1-1-1982

Design of an interactive system for real time simulation of tool motion of numerically controlled milling machines.

Sunil Kumar Bhalla

Recommended Citation

Bhalla, Sunil Kumar, "Design of an interactive system for real time simulation of tool motion of numerically controlled milling machines." (1982). *Theses and Dissertations.* Paper 2453.

DESIGN OF AN INTERACTIVE SYSTEM FOR REAL TIME SIMULATION OF TOOL MOTION OF NUMERICALLY CONTROLLED MILLING MACHINES

by

SUNIL KUMAR BHALLA

A Thesis
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science

in
Mechanical Engineering

Lehigh University
1982

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

CERTIFICATE OF APPROVAL

This Thesis is accepted and approved
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Mechanical Engineering

Sept 2, 1982
(date)

Professor in Charge

Chairman of Department

ACKNOWLEDGMENTS

I would like to express my sincere thanks to Dr. John B. Ochs for his constructive suggestions, encouragement and support throughout the course of this study. I must also express special appreciation for the assistance and guidance provided by Dr. Tulga Ozsoy. His technical expertise in the field of computer graphics provided me invaluable assistance and allowed this work to be brought to a successful conclusion.

I am indebted to my parents, Mr. and Mrs. R.S. Bhalla, for the unwavering motivation they provided me, and to my wife, Bindiya, whose direct personal sacrifices and continuing encouragement throughout my Master's program, made this work a reality.

Table of Contents

List of Figures

ABSTRACT

Direct view storage tube technology has long been the standard display hardware for CAD/CAM systems. With the improvement in man/machine interaction provided by raster-scan displays, applications are now being developed to exploit raster-scan technology's capabilities. A system has been designed for real-time interactive display of tool-motion of numerically-controlled milling machines. The mill simulation of Unigraphics CAD/CAM system that employs a storage tube display, has been transferred to a raster-scan type display, which has the capabilities of real-time dynamic graphics, 16-color representation, and multi-channel operation.

The factors which have been considered important in the system design are: system interfacing, display format and user interaction. The data is stored in a neutral format, in unformatted files, from which it is directly-accessed for display. The screen of the raster-scan display terminals has been divided into a message area, command area and picture display area, thus, using it both for display and command input. User interaction is achieved through dynamic, hierarchical menus.

Machining operations performed on Unigraphics can now be transferred and displayed on the VS11 terminals, making use of all the advantages, such as, zooming,

translation, rotation, scaling, and full-color capability. The displayed picture is not obscured by the lines traced by the tool and it is possible to detect visually the interferences between the tool, part, clamps and fixtures. The designed part can be oriented in any direction and the machining operation is simulated with a display that has improved user control and in general has been found to be "user-friendly".

CHAPTER 1

INTRODUCTION

1.1 Background

One of the major developments in computer technology which has grown for aerospace applications and is now spreading to many other industrial applications is computer-aided design and computer-aided manufacturing (CAD/CAM) and the more general computer-aided engineering (CAE). CAD/CAM can be defined most simply as the use of computer hardware and application software designed to translate a product's specific requirements into the final product. It plays a key role in areas such as design, analysis, drafting, documentation, N/C programming, tooling, fabrication, assembly, quality control and testing.

CAD/CAM as a concept emerged from the development of "electronic sketching" -- interactive computer graphics. Computer graphics is a topic of rapidly growing importance in the computer field [10]. It has always been one of the most visually spectacular branches of computer technology, producing images whose appearance and motion make them quite unlike any other form of computer output. One of the uses of computer graphics has been to improve the understanding of what might typically be presented in tabular form. It is an extremely effective method of communication between man and computer; for instance, a graph from a table of numbers enables relationships to be better understood, the trends identified and anomalies highlighted. The clarity of the image that can be displayed depends largely on the computer hardware and software used to generate it.

The last decade has witnessed numerous changes and enhancements in the hardware and software used for computer applications. The cost of computer hardware, as well as its size, has been decreasing. Application software is being produced that is designed with the end-user in mind, and not the computer scientist. The low cost of "user-friendly" systems has spurred the growing applications of computers and computer graphics.

The reduced chip sizes have led to design of better display devices by incorporating microprocessors into the graphics terminals itself. These terminals provide dynamic, real-time multi-color display versus single-color static displays of the part, at a competitive cost. A system for computer-aided design, analysis, drafting or manufacturing consists of one (or more) input devices, graphic display, computer, mass storage and output devices. Most of these systems are stand-alone types, providing such features as designing of mechanical parts or electrical circuits, finite element analysis, drafting and dimensioning, and manufacturing functions.

Some of the main vendors of CAD/CAM systems are: Applicon, Inc., Autotrol, Inc., Calma Co., Computervision Corpn., IBM Corpn., Intergraph Corpn., McDonnell Douglas Automation Co., Manufacturing Data Systems, Inc., and others. Each of these companies espouse a philosophy of an integration of the process from conceptual design through manufacturing with information accessed, generated and shared by what in the past were separate disciplines.

1.2 Current Approach to CAD/CAM Integration

As CAD/CAM is currently implemented in some industries today, the computer-aided design and computer-aided manufacturing functions are planned and implemented by management as separate systems. Nevertheless, as the separate systems become more numerous, users are becoming aware of growing inefficiencies: it is troublesome to learn the peculiarities of different systems; it is costly to store often overlapping data for each separate system; it is a source of error to extract data from one system and enter it manually into the next one. The need for a broader focus has been stressed by Myers [9].

The block diagram of Figure 1-1 shows the different functions of a CAD/CAM system. Each of these is described below:

Design: Starting from a rough sketch or possibly just an idea, the designer uses the graphics display to

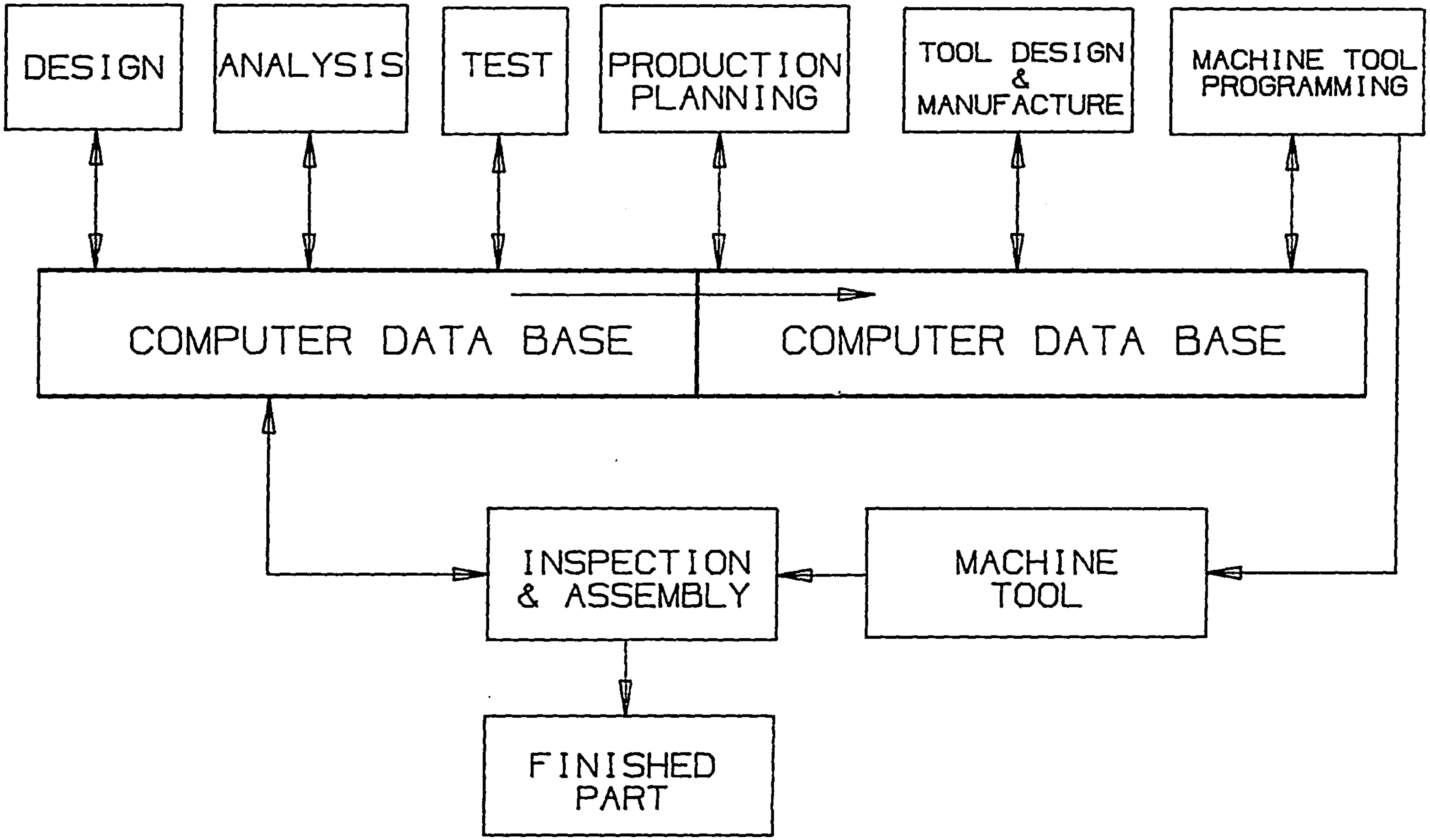


Figure 1.1: Functions of a CAD/CAM System

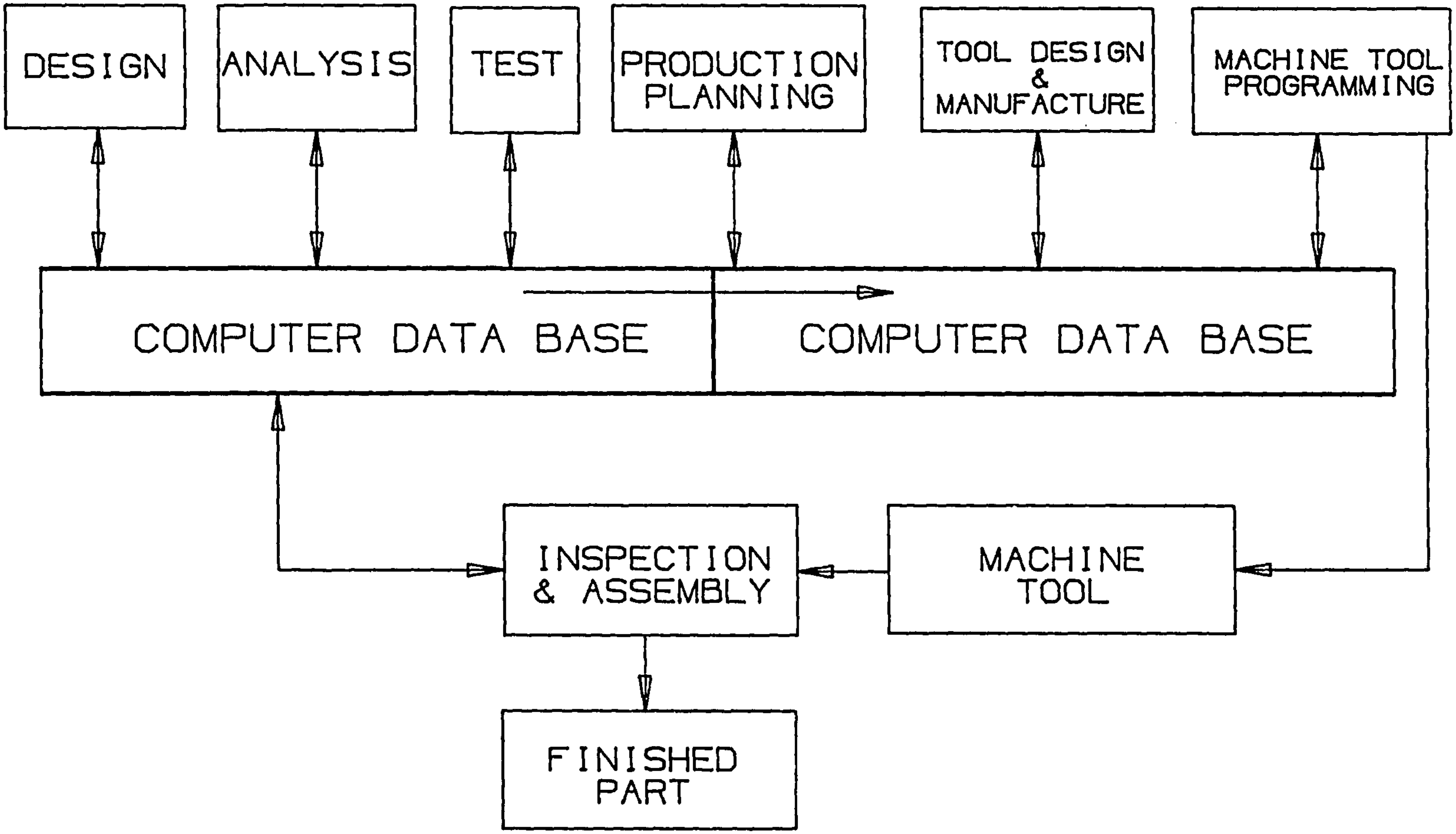


Figure 1.1: Functions of a CAD/CAM System

construct a 3-Dimensional model of a design. CAD is at its best in designing families of parts. These parts can be designed so that the parameters are selected by the designer as input to a process (parametric design). In this way the design of the entire family of parts can be completed in little more time than is required to produce the design for one part, with the CAD system checking to see if any design rules are violated. Some typical design applications are: the design of structures, machine parts, castings, sheet metal parts, electrical systems mechanical linkages, and many others.

