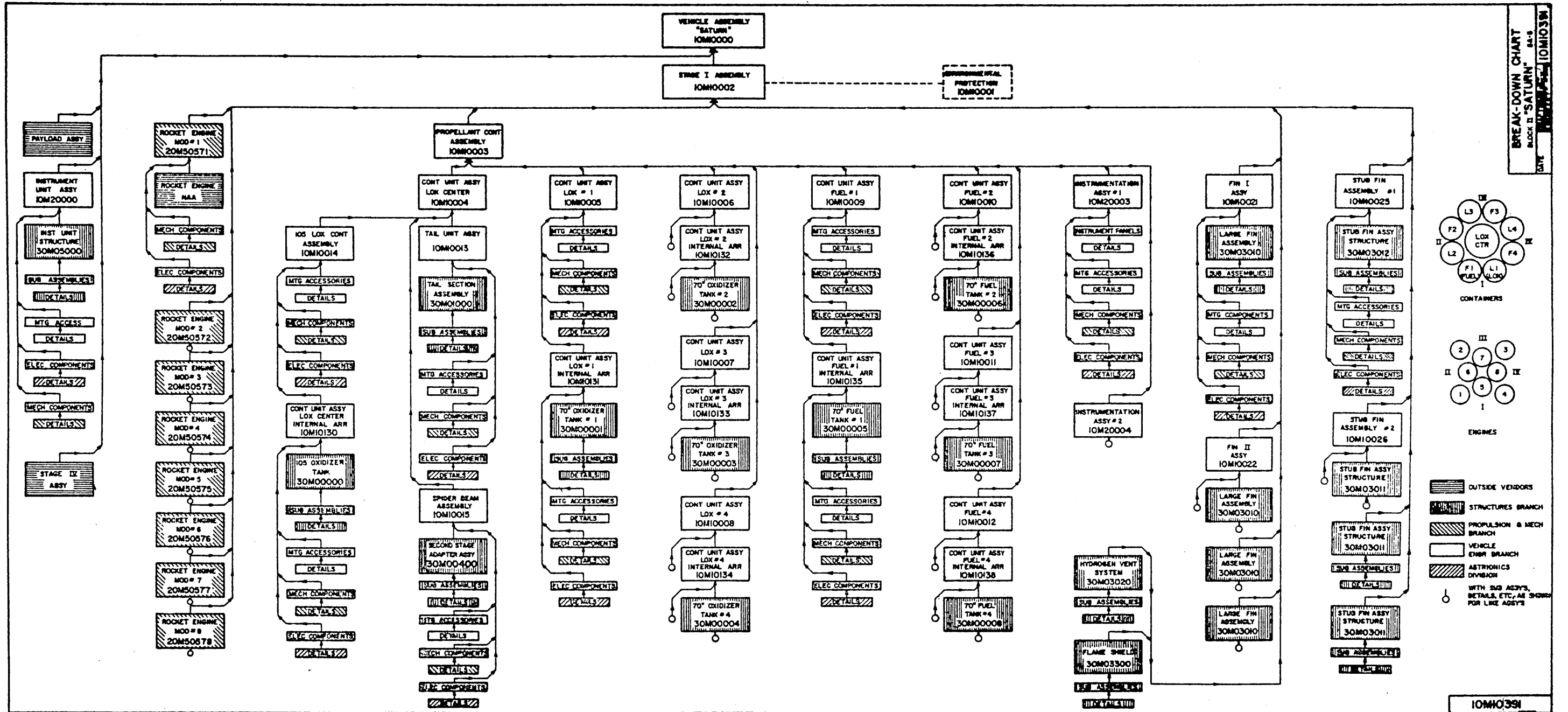


Fig. 5-12 Breakdown of Saturn SA-5 Payload and Stage I Assemblies

input was a generic name such as "power supply." In order to discriminate between power supplies, the list of such components in the structure would be displayed, by drawing name and number, to permit the user to choose the individual item he needs. If he is still in doubt after seeing the list, the EXPLODE, INTERFACE, and USED IN operations will help him to isolate the desired package.

The second case of ambiguity arises when the user identifies a component which is used in more than one place in the structure as the starting point in his search. Since he has just used the START operation, the retrieval system will assume that the user is not interested in the component designated but is using it only as an entry to the structure. (If this is not the case, and the component itself is the end-item of the search, the user can enter its identification again to get the package.) The next display will identify all the next-higher assemblies using the given component.

If this still does not discriminate sufficiently between uses of the component (the same higher assembly uses the same package more than once) the identification of the higher assembly and use of the USED IN operation will move the process one step upward. This can be repeated until the appropriate larger assembly is identified, at which point repeated use of the EXPLODE operation can be used to get to the unique desired starting point.

A function which permits availability of data to the user about more than one structure (for example, Saturn V as well as Voyager might be represented in the files) would be called SELECT. At any time the system user wishes to start a new search, or change the subject of his search from one structure to another, he uses the SELECT operation. The result will be a display of the names of the structures represented in the data file. He then indicates the option he desires and uses the START operation to begin his search.

In summary, these operations, operands, and their consequences are as follows:

<u>Operation</u>	<u>Operates On</u>	<u>Results</u>
SELECT	Total data base	Selects the pyramid relevant to a given structure
START	Structure data base	Selects a starting point in the structure
DISPLAY	Data package or item in package	Displays a list of items in package, then, if desired, details are of the items listed
EXPLODE	Data package	Displays component packages
USED IN	Data package	Displays identification of next-higher assembly
INTERFACE	Data package	Provides a graphic display of packages with direct interfaces with that given, and interface control information if applicable

Several further operators are needed. The first is not strictly a part of the retrieval function but serves to record the interim or final results of a search. The PRINT operation causes the contents of the display to be transferred to a permanent medium.

Two other operators, not strictly needed to carry out a search but without which searching is extremely tedious, are called RESTORE and TRACE. The RESTORE function brings back the display seen just prior to the current one. The TRACE operation displays a list of the last ten steps the searcher has made, identified by package number and name, to permit the searcher to return to a previous step. This is extremely useful when the user discovers he has taken a wrong turn in tracing through the data base.

5.4.2.5 File Contents. In Paragraph 5.4.2.2 two principal problems were discussed; the problem of selecting the data to be placed in the retrieval file, and that of

specifying a retrieval technique. The ensuing description of a solution to the second problem indicates some of the constraints which must be placed on the solution of the first.

The file must be organized by structures, and its elements are what have been called data packages. It must be complete in the sense that there is an uninterrupted linkage from the most general item (the name of the structure) to the most specific level included.

The data package, identified by a number and a name, contains at least a list of components (if it is not itself a single indivisible component). It will contain, if appropriate, the locations of the associated drawing, specification, standards, tolerances, inspection criteria, etc., if these items of information are not included in digital storage. The decision as to which elements should be in literal form in digital storage and which should be represented by identifying information is an economic one. Among the parameters which will influence this decision are:

- The number of data packages needed to describe the structure
- The volume of data per item of information (e. g., specification)
- The cost of converting these items to digital form
- The cost and size of the peripheral storage devices needed to maintain the files
- Whether or not each file has to be continuously available (that is, whether a storage device such as a data cell can be loaded with specified files at specified times)

Among the data will be packages specifying interfaces as well as those specifying components. Their retrieval can be achieved directly if they are identified by drawing number or name, or by association if they are identified by finding the components they relate to and the INTERFACE operation is executed.

5.4.2.6 File Creation and Maintenance. The file would be large for structures of the scale of Voyager. Its size would depend on the amount of direct data included, as opposed to the number of data references, but even at a minimum it would be large

enough that initial on-line input of the entire volume of data would be extremely expensive. On the other hand, the file maintenance could well be performed on-line.

The initial data base would then be loaded as a batch process, and changes to the file would be performed using a graphic console. To assure that only authorized changes are made, only designated consoles would be used to update the file.

The update operations, which would be implemented in such a way that they would be available for either on-line or off-line use, would be ADD, DELETE, REPLACE, and INSERT, and they would refer to one of three categories of data – DATA, ITEM, or PACKAGE.

In each use of one of the operation/data-category pairs, a data-set identification is required. If the data set is an entire package, the package number is sufficient. In the case of smaller sets the first step in updating is to retrieve and DISPLAY the package which contains them. If the data set is an ITEM, the corresponding item identification number is then used to point to the item to be modified. If the set is smaller than the ITEM level, the DISPLAY operation is used again, this time in relation to the item containing the data set to be altered. The character at which the alteration is to start is indicated by light-pen, the corresponding operation button is pressed (e.g., INSERT), the data-set button DATA is pressed, and the entry is made  
In summary:

<u>Operation</u>	<u>Data Set</u>
ADD	{ DATA – Character in item in package ITEM – Item in package PACKAGE – Package I. D.
DELETE	
REPLACE	
INSERT	

The update operation would include a safeguard to protect users of the file. This might take the form of actually inhibiting all references to the part of the file being updated, or might merely inform anyone examining a package being altered that alterations are in progress.



5.4.2.7 Retrieval Equipment Configuration. It has been implicitly assumed that retrieval would be performed using a CRT console with light pen, having an alphanumeric keyboard and a few function buttons (which might be selected from little-used characters on a standard keyboard).

In this configuration, after an initial alphanumeric entry to the data structure was achieved, the rest of the search and data extraction process could be carried out using the light pen to select the appropriate items and the function buttons to perform the desired operations.

The only consoles which must have graphic capability are those which are assigned to update the files. Consoles used strictly for retrieval do not need graphic-input capability and are called upon to display only very simple patterns (boxes connected by lines).

5.4.2.8 Notes and Comments on REDI. There has been no discussion of "effectivities" in this conceptual outline, but they would certainly be included in the data packages. This would not only be a tool for the designer but would make the retrieval system a major tool in configuration control, as well as to facilitate historical reviews on previously assembled structures.

The system as outlined is not complete in the sense that it supplies the designer with all the information he might conceivably use but it could be supplemented with other systems supporting ancillary data sets. Some of these are discussed in the next subsection.

The previous descriptions imply that the structure to be factored, the "top assembly", might be a complete vehicle. It should be noted that this idea is not an essential part of the plan, might be preferable to start at the "main assemblies" level, as for example, one sequence of Voyagers might use a different booster from that used by another.

5.4.2.9 Retrieval of In-Process Design Information. This problem, cited in Paragraph 5.4.1 as the "second problem", is that of supplying the user of an on-line machine-aided design system such as Digigraphics with the data he needs to work efficiently. This data includes conceptual versions of his drawings, related drawings, tolerances, standards, inspection attributes, preferred parts lists, etc., and is similar to that which can be handled by the REDI system and its retrieval can be accomplished in the same manner.

Other data is involved however, and the nature and structure of this further data indicate that other retrieval techniques will be needed to deal with them. Among the more obvious categories of design data are those contained in engineering handbooks, drafting manuals, etc. The retrieval of this kind of data is most naturally performed by the display of a back-of-the-book type of alphabetized index. This technique, if not entirely adequate, could be supplemented by coordinate indexing. This would require that skilled indexers assign descriptive terms from a controlled vocabulary to major topics covered in the handbook.

The current users of the LMSC computer graphics system have indicated some more specialized requirements calling for more exotic approaches to retrieval. The group doing orbital analysis foresees the storage of Venus swingby (to and from Mars) flight parameters and summaries of past performance studies. The structural analysis group specifies a system which will permit the design engineer to retrieve reports on performance data by structure type (honeycomb, waffle, etc.) as well as the kind of data to be handled in the REDI system. The network analysis group would like to have a Preferred Parts Handbook in computer storage. They require that the results of network analyses, as well as the parameters of derived equivalent circuits, be stored and retrievable. They stated, incidentally, that in their opinion the retrieval of their data could be adequately performed by reference to the functional use or discussion of the circuit.

Many of these requirements can be met by a "structural" system such as REDI. Most of the remainder can be met by a system designed to retrieve data from a handbook. The balance indicate a need for a library type of reference retrieval system based on

coordinate indexing. The first and third of these techniques have been implemented, the third widely, but methods for handbook data retrieval are still in the research stage.

A final note on this problem concerns archival rather than work-in-progress storage of this design data in digital form. References to designs having a direct bearing on structures could be added to the corresponding files in the REDI system, but separate indexes should be prepared for such special data as swingby orbits.

#### 5.4.3 Program Description

Following is a description of a set of program segments designed to implement the retrieval system developed in Paragraph 5.4.2. It is assumed that these program segments would operate under the control of a real time, multi-processing monitor such as the Univac EXEC8, the IBM system/360 OS, the CDC Digigraphics system, or the LMSC LACONIQ (for System/360). A program segment is a computer program which is called into main storage by the monitor, performs one or more functions, and then returns control to the monitor. Real-time programs typically consist of a number of program segments, each of which is relatively small.

It is assumed that the monitor will permit a program segment to communicate with the monitor and with other program segments. For example, a program segment should be able to designate its successor segment and it should be able to transmit data and program flags to other program segments servicing the same input-output terminal.

The following items are terms and abbreviations used in this section:

- Package number (PKN) is defined as the number used to identify a package or item in a package.
- Basic PKN is defined as the portion of the PKN to the left of the "dash".
- Data sent to an input-output terminal as a unit shall be called a message.

The program logic was developed with the assumption that data would be organized into four files. The data need not necessarily be segregated in the storage device but each file must have its own directory. Furthermore, each structure shall have its own set of directories. The following is a description of the files.

- a. File A: This file contains lists of the "dash number" versions for each basic PKN. Entry to the file is by basic PKN. Individual records contain a description of the item and a listing of the dash numbers with their effectivities.
- b. File B: This file contains the PKN detail. If the PKN is a document such as a specification, the record will contain the text of the document. If it is a hardware item, the record will contain parts list information such as description, unit weight, a list of the next level assemblies of which this PKN is a component, with quantity per assembly, and effectivity. In addition, the record will list the items in the package and such information as description, quantity per assembly, drawing release status, etc. Entry to the file is by PKN.
- c. File C: This file contains interface information. Entry is by PKN.
- d. File D: This is an "inverted file" which provides a cross-reference to PKNs by their generic names. Entry is by generic name. Records contain basic PKNs and descriptions.

#### 5.4.3.1 Program Segment Descriptions.

##### Segments SL1 and SL2

Function: SELECT

The purpose of this segment is to permit the user to select a structure. Once the selection is made it is stored in an area associated with the input-output terminal where the selection was made. Each time data is retrieved by other program segments this area is examined to determine the structure. If no structure had been selected an error message would be sent to the console.

The user may change to another structure by pressing the SELECT function button at any phase of his inquiry. Selection of a new structure will cancel all requests for data on the previous structure still waiting to be processed.

Only one structure may be selected at a time.

### Segment Outline

#### Segment SL1 (Ref. Fig. 5-13)

1. Notify the monitor that program segment SL2 is the successor segment.
2. Send a pre-formatted message listing the various structures and their codes to the input-output terminal.

#### Segment SL2 (Ref. Fig. 5-13)

Initialize the request queue. This is a request queue maintained by the program, not the monitor. Initializing the queue will cancel any requests waiting to be processed.

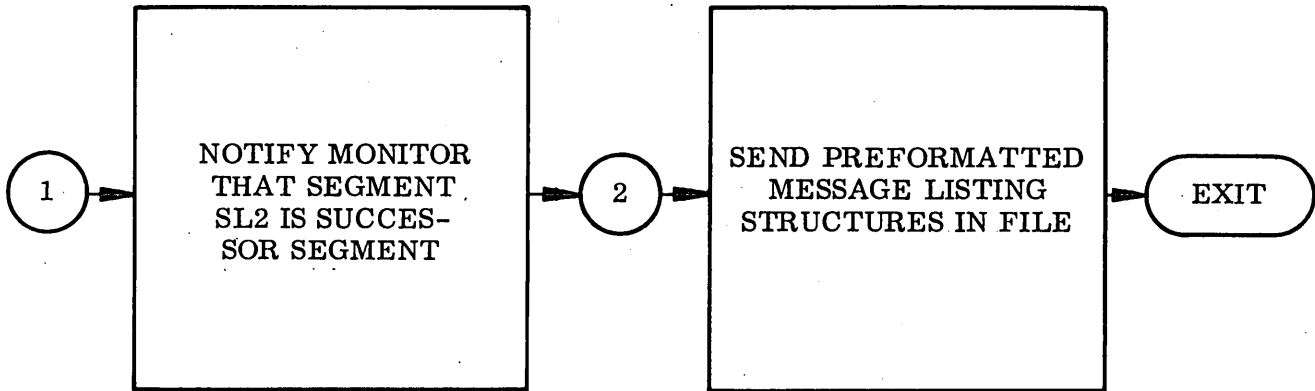
### Segment S

Function: START

The purpose of this program segment is to permit the user to initiate a search. The user may enter any one of the following type of parameters in conjunction with the START function button:

1. A PKN with a "dash number"
2. A basic PKN
3. One of a selected list of package descriptions
4. One of a selected list of generic names such as fuel pump, heat shield, bracket assembly, etc.

SEGMENT SL1



SEGMENT SL2

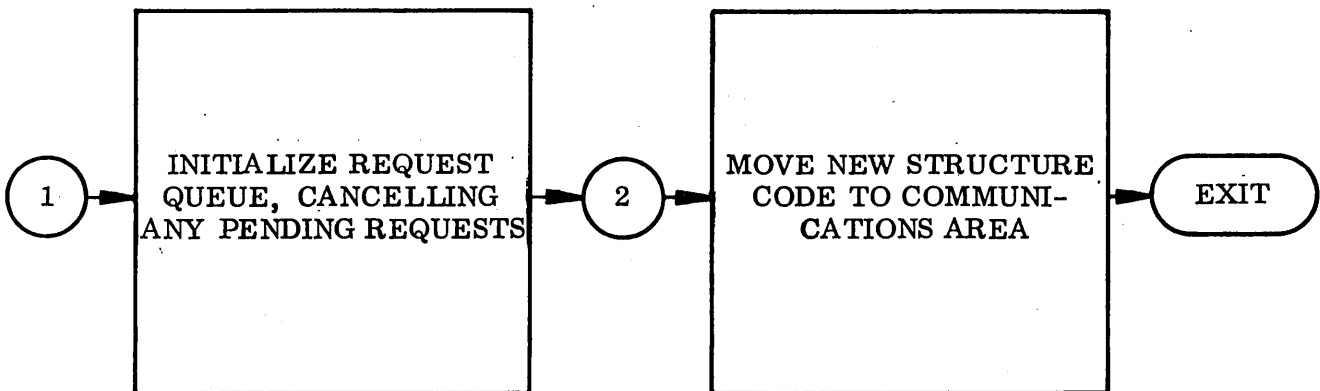


Fig. 5-13 Logical Flow Chart Segments SL1, SL2

The segment responds by displaying a list of PKNs on which the user may operate with such function buttons as START, DISPLAY, EXPLODE, USED IN, INTERFACE, etc.

If the user enters a PKN with a "dash number," the segment will display the description and effectivity of the PKN. If the user enters a basic PKN, the segment will list all of the "dash numbers" (including the original, which has no "dash number") with their effectivities.

If the user enters a package description, the segment will display the basic PKN for that segment. If he enters a generic name, the segment will display the basic PKNs indexed by that name and their descriptions.

#### Segment Outline (Ref. Fig. 5-14)

1. If the input parameter is not a PKN, with or without a "dash number," go to Step 7.
2. Separate the PKN into its basic PKN and "dash number" if it has one.
3. Retrieve the basic PKN record from File A.
4. Format header portions of message. This will include the description.
5. If the input PKN was a basic PKN, go to 6. If it was not, position the data for the PKN in the message, send the message to the input-output terminal, and exit to the monitor.
6. The input parameter was a basic PKN. List all of the "dash numbers" in the message, send the message to the input-output terminal, and exit to the monitor.
7. The input parameter was not a PKN. If the input parameter is in the selected table of package descriptions. Pick up the PKN and transfer to 3 for further processing. If not, continue to 8.
8. By elimination, the parameter is assumed to be a generic term. Retrieve its record from File D.
9. Format the header portion of the message. List the basic PKNs catalogued under the generic term with their descriptions, send the message to the input-output terminal, and exit to the monitor.