Analysis: After completion of geometric definition, analysis functions are invoked to determine various properties. The results of the analysis often require changes in the design geometry, and a fresh analysis of the new design. This iterative process is performed at the CAD workstation, without the traditional shuttling of the part design between the engineering analysis and design groups. A complete system, such as a four bar linkage, can be first modelled and analysed. This may be broken up into individual components which can then be designed and analysed. The commonly available design analysis functions are: mass properties calculations, finite element analysis, dynamic and kinematic analysis, tolerances, stacking, static clearances, etc.

Review or Test: Most companies have model shops, where actual prototypes are built, tested, and more often than not, sent back to be redesigned. Now, 3-dimensional moving computer models can be built, with only a final prototype being constructed and tested. Computer-aided testing has been used to control the simulation of working conditions of the constructed physical model. Computer data taking and networking of systems has allowed fast and reliable feedback to the engineer, to quantify the computer modelling and analysis. With split screen terminals the designer and engineer can perform A-B comparisons between his final computer model and the physically constructed and tested model.

Tool Design and Manufacture: A tool engineer retrieves the part from the data base and designs tooling for the part. The CAD system generally enables the designer to rotate the part on any axes and measure from any datum. This makes it simple to measure dimensions which are necessary to design tooling and extremely difficult to obtain by ordinary methods. Typical applications are the design of holding fixtures, dies, templates, etc.

Machine Tool Programming: In producing a numerically-controlled (N/C) part program, the accurate geometric definition of the part that is present in the data base is employed by the part programmer. The definition of the cutter path is then normally specified interactively, rather than by the traditional APT process of mentally envisioning the cutter path. The machine tool programming can often be done through a graphic presentation of the machining operation. Typically N/C programming can be done for 2, 2&1/2, 3, 4, and 5 axis machining on machines including milling machines, lathes, drilling presses, etc.

Inspection: Inspection of parts with complex shapes can be aided through the use of original CAD data. The use of various cutting planes allows the part to be measured and inspected automatically since the CAD/CAM 3-dimensional data base allows generation of any view of the part design as specified during the inspection process. This allows rapid decisions to be made as to whether the part is within specifications.

Computer Data Base: Last, but not the least, is the importance of a common data base that forms the thread to link all these functions. Once the designer is satisfied that the part design has been sufficiently optimized, the model is entered into the data base. The data base contains all the information pertaining to the part. This is usually a 3-dimensional geometric definition which can be accessed from any of the above functions.

With the varieties of systems currently available, it is possible to concieve the complete sequence of functions from engineering design through manufacturing as an integrated system. Once the different functions in a system are integrated, it is desired that the systems designed by different manufacturers should also communicate. An early step toward that end would be for one system to be able to translate its particular data format into a neutral or general format, which another system could then interpret and use. Next, a series of steps can be envisioned that would integrate one system with a closely related one, as groups of systems are brought together and as the division between CAD and CAM is breached. As different systems are connected together, the value of having a common data base from which systems serving various functions could draw would become evident. This, then, is integration -- connected systems served by a common data base.

1.3 Problem Statement

Within an integrated CAD/CAM environment, the work described in this thesis has been designed to improve the numerically-controlled, (N/C), mill simulation by incorporating the real time dynamic motion and selective erasure capabilities of new raster technology. Here, a common data base has been designed which is shared by two systems and is used to display the same data on different display devices. The systems in question are Unigraphics[1] CAD/CAM system and the VS11[2] graphics systems.

---

[1] Unigraphics is the trademark of McAUTO

[2] VS11 is the trademark of Digital Equipment Corporation

Unigraphics is a stand-alone system with a graphics package having capabilities for design, drafting, analysis and manufacturing. The system uses a prompted menu user-interface, and is based, like many current CAD/CAM systems, on the work done by Hanrathy.

The display device for this system is the Direct View Storage Tube (DVST) and input devices are keyboard, thumb-wheels, and pen and tablet. In storage tube type of terminals, an electron beam 'paints' a static image on the screen by 'stroking' vectors on the screen's phosphor surface. Resolution is very good in these terminals, typically providing an addressable matrix of 4096*4096 points on a 19" diagonal screen. These tubes will electronically maintain a picture of any density that is created on the screen. However, no selective erasure is possible, pictures displayed are of a single color and line intensity, and limited dynamic motion can be performed.

The VS11 is a system which can be used both for applications programming and graphics display. Graphics on this system is mainly run through the VS113D Graphics Package [11], developed inhouse and explained in detail in the following chapter. The display device is raster-scan terminals with 16-color capability. The basic components of the system are: a high speed display processor, image memory modules and sync generator/cursor controls. The system features multi-channel operation providing color dynamic motion capability. The input device in this case is the rate-type joystick which controls the cursor and also has an interrupt switch.

Raster-scan terminals operate by sending an electron beam across the screen, left to right and top to bottom at a continuous sweep rate. An image is created by the system on activating a light or dark dot at any given spot on the screen by turning the beam on or off as the beam passes it. Resolution is usually not as good with a raster terminal as it is with a storage tube, typically with an addressable matrix of 512*512 points on a 21" diagonal screen. However, these display devices usually have multi-color capability, selective erasure is possible and these can run real-time dynamic graphics.

Once a part has been designed on the Unigraphics CAD/CAM system, the user can enter the numerical-control (N/C) machining sub-function and graphically simulate the machining using either the lathe or the mill module. Along with machining of the part, the system also creates a Cutter-Location (CL) Source File. This file contains tool motion statements, postprocessor statements and auxiliary statements to control tool motion, coolant flow, spindle speed, etc. Before sending the CL-Source File to the postprocessor, the machined part is verified for correctness of the tool path; this is first done visually on the display terminal and then sent to the plotter for final verification. Since the Unigraphics terminals are of the storage tube type, visual verification is particularly difficult. All entities, as well as the tool path are displayed in a single color. Moreover, as the tool moves over the part it draws lines along its path. As a result, by the time the tool finishes the machining simulation there are innumerable lines on the screen, which makes the distinction of the part an impossibility in most cases. Examples of these cases are shown in Chapter 2.

Additionally, the tool cannot be stopped during the simulation of the machining operation to check its location. The designed system eliminates most of these drawbacks. Although the software discussed herein has been written specifically for the above system, it can be adapted to other CAD/CAM systems and other raster display devices.

1.4 Approach to Problem

The system designed and discussed in this thesis is capable of transferring part geometry and tool-paths from the Unigraphics workstations to the the VS11 terminals, by working through a neutral data base. The parts may consist of entities such as points, lines, arcs, splines, or conics. The transfer of surfaces from Unigraphics to VS11 is in the development stages. However, once the edge curves of a surface are transferred, a surface can be created on the VS11 terminals by using the VS113D

Graphics Package [11].

The tool-paths are those that are generated under any option of the Mill Module in the Unigraphics package. These include functions such as drilling, profiling, and pocketing performed on 2, 2&1/2 or 3 to 5 axis milling machines. Once transferred to the VS11, the parts can be translated rotated, zoomed or oriented in any direction and the tool path displayed. Also displayed are the location-coordinates of the tool and its position in the Cutter-Location Source File at any instant during the milling operaton; which provides the user with the capability to edit the tool-path. The user has individual control over the color in which each entity and the tool is displayed. Options are provided to switch-off any entity, provide a tool- trace, tool-erase and control the speed of the tool during the milling operation.

In designing the system, special attention was paid to the following factors:

1) System Interfacing,

2) Display Structure, and

3) User Interaction

Liewald and Kennicott [7] have discussed the need to interface applications and share data among different systems. Interfaces depend on a standard for digital representation of a product; they are a first step toward the real goal, integration. Numerous standard bodies [6] are attempting to provide flexible, yet stable methods, with which to represent products. However, because in the case discussed herein, the operating system for both the systems is the same, and data is to be exchanged between two graphics software packages, a neutral format has been designed through which the systems communicate. Before considering the exchange of CAD/CAM data, it is first necessary to establish the contents of the data [6]. Three fundamental components of data -- format,representation and meaning -- are described below in order to identify the issues that have to be resolved by the designed system.

Format: The huge number of software/hardware systems is matched by an equal diversity in representation for integer, floating-point, and character definitions. This variety of methods manifests itself in the definition of numeric and character-set standards and in the establishment of the basic accuracy of the model. Additional difficulty in sharing or transferring CAD/CAM data is due to the inability to determine how good the data is. In processing the data downstream from the generating application, one must be aware of the basic computing accuracy achievable on the generating hardware or the meaningful accuracy on the display device, and the accuracy sought by the user. Moreover, not all the information generated by one system is required by another, espicially in our case. The unnecessary data has to be picked out and rejected.

Representation: There are several distinct methods with which to represent the geometry of a part. One is the "edge boundary" or wireframe, in which the extremes of the part are represented by a collection of space curves. Another is "surface representation", which allows a more precise definition of geometry, particularly for location not on an edge boundary. This can be difficult to display. The method of representation on the system discussed here is the commonly used hybrid implementation, the "edge-surface representation". In this method, the principal geometry is defined as a wireframe; surfaces can be developed between the edges to provide additional detail. Since representation for both systems is the same no additional difficulties in interfacing are encountered.

Meaning: The meaning component expresses the design intent of the data by associating the object representation with the real- world product. That is, in a part composed of entities, such as, lines, arcs and points, the method of grouping together of these entities to give the desired shape to the real-world part is what gives the data its meaning. Thus, meaning addresses both individual data elements and collection of elements.

The above-mentioned complexities in the data transfer are taken care of by the application programs explained in Chapter 3.

The inherent differences between the two systems to be interfaced necessitate the design of a different display structure or format. While display on the Unigraphics system is through a display monitor (the DVST) and a separate message monitor, that on the VS11 has to be achieved through the display screen only. This can be accomplished if the display monitor is also used for messages. The screen is divided into three parts: the areas for the display of part, the user-interaction portion and message area for tool location. The parts remain fixed to the areas assigned on the screen. However, the screen can be cleared of the menu-area, if it interferes with the displayed part, by issuing a simple command. Such a display structure has also been suggested by Gingerich, et. al. [1] and Price [13].

Perhaps no single component of an interactive program is more unpredictable in performance than the user interface, that is, that part of the program which determines how the user and the computer communicate. The user interface is most commonly divided into four components [10]. The most important of these is the "user's model", the conceptual model formed by the user, of the information he manipulates, and of the processes he applies to this information. The model enables him to develop, even with little or no knowledge of computer technology, a broad understanding of what the computer is doing. With the model's help he can anticipate the effect of his actions and devise his own strategies for operating the system.