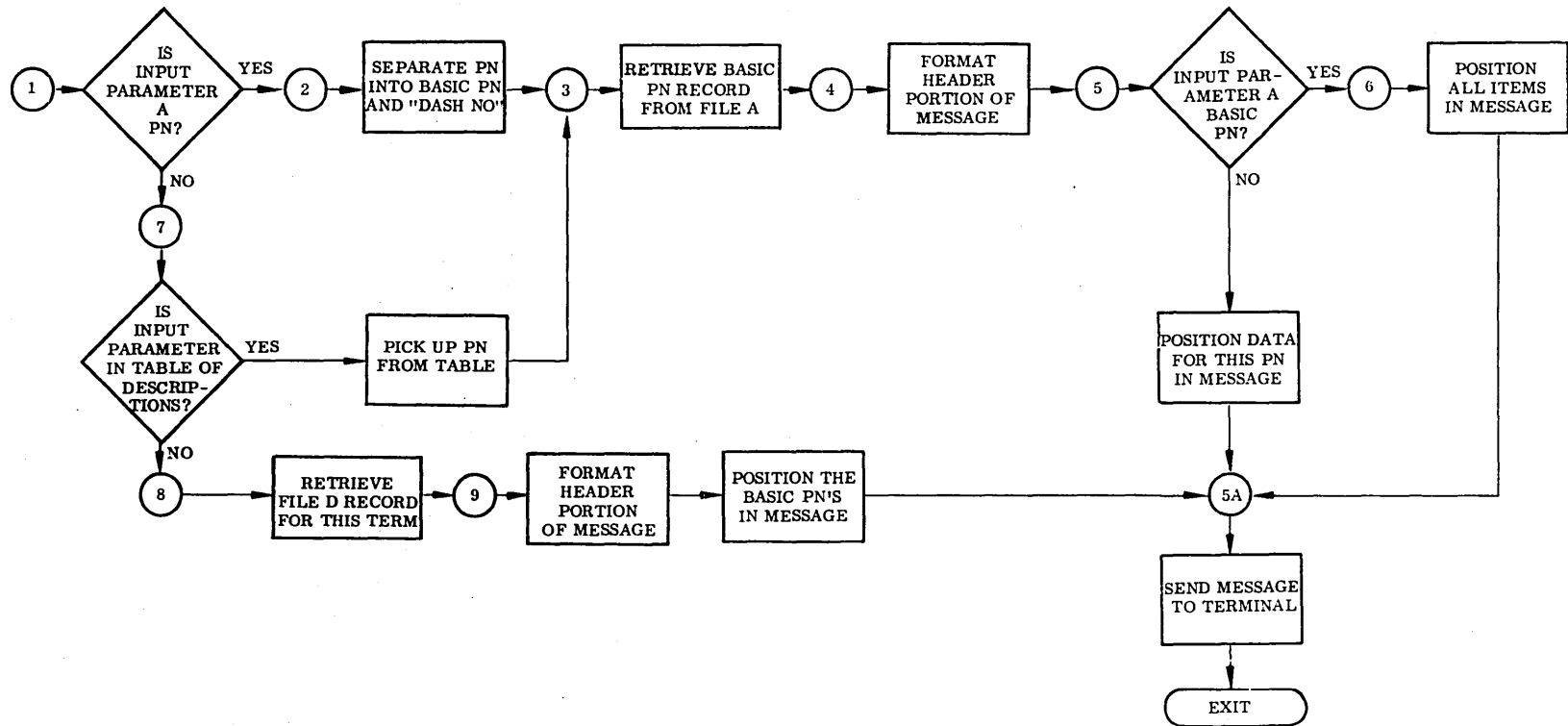


Fig. 5-14 Logical Flow Chart, Segment S



## Segment D

Function: DISPLAY

The purpose of this segment is to display the File B data for a given PKN. The output format will vary with the nature of the PKN. For example, if the PKN were a document, and the document was in digital storage, the text of the document would be formatted and displayed. If it were a parts list, the header and item detail would be formatted and displayed.

### Segment Outline (Ref. Fig. 5-15)

1. Retrieve the File B record for the input PKN.
2. Determine the type of record and format the message in an appropriate manner.
3. Send the message to the input-output terminal and exit to the monitor.

## Segment E

Function: EXPLODE

The purpose of this program segment is to select and list the components of a given PKN.

### Segment Outline (Ref. Fig. 5-16)

1. Retrieve the File B record for the input PKN.
2. Format the header portion of the message. Select and position the items in the message.
3. Send the message to the input-output terminal and exit to the monitor.

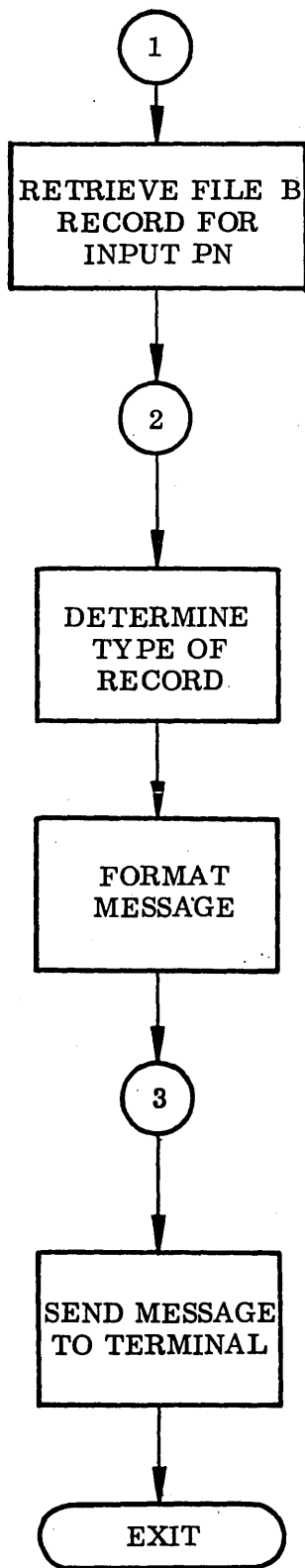


Fig. 5-15 Logical Flow Chart, Segment D

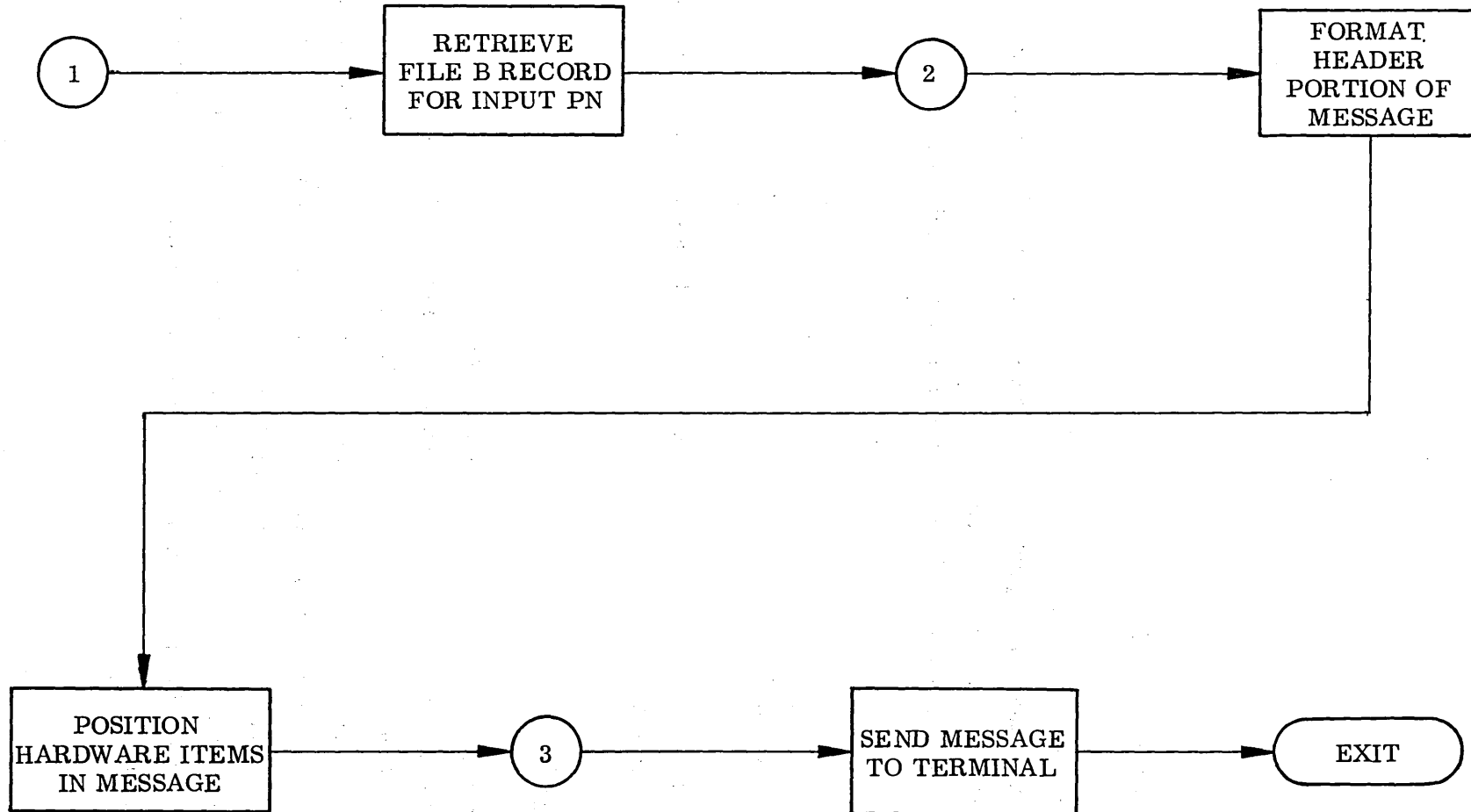


Fig. 5-16 Logical Flow Chart, Segment E

## Segment U

Function: USED IN

The purpose of this program segment is to list next higher level assemblies of which a given PKN is a component, together with the quantity per assembly and effectivity.

### Segment Outline (Ref. Fig. 5-17)

1. Retrieve the File B record for the PKN.
2. Format the header portion of message. Select and position the "next assembly" data in the message.
3. Send the message to the input-output terminal and exit.

## Segment I

Function: INTERFACE

The function of this program segment is to format and display a diagram in response to the INTERFACE function button.

### Segment Outline (Ref. Fig. 5-18)

1. Retrieve the File C record for the PKN.
2. Initialize index used to step through chain of interface.
3. Construct first box and position the PKN in it.
4. If the last item has been processed, determine the section of the construction grid in core which will be sent to the input-output terminal, send the "message", and exit to the monitor.
5. Position first circle. If the record contains data describing the interface, such as a document number or descriptive term, position the data in the circle.

5-63

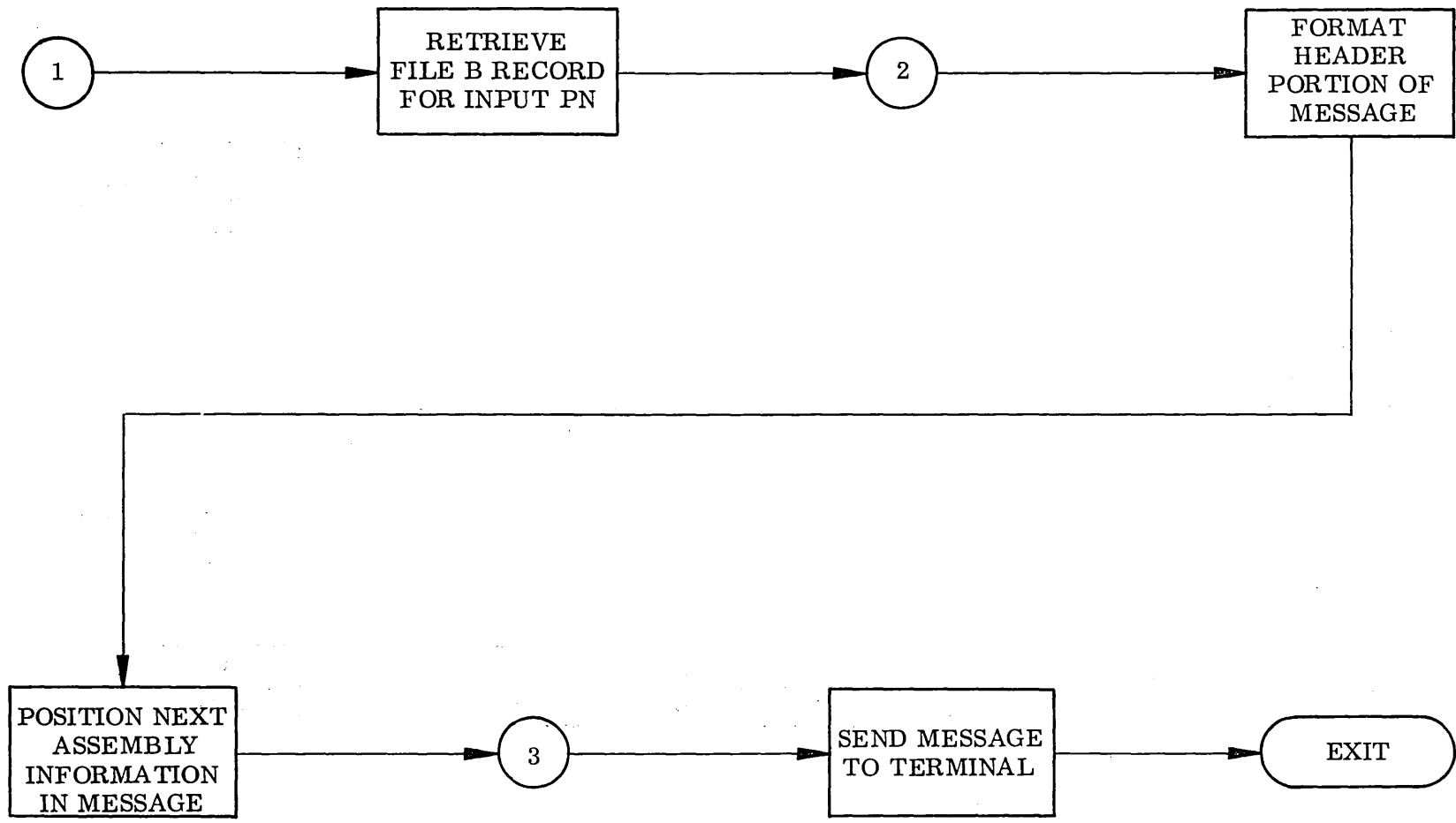


Fig. 5-17 Logical Flow Chart, Segment U

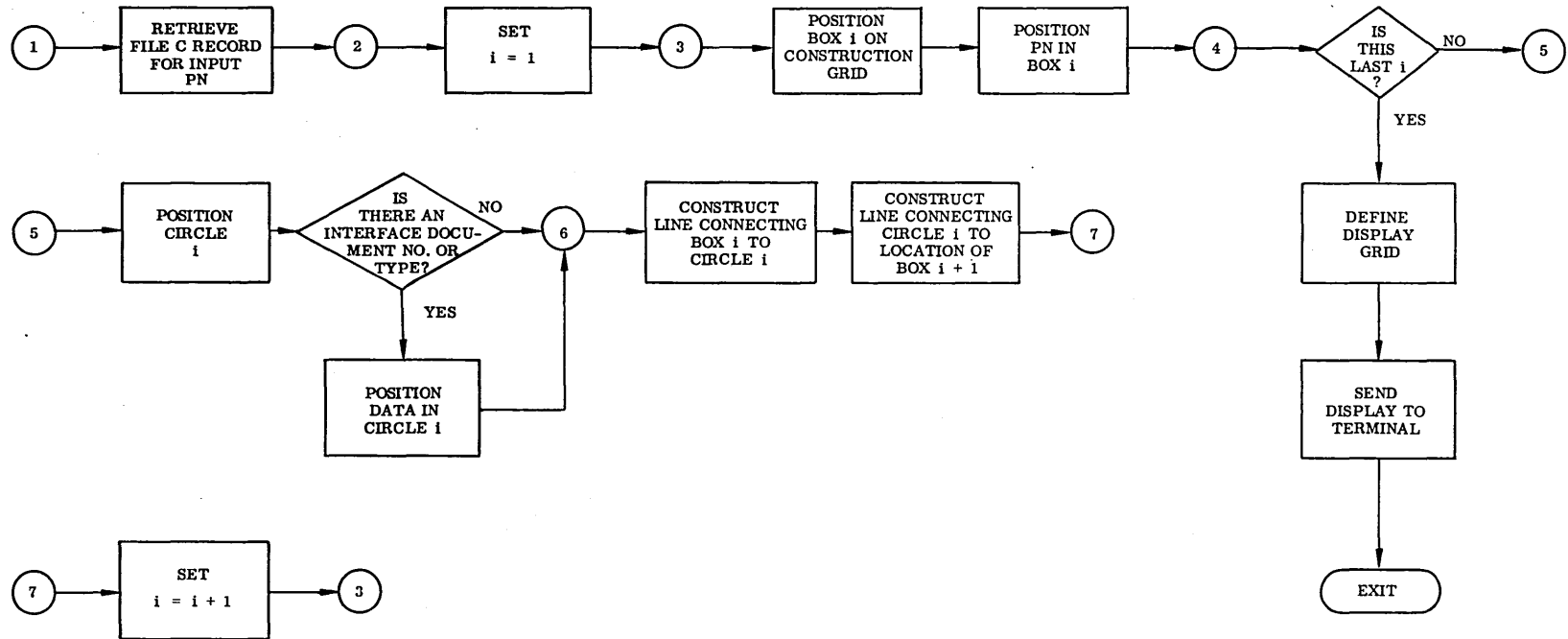


Fig. 5-18 Logical Flow Chart, Segment I

6. Construct lines between first box and first circle and between first circle and the boundary of next box.
7. Increment index and transfer to 3 to process the next item.

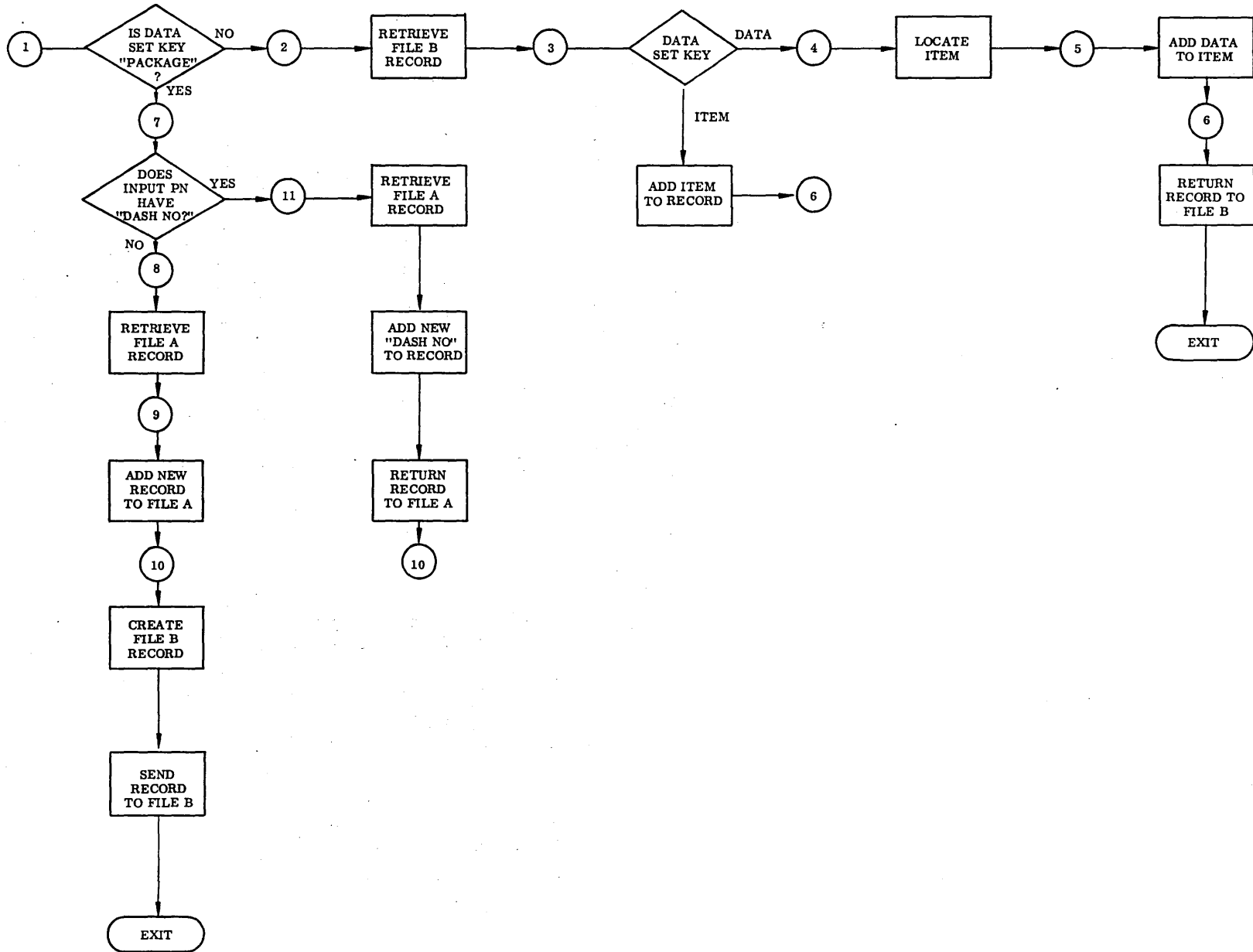
### Segment A

Function: ADD

The purpose of this program segment is to add DATA to an item, add an ITEM to a package, or add a complete PACKAGE. The segment is called by the ADD function button. The scope of the operation is determined by the choice of one of the following DATA SET keys: DATA, ITEM, or PACKAGE.