Once the user has understood the model he needs "commands" with which to manipulate it. The systems of commands provided can be called the 'command language' and forms the second component of the user interface. Ideally the computer commands should have natural command languages, so that the user is not conscious of learning a new language. Thus, the designed system employs menu-selection to form an interactive dialog between the user and system in which a list or menu of valid commands is displayed and selected. A form of online documentation, menus free the user from remembering keywords and syntax. Menus may be static or dynamic. Dynamic menus display a much smaller command set at any one instant, but change frequently as interactive dialog progresses.

Dynamic menus may also be hierarchical -- and are so in our designed system. In these, the initial menu may display a system's major components. Each of the commands in the main menu may lead to another sub-menu, giving the user more options to choose from. For example, in the designed system, two of the six choices in the main menu are 'option' and 'draw'. If the user selects 'option', the succeeding menu will show the available options, that is, color,tool parameters etc. If 'draw' is selected, the menu will display the things which can be drawn -- part or tool-path. This dynamic, hierarchical menu was preferred to static or single menus because: 1) certain entered commands can be used to predict the following commands and 2) a small, changing menu is preferable to a large menu that includes the entire command set.

The third component of the user interface is "feedback", with which the computer assists the user in operating the program. Feedback comes in many forms: acknowledgement of receipt of commands, explanatory messages, or indication of selected objects. An expanded model of the interactive process showing the feedback paths is shown in Figure 1-2 [10]. The designed system prompts the user if the command entered is not contained in the menu, asks for colors required for different entities, and so on. Feedback helps the user to be sure that his commands are accurately received and fully understood by the program. It tells him little about their real effect. A fourth component, "information display", is therefore necessary to show him the state of the information he is manipulating. Here we are concerned with organising the displayed image to convey the information as effectively as possible. The image is a confirmation to the user that the model is correct, and it is therefore designed strictly in accordance with the model that has been chosen.

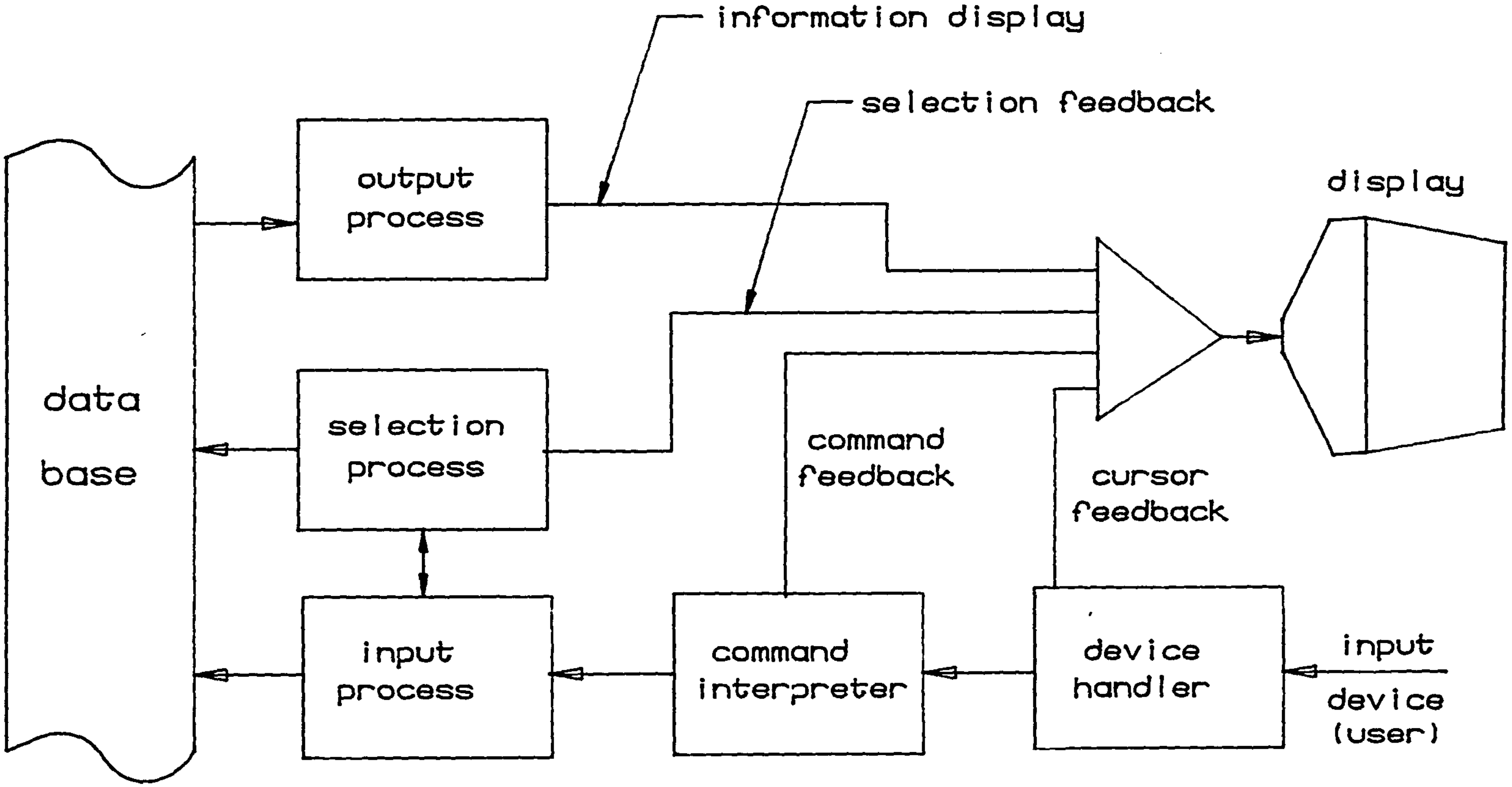


Figure 1.2: Feedback in Interactive Systems [10]

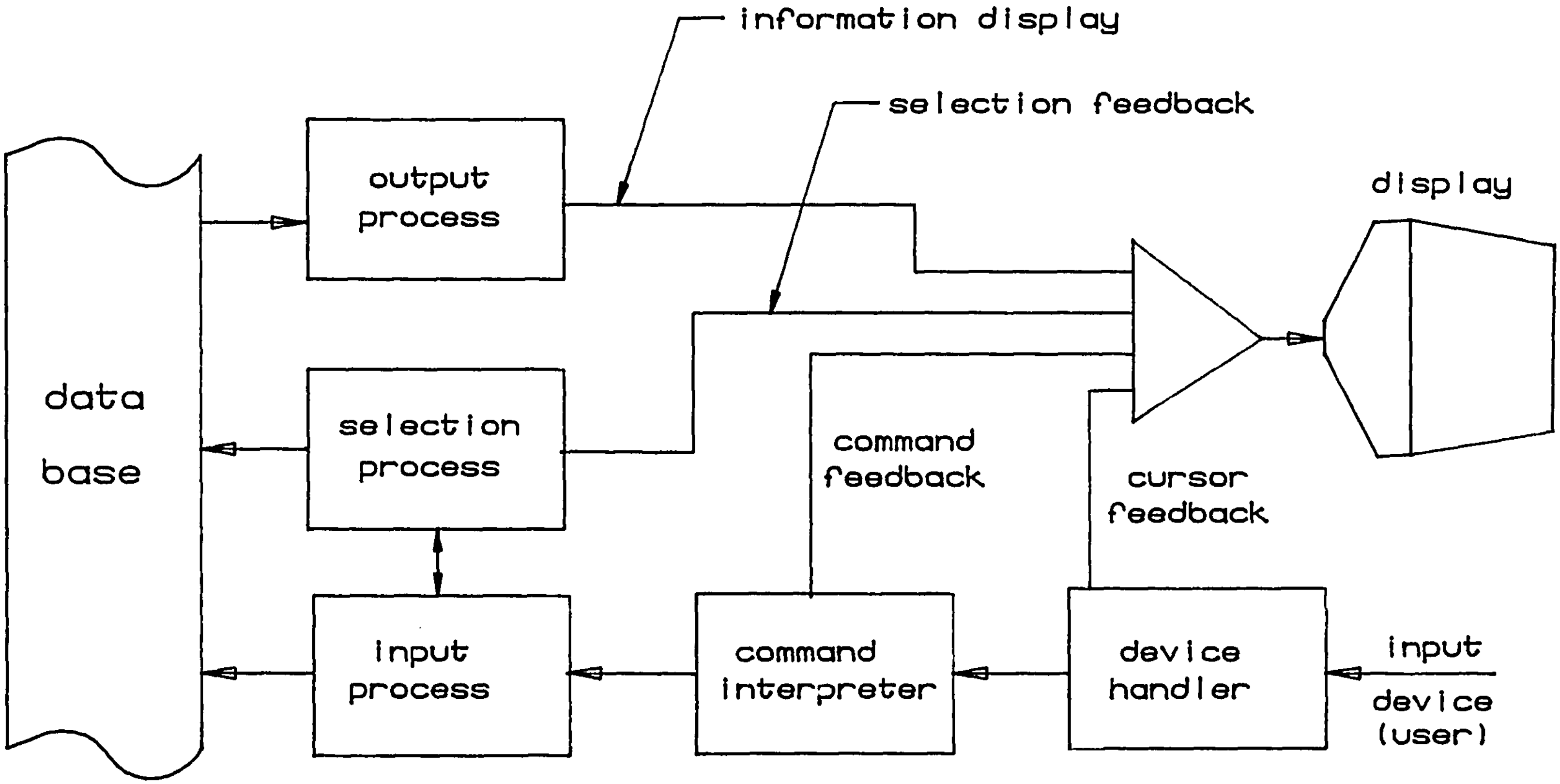


Figure 1.2: Feedback in Interactive Systems [10]

1.5 Organisation of Thesis

To provide the reader with a thorough understanding of the existing and the designed systems, and the effects of the new design, this thesis has been divided into four parts. The next chapter gives details of the existing system, explains the hardware and software available for both systems and draws a model of the current system. Chapter 3 explains the designed system configuration, with a detailed discussion on how data is transferred and subsequently displayed. Chapters 4 and 5 provide a case study and conclusions. The case study shows the user-interface and the final advantages achieved by using the VS11 Graphics system.

CHAPTER 2

EXISTING SYSTEM MODEL

For the purposes of this study, the first step has been the development of a model of the existing system. This model allows one to envision the operation of the entire system easily, and thus, point out areas that can be improved. The first section in this chapter discusses the hardware and software of the Unigraphics CAD/CAM system. Also described is the present method of approach of simulation of numerically-controlled (N/C) machining of a part. The second section discusses the advantages and disadvantages of this system, and the need to improve its capabilities. The last two sections detail the capabilities of the color raster display, explain the hardware and software that form the basis of this work, and discuss the advantages of real-time dynamic display of numerically-controlled (N/C) milling machine tool motion.

2.1 System Characteristics

In this section, the Unigraphics CAD/CAM hardware and software are described, along with the process used to create and graphically display the tool motion for N/C machining. As stated previously, the Unigraphics CAD/CAM system should be treated as a generic system, with capabilities that are typical of current systems.

2.1.1 Hardware

The current Unigraphics system runs on a host computer, the VAX 11/780[3], with 32-bit addressing and a virtual memory operating system, to provide essentially unlimited program space. The system's data base is maintained on magnetic disks along with the system software. A magnetic tape drive is available for back-up operations. As shown in Figure 2-1 the host computer supports six Unigraphics work-stations of the direct view storage tube type (Tektronix 4014's)[4] and another six work-stations of the raster scan type (VS11 terminals). All twelve terminals can be used for both graphics as well as applications programming. Also supported currently are six VT100[5] terminals, used mainly for program development.

The VAX based system supports general engineering applications as well as specific applications packages, such as Unigraphics mechanical CAD/CAM software. Unigraphics is a stand-alone system having capabilities for design, drafting, analysis, and manufacturing. Currently, the VAX system supports six design stations

---

[3] "VAX" is a trademark of Digital Equipment Corporation

[4] "Tektronix" is a trademark of Tektronix, Inc.