### Segment Outline (Ref. Fig. 5-19)

1. Identify the data set key. If it is PACKAGE, go to 7; if not, go to 2.
2. Retrieve the File B record for the input PKN.
3. Identify the data set key. If it is DATA go to 4. If it is ITEM, add the new item to the record and go to 6.
4. Data set key was DATA. Locate the item in the record.
5. Add the new data to the item.
6. Return the updated record to File B and exit to the monitor.
7. Data set key was PACKAGE. Analyze input PKN. If it has a "dash number" go to 11; if it does not, go to 8.
8. Create a new File A record for the basic PKN.
9. Add the new or updated record in File A.
10. Create File B record, deposit it in the file and exit to the monitor.
11. Retrieve the File A record for the basic PKN. Add the new "dash number" to the record, return the record to the file and transfer to 10.



5-66

Fig. 5-19 Logical Flow Chart, Segment A



## Segment DL

Function: DELETE

The purpose of this program segment is to delete DATA from an item, an ITEM from a package, or a complete PACKAGE. The segment is called by the DELETE function button. The scope of the operation is determined by the choice of one of the following DATA SET keys: DATA, ITEM, or PACKAGE.

### Segment Outline (Ref. Fig. 5-20)

1. Identify the data set key. If it is PACKAGE go to 7; if it is not, go to 2.
2. Retrieve the File B record for the PKN.
3. Locate the item to be operated on.
4. Identify the data set key. If it is DATA go to 5. If it is ITEM delete the item from the record and go to 6.
5. DELETE the selected data from the item and reformat the remainder of the item if necessary.
6. Return the record to File B.
7. Data set key was PACKAGE. Retrieve File A record for basic PKN.
8. Determine whether input PKN has a "dash" number. If it does not, go to 10. If it does, delete the "dash" number item from the record and return the updated record to File A.
9. Delete the record from File B and exit to the monitor.
10. Input PKN does not have a "dash" number. Delete the File B records for each of the "dash numbers," including the original PKN, which would have no "dash number."
11. Delete the File A record and exit to the monitor.

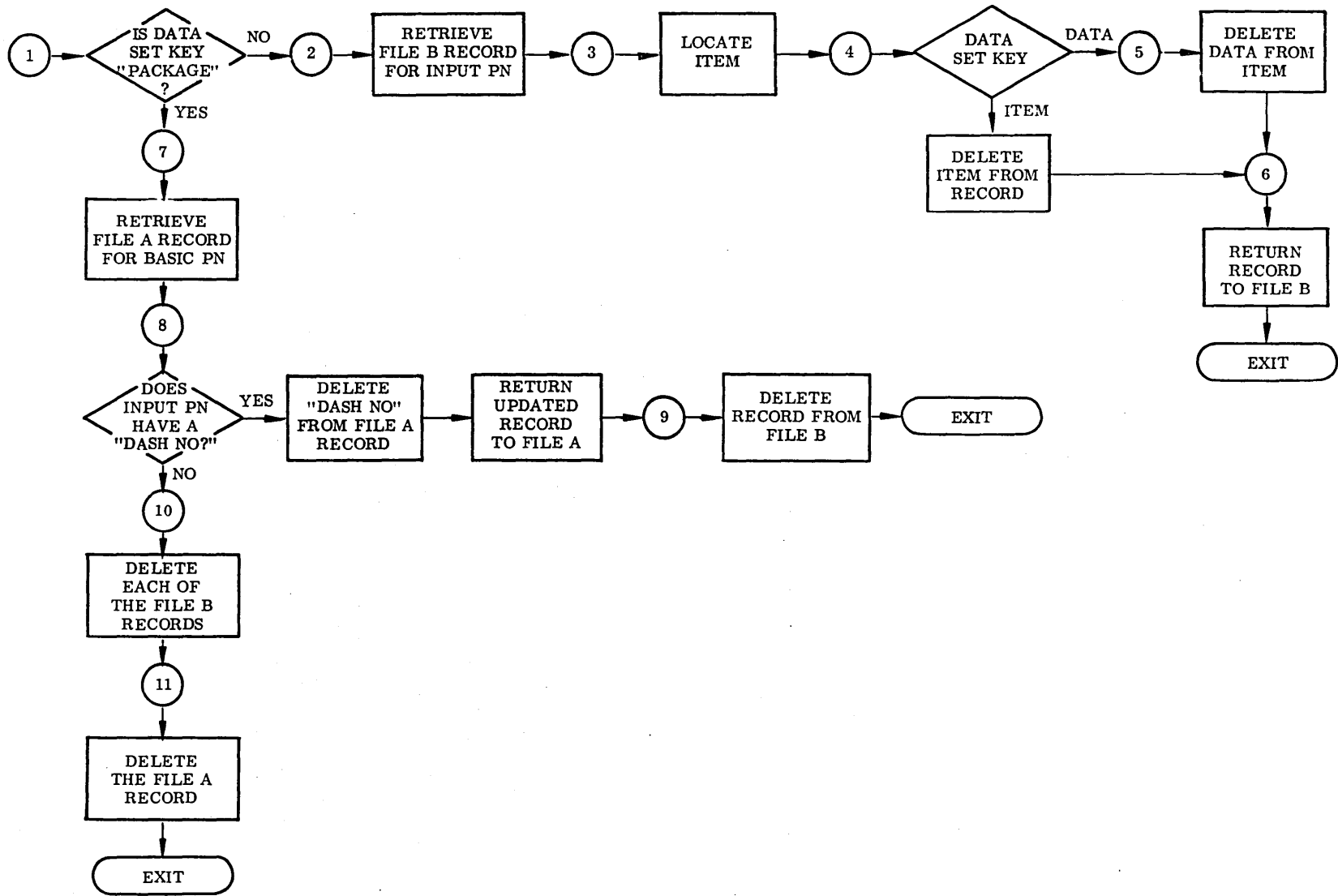


Fig. 5-20 Logical Flow Chart, Segment DL

## Segment R

Function: REPLACE

The purpose of this program segment is to revise DATA in an item, replace an ITEM in a package, or replace an entire PACKAGE. The segment is called by the REPLACE function button. The scope of the operation is determined by the choice of one of the following DATA SET keys: DATA, ITEM, or PACKAGE.

### Segment Outline (Ref. Fig. 5-21)

1. Identify data set key. If it is PACKAGE, go to 7; if not, go to 2.
2. Retrieve the File B record for the input PKN.
3. Locate the item which is to be revised or replaced.
4. Identify the data set key. If it is DATA, go to 5. If it is ITEM, delete the old item from the record, insert the new item, and go to 6.
5. Data set key was DATA. Reformat the item, substituting the new data for the old.
6. Return the updated record to File B and exit to the monitor.
7. Data set key was PACKAGE. Create the new File B record and transfer to 6.

## Segment IN

Function: INSERT

The purpose of program segment IN is to create or revise an interface record (File C). It must be preceded by a "clear screen" or INTERFACE command. Figures, such as boxes, circles, lines, etc. may be "drawn" by the light pen and alphanumeric data may be entered by the terminal keyboard.

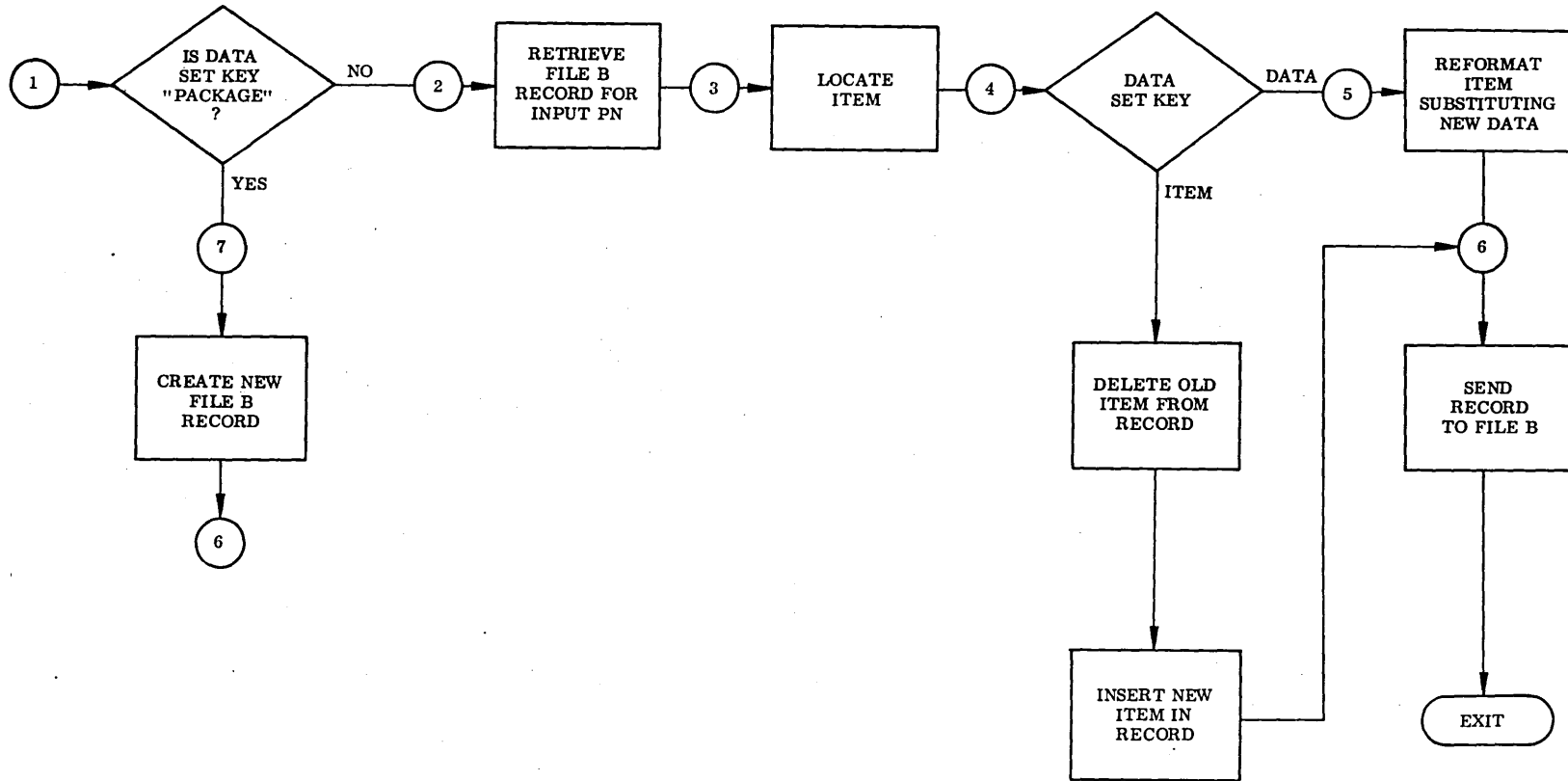


Fig. 5-21 Logical Flow Chart, Segment R

Segment Outline (Ref Fig. 5-22)

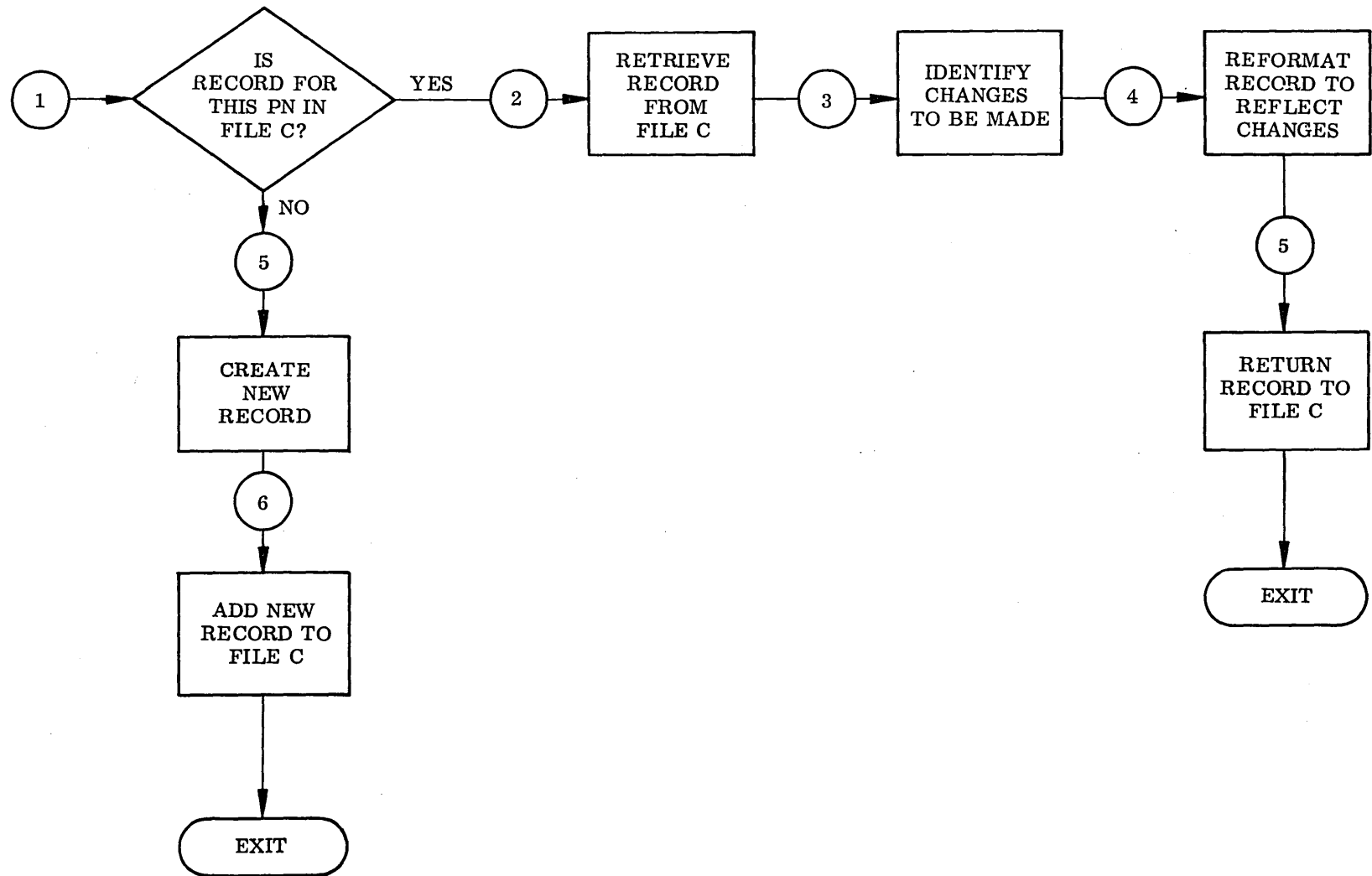
1. Determine whether there is a record for the input PKN in File C. If there is not, transfer to 6 to create a new record. If there is, go to 2.
2. Retrieve record for input PKN from File C.
3. Identify changes to be made. The technique for analyzing the graphic input will depend on the particular hardware and software used.
4. Reformat the record to reflect these changes.
5. Return the updated record to File C and exit to the monitor.
6. New record. Create a new record from the input data.
7. Add this new record to File C and exit to the monitor.

5.4.3.2 Typical Display Formats. Figures 5-23, 5-24 and 5-25 show a typical console display screen representation resulting from the operations described above. Figure 5-23 shows a typical interface structure diagram resulting from the INTERFACE operation. Figure 5-24 could represent the final response to an inquiry for a digitized drawing and Fig. 5-25 shows a typical response to an inquiry for a data package structure.

5.4.4 Implementation: Data Base

5.4.4.1 Introduction. The creation of a data base to implement the REDI system for the retrieval of design documents of MSFC could be performed in two steps. The first depends on a maximum use of existing machine-readable information and the second is that of adding the further data needed for the complete implementation of the retrieval system.

5.4.4.2 Data Base Requirements. In order to prepare a data package it will be necessary to examine the master documents to identify their supporting documents, such as process specifications, test specification procedures, inspection attributes, etc. It is then necessary to relate these data packages sequentially in their component-assembly chains and laterally in terms of item interfaces.



5-72

Fig. 5-22 Logical Flow Chart, Segment IN

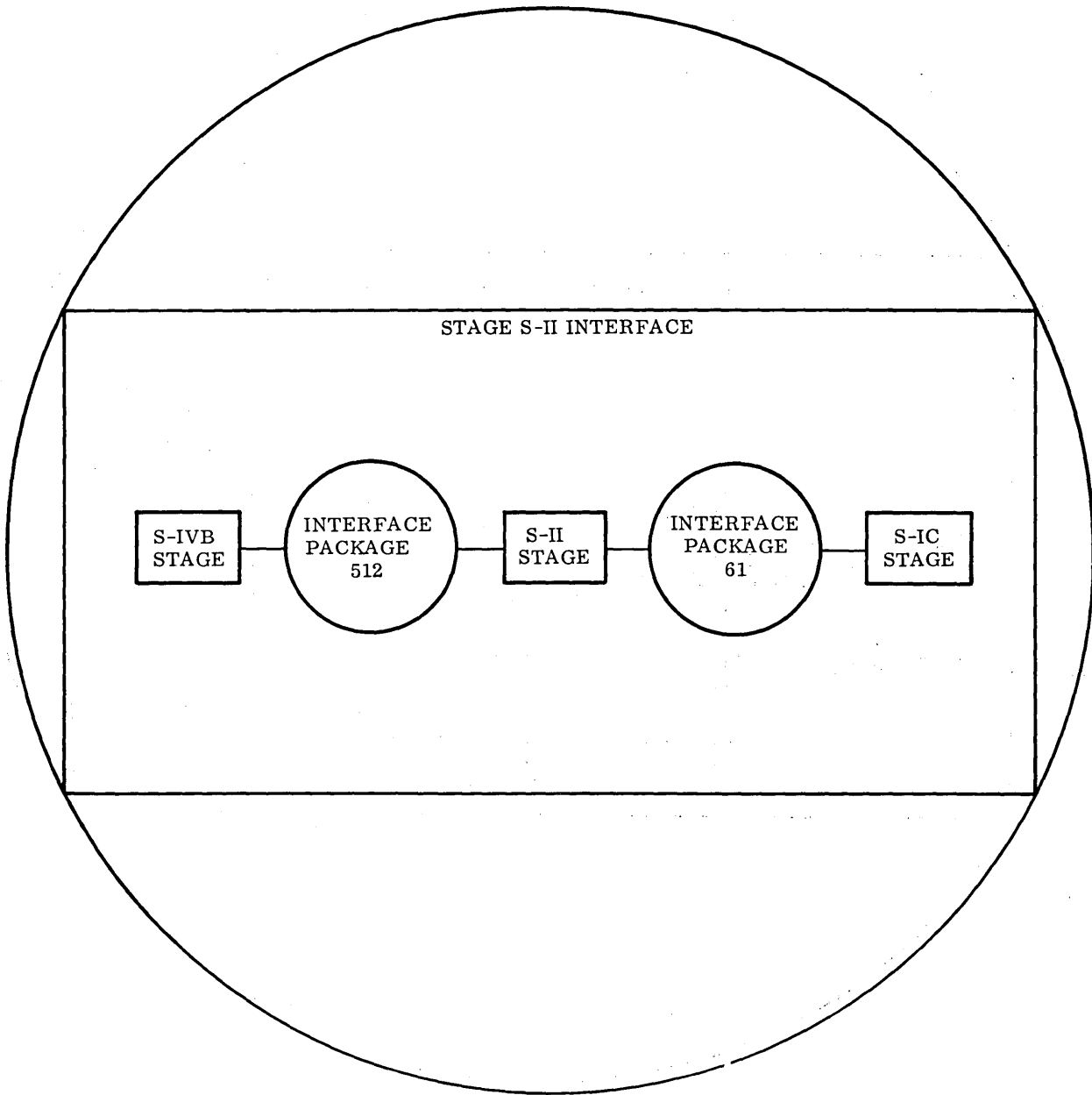


Fig. 5-23 Typical Display – Interface Structure

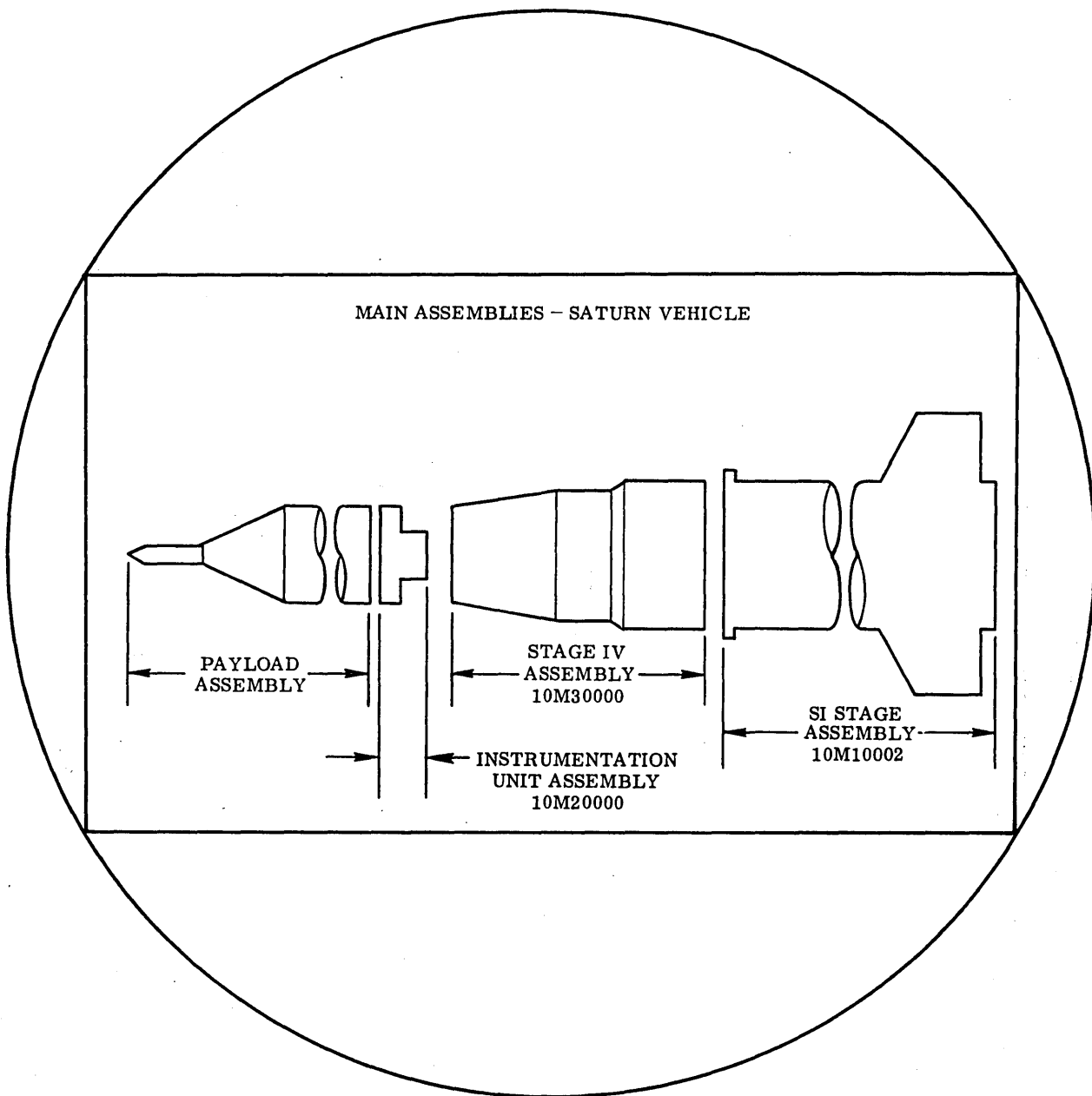


Fig. 5-24 Typical Display - Digitized Drawing



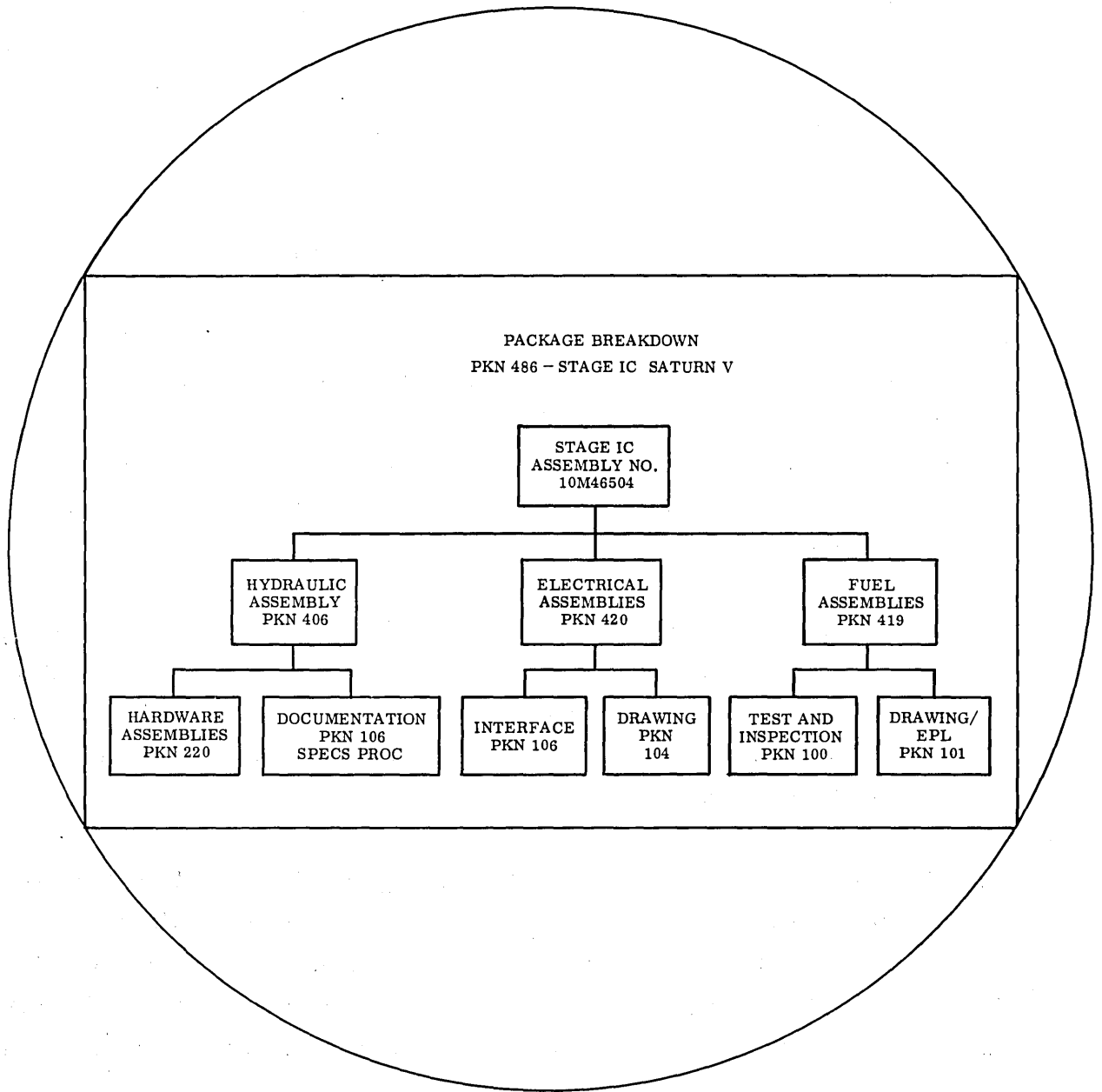


Fig. 5-25 Typical Display - Structure Breakdown

5.4.4.3 Existing Machine-Readable Data. The pertinent machine readable documents used by the MSFC are identified in Table 5-3. The "next assembly" data entry enables the creation of the "used in" file and the separate engineering parts lists provide the basic "explode" capability. The existence of a manufacturer code identification may point to interface documents that exist as the result of the documentation requirements imposed on the contractors.

Not all the documents shown are pertinent to the retrieval system implementation; the full list is given for completeness. Some documents of special interest are the Generation Breakdown or "Christmas Tree" list which provides direct identification of the drawing level, the specific program, and the vehicle number. The Saturn and Apollo Interface Control Document Logs provide Contract End Item (CEI) identification plus release status. The CEI entry enables the retrieval process to proceed directly to the highest assembly level. The Engineering Parts List and Separate Parts List (EPL/SPL) identify the operating division responsible for preparing the drawing as well as naming the design engineer.

Since the engineering function is predominantly one of processing system changes, each drawing in the system typically includes several Engineering Orders (EOs), so that a majority of the supporting documentation consists of EO transactions. The machine-readable document generated when such a transaction takes place represents a primary source of data for entry in the REDI system.

5.4.4.4 Use of Existing Data Systems as Retrieval Aids. The use of existing reference documents to provide entries to the REDI files could be a significant tool. One such reference is the Apollo Document Index, Section 2, TALK (Titles Alphabetically Listed by Keywords), which provides direct reference to interface documents, system reports covering the evaluation of systems and subsystems, and test reports.



## 5.5 HARDWARE STATE-OF-THE-ART

### 5.5.1 Mechanized Pictorial Storage and Retrieval

In surveying the equipment for mechanized pictorial storage and retrieval, a startling variety of microforms and means for retrieving these forms is found. For example, the tabulation given in Van Dam (Ref. 5-1) includes 20 different systems. Other exhaustive listings can be found in Refs. 5-2 and 5-3. The systems described here are typical of some of the larger ones, and have been selected to represent a variety of microforms and retrieval approaches. Good general reviews of the microfilm field can be found in Refs. 4 and 5.

**5.5.1.1 FMA-Filesearch Device.** The Filesearch searches a 1000-ft roll of 35mm film with a maximum of 32,000 document pages per reel. The pages are recorded at a reduction of 25:1, and optically encoded indexing information accompanies each film frame. A 32,000-page reel can be searched at 200 ft per minute (6400 pages), and output may be displayed, printed out as hard copy, or recopied onto roll film. The search operation consists of typing a search descriptor or number in a punched card and then using the card to set up the search pattern in the machine.

**5.5.1.2 Eastman Kodak Minicard System.** The Minicard subsystem of the Defense Intelligence System, one of the largest and most expensive systems developed, uses a unit-record film chip, 16 by 32 mm, as the storage medium for up to 12 pages of input and associated photographic indexing data. The index codes are at one edge of the film chip in a matrix of from 252 to 2730 code bit positions. The film chips are arranged by their respective subject areas, which divides the file into pre-ordered categories which can then be searched separately without resorting to a sequential sort of the entire file. The file may be searched at 1200 chips per minute (up to 14,400 pages). The question statements are supplied on paper tape, and logic is stated by plugboard wiring.

**5.5.1.3 Mosler Selectriever.** The Selectriever provides random access to anyone of 200,000 unit documents – tab cards, aperture cards, or microfiche cards – in

less than 10 sec. The 3-1/4 by 7-3/8-in. card is coded with 35 round holes along the bottom edge, and holes are notched out to identify the card for automated handling. A cartridge holds 100 unit documents and is housed in one of two parallel honeycombed walls, each holding 1000 cartridges. A lightweight cartridge retrieval mechanism operates between the two walls, moving at high speed to retrieve cartridges in response to commands entering the system via keyboard, punched paper tape, or computer interface. The time from command to readout averages 6-1/2 sec; the card is automatically returned to its cartridge; and the cartridge is returned to its storage location in 3 sec.

5.5.1.4 Houston-Fearless Microfiche Retrieval-Display Unit. The Houston-Fearless card retrieval-display unit stores 750 microfiche cards containing 70,000 pages of information. Ten category pushbuttons, each corresponding to a "book" of information, select the classification topic. A "section" set of keys selects the section of the book. The section index is automatically displayed on the screen so that the actual page of information may be selected. The entire selection cycle will not exceed four sec for an operator who has minimal training.

5.5.1.5 Magnavox Magnavue System. The Magnavue system of the U.S. Army Missile Command uses a 35 mm by three-in.-long mylar base film chip with a code field that can store 80 alphanumeric characters (560 bits). The chips are stored 300 per magazine in 300 magazines, providing line random access to 900,000 chips. The control unit can handle two such files for a total of 1,800,000 chip capacity. The photographic codes on the request card are matched against the codes in the master file, and the "hits" are photographed onto a working film or copy card. Response to "random" inquiries on the Magnavue is performed in 60 sec.

## 5.5.2 New Storage Concepts

The advantages of microfilm as a storage medium include an established and relatively inexpensive technology; compact storage; many shapes, forms, types of film; easy reproduction; and permanence. The single disadvantage, however, is the nonerasability of the medium. In an effort to overcome this liability and to create an even

larger storage capacity (or real-time processing), new media have been investigated, including thermoplastic tape, video (magnetic) tape, and photochromic materials.

5.5.2.1 Thermoplastic Film. For several years the General Electric Company has been investigating a type of film known as Thermoplastic Film. This film, like the Photochromic Film, requires no separate development process and is erasable and reusable and can accommodate high reduction ratios.

Recording on this film is accomplished by an electron beam operating in a vacuum. The electron beam forms a charge pattern on the thermoplastic, and the plastic's insulating qualities cause the charge to dissipate slowly. While the charge pattern is on the surface, the thermoplastic is liquified by heating to permit the electrical attractive forces between a conducting layer in the film and the electrons in the charge pattern to form distinctive impressions in the thermoplastic material. When the film cools, the impressions are solidified virtually instantaneously to form the image.

Xerox Corporation is conducting research with a material that works in much the same manner as the G. E. Thermoplastic Film and has the same basic characteristics. Xerox calls its process "Frost."

5.5.2.2 Photochromic Storage. Photochromic materials (PCMI) are light-sensitive organic dyes placed as a molecular dispersion in suitable coatings and applied to a surface. With photochromics, the molecular coating can be exposed with ultraviolet radiation and erased with white light. It is this reversible characteristic that makes PCMI attractive. Furthermore, it is grain free, thus permitting the recording of very high density micro-images.

A reduction ratio of 200 to 1 allows for the storage of 2500 pages on one three by five in. transparency. At present, the PCMI process to produce micro-images is available at National Cash Register on a service bureau basis. A three by five inch master plate costs \$250 with contact prints made from the master at \$0.50 to \$1 each depending on quantity. Thus, there is the ability to mass produce micro-image dissemination files economically. Thus far, document retrieval systems have tended in the direction

of providing a central store of images that could be placed on-line through remote consoles. PCMI would allow nonreturn copies to be sent out to the remote stations.

5.5.2.3 Video Recording. The Videofile system, an automated image storage system using magnetic tape developed by Ampex Corporation, has the capability of storing a large amount of graphic material with fast access to any particular required document. The stored images can be added to, updated, or purged under the application of external controls with tape images immediately ready to view without processing. They have an archival permanence as long as they are needed, and they are of high quality on viewing. The techniques employed in this system are based on principles similar to those used in commercial television recording. The documents put into the system are scanned by a high resolution television camera and recorded electronically on two in. wide magnetic tape as a video signal.

Each image occupies either a 1/3-in. length of tape (for 11-1/2-by 14-in. documents) or a 1/6-in. length of tape (for 5-1/2-by 8-1/2-in. documents) depending on which of two standards are employed. For the larger documents (or higher resolution of smaller documents), 1280 scan lines at 15 frames per second are used. For smaller documents, 640 scan lines at 30 frames per second are used. The standard involved determines the number of document images that can be stored on each reel.

With each image written on the magnetic tape is a digitally coded address to identify that image. The address can consist of a combination of either 12 alphanumeric characters or 18 numbers.

On recall, the tape with the stored images is traversed at high speed until the location of the required document is reached. The image is then read from the tape in the form of a video signal and can be displayed on a monitor (soft copy) and/or printed out (hard copy).

5.5.2.4 Dynamic Updating of Filmed Base. An interesting new concept which has been fabricated into prototype hardware is reported by the MITRE Corporation (Ref. 5-6). Two magnetic tracks stripped on roll microfilm are used to allow

dynamic updating of the filmed base. Consecutive search capability (i. e. , store the present position and search forward or reverse from it) provides a significant decrease in the retrieval response time.

A basic problem with this approach seems to be the reliability of the magnetic striping for storing digital codes.

### 5.5.3 The AMACUS System

The maintenance of engineering design data in hard copy media is time consuming and costly. AMACUS is a system that enters drawings to the computers via microfilm operations cards and an optical manner. The Automatic Microfilm Aperture Card Updating System (AMACUS) is currently being developed by General Precision, Link Division, for the U. S. Army Weapons System Command.

AMACUS is used primarily as a system for incorporating design changes into existing design. Graphic data are entered into a computer via an aperture card read by an optical scanner. The image is displayed simultaneously on an aperture card viewer and on a CRT. The CRT unit is equipped with a light pen for altering the geometric pattern being displayed on the tube. The selection of one of three line widths permits matching the alteration with the original design. An alphanumeric keyboard and a keyboard containing standard geometric patterns and electrical symbols is used for entering applicable data.

AMACUS is currently scheduled to be installed at the U. S. Army Weapons Command during December 1967. An advanced AMACUS with storage and remote retrieval/input capability is currently being developed as part of a total data system for the U. S. Army Weapons Command. This development is directed by Mr. M. D. Silkiner, Chief, Management Science and Data Systems Office.



## 5.6 REFERENCES

1. A. Van Dam, A Survey of Pictorial Data Processing Techniques and Equipments, University of Pennsylvania, AD 626 155, Aug 1965, 160 pp. (plus extensive tables). This study was performed for the Bureau of Supplies and Accounts of the Navy, and covers the entire range of pictorial data processing including display, transmission, storage and retrieval, optical pattern recognition, and man/machine interaction. The study covers the principles of operations of the various devices, and then presents extensive tabulations of equipment characteristics for the various categories of device. The scope of this report is quite broad, and should be required reading for analysts working with pictorial data systems.
2. Libraries and Automation, Library of Congress, 1964  
This is the Proceedings of the Conference on Libraries and Automation covering seven topics: the library of the future, file organization and conversion, file storage and access, graphic storage (current status of graphic storage techniques with application to library mechanization), output printing, library communications networks, and automation of library systems.
3. C. P. Bourne, Methods of Information Handling, New York, Wiley, 1963  
This book provides an illustration of the tools, equipment, and methodology that can be applied in the design of information systems. Of particular interest to retrieval sciences are the chapters "Classification and Indexing" and "Microfilm and Image Handling Equipment."
4. E. J. Menkhaus, "The Many New Images of Microfilm, "Business Automation, Oct 1966, 11. 32-43  
A good general review of the microfilm field.
5. G. M. Lear, "A Review of the Status of Existing and Advanced Micro-image Systems," Proceedings of Air Force/Industry Management Symposium, published by Ballistics Systems Division, Norton Air Force Base, Calif. , N 66 38742, Sep 1965, pp. 257-266  
An excellent, concise review of the microimage state of the art and its relation to Air Force operations.

6. G. Barboza, Summary of Efforts Expended on the Concept of Dynamic Update of a Microfilm Film, The MITRE Corp., AD 620 279, Jul 1965, 84 pp.  
Description of the prototype hardware that was designed and fabricated to prove the feasibility of combining magnetic and photographic techniques of information storage.

Section 6  
STUDY OF MAN/MACHINE SOLUTION OPTIMIZATION

6.1 INTRODUCTION

The man/machine relationship must be a complementary one with each assisting the other in the things each does best. The relationship can be thought of as a chain formed of the basic links:



in which the man makes decisions which determine succeeding computer operations. It may be helpful to think of the human nervous system and the electronic system as two computers with vastly different characteristics. One of the distinctive qualities of the man in this context is that he requires far less detailed instructions than a machine and can make decisions which, if they were to be programmed into a machine, would lead to an immensely complicated decision logic.

The man, nevertheless, does require "programming" or instructions to aid in fulfilling his role. It is assumed that he is instructed in the techniques of dealing with the machine but, in addition, he must have a plan or program for dealing with the content of his work and the alternate decisions he may make as a result of the outcome of computing operations.

6.1.1 Objective of the Program

The purpose of the work reported here is to establish the criteria of cooperation between the man and the machine to maximize the effectiveness per hour of machine utilization. For this purpose it appeared essential to use an already established

program in which the role of the operator was not so involved as to obscure the essentials of his performance, and yet simple enough so that personnel unfamiliar with the program could learn the procedures and perform the cooperative tasks.