[5] "VT100" is a trademark of Digital Equipment Corporation

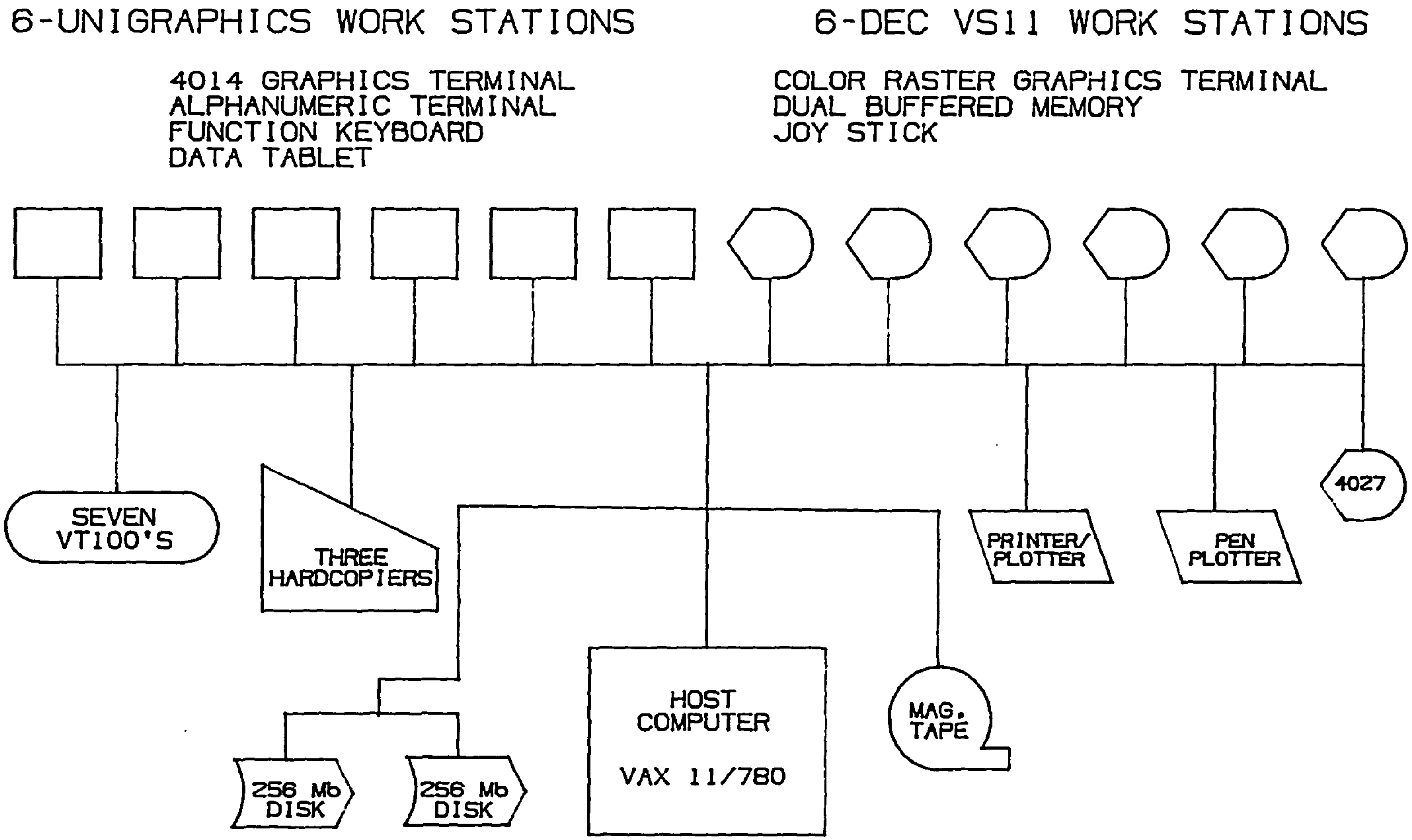


Figure 2.1: Hardware Available

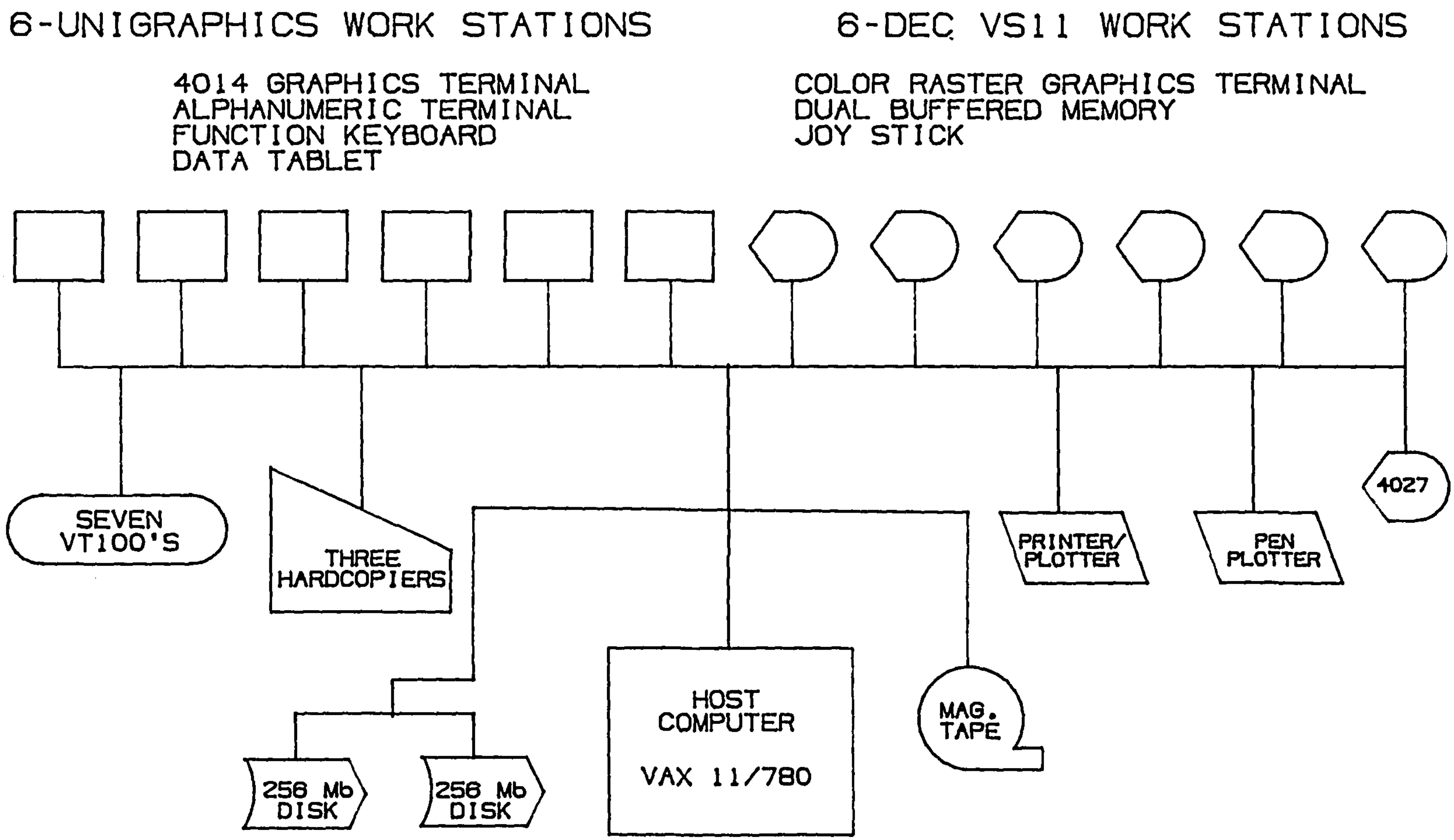


Figure 2.1: Hardware Available

for Unigraphics, each includes a Cathode Ray Tube, (CRT) display, message monitor, function and alphanumeric keyboards, and a pen and tablet for optional input, see Figure 2-2. Each design station operates independently of the others. The viewing screen of the storage CRT is used forgraphic display only and has a resolution of 4096* 4096 addressable points. The message monitor is the prompting device used to communicate with the user and to keep the storage tube clear for graphics. The user responds to the lists of options (menus), or requests for input, which are displayed on the message monitor screen. The Program Function keyboard is used for option selection. Each button on the keyboard represents an operation or function. The function or numbered buttons are lighted in accordance with the display on the message monitor. The alphanumeric keyboard is used to enter required parameters and text. The keyboard also has a pair of thumb-wheels which control the horizontal and vertical movement of the position indicator. The thumbwheels are used in conjunction with the space bar to indicate positions on the screen.

For a detailed description of the Unigraphics hardware, refer to Unigraphics Users Manual(2).

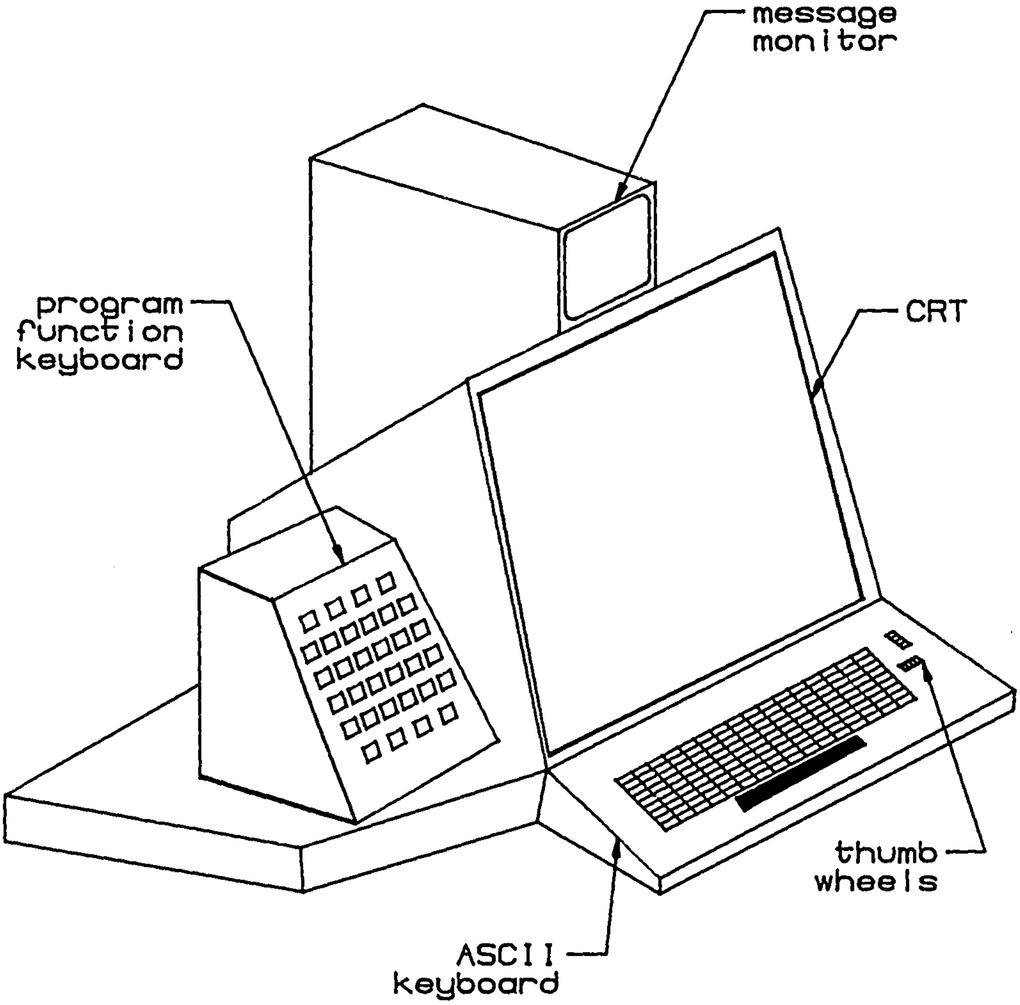


Figure 2.2: A Unigraphics Workstation

2.1.2 Software

The Unigraphics software supported on the system and used as the starting point for this thesis is briefly outlined below. Also discussed, is a brief description of the various modules on the Unigraphics system.