## 6.2 DESCRIPTION OF PROGRAM

The 144 Curve Fit Program was chosen for the man-computer experiments because it is straightforward enough so that the principles of investigation could be recognized without being obscured by excessive detail. This program fits experimental data by up to fourth order polynomials; the X and Y variables can be transformed by log, exponential or reciprocal transformations. This amounts to 144 possible curves. The program computes the mean square fit for all forms simultaneously and lists the 10 best fits on the surface of the CRT. The man can then choose any of the 144 curves for the graphical display CRT. The curve chosen for permanent record can be reproduced (i.e., by a Polaroid camera or by tape onto an SC-4020 for computer prepared plot); the set of points can be corrected on the CRT at the display console; and further runs can be made if desired by the operator. A user's flow diagram for subsequent experiments is shown in Fig. 6-1. The program handles 31 points.

### 6.2.1 Boxes D and J (Fig. 6-1)

The man can predict that the plots will follow certain functional forms on the basis of physical knowledge or inspection of a graph of the data, for example:

1. For chemical rate equations,  $\log \text{ rate} \approx (1/\text{temperature})$ .
2. Scaling laws are frequently straight lines, in suitable coordinates.
3. Inspection of Fig. 6-8 (shown later) suggests the form  $Y = \text{constant} + 1/X$ .\*

---

\*If high confidence is held for the correct form, the computer could be instructed to consider this form only.

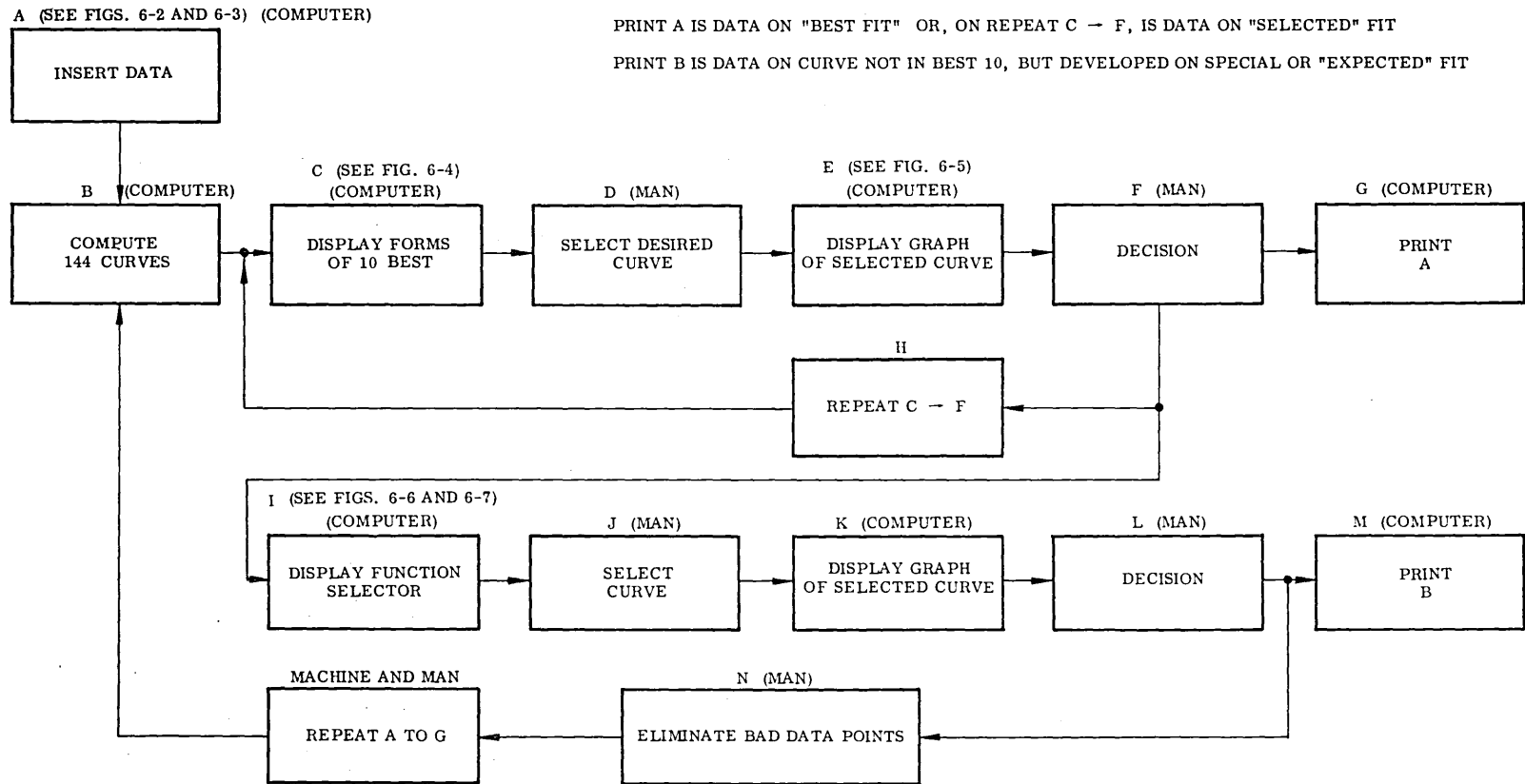


Fig. 6-1 Flow Diagram Used for the 144 Experiment

### 6.2.2 Original Instrumentation Errors

The man has information on original instrumentation errors based either on knowledge of the measuring instrument or inspection of the graph. This can influence his acceptance of a given machine plot. For example, if the residual variance is not much greater than that expected from instrumentation error, a curve can be accepted even though it is not a best fit. (In principle, decision rules could be formulated and programmed into the machine, but these rules might depend on the type of data.)

### 6.2.3 Boxes F and L (Fig. 6-1)

With the possibility of alternate forms, the man might choose a form more physically acceptable even though it was not the best fit, for example:

1. Many physical laws can be expressed as straight lines in suitable coordinate systems and the resulting intuitive tendency is to accept a linear solution over a higher order form.
2. If the formulas are going to be used in complex systems analyses, the advantages of a simple formula may outweigh considerations of accuracy.

### 6.2.4 Box N

The man can use his knowledge of instrumentation error or inconsistencies of plotted data to recognize wild points. For example, the circled point shown later in Fig. 6-12 is obviously suspicious. This point could be deleted prior to entry into the machine. (Decision rules could be programmed for such conditions if these situations were frequent enough to warrant the programmatic complexity.)

## 6.3 EXPERIMENT DATA AND OPERATIONS

These runs were made to verify the adequacy of the instructions, to establish characteristic times for the functions shown in the boxes of Fig. 6-1, to see what type of

decisions were best made by the man, and to use the results in formulating suitable criteria for instructions to be generated for making decisions in the general case.

For these experiments sets of pairs of data points (X, Y) were put on punch cards. These sets were collected from sources having known relationships and statistical error magnitudes in the point determinations. The three trials reported used three of these sets of data. Plots of the three sets of data were made, and, with estimates of the best fit curves, are shown in the following figures. Each data run presented the displays having items similar to those shown in Table 6-1. The X variable is resistance and the Y variable is transconductance. They are expected to be of the form  $Y = F(1/X)$ .

### 6.3.1 Discussion of Table 6-1

Item 1 is a machine operation of accepting the cards and displaying the pairs of numbers in X and Y (Fig. 6-2). The recorded time is from insertion to display. The man can take as long as is needed to make a comparison to some other data list. (Usually, comparing the cards with the data list should be done in advance, and the later point display in Item 4 is a better comparison.)

Item 2 is a machine operation ordering the cards in increasing value of X (Fig. 6-3).

Item 3 is a machine operation displaying the 10 forms of the equations giving the best fits (Fig. 6-4). (This will not always be a quartic as might be expected since these are ordered in increasing standard deviation in which the sum of the squares of the individual residuals is divided by the number of points minus one, minus the order of the curve.)

Item 4 is the display of a curve selected from the best 10 (Fig. 6-5). A small part of this time, e.g., five sec, is used by the operator in selecting the curve. The greater part of the time is used in computing, storing and erasing the display, and displaying the curve.

Item 5 consists of recalling the function display, five sec, selecting a new curve form, 10 sec, and generating and displaying a second curve (Fig. 6-7).

	X		Y		DATA SET
				144	CURVE FIT PROGRAM
1.	2.196000000E	03	5.027000000E	02	
2.	2.311000000E	03	4.720000000E	02	
3.	2.320000000E	03	5.126000000E	02	
4.	2.739000000E	03	4.379000000E	02	
5.	2.991000000E	03	4.767000000E	02	
6.	3.053000000E	03	4.246000000E	02	
7.	3.543000000E	03	4.251000000E	02	
8.	3.755000000E	03	4.519000000E	02	
9.	3.774000000E	03	4.227000000E	02	
10.	5.594000000E	03	3.943000000E	02	
11.	5.795000000E	03	3.822000000E	02	
12.	5.820000000E	03	3.941000000E	02	
13.	6.616000000E	03	3.565000000E	02	
14.	9.196000000E	03	3.444000000E	02	
15.	9.585000000E	03	3.581000000E	02	
16.	1.112000000E	04	3.352000000E	02	
17.	1.131000000E	04	3.513000000E	02	
18.	1.233000000E	04	2.827000000E	02	
19.	1.336000000E	04	2.710000000E	02	
20.	1.363000000E	04	2.970000000E	02	

Fig. 6-2 Complete Data Display



6-7

DATA SET			
CLAVE	DTT	PROGRAM	
1.363000000E	04	2.970000000E	02
1.336000000E	04	2.710000000E	02
1.233000000E	04	2.827000000E	02
1.131000000E	04	3.513000000E	02
1.112000000E	04	3.352000000E	02
9.583000000E	03	3.581000000E	02
9.196000000E	03	3.444000000E	02
6.616000000E	03	3.363000000E	02
5.420000000E	03	3.941000000E	02
5.793000000E	03	3.822000000E	02
5.394000000E	03	3.843000000E	02
3.774000000E	03	4.227000000E	02
3.755000000E	03	4.319000000E	02
3.543000000E	03	4.251000000E	02
3.055000000E	03	4.246000000E	02
2.991000000E	03	4.767000000E	02
2.739000000E	03	4.379000000E	02
2.320000000E	03	5.126000000E	02
2.311000000E	03	4.720000000E	02
2.196000000E	03	5.027000000E	02

Fig. 6-3 Display of Data Before Acceptance Into Data Set

BEST CURVES

CURVE	FUNCTION	TRANSF	ORN	DEGREE
$Y=1/LG(P(X))$	$Y=1/LG \quad F(X)$	$F(X)=P(X)$		3
$Y=1/P(X)$	$Y=1/F(X)$	$F(X)=P(X)$		3
$Y=E(F(X))$	$Y=E(F(X))$	$F(X)=P(X)$		3
$Y=F(X)$	$Y=F(X)$	$F(X)=P(X)$		3
$Y=P(LG \quad X)$	$Y=F(X)$	$F(X)=P(LG \quad X)$		3
$Y=E(F(X))$	$Y=E(F(X))$	$F(X)=P(X)$		4
$Y=1/P(X)$	$Y=1/F(X)$	$F(X)=P(X)$		4
$Y=1/LG(P(X))$	$Y=1/LG \quad F(X)$	$F(X)=P(X)$		4
$Y=P(X)$	$Y=F(X)$	$F(X)=P(X)$		4
$Y=P(1/LG \quad X)$	$Y=F(X)$	$F(X)=P(1/LG \quad X)$		5

8-6

Fig. 6-4 Display of Functions of 10 Best Curves

	X		RESIDUAL	
1.	2.196000000E	03	8.131521635E	00
2.	2.311000000E	03	-1.533317568E	01
3.	2.320000000E	03	2.581433256E	01
4.	2.739000000E	03	-2.606810293E	01
5.	2.991000000E	03	2.427174433E	01
6.	3.055000000E	03	-2.512129514E	01
7.	3.543000000E	03	-6.532116465E	00
8.	3.755000000E	03	2.690805215E	01
9.	3.774000000E	03	-1.728311844E	00
10.	5.594000000E	03	6.738309965E	00
11.	5.795000000E	03	-2.744741976E	00
12.	5.820000000E	03	9.467393957E	00
13.	6.616000000E	03	-1.943518160E	01
14.	9.196000000E	03	-8.914993420E	00
15.	9.585000000E	03	8.786885761E	00
16.	1.112000000E	04	6.098946534E	00
17.	1.131000000E	04	2.527477429E	01
18.	1.233000000E	04	-2.452767854E	01
19.	1.336000000E	04	-1.365543430E	01
20.	1.363000000E	04	1.875371826E	01

CURVE  
 $Y=1/LOG(F(X))$   
 DEGREE 3  
 COEFFICIENTS  
 A= 1.002843281E 00  
 B= 1.072517610E-06  
 C= -1.220266330E-10  
 D= -1.220266330E-10  
 E= 5.338457911E-15

STANDARD DEVIATION  
 1.957329741E 01

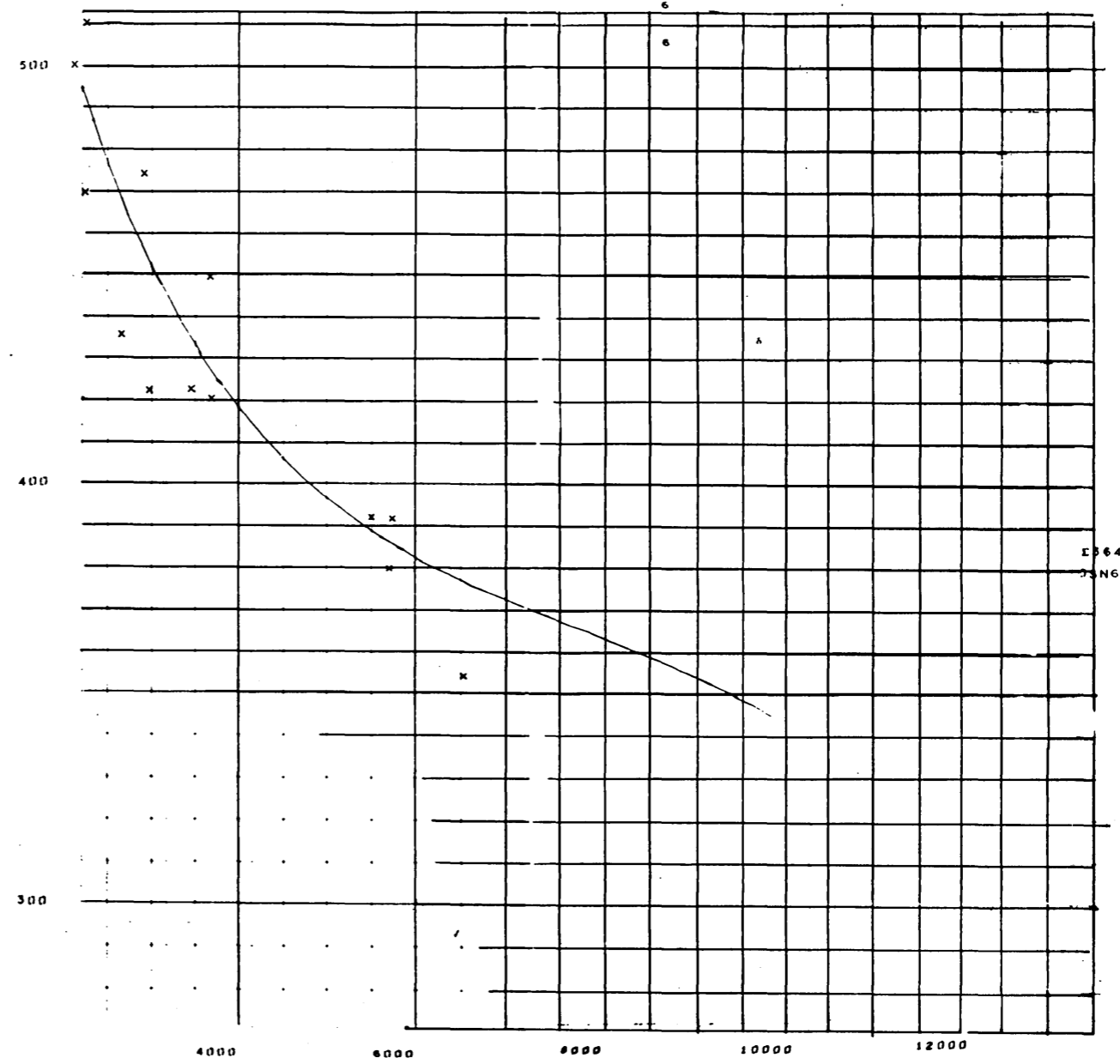


Fig. 6-5 Display of Best Curve  

$$y = \frac{1}{\text{Log}} (A + BX + CX^2 + DX^3)$$

Table 6-1  
TIME ELEMENTS

Item	Function	Operator	Time (sec)
1	Enter Cards and Compare to Data List	Computer	
2	Accept Data and Reorder Cards in "X"	Computer	5
3	Compute 144 Fits, Select and Display 10 Best Fit Forms	Machine	10
4	Select Specific or 10 Best and Display Curve (1)	Man- Machine	10 134
5	Select Specific from Function Display, Compute and Display Curve (1)	Man- Machine	35 135
6	Enter Title	Man	120
7	Print Out (2)	Machine	4

- (1) The curves of which Items 4 and 5 are examples can be recorded on magnetic tape and printed out on a curve plotter, i. e., SC-4020.
- (2) The equations of the curves and the 10 best curves can be determined in 10 sec since printout can be done anytime after Step 3. The time after this step involves about 2 min for each alternate called up for examination by the man.

Item 6 consists of selecting letters, spaces, etc., on the display tube using a light pen. The time to display a full title is large and the man would do well to use a short coded identification, filling in the title later (Fig. 6-6).

Item 7 consists of recording on paper tape the 144 curves, the 10 best, and any others of interest (Fig. 6-7).

### 6.3.2 First Trial Runs

The first trial was designed to establish the completeness of the set of instructions written for the operator of the 144 Curve Fit Program. Three separate operators,

	FUNCTION	
Y=F(X)	Y=1/F(X)	Y=E (F(X))
Y=E(1/F(X))	Y=LG F(X)	Y=1 /LG F(X)
	TRANSFORM	
F(X)=F(X)	F(X)=F(1/X)	F(X) =P(LG X)
	DEGREE	
F(X)=F(1/LG X)	F(X)=F(E(X))	F(X) =P(E(1/X))

Fig. 6-6 Display for User to Select a Function, Transform and Degree

	X	RESIDUAL	
1.	2.196000000E	2.775402436E	03
2.	2.311000000E	8.118927479E-02	03
3.	2.320000000E	4.091647050E	03
4.	2.739000000E	-2.308369545E	03
5.	2.991000000E	2.192089826E	03
6.	3.055000000E	-2.862983675E	03
7.	3.543000000E	-1.665499960E	03
8.	3.755000000E	1.495090254E	03
9.	3.774000000E	-1.382348063E	03
10.	5.594000000E	-4.969563425E	03
11.	5.795000000E	-1.334151898E	03
12.	5.820000000E	-9.826942235E-01	03
13.	6.616000000E	-2.451040811E	03
14.	9.196000000E	2.823618211E	03
15.	9.585000000E	2.177202122E	03
16.	1.112000000E	1.809836072E	04
17.	1.131000000E	3.642636046E	04
18.	1.233000000E	-2.072853819E	04
19.	1.336000000E	-2.168563170E	04
20.	1.363000000E	7.005786851E	04

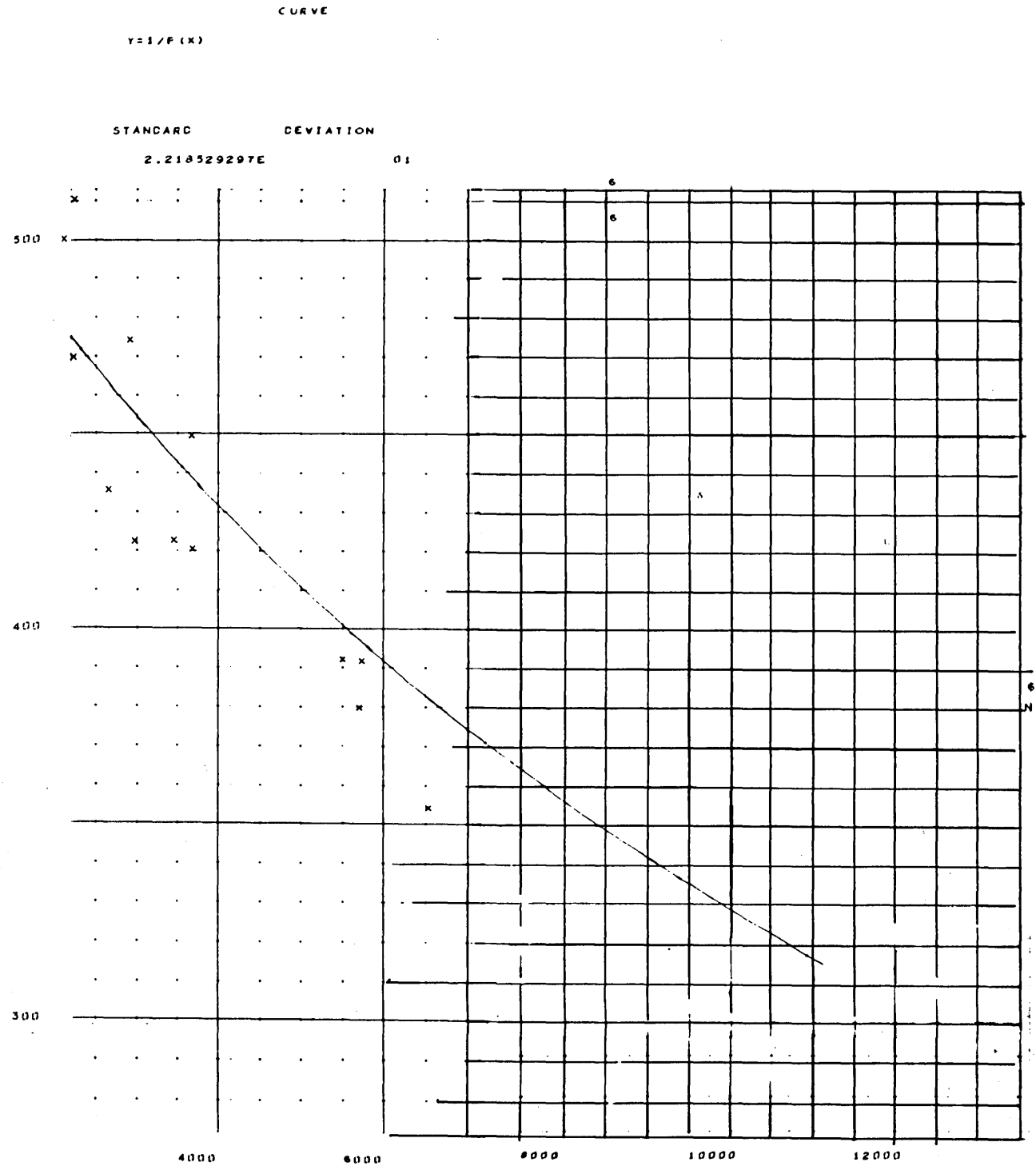


Fig. 6-7 Display of Chosen Curve From Box I of Fig. 6-1 ( $y = A + B/X$ )

none having previous experience on the program, performed the operations of the Curve Fit Program. Each operator, with less than an hour of experience, could operate the program with times approaching those shown in Table 6-1.

The points of the curves, the X-Y pairs, were chosen from a source which could be expected to produce a curve where Y was proportional to the reciprocal of X, including a moderate value of statistical error.

Figure 6-8 shows the hand plot and the operator's best estimate of the curve fitting the points. Figure 6-9 shows the machine fits, Curve A being the best fit and B being the best of the expected form.

In Fig. 6-8, the origin is not suppressed as it is in Fig. 6-9, so that the visual comparison of the curves is not easily accomplished; however, it is obvious that the form of the curve can be identified and the correct values of the coefficients established by the 144 Curve Fit Program.

### 6.3.3 Second Trial Run

The second trial was performed to see if a set of points could be recognized by both the operator and the machine. The points represent a cubic relationship between Y and X. Figure 6-10 represented best estimate of one operator and Fig. 6-11 presents the curve generated by the Curve Fit Program. (The randomness introduced into this set of curves was so great that the set would probably be discarded if it were being used in any realistic situation.)

### 6.3.4 Third Trial Run

The data used in the third trial contained a random point which, by examination of Fig. 6-12, is seen to be a wild point and can be excluded immediately if the data is plotted. Figure 6-13 contains the best fit curves including the point Curve A and

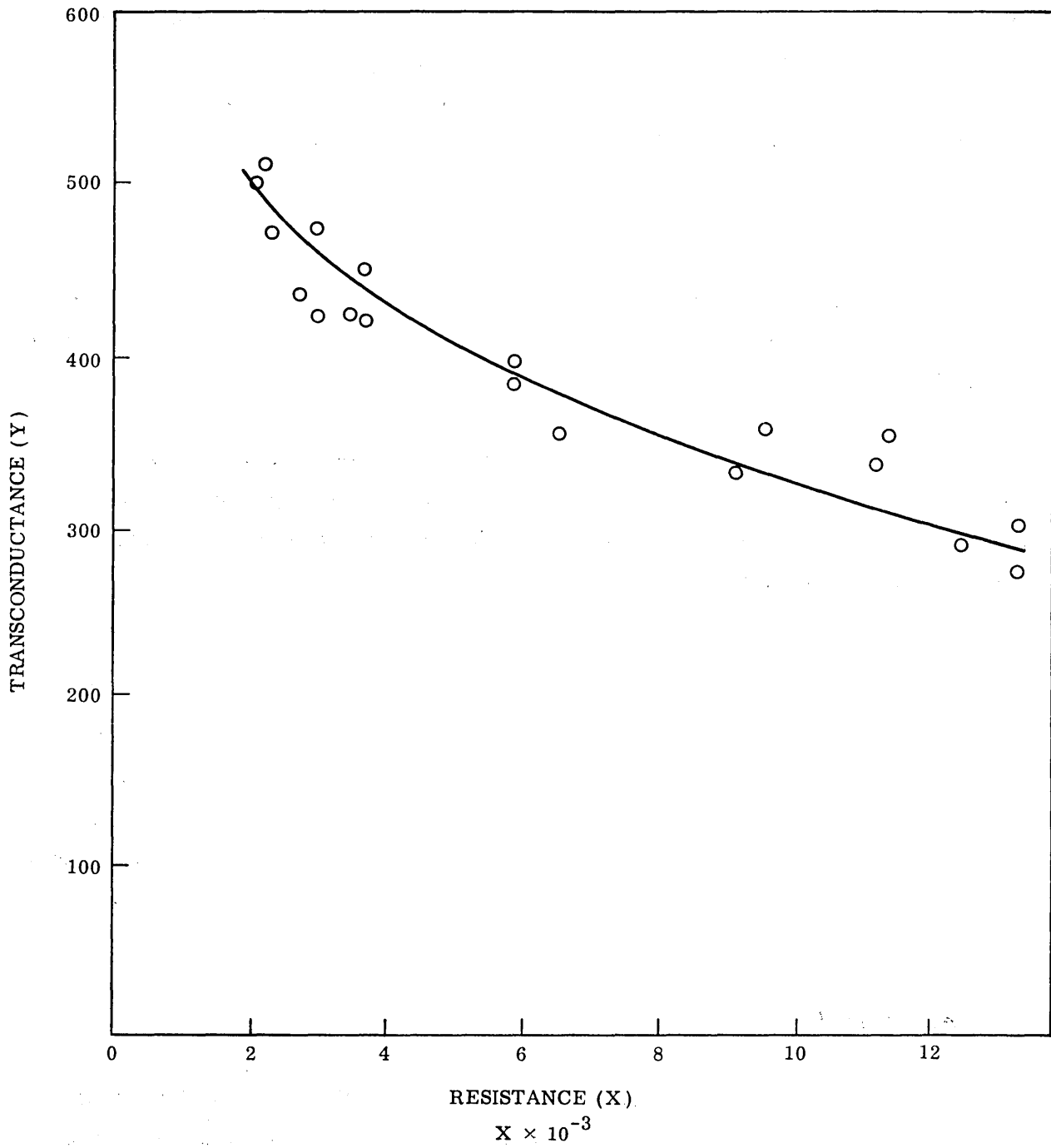


Fig. 6-8 Run 1, 144 Curve Fit Program, Preplot and Curve



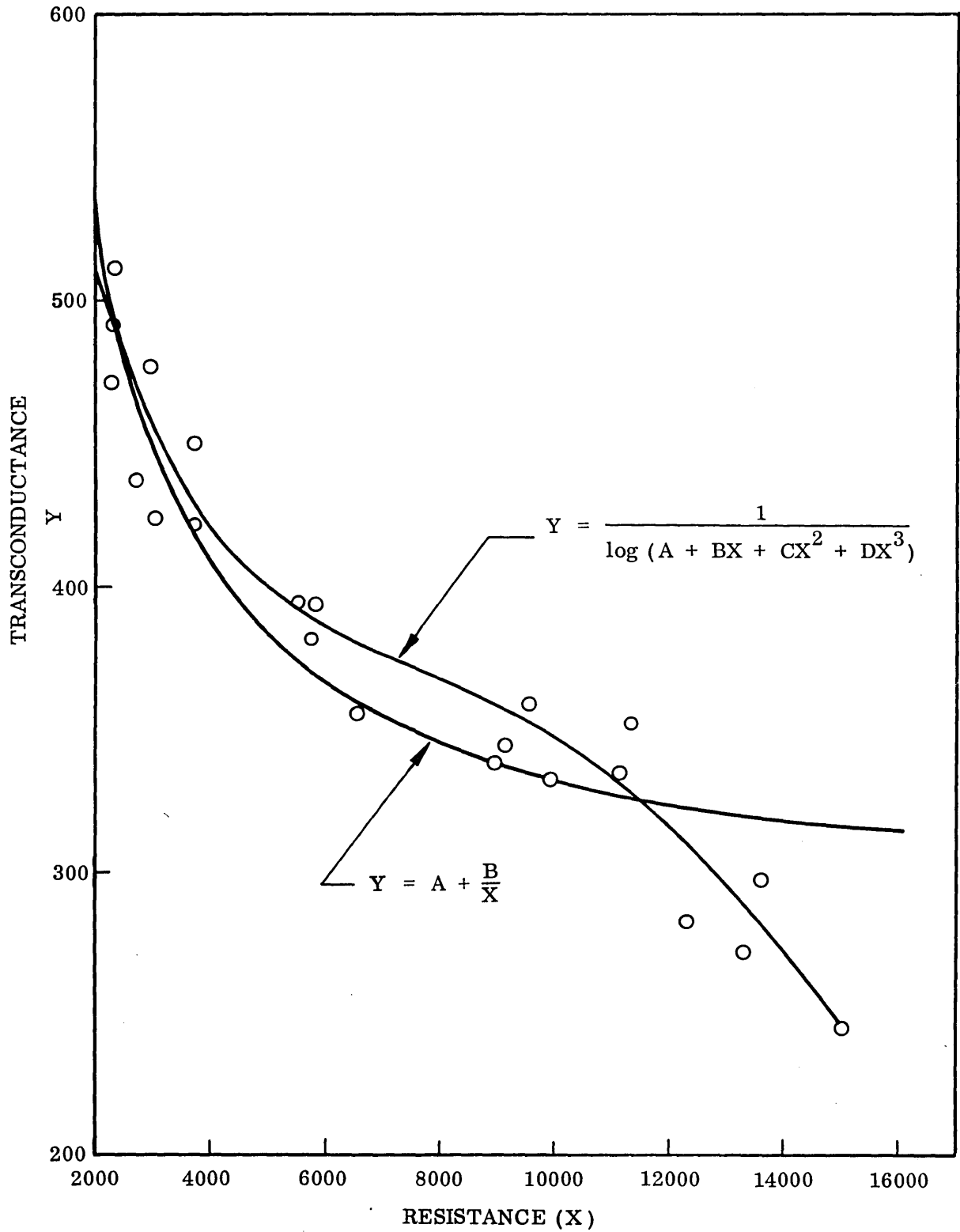


Fig. 6-9 Run 1, Computer Output

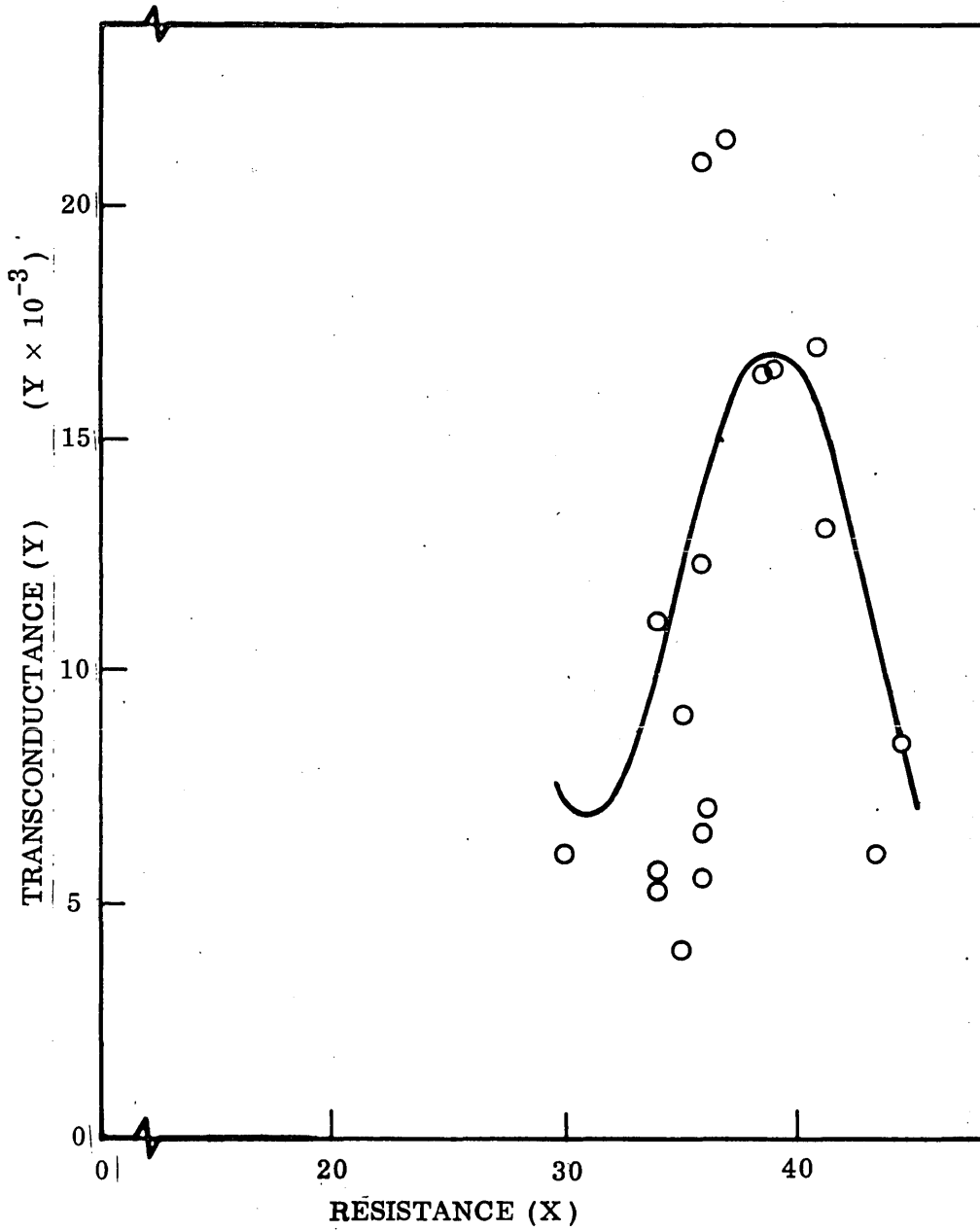


Fig. 6-10 Run 2, 144 Curve Fit Program

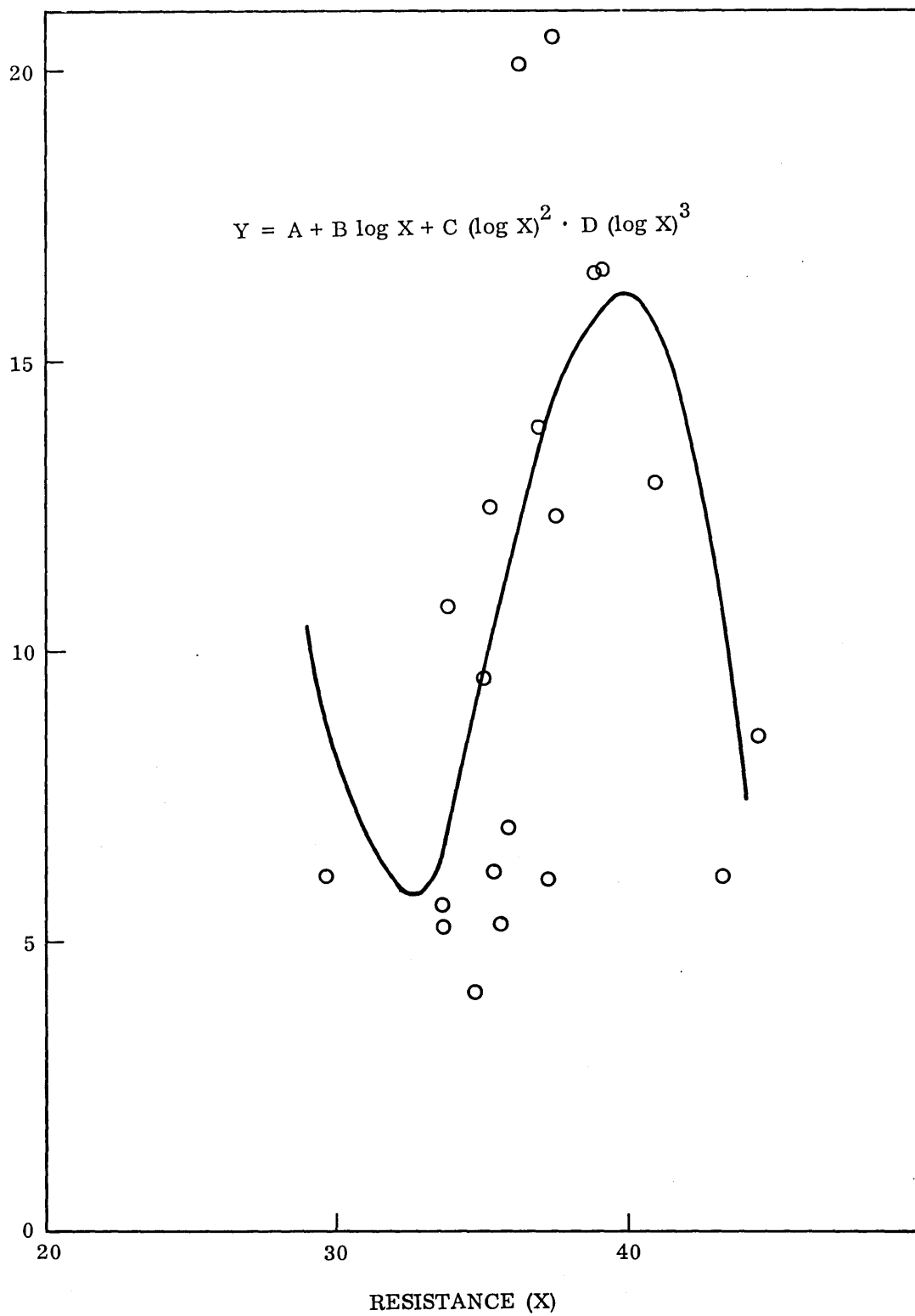


Fig. 6-11 Run 2, Computer Output

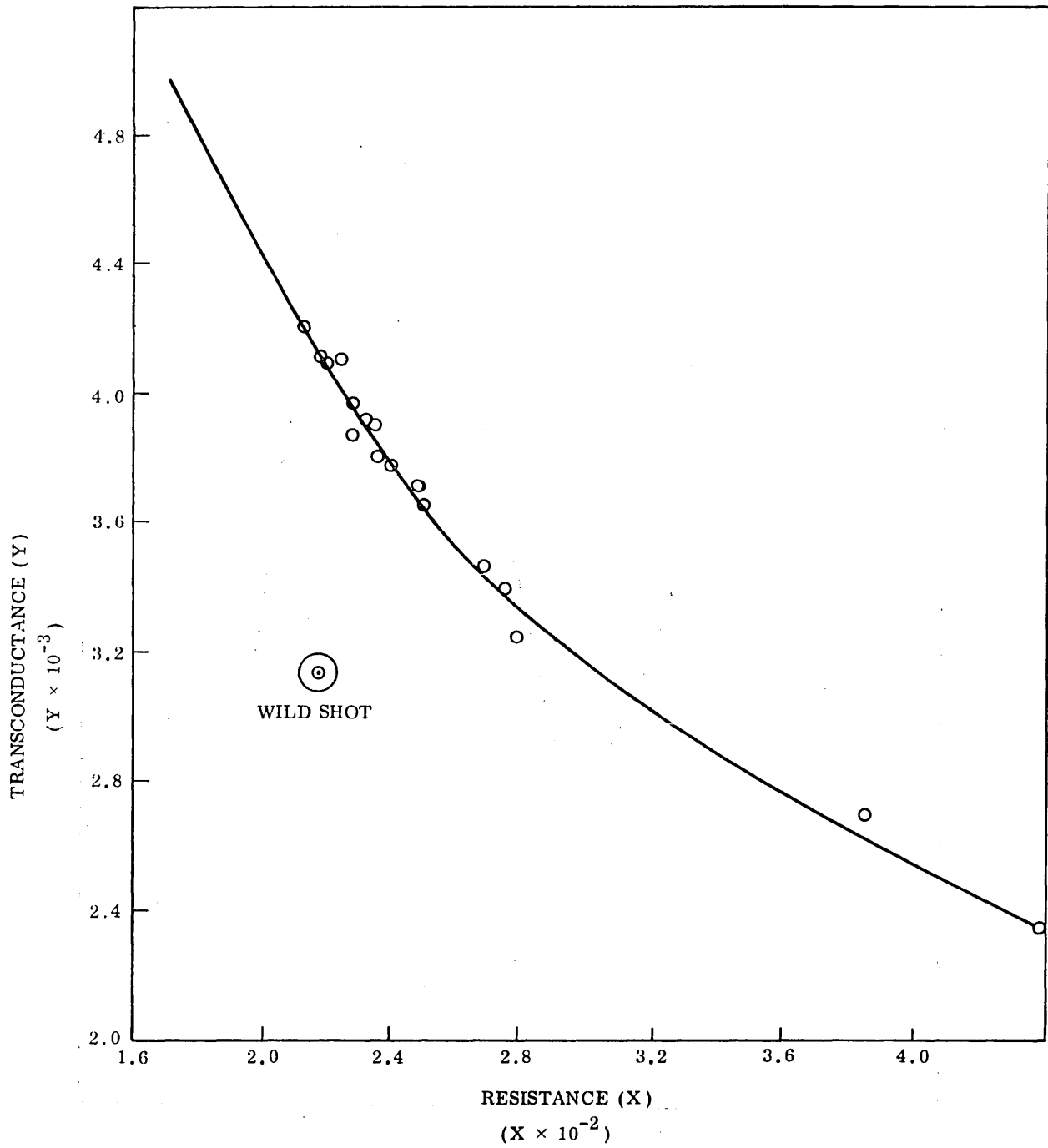


Fig. 6-12 Run 3, 144 Curve Fit Program, Preplot and Curve

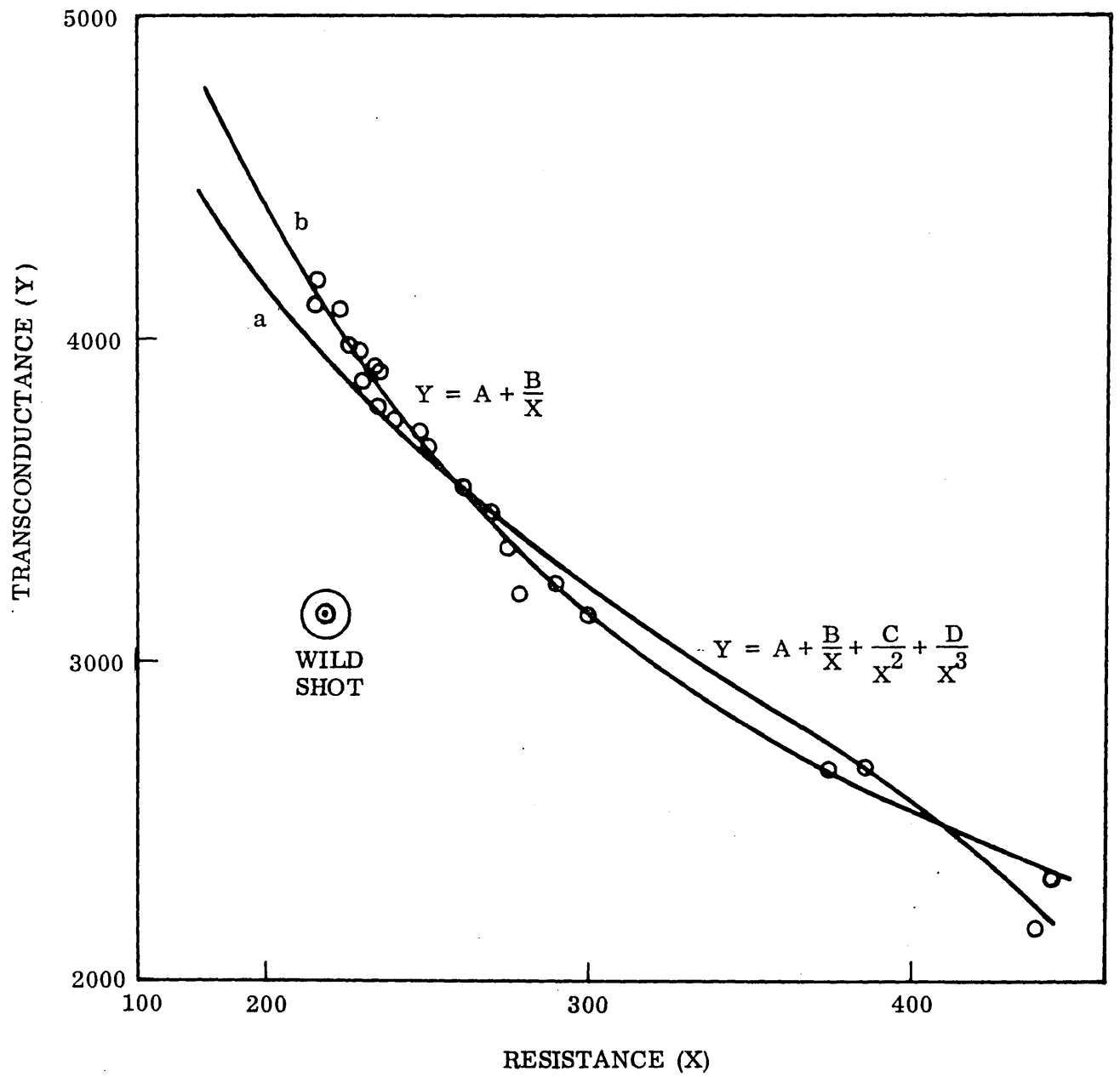


Fig. 6-13 Run 3, Computer Output

excluding the point Curve B. This trial demonstrates the difficulty of programming the machine to ignore bad data as compared with the capability of the operator.

From these results it can be seen that the man can be more discriminating concerning items which are not programmed into the machine. He can make a better selection of "wild points." He can force the acceptance of curves which might be rejected by the computer and he can choose whether or not to use data at all. The time results also show how the user can save time by exercising discipline in choosing which curves to display.

#### 6.4 INSTRUCTIONS

The operator should prepare himself for the particular program before working at the machine. In the present program, he should have plotted the points estimated for the best fit curve, its form and degree, and ask for this immediately from the machine, thereby reducing the total display time. The data should be prepared for input so that the fastest approach is used. In the present case, the use of punched cards reduced the input time from 10 min (console/light pen input using A/N Register vs cards) to about four sec. In general, this preparation should include (at the minimum):

- a. Before Going to the Graphics Console:
  1. Graph data
  2. Mentally review physical considerations
  3. Decide on most likely functional form
  4. Select possible alternates
  5. Estimate accuracy of data
  6. Clean up obvious wild points
- b. Use of Computer:
  1. Punch cards (Do not use light pencil for original input.)
  2. Run through steps A - F

3. If first choice appears in top ten, call for display:
  - (a) If this curve is obviously acceptable, print and leave.
  - (b) If curve is not obviously acceptable, call for alternates.
  - (c) If one or more alternates are obviously acceptable, print and leave.
  - (d) If some alternates cannot be obviously accepted or rejected, print out and go to desk for further consideration.
4. If first choice does not appear, either call for first choice and look at it or call for alternates.
5. If no alternates appear on display, go back to desk and consider possible reasons.

NOTE: Only decisions which are obvious (i.e., can be made quickly) should be made at the console.

#### 6.5 SUGGESTIONS FOR MORE EFFICIENT DIVISION OF LABOR BETWEEN MAN AND COMPUTER

The original designer of any man-machine program should keep in mind the speed of the computer, the speed of the mechanism which communicates with the man, and the speed of the man in making decisions as well as the speed with which the man can communicate with the machine.

This implies that the computations should be done by time-sharing if a high-speed computer is used since a very small fraction of the time is generally spent in calculations.

As can be seen from Table 6-1, the longest times are associated in generating a change in the display, i.e., for generating curves as in Items 4 and 5, or of generating a title as in Item 6. Part of this time is utilized in erasing the existing display; however, as can be seen from Item 2, this is not too great an interval. A major portion of the time consumption is in generating the displays of curves as shown in Items 4 and 5. It appears that this is because we have copied too slavishly man's method of operations.

Since the psychology and physiology of the computer differs markedly from man's, it can be expected that suitable strategies differ correspondingly. Also, since the man is more flexible than the computer, he should expect to adapt himself to the limitations of the computer.

With respect to curve plotting, the man does not like to compute points. He has a good interpolator in the French curve. He also needs coordinate paper as an aid to placing points. On the other hand, the computer is extremely efficient in computing points and does not need paper. It does, however, have difficulty in stringing them into a continuous curve. The curve generation could be shortened by dropping the coordinate grid and printing only discrete points (50 might be enough) and allowing the man to do the interpolation visibly as part of his evaluation of fit. Part of man's adaptability is this capability to visualize the connections between close points.

## 6.6 SIMULATION

Lockheed has prepared a simulation code, LOMUS I, which simulates operations of a complex "job shop" where long predictable runs are not realized. A large computer facility has such a character. The output can answer such questions as:

1. Will the cost of adding another machine, component, subsystem or operation be more than offset by the increase in service?
2. Should a system structure of type A or type B be acquired?
3. Could system performance be improved through the use of different scheduling rules and information networks?
4. What procedures should be followed to minimize the disruptions due to a computing machine failure?
5. What is the probable impact of a priority scheme upon system performance?
6. How do manpower variations affect system cost?
7. How are the dynamics of queues affected by work load variations?