The Unigraphics software package consists of several different modules which are linked together as shown in Figure 2-3. The Unigraphics system can be used to generate, store, retrieve, and manipulate geometric and graphic information. It can produce computer read-outs and pictures of three-dimensional figures on a DVST screen, or on a hard-copy drawing and generate Cutter-Location (CL) data for numerical control (N/C) processing. A design that is a complete three-dimensional model of a part may be developed on the Unigraphics system. The user creates geometry to describe the object being designed, and this description is stored in the computer memory, under an appropriate naming or number convention. The stored model or part can be used to produce fully-dimensioned engineering drawings, or to generate numerically controlled instructions for machining and input for analytical processes, such as finite element analysis. A brief description of the different modules shown in Figure 2-3 follows:

N/C MODULES:A single function button, "N/C Machining", allows the user to select machine tool operations and set

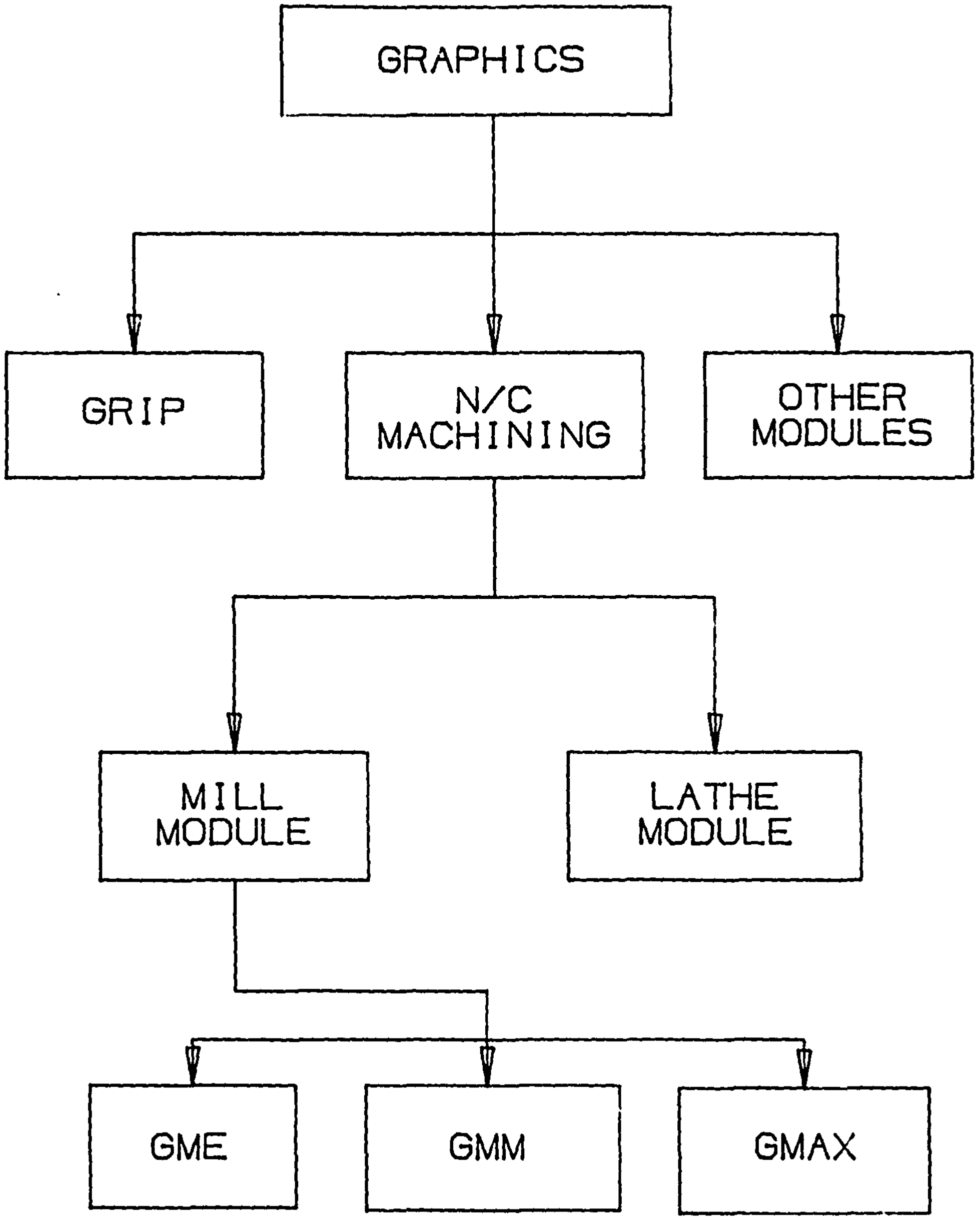


Figure 2.3: Various Modules in Unigraphics

control parameters. Unigraphics machining modules provide an interactive graphics alternative to conventional N/C programming techniques. After construction and verification of the geometry of a part, that same geometry is used as input to the machining modules.

The main machining modules suported by this package are the Mill and and the Lathe modules. The Mill Module is further sub-divided into: 1) GME (Graphics Machining Editor) -- This module is used for drilling operations as well as for editing of the CL-Source Files. 2) GMM (Graphics Mill Module) -- This module is used for profiling and pocketing of parts on 2 and 2&1/2 axis machines only. 3) GMAX (Graphics Multi-Axis Module) -- This is used for all milling operations performed on 3 to 5 axis machines. After input of manufacturing data, such as tool dimensions, speeds, feeds and tolerances, the system automatically generates tool paths and then displays the tool motions, simulating the machining of the part.

GRIP: The GRIP (Unigraphics Interactive Programming) language is an optional extension to Unigraphics which provides the user with an alternative means of operating the system. Almost any operation which can be performed interactively in Unigraphics can also be performed by executing the commands of a GRIP program. For example, GRIP commands are available to create geometry and dimensions and to perform transformations and file management. In addition, there are several commands which allow a GRIP program to communicate with the user as it is running by displaying messages and menus.

GRIP is a higher level language; it performs mathematical computations, includes various branching and looping capabilities, and allows use of separately compiled subroutines. Any systematic design procedure can be encoded as a GRIP program which will then carry out the procedure automatically, prompting the user to make decisions and enter data only when necessary. GRIP programs can provide a wide range of capabilities from minor custom enhancements to elaborate family-of-parts systems which will automatically produce complete sets of engineering drawings. GRIP is being used in this thesis to draw clamps for the parts to be milled. These clamps are drawn as family-of-part systems, so that the user can type in the size of the clamp suitable for the particular job.

OTHER MODULES: The Unigraphics Finite Element Module (GFEM) is an optional module which provides finite element users with a graphics modelling tool. Through GFEM, an engineering group can do both design and modelling on the same system. Once the design phase is completed, the existing part can be used to generate nodes and finite elements.

The optional Unigraphics Schematic Module (GSM) enables the user to create schematic drawings and then generate Bills of Materials and Connector Lists from those schematic drawings. GSM has three main capabilities: 1) Producing and maintaining a user defined symbol library. 2) Producing and maintaining schematic drawings. 3) Producing inventory related information

For a more detailed information on the above modules refer to Unigraphics Users Manual(2).

2.2 Current Methodology

A block diagram of the procedure to design and mill a part on the existing system is shown in Figure 2-4. The user first creates a 3-dimensional model of thepart to be machined, interactively, by responding to the messages on the message monitor. User response to the system is via the Program Function and the alphanumeric keyboards. After construction and verification of the tool geometry, that same geometry is used as an input to the machining modules. The user enters the "N/C Machining" sub-function through the Program Function Keyboard. The appropriate sub-modules, GME, GMM or GMAX are then used to obtain the tool-paths required for machining of the part and all manufacturing data, such as, tool dimensions, speeds, feeds, and tolerances is then input. Once this is done, the system automatically generates the desired tool paths and displays the tool motions,

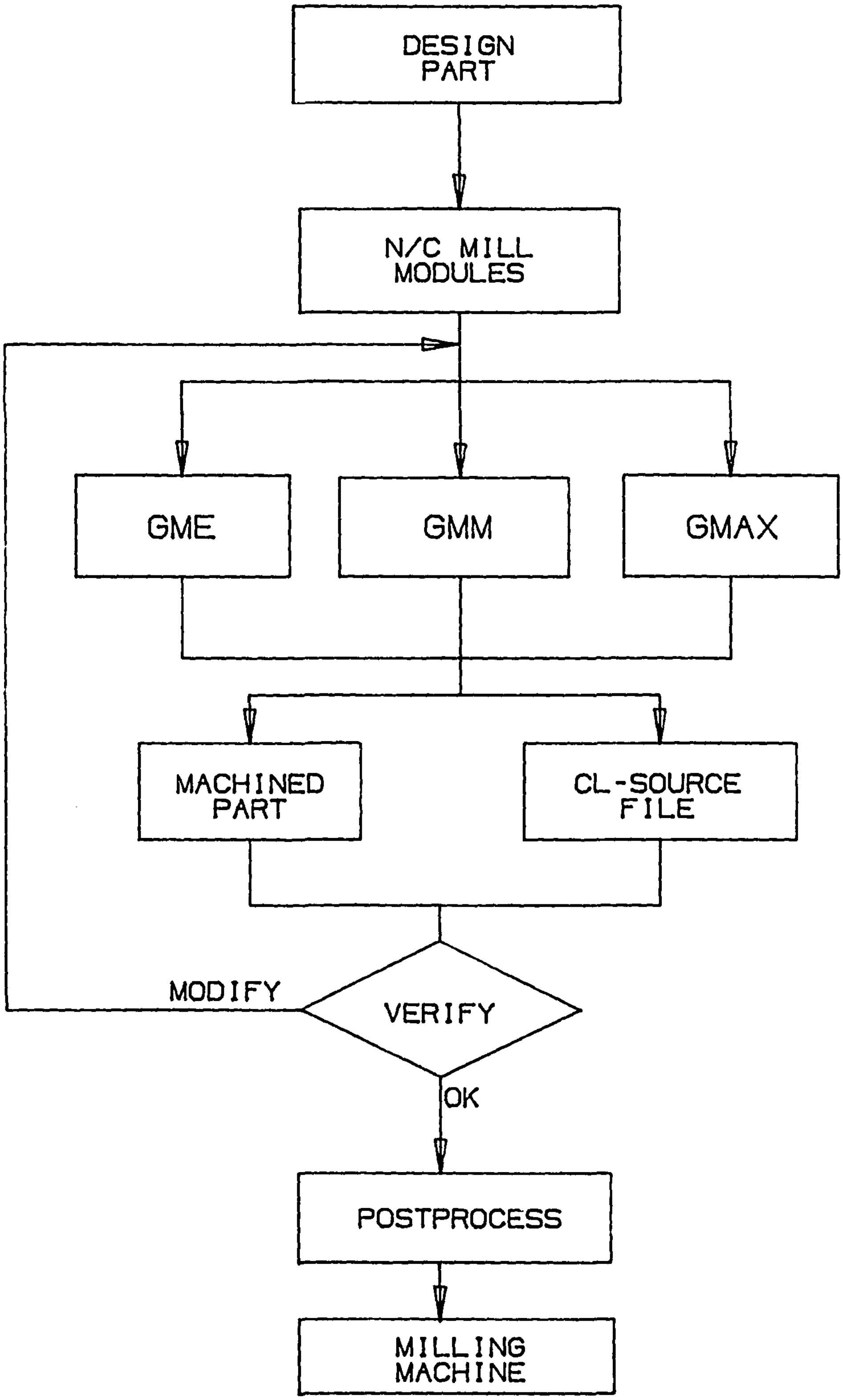


Figure 2.4: Milling Procedure on Existing System

simulating the machining of the part. Simultaneously, the system also creates a Cutter Location (CL) Source File in the users directory, which can be accessed later. The CL-Source File contains all the information about the tool and tool motions, such as tool dimensions, names of tool paths, feed rates, tool locations, etc.