8. How does the average system response time vary with work load?
9. How does the input job mix affect machine utilization?



The LOMUS I program requires information based on operating experience. This information includes data tables and parametric input. A list of these from LOMUS I is as follows:

#### Data Tables

- Manning
- Workload initialization
- Software
- Transit time
- Function volume relationships
- Distributions
- Arrival rates
- Machine name and cost

#### Parameters

- Workload initialization and rate
- Debug percentage
- Scheduling rule
- Routing control
- Manpower loading
- Constraint complexity
- Priority allocation
- Value control
- Application area

The addition of the man in the loop would emphasize the variability of estimated time to complete a job due to choices which a man can make during the running of the program. The other inputs would be the same. Therefore, no modifications to the program are recommended at this time.



## Section 7 HARDWARE ANALYSIS

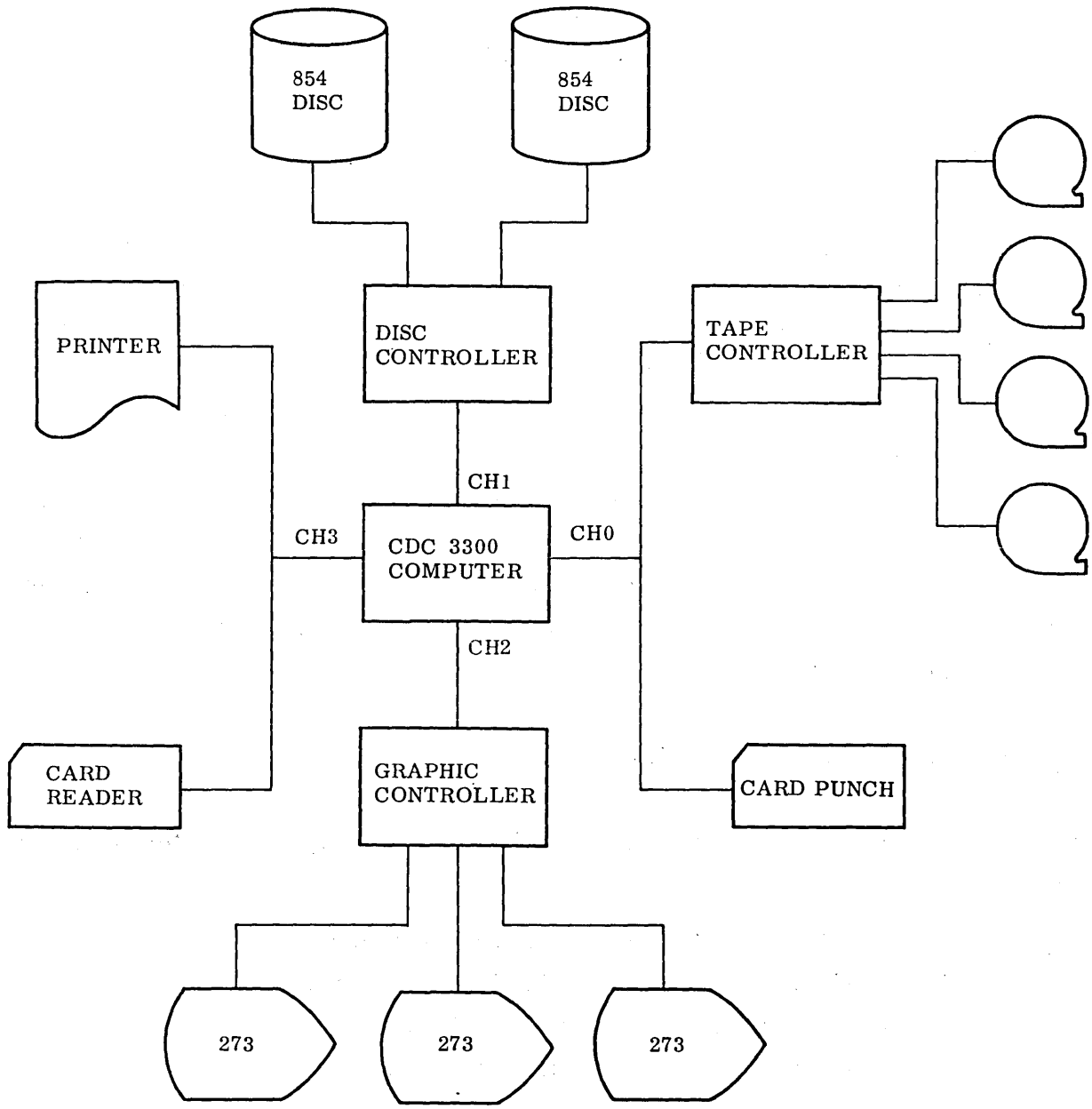
### 7.1 INTRODUCTION

It is the purpose of this section to discuss the compatibility of the programming specifications contained in this report with the Univac-supplied Advanced Mariner Display as described in the Univac Product Description S-90060. Because the Univac specifications were not available until late in the contract period, all specifications were written assuming a generalized hardware and software system. The Control Data Corporation (CDC) Digigraphic 270 system typifies most of the desirable features of a computer graphics configuration. It was used as a standard system with which LMSC had had experience and it could be easily written about. Therefore, this section will describe the CDC system, discuss the influence some of the assumptions have had on the programming specifications, and evaluate some of the changes which will be required by the use of the Univac system.

### 7.2 HARDWARE SYSTEM CONFIGURATION

The assumed hardware system as configured in Fig. 7-1 comprises the following types of units:

- Three CDC 273 display units and a controller
- A CDC 3300 computer with 32,768 words of memory
- Random storage on two, 2-million word disc drives
- A high-speed printer (1000 lines per minute)
- Four tape units
- A card punch



3 DISPLAY CONSOLES

Fig. 7-1 Hardware Configuration

Assumption of this system will influence such decisions as the following: a. Should the display console operator be given the option of entering parameters on cards or must he enter all parameters through the display interface? b. Can the operator get immediate access to printed output or must the display console software provide for paging through lists of output parameters? c. Will the program fit into the memory of the machine in one segment or must it be partitioned?

Upon configuring the target Univac system, each of these questions and others will have to be asked again and the answers reevaluated.

### 7.3 SOFTWARE SYSTEM

The assumed software support is based on the CDC Functional Control Package. This software will support up to three display consoles on a time-sliced operating basis under which the consoles may share application and software support programs. The system provides several features described below. All of these features have been assumed in writing the programming specifications and all of them are considered to be necessary for harmonious multi-console operation. The features are as follows:

- A basic software operating system comprising system and application programs to lend homogeneity to the operation. This permits any application programs to be run from any one or several consoles simultaneously with any other program functioning at another console, and with a minimum duplication of programming effort and computer memory content.
- A basic package of programs that can be called from any console to perform certain universal operations, e.g. entering parameters, drawing lines, collecting groups, transforming or erasing graphic entities, storing and retrieving displayed graphics, etc.
- A standard model structure which will provide a link between the hierarchy represented by the world as shown on the display window and the universe as defined within the computer. On the CDC system, the model defines a "universe" eight-million addressable units square while at the window of the display (only

four thousand units square) the viewer may examine any part of this world with a useful range of display magnification of 1 to 32,768. For example, working at an accuracy of  $1 \times 10^{-4}$  in. per unit, the universe may contain a drawing 70 ft long by 70 ft high.

- A package of programs which can be called by an application to perform manipulations of the model and of the display such as: entering a graphic entity or other data into the model, displaying a light button, or copying and moving a template group.

The basic specification decisions influenced by these assumptions are as follows: a. What is the operating environment of the application program, and how is it called? b. Which standard console operations may the application program take advantage of? c. How should the problem be structured by the program to best use the model provided? d. Which operations are easy to perform on this model and which are difficult?

#### 7.4 DISPLAY OPERATIONS

The facilities discussed here are assumed to be available to the display console operator. The following paragraphs describe the functional features and areas of the console shown in Fig. 7-2:

- **Display Working Surface:** A section of the display designated for presentation of the window through which the window is viewed. It is on the window area that the lines or other graphic entities are displayed as the operator views and/or operates upon them.
- **Display Control Area:** A section of the display designated for presentation of general operator controls such as light buttons or registers for entry or display of parameters.
- **Light Pen:** The pen is used to select graphics, registers, or light buttons to perform specific functions. The pen is also used to select and move the tracking cross.

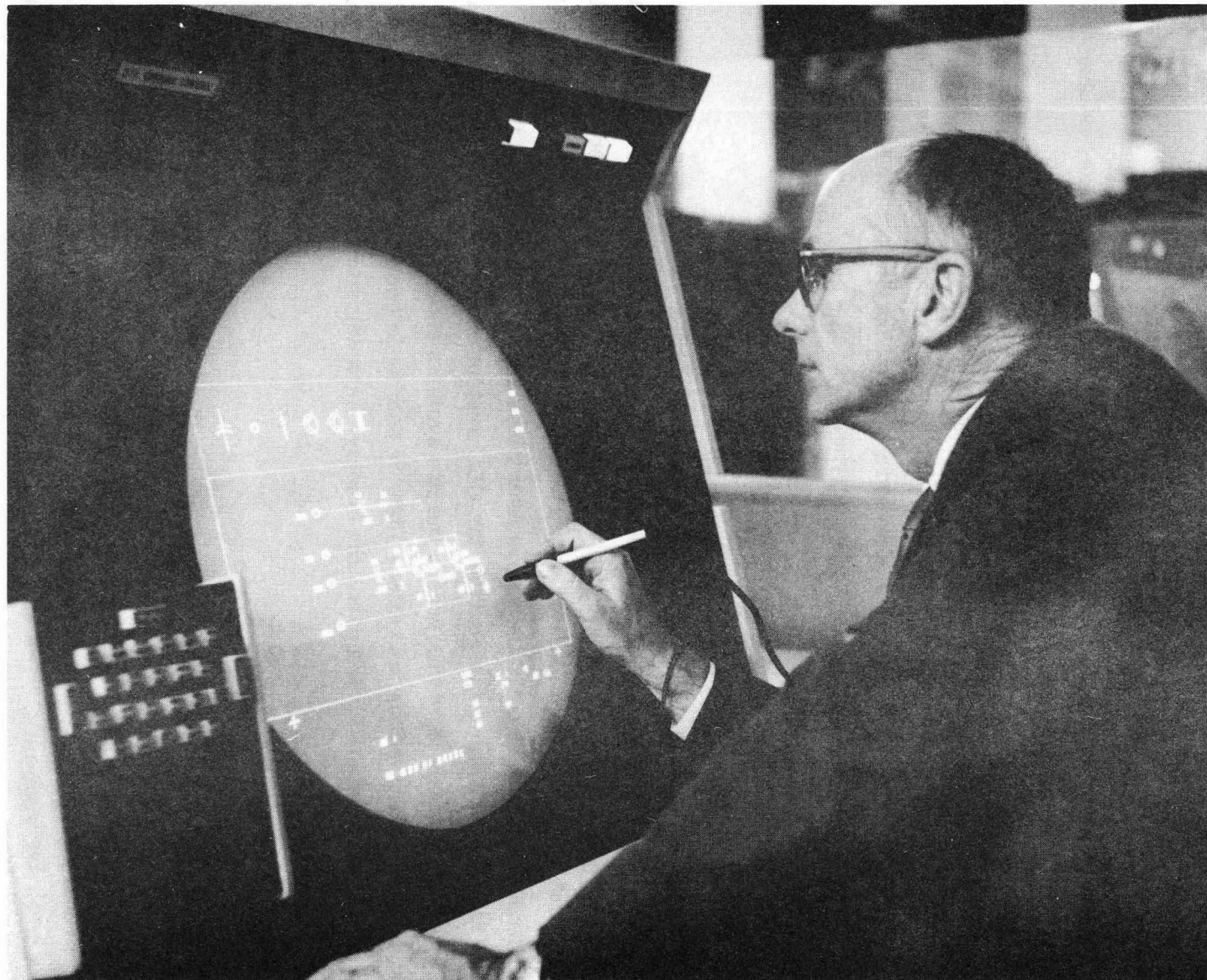


Fig. 7-2 Display Console With Permanent and Programmed Controls Shown

- **Tracking Cross:** A special graphic entity in the form of a cross which may be moved with the light pen and is used to designate an area of the working surface which is not illuminated and thus not selectable with the light pen.
- **Function Keyboard:** A hardware box of keys that the operator may press to indicate an operation to be performed. The definition of each key is changeable under software control.
- **Light Button:** A special graphic entity which is used like a function key. When selected by the light pen, an operation is called.
- **Alphanumeric Keyboard:** A keyboard for entering alphanumeric data. On the CDC system this consists of an array of light buttons. No difficulty is foreseen in substituting a hardware keyboard.

## 7.5 CONVERSION CONSIDERATIONS

The following discussion will give consideration to the differences between the CDC 270 system with the Functional Control Package software and the Univac Advanced Mariner Display. These differences will be evaluated in light of their influence on the programming specifications depicted in this document. It should be noted that a philosophical difference makes direct hardware comparison difficult; this difference is the trade between software display generation on the CDC system and hardware display generation on the Univac system. For this reason the two systems should each be evaluated as a hardware/software combination. Since little or nothing is known of the Univac software, this is difficult. Table 7-1 compares the display characteristics of the two systems.

### 7.5.1 Hardware

It is readily seen that hardware for both systems is comparable in size and speed. Any speed differences are considered insignificant because both systems can display a satisfactory quantity of data in regeneration cycle time. Size considerations, however, are more severe since many of the programming specifications make use of light buttons and registers. (The 12-in. -square display area does not allow much space



Table 7-1

## HARDWARE COMPARISON - CDC 273 VS. UNIVAC MARINER DISPLAY

Parameters	CDC 273	Univac
Basic Display Generating Scheme:	Hardware displays as commanded by drum memory. Display synchronized to drum revolution. Nonlinear analog display characteristics compensated for by software when commands are written. Regeneration fixed at 30 Hz.	Hardware display as commanded by core memory. Memory synchronized to display characteristics; next display instruction waits for completion of previous. Nonlinear display characteristics do not manifest themselves. Regeneration rate set by software in display controller.
Display Dimensions:	CDC 273 (with standard software)	Univac
Usable Display Area:		
Shape	20.5 in. diagonal round	12 × 12 in. square
Area	323 in. <sup>2</sup>	144 in. <sup>2</sup>
Display Working Surface:	11 × 17 in.	Full screen
Display Control Area	Circular sectors outside of the 11 × 17 in. construction area	Minimal <sup>(b)</sup>
Addressable Units	4,096 × 4,096	1,024 × 1,024
Inches Per Unit	0.005 in.	0.010 in.
Display Timing <sup>(a)</sup>		
Random Reset Positioning	25 μs full screen, 20 in.	15 μs full screen, 17 in.
Long Line (17 in.)	44.08 μs	32 μs
Short Line (0.5 in.)	4.04 μs	2 μs
Character (average)	10 μs	5 μs
Character Generation Characteristics:		
Character Size:		
Small	0.14 × 0.1 in.	0.14 × 0.095 in.
Large	Not programmed	0.28 × 0.190 in.
Characters Per Line:		
Small	146	80
Large	Not programmed	40
Character Rotation	Not programmed	90 deg
Generation Method	Software vectors, variable font	Hardware generator, fixed font
Console Controls for Operator Action:		
Function Keyboard (Program Definable)	24 keys	35 or 40 keys (Text says 40; figure shows 35)
Alphanumeric Keyboard	Software feature: Variable font displayed in lower control sector	Hardware keyboard fixed alphanumerics
Light Pen	Standard	Optional
Light Buttons	As desired in Control surface sector	Very little room for light buttons, but usable <sup>(b)</sup>

(a) The times shown for the CDC 273 are average times for the specified length and are dependent on how the software package chooses to generate the beam driving instructions to produce a smooth line.

(b) If graphic entities are displayed simultaneously with control functions, minimal usable area is available for control (light buttons and registers).

for other than graphic data presentation). The user of the Univac system would have to make greater use of the programmable function keys as a less satisfactory substitute for the light buttons. With respect to the alphanumeric keyboard and the hardware generation of characters, the Univac equipment would seem to have an advantage unless a variable type font is required. In summary, the hardware considerations cannot be fully assessed without a detailed hardware/software description, but it appears that there will be no major problems.

#### 7.5.2 Software

The software system assumed in these specifications for the various applications discussed in this report is considered to be near minimum level for operation of a display system for a diverse group of users operating simultaneously. Without such a basic, comprehensive software package, each application must provide all distinct operating procedures, console controls, model manipulations, etc.

The graphic programming language outlined in the report of Adams Associates to Univac would be a most satisfactory basis for such a comprehensive software system, however it is not all clear from this report or from the Univac documentation that such a language will actually be provided. It is a requirement that if a comprehensive package is not provided by Univac with the delivered hardware, and differing and distinct applications such as those outlined in this report are to share the graphic facility, then a basic package of software and manipulative routines must be specified and written to ensure homogeneity of all efforts. It is estimated that a software effort of this size will require eight to nine man-years of programming effort.

Appendix  
GLOSSARY

- AC ANALYSIS: The analysis of the behavior of a circuit which is subjected to steady, single-frequency, sinusoidally varying waveforms.
- ACCESS, RANDOM: Access to storage such that the next position from which information is obtained is not dependent on previous location.
- ACCESS TIME: The time interval between the instant at which the arithmetic unit requires information from storage or memory unit.
- ACTIVE ELEMENT: A source of current or voltage.
- ADDRESS: A label, name, or number.
- ALGORITHM: A fixed, step-by-step procedure for performing some data processing function, usually a simplified procedure for accomplishing a complex result.
- ALPHANUMERIC: A generic term for alphabetic letters, numerical digits, and some special symbols.
- AMPERE: A unit of electric current, abbreviated as amp.
- ANALOG COMPUTER: A computer which calculates by using physical analogs of the variables.
- APPLICATION PROGRAM: Existing analytical engineering program(s) or one to be developed.
- ARITHMETIC UNIT: The section of the hardware of a computer where arithmetic and logical operations are performed on information.
- BATCHING: Grouping a number of similar input jobs (tasks) together for processing, thereby using a single setup or program for a longer period in a single run.
- BIT: Contraction for binary digit.

**BYTE:** A set of consecutive binary digits, usually representing either a character code (BCD, BCC, etc.) or some easily manipulated set of digits or portion of a machine word; e.g., an eight-bit or six-bit byte.

**CHORDING:** The act of depressing more than one keyboard key at a time.

**CIRCUIT:** A communications link between two or more points.

**CODE:** A system of symbols for representing information in a computer and the rules for associating them.

**COMPILER:** A program-making routine which produces a specific program for a particular problem.

**COMPUTER:** A machine which is able to calculate or compute, i.e., one which will perform sequences of reasonable operations with information, mainly arithmetical and logical operations.

**COMPUTER WORD:** A 24-bit word (for discussion purposes).

**CONTROL SURFACE:** The area on the CRT display surface used for light buttons and light registers. That area on the CRT not reserved as working area.

**CONTROL UNIT:** That portion of the hardware of an automatic digital computer which directs the sequence of operations, interprets the coded instructions, and initiates the proper signals to the computer circuits to execute the instructions.

**CRT:** Cathode-ray tube.

**CURRENT:** Refers to an electric current which is the time rate of flow of electric charge across a surface.

**DATA BASE:** A data storage scheme.

**DATA ITEM:** One input variable. It can be an array of values or a single value.

**DC ANALYSIS:** The analysis of the behavior of a circuit which is subjected to steady, direct current voltage and/or current sources.

**DEPENDENT CURRENT SOURCE:** In ECAP, a transistor. A dependent current source is always associated with two branches; it is placed in one branch and is

controlled by the other. The assumed direction of positive current flow for the dependent current source must be the same as that of the branch in which it is placed.

**DESCRIPTOR:** The first three 24-bit words related to an entity stored in the Computer Graphic List. The descriptor contains the key, category, reference, and parental pointer fields for the specific entity stored therein. Every entity in the list is headed by a descriptor.

**DESIGN, SYSTEM:** A plan or outline for a system covering its aims and purposes, environment, functions, the phenomena to be processed, and the partitioning of the system into subsystems and components.

**DIGITAL COMPUTER:** A computer in which information is represented in discrete form and which calculates using numbers expressed in digits, with yes and no expressed in 1's and 0's to represent all the variables that occur in a problem.

**DIRECT CURRENT:** Unidirectional current as produced from batteries, from dynamo machinery equipped with commutators, or by means of rectifiers.

**DISPLAY:** Presentation of information in a form comprehensible by a human being, as a chart or graph, hard copy, cathode-ray tube pictures, letters or figures, meters, dials, etc.

**DISPLAY, CORE:** A method of graphic display using information stored in the computer core memory. Core display is synonymous with on-line display.

**DISPLAY SURFACE:** The 20-in.-diameter area on the CRT face utilized for man-machine communications.

**DRUM, MAGNETIC:** A rotating cylinder coated or impregnated with magnetic material upon which information is stored by polarized magnetic dots.

**ELECTRONIC:** Pertaining to the motion, emission, and behavior of currents of free electrons, especially in vacuum, gas, or photo tubes and special conductors or semiconductors. Contrast this with electric, the flow of currents in wires only.

**ELECTRONIC COMPONENT:** A device used for controlling current and/or voltage in an electronic circuit, e.g., a diode, tube, or transistor.

**ELEMENT, ELECTRICAL:** An electrical device which has only one electrical property e.g., an inductor has only the inductance property and no resistance.

**ELEMENT, GRAPHIC:** Copies of the templates displayed on the working surface as part of the circuit schematic.

**ENTITY:** The smallest addressable information unit within the Computer Graphic List. It is a variable-length block containing a range of data specified by the entity type. There are five classes of entities and within each class there are a discrete number of entity types. (See ENTITY CLASSES.)

**ENTITY, ALPHANUMERIC:** A record of alphanumeric data for placement anywhere on the display surface.

**ENTITY, APPLICATION:** A record for storing general purpose information.

**ENTITY, CHILD:** An entity which has been designated as a member of a logical unit called a group. Each child entity is related to the parent entity by its parental pointer field.

**ENTITY CLASSES:** There are five general classes of entities: (1) control surface, (2) alphanumeric, (3) graphic, (4) linkage, and (5) application. Each class consists of a discrete number of entity types.

**ENTITY, CONTROL SURFACE:** Within this class four types of entity are defined: (1) tracking cross, (2) frame, (3) light register, and (4) light button.

**ENTITY, GRAPHIC:** Within this class five types of entity are defined: (1) dot, (2) line, (3) circle, (4) circular arc, and (5) polystring.

**ENTITY, LINKAGE:** Within this class one type of entity is defined: group control.

**ENTITY, PARENT:** An entity consisting of a group of child entities.

**FIELD, CATEGORY:** The last six bits of the first 24-bit word of the descriptor for an entity stored in the Computer Graphic List. The category field has been established for use by the application programmer.

**FIELD, KEY:** The first 12 bits of the first 24-bit word of the descriptor for an entity stored in the Computer Graphic List. The key field defines the entity type.

**FIELD, PARENTAL POINTER:** The third 24-bit word of the descriptor for an entity stored in the Computer Graphic List. Zeros in this field indicate that the entity is not currently grouped. When the entity is grouped, the field contains the list address of its parent entity.

**FIELD, REFERENCE:** The second 24-bit word related to an entity stored in the Computer Graphic List. This field may only be used by the application program.

**FIXED CURRENT SOURCE:** A current source that has no provision for varying its output.

**FIXED POINT ARITHMETIC:** An arithmetic notation or representation in which all numerical quantities are expressed to the specified number of digits, with the point implicitly located at the same specified position.

**FIXED VOLTAGE SOURCE:** A voltage source that has no provision for varying its output.

**FLOATING POINT ARITHMETIC:** An arithmetic notation taking into account varying locations of the base point, writing each number by specifying its sign, its coefficient, and its exponent affecting the base.

**FLOW DIAGRAM:** A graphical representation of a sequence of operations.

**FRAME:** Usually a rectangular display on the CRT display surface which encloses the working surface; however, the frame dimensions can be changed by FCP common parameters.

**FREQUENCY:** The rate at which a current alternates, usually measured in cycles per second.

**GRAPHIC APPLICATION PROGRAM:** A computer graphic program written for an application program.

**GRID, CONSTRUCTION:** An area consisting of 8,388,607 addressable points in the X and Y planes. This is the area upon which the user constructs his model. The construction grid can be scaled by the user for his particular application.

**GRID, DISPLAY:** An area consisting of 4,096 addressable points in the X and Y planes. The display grid circumscribes the display surface so that any combination of points in the X and Y planes can be given on the display surface.

**HEURISTIC:** An intuitive trial and error method of attacking a problem (as opposed to the algorithmic or set procedure method).

**HIERARCHY:** A specified rank or order of items, personnel orders, nested sub-routines, etc.

**IMPEDANCE:** The term expresses the relation between a sinusoidally varying quantity (such as force, pressure, voltage, electric field strength, temperature) and a second quantity (such as velocity, current, magnetic field strength, or heat flow) which is a measure of the response of a physical system to the first.

**INDEX, ZOOM:** Binary logarithmic ratio of display grid units to construction grid units provided for use by the operator to magnify the display on the working surface. The magnification can be increased or decreased in integral increments of powers of two.

**INPUT:** Information transferred from outside the computer, including secondary or external storage, into the internal storage of the computer.

**INSTRUCTION:** A machine word or a set of characters in machine language which specifies that the computer take a certain action.

**INTEGER:** A whole number; not fractional or mixed.

**INTERFACE:** A common boundary between systems or parts of a single system.

**I/O EQUIPMENT:** The set of devices used to present information to and receive information from a computer, e.g., card readers and punches, printers and typewriters, paper tape readers and punches, magnetic tapes, data channels, CRT and console displays, mechanical switches, and magnetic drums.

**JOB:** An operation or series of tasks to be performed on a computer or in a computing facility.



**KEYBOARD:** Consists of 25 manually operated keys and is part of the Computer Graphics Console. The functions of the keys are defined by the software but can be modified or completely changed by the user's application program.

**LANGUAGE:** A system consisting of a carefully defined set of characters, rules for combining them into larger units (words or expressions), and specifically assigned meanings, used for representing and communicating information or data among a group of people, machines, etc.

**LIGHT BUTTON:** Software-defined functions displayed on the control surface. There are primary and secondary light buttons; a primary may have up to 16 secondaries associated with it.

**LIGHT PEN:** A pencil-like bundle of fiber-optics and associated hardware. It is used by the operator to interact with data on the display surface. It does not write on the display surface; rather, it senses the current X and Y display grid coordinates of the light beam and, under appropriate program control, the display can be updated with a light-trace of the light pen's path.

**LIGHT REGISTER:** There are several FCP-implemented light registers displayed on the control surface. They are provided for operator input of data and for output of program-generated values or data.

**LIST, COMPUTER GRAPHIC:** A data base containing the parameters which define the entities created by and related to the Computer Graphic System. It is kept in book form.

**LOGIC:** The science that deals with the canons and criteria of validity in thought and demonstration of fact; the formal principles of reasoning.

**MACHINE LANGUAGE:** Information in the physical form which a computer can handle.

**MEMORY:** Any device into which information can be introduced and then extracted at a later time.

**MODULATION:** Process by which certain characteristics of a wave are modified in accordance with a characteristic of another wave or signal.

**OPERATION, REAL TIME:** The processing of data in concert or concurrent with a physical process so that the results of the calculation are useful in the physical operation; continuous data processing in synchrony with an attendant upon another operation.

**OPERATOR:** The person operating the graphic console. Same as USER.

**OPTIMIZE:** To arrange phenomena to achieve a maximum operation or effect with regard to some variable.

**OUTPUT:** Information transferred from the internal storage of a computer to secondary or external storage or to any device outside the computer.

**PAGE:** Sequence of N consecutive computer words where N is arbitrary but fixed for a particular application.

**PICKING:** The FCP function which identifies specific graphics or points on the graphics through interpretation of light seen by the light pen.

**PICK TABLE:** A table in which the item pointed to on the screen is placed for program interpretation and execution.

**PLUGBOARD:** A removable board holding many electric terminals.

**POINTER:** Mechanism for obtaining an address; usually the address itself.

**POINT WRITING:** A dot displayed in the upper left quadrant of the tracking cross. Its purpose, as the location of the tracking cross, is to enable light pen definition of specific points such as the center point of a circle or the starting and end points of a line.

**PRINTOUT, DYNAMIC:** A dump or printout taken during the course of processing.

**PRINTOUT, HARD COPY:** The printing of information onto paper from another storage medium or from a processor.

**PROGRAM:** A precise sequence of coded instructions enabling a digital computer to solve a problem.

**PROGRAM, UTILITY:** A program necessary to support the development and maintenance of other programs or program systems, or to support a computing and programming facility.

**PROGRAMMING:** The act of planning, flowing, coding, debugging and testing a program.

**PULSE:** In general, a sharp difference, usually over a relatively short period of time, between normal level of some physical variable corresponding to the average level of a wave or waves and a high or low level of that physical variable corresponding to the crest or trough of the wave; often a sharp voltage change.

**RANDOM ACCESS:** Access to the memory or storage under conditions where the next register from which information is to be obtained is chosen at random – in other words, it does not depend upon the location of the previous register.

**SCANNING, OPTICAL:** Reading information from cards or papers via a light-sensitive device.

**SCISSOR:** The act of dropping out an entity from the display when, as a result of demagnification due to the zoom index, its coordinate parameters exceed the range of the display grid. This is a software function.

**SIGN-ON:** The last part of the FCP initialization procedure which initiates Computer Graphic System operation by calling the SIGNON overlay, transferring control to a display system executive routine, and enabling all keyboard and light button controls. The frame, light buttons, and light registers are then displayed.

**STRING:** A sequence of groups of consecutive computer words in which one word in the group is a pointer to the next group of words. The string is terminated with a null pointer.

**SURFACE:** The area reserved for light buttons and light registers. The area on the CRT display surface exclusive of the working surface.

**SURFACE, DISPLAY:** A 20-in.-diameter area on the CRT screen utilized for man-machine communications. A light blue, field-free display is presented to the operator due to components of the P7 phosphor coating deposited on the inside surface of the CRT screen.

**SURFACE, WORKING:** One of two divisions made on the CRT display surface. The working surface is enclosed by a frame (viewing window) which is a displayed graphic.

**TABLE, PICK:** Used to store references to specific parameters selected for subsequent graphic construction or processing by either FCP or application programs.

**TEMPLATES:** Graphic models of components. A circuit diagram.

**THESAURUS:** A collection of words or other stored items arranged by concept rather than by alphabetic order, rank, or sequence.

**TIME SHARING:** The use of a computer or other device for two or more purposes during the same overall time interval through multiprocessing, interleaving, alteration of tasks, etc.

**TRACKING:** The FCP function which maintains cognizance of the position of the light pen as it moves across the display surface. A core-displayed tracking cross is used as the light source for the light pen.

**TRACKING CROSS:** There are two tracking crosses displayed, one at the lower left and one at the lower right of the display surface and just outside the working surface. They are an integral part of the frame and serve as a light source for the light pen while tracking.

**UNIT, CONSTRUCTION GRID:** The spacing between the 8, 388, 607 points on the X and Y axes of the construction grid.

**USER:** The person operating the graphic console. Same as OPERATOR.

**VOLT:** A unit of electrical potential difference, abbreviated V or v.

**WINDOW CONCEPT:** Method used whereby selected portions of a picture can be displayed on the CRT.

**WORD:** A set of characters which occupies one storage location and is treated and transported as a unit by the computer circuitry. Words may be fixed or variable in length depending on the construction of the particular computer.

**WORD SIZE:** The number of bits or characters in a computer register.