The next step is to verify that the part has been machined correctly. This is done visually on the display terminal and can be sent to the plotter for additional verification. The CL-Source File can be edited to provide for any modifications in the tool path, as well as to enter some post-processor and auxiliary commands which are specific to a given machine. Now the post-processed file can be sent via electronic file transfer, cassette tape, or punched paper tape to the specific milling machine for the actual creation of the part.

2.3 Advantages/Disadvantages of Current System

In most applications of computer graphics the quality and features of the displayed image is very important. The present Unigraphics terminals are of the Direct View Storage Tube (DVST) type and represent pictures as line drawings/wire frame drawings. As mentioned previously, these terminals have a high resolution (4096*4096 addressable points) and display a bright green picture. However, as with most DVST's these can generate images of a single level of line intensity and single color only, due to the limitations of the phosphor.

Moreover, an inherent problem with the Direct View Storage Tube is the difficulty in removing the storage charge to erase the picture. The normal erasing method, here, is to apply a positive voltage to the storage tube, which removes all the charge but also generates a rather unpleasant flash over the entire screen surface. This erase problem is a serious problem of the DVST, for it prevents the use of the device for dynamic graphics applications. This disadvantage is indubitable specially in the case of the Mill Module where the tool draws lines over the part to be machined, as a result of which it is almost impossible to distinguish the part among the maze of lines on the screen. The problem is aggravated by the fact that all the lines are of the same color. Figures 2-5 and 2-6 illustrate an example in support of the above discussion. Figure 2-5 shows a part which has been designed on the Unigraphics system and displayed on the DVST. And Figure 2-6 shows what it looks like after the required machining operations have been performed.

Visual verification of the tool path on the display device is a must, because it eliminates faulty runs on the machine, thus saving both machine time, operator time, and material. Here the DVST has an advantage because of its high resolution. But the inability to stop the tool in any position during the machining

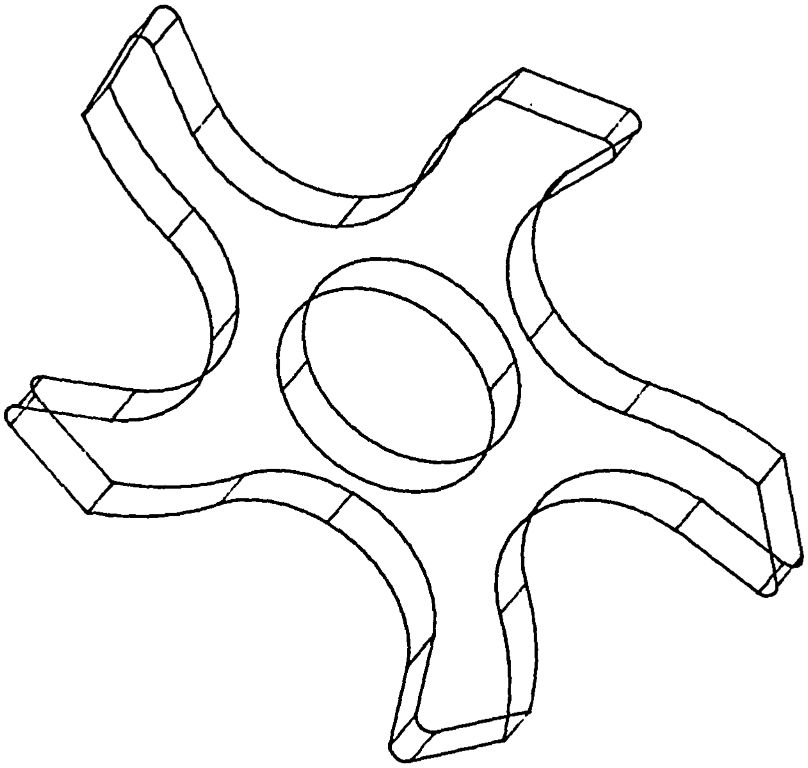


Figure 2.5: Part Designed on Unigraphics

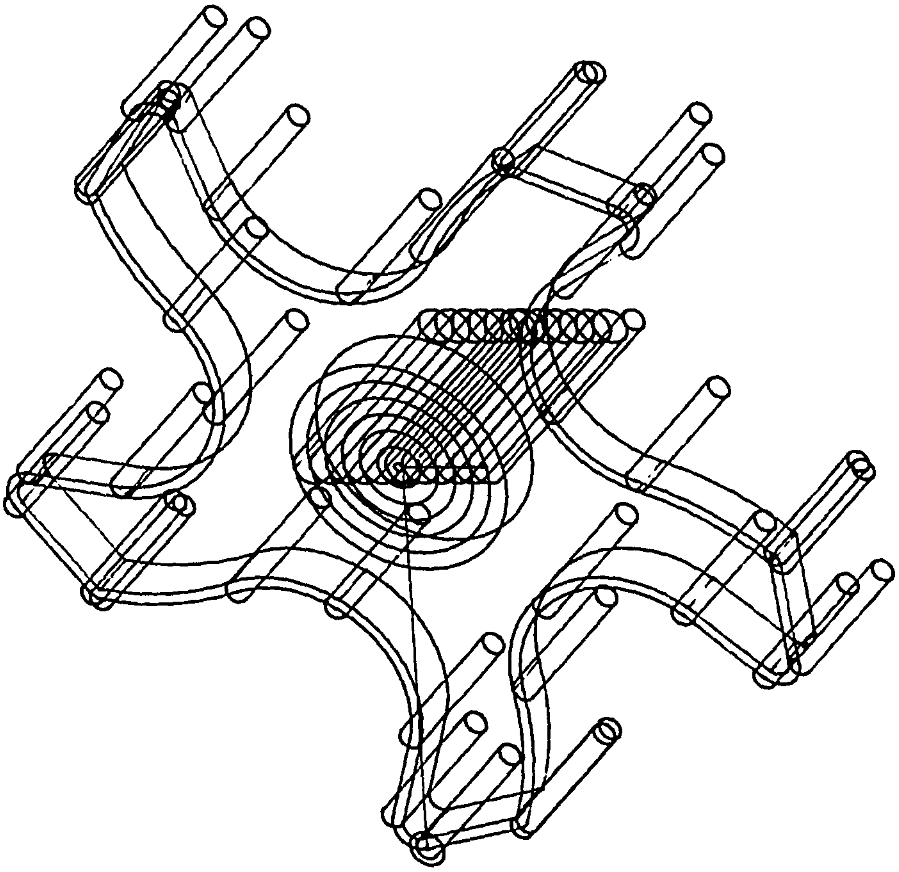


Figure 2.6: Machined Part

operation is its serious drawback. On these terminals, once the manufacturing operation starts, it is impossible to stop it in-between and verify the location of the tool. Also, it is not possible to run dynamic graphics because of the inherent limitations of the hardware. The data base as a wire frame is also a problem, since without surface definitions the design cannot detect interferences with fixtures and clamps, requiring interference to be detected visually.

2.4 Description of VS11 System

The VS11 system is a color video graphics system, consisting of three modules and a joystick assembly, supported on the VAX 11/780 minicomputer. Most of the drawbacks of the Unigraphics system can be eliminated by the use of a raster scan terminal of the VS11 type. The following two sections explain the hardware and software of this system. Many of the features of the VS11 system are generic of raster scan technology.

2.4.1 VS11 Hardware

As discussed in the VS11 Installation Guide [22], the basic components of the system are: a high-speed display processor, image memory modules and sync generator/cursor controls. The system uses a high-speed microprocessor as the display processor and an image memory to store the pixel data output of the display processor. The image memory is then continually "read" to the system monitor for the display of pixel information. The use of an image memory eliminates the need for the storage of a display file in the host Central Processing Unit (CPU) and allows the display file to be changed after the first read pass.

Four of the six VS11 terminals have a single memory channel each. Dynamic graphics can be run on these as well as on the fifth and sixth terminals -- which have two memory channels -- resulting in smoother dynamic displays. Thus, for the dual memory terminal while one memory channel is being written to by the display processor, the other memory channel can be 'read' to the system monitor, and vice-versa. This results in less time lost between the creation and displaying of the image, giving a smoother display. All terminals are capable of displaying sixteen different shades of color. The single memory channels have a pixel resolution of 512*512 and intensity of 4 bits for each system monitor, while two memory channels have each a 512*256 bits of resolution and 4 bits of intensity.

Figure 2-7 is a block diagram of the basic VS11 Graphics System. The three major functions that appear on this diagram are the three modules of the system. These are:

1) the Display Processor

2) the Image Memory

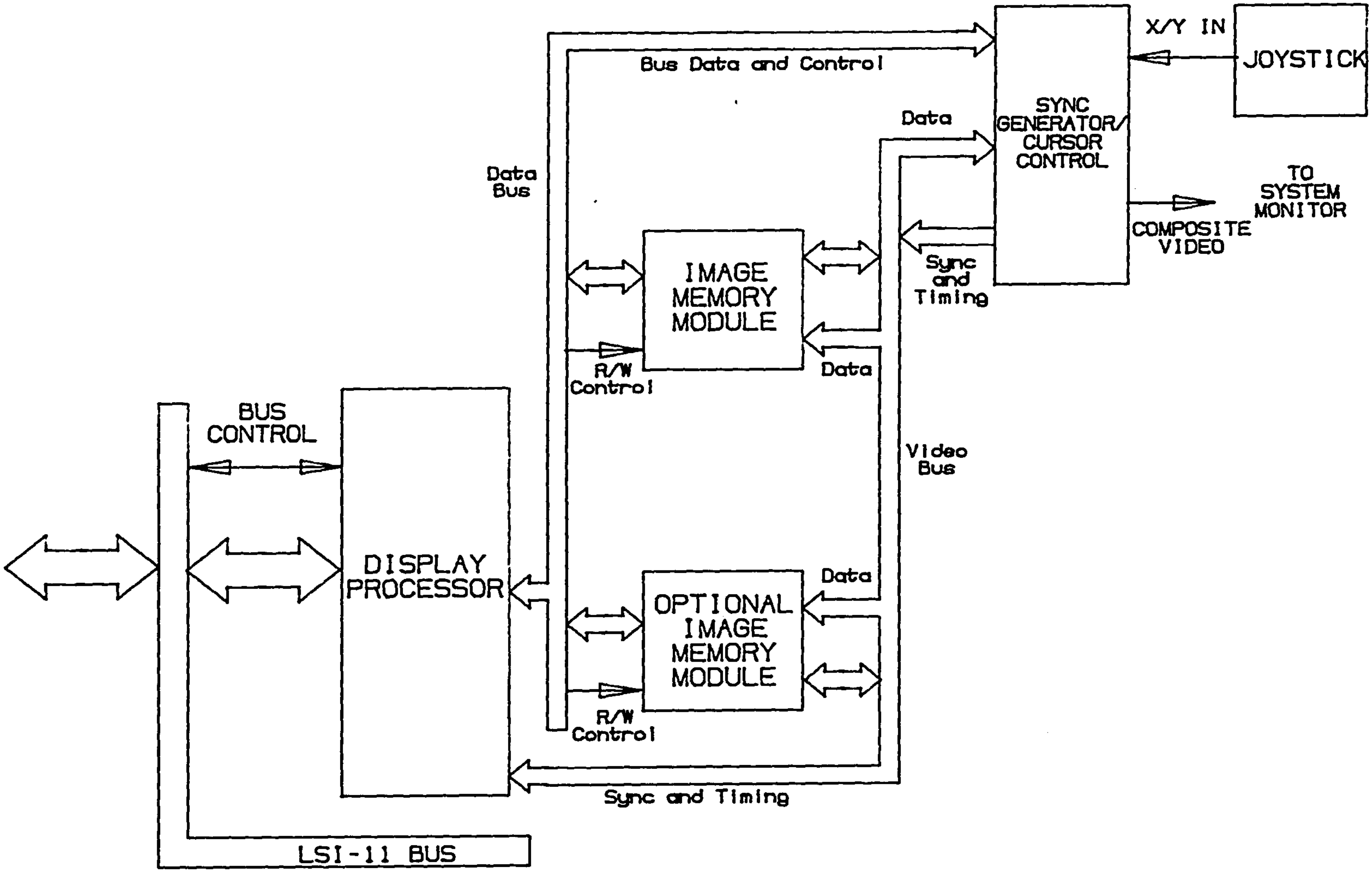


Figure 2.7: VS11 Graphics System Hardware

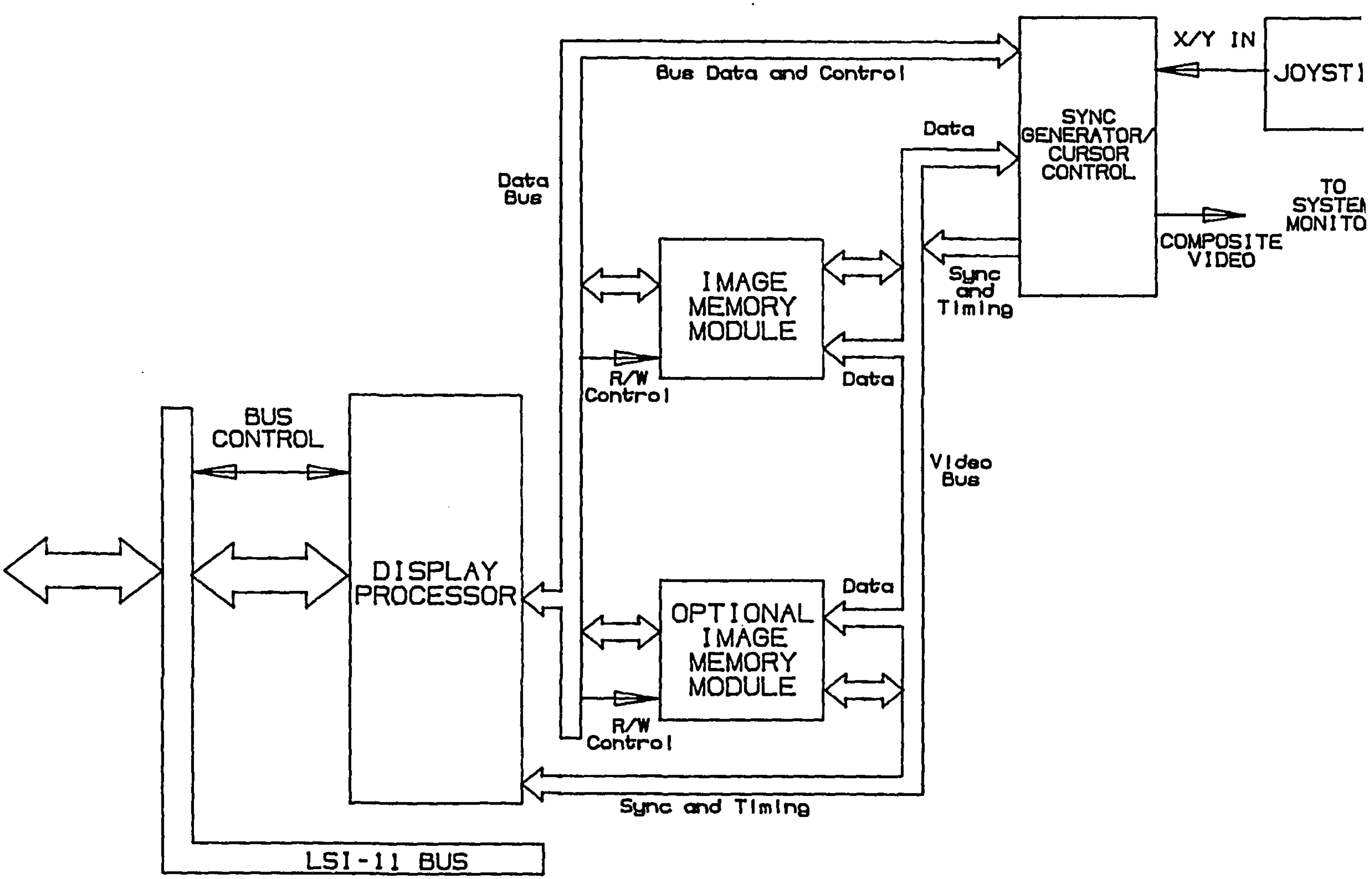


Figure 2.7: VS11 Graphics System Hardware

3) the Sync Generator/Cursor Control

The figure also shows optional additional Image Memory and the joystick, which is connected to all systems and can be used to control the cursor, if desired.

The VS11 system operates as a Direct Memory Access (DMA) device on the system bus. This way it avoids taxing the CPU; that is, it reads the display data from memory independently of the CPU, from which it just 'steals' a memory cycle whenever it needs an instruction. After constructing a display file in the CPU memory, the starting address of the file is transferred to the Display Processor's Display Program Counter (DPC). The Display Processor then makes Non-Processor Requests (NPR's) to gain use of the bus. Once bus master, the Display Processor, has direct memory access (DMA) to the CPU memory, it reads the display file which consists of graphics, data, and control instructions. These instructions are processed by the display processor and set-up the graphic mode, status registers, and control functions, and contain the pixel (picture element) information for display. Picture information processed by Display Processor is sent to the image memory for storage and eventual display on the system monitor.

The Image Memory stores an "image" of the pixel information to be displayed on the system monitor. Pixel data received from the Display Processor is stored in random access memories (RAM's) on the Image Memory.

Digital data for display is read from the Memory to the Sync Generator/Cursor Control via the video bus. Because the Image Memory stores the picture to be displayed, the Display file in the CPU memory is not required for refreshing the display and can be changed after its contents are transferred to the VS11. Optional Image Memory Modules can be added to the basic VS11 system to increase pixel resolution, to run dynamic graphics, or for multi-monitor and/or multi-channel operation.

The Sync Generator/Cursor Control provides digital-to-analog conversion of the pixel data from the Image Memory for display on the system monitor. Cursor control circuits within the Sync compute positions for display on he system monitor based on the method of data input -- the joystick or display file primitive can control the displayed cursor position. Joystick position is read back to the display processor via the memory bus/DBUS during a switch interrupt. The joystick is a rate type, that is, small movements of the joystick produce slow movements of the crosshairs. Large joystick movements cause the crosshairs to move rapidly.

For a detailed description of the VS11 graphics hardware refer to VS11 Installation Guide [22].

2.4.2 VS11 Software

Data to be displayed by the VS11 is first organized into a display file in the memory of the host CPU. The file consists of a series of VS11 instructions (graphic,control, and data) which define an image. After the display file is constructed, the VS11 display "program" is initiated by moving the display file starting address to the VS11 Display Program Counter (DPC). Once initiated, the VS11 issues non-processor requests to sequence through the display file instructions and generate the desired image on the system monitor. Since the VS11 contains an Image Memory, continuous image refresh from CPU memory is not required. After one pass through the memory file, the display file may be altered or removed.

Pictures are produced by writing pixel data into one or more image memories. The memories to be written must be enabled for writing by the program via a control instruction, and since more than one memory can be write-enabled at a time, the program must make sure that the memories not to be written are placed in a Read-only or Protect state. Once the graphis data is written, the program must assure that the memory is placed in a Read-only or Read/Write state so that pixel data is "read out" to the monitor.

The VS11 3D-Graphics Package [11] takes care of most of the intricacies presented above. This package is designed to create and display user-defined pictures on the VS11 terminals. The basic geometric entities to draw pictures may consist of points, lines, arcs, conics, or splines. It is also possible to represent ruled surfaces, tabulated cylinder, and surfaces of revolution. The VS113D Graphics Package is a collection of Fortran-callable sub-routines developed at Lehigh, to support graphics on the VS11 terminal.

A sample program to draw the picture of Figure 2-9 is shown in Figure 2-8. This method of graphics representation does not require the user to be too familiar with the system hardware. Most of the software details are taken care of by these routines and are transparent to the user. For instance, the sub-routine 'INIT', initializes the VS11 driver so that it is ready to receive other graphics and control instructions. This also puts both the memory channels in read/write mode enabling data to be displayed from both channels. The sub-routine 'SETWIN' defines the minimum and maximum X, Y and Z limits of the window and assigns the window a number. The 'POINT', 'LINE', and 'CIRC3D' sub-routines draw points,lines, and circles resp., at the designated positions. The call to 'AXES' draws a co-ordinate axis at the position desired and the call to 'SEND' dumps the contents of the frame buffer to the system monitor, thus drawing the picture represented by calls made to other sub-routines before 'SEND'. The result in this case is Figure 2-9.

```
LOGICAL T,F
T=.TRUE.
F=.FALSE.
CALL INIT
CALL SETWIN(0.,511.,0.,479.,-500.,500.,1)
CALL DISWIN(1,1)
CALL NEWCOL(2)
CALL POINTA(-75.,-75.,0.,T,IO,0,1)
CALL DRAWR(150.,0.,0.,T,1,1)
CALL DRAWR(0.,150.,0.,T,1,1)
CALL DRAWR(-150.,0.,0.,T,1,1)
CALL DRAWR(0.,150.,0.,T,1,1)
CALL NWCSXY(0.,0.,0.,1.,0.,0.,0.,1.,0.,1)
CALL CIRC3D(0.,0.,0.,40.,T,1,1)
CALL AXES(1,1,20,T)
CALL SEND
STOP
END
```

Figure 2.8: Program to Draw Picture of Figure 2.9

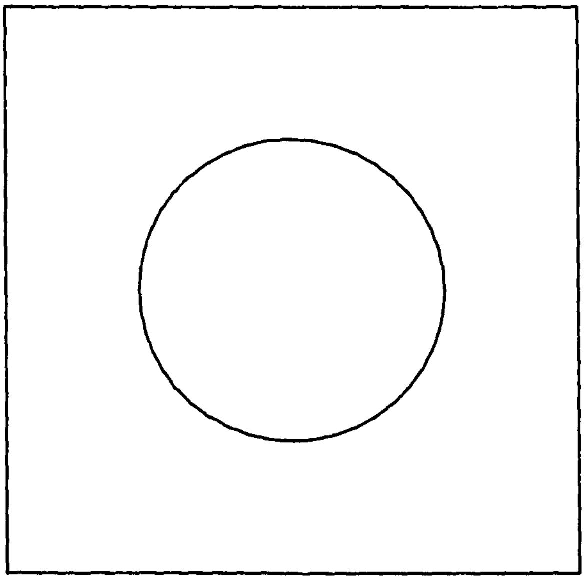


Figure 2.9: Picture Drawn by Program of Figure 2.8

It is possible to display the pictures dynamically, as well as do translation, rotation, zooming, or scaling of the display. For an additional description of the package refer to the VS11 3D-Graphics Package [11] or the VS11 3D-Graphics Example manual [12].

2.5 Advantages/Disadvantages of Dynamic Graphics/VS11 system

It should be emphasized at this point that the two graphics systems discussed above are predominantly independent of each other and at the same time are complete packages in themselves. However, both the systems have different capabilities, some tasks which can be accomplished much better on one cannot be done so well on the other, and vice-versa. There also are some jobs which at present can be executed on one system only but would appear much more efficiently on the second system. Our goal, thus, is to integrate the two systems in such a way that most, if not all, the jobs can be done on both systems.

The design discussed in this thesis is one of the steps in that direction. Here we have taken the Mill Module and interfaced it to the VS11 terminals, so that all the numerically controlled machining which has been done on Unigraphics terminals can now be displayed on the raster scan VS11 terminals. The pictorial representation of tool motion can be manipulated interactively, allowing the user to such facilities as rotation, scaling, zooming, and scaling of the tool and the part to be machined.

The user has complete control over the colors which can be assigned to different entities. The tool can be stopped at any position and its location verified from the displayed co-ordinates. Since the tool does not leave a trace along its path of travel (this is kept as an option), the part to be machined is not obscured by the lines drawn by the tool, as on the storage tube terminals. The part can be oriented in any direction for better display, the machining operations performed, and checks can be made for interference. If desired, the user can zoom in on any particular portion of the part to be machined to take a "closer-look" at the machining operation. All these advantages make the Mill Module more versatile. The disadvantage of the VS11 display is its low resolutionas compared to the storage tubes. The dual buffered terminals have a still lower resolution. A comparison of the resolution available on the storage tubes, single (512*480*4), and dual buffered (256*480*4) raster scan terminals can be seen in the Figure 2-10.

The system design to transfer and display the tool motions from Unigraphics to VS11 terminals is discussed in the next chapter.

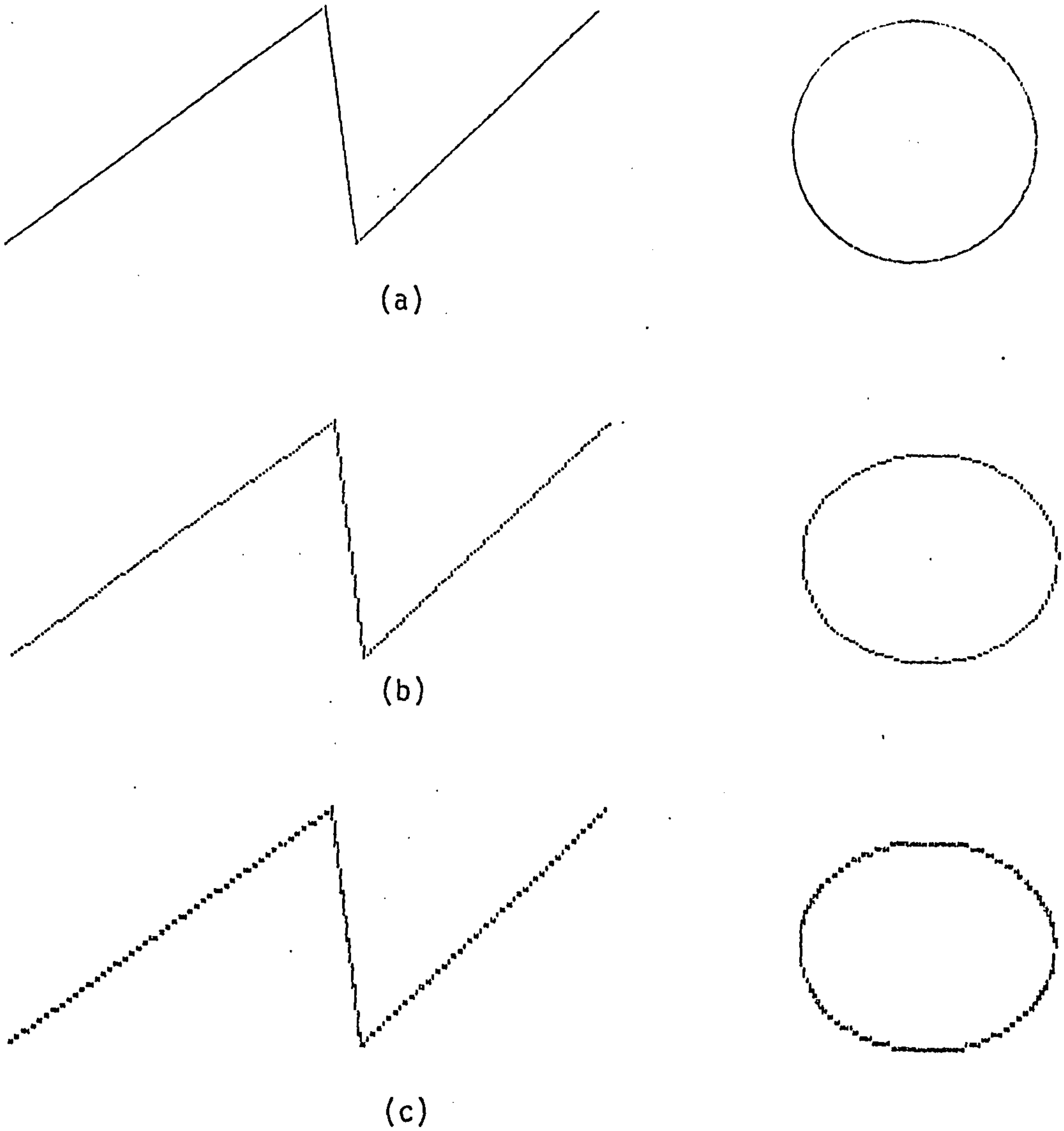


Figure 2-10: Comparison of resolution of (a) storage tube display (b) 512x480 raster display (c) 256x480 raster display.

CHAPTER 3

DESIGNED SYSTEM CHARACTERISTICS

The characteristics of an interactive system designed to display the tool-motion on numerically controlled milling machines, emloying real time dynamic interactive graphics, are presented in this chapter. The system has been developed to take advantage of the present display technologies and some software features that are currently in the developmment stage.

The designed system characteristics are discussed in three parts. The first section is concerned with the basic configuration of the system, including some software aspects and user interface options. The actual method of generation of data, its retrieval from the database of one system and its transfer and storage to a neutral format is discussed in the second section. Finally, the third section discusses the capabilities that the system has to generate and display the transferred data. The command menu structure and the options which the user has to control the display of the tool are also explained. The last two sections also give the flow-charts of the programs designed to perform the required operations.

3.1 System Configuration

A block diagram of the steps needed to display the tool path in the described system is shown in Figure 3-1. First, the Unigraphics CAD/CAM system is used to generate the part to be machined. The required machining operations are then performed on the designed part with the Mill Module, and the system displays the tool motion. To transfer this data and appropriate files so that they can be displayed on the VS11 terminals, the user executes the transfer program (TOOL-TRANSFER), and then the display program (TOOL-DISPLAY). The system works in an interactive mode in which the user communicates and selects menu options to obtain the desired display. The man/machine interface is accomplished with a user selectable menu which appears on the graphics screen and prompts the user to interact in real time.

Several factors were considered carefully in designing an efficient, user-friendly, and device independent system. These factors divide naturally into four components:

1. Data Base Handling Techniques
2. Display Format
3. Command Menu Structure

For each of these items, the characteristics and capabilities as described in Chapter 2 have been considered. For the system design, a clear understanding was required of the Unigraphics data base. Once this was

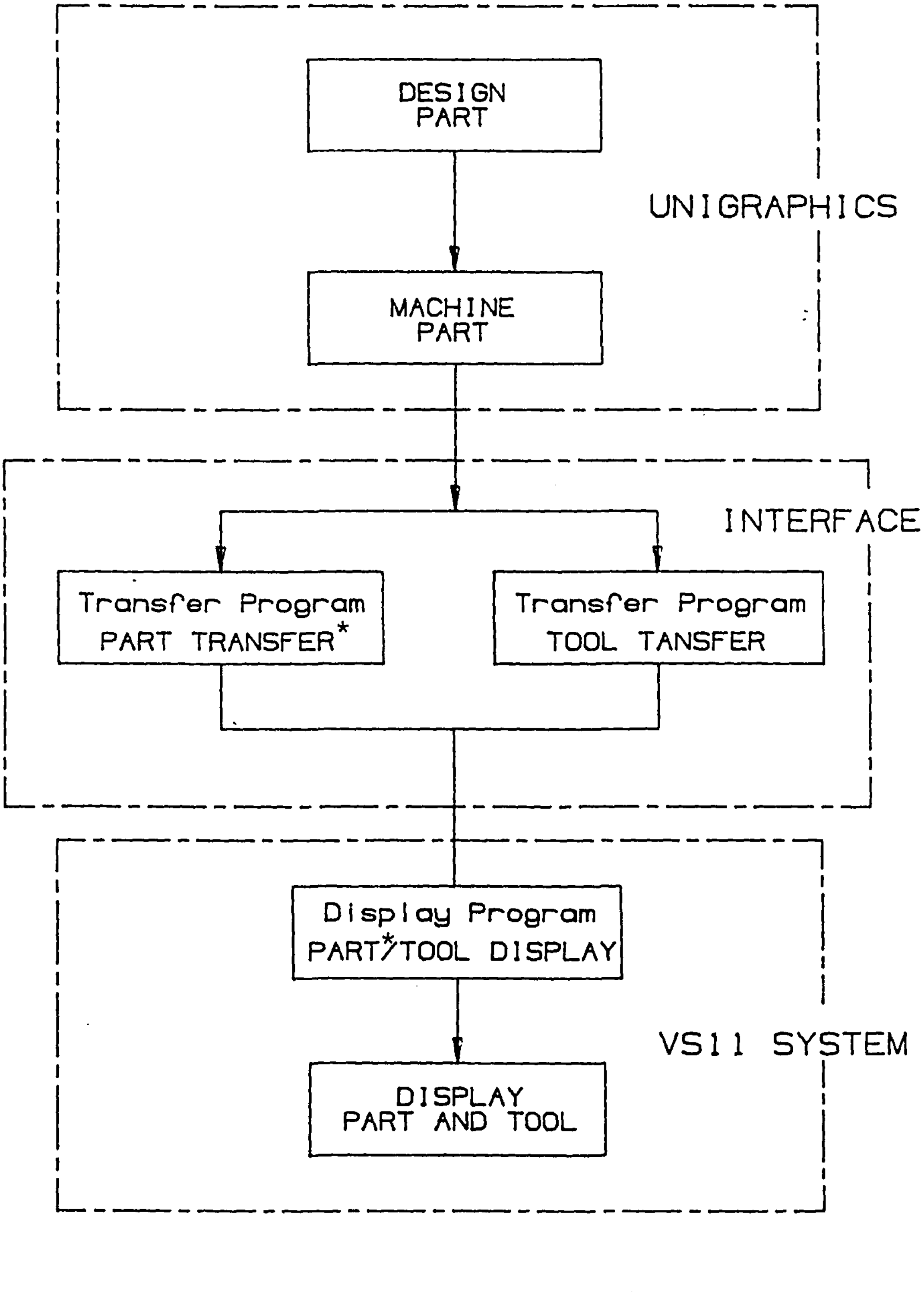


Figure 3.1: Tool Display Procedure on VS11 (*Programs developed by Prof. Ozsoy as part of the project.)

obtained, various ways of transferring this data to the VS11 terminals were considered, including the Initial Graphics Exchange Specification [6] and the Tornado Data Base Structure [19]. Since the data transfer in this system involved only the Mill Module, it was decided to store the data in a neutral format, in unformatted files, which could be accessed directly by another program for display on the monitor. This eliminated any need for the development of graphics pre- or post-processors, also saving a lot of memory space and increasing the execution speed of the transfer and display programs.

The display on Unigraphics is through the Direct View Storage Tube and messages are displayed through the system monitor. The user responds to the system through the Program Function and the terminal keyboards. In contrast, the VS11 displays both the messages/commands and the graphics pictures on the display monitor only. The joystick has not been used and only the keyboard was considered sufficient for user responses. The switch on the joystick was used as an interrupt.

The command menu structure is of the hierarchical dynamic type. Menus are not displayed on the screen until the user desires to do so. The menus go to a maximum of three levels with each menu consisting of no more than six options. The system capabilities include scaling, translation, rotation, and zooming of the display as well as individual color control over each entity and the tool. The part displayed can be viewed from any angle and the machining operation performed on it. The display speed of the tool is controllable -- rapid or slow. Standard clamps have been designed to hold the part while the machining is done to give a realistic picture of machine tool motion.

3.2 Data Generation, Transfer and Storage

The first step is to design a part on Unigraphics and machine it with the Mill Module, which automatically generates a Cutter-Location (CL) Source File. This generates data for the part and the tool path in the Unigraphics data base.

The next step is to read this graphics data and convert it to a standard or neutral format. This can be stored somewhere and retrieved later for eventual display on the desired system. Essentially, the data has to be transferred in two phases. The first phase consists of transferring the part to be machined and the second is to transfer the CL-Source File. To transfer part files, the Unigraphics/Files-11 Interface Package(UFIP) [18] as documented in the Unigraphics Programmers Manual [20], has been used. UFIP consists of a subroutine package which enables one to store and retrieve Unigraphics data files between the Unigraphics libraries and the Files-11 format on the VAX 11/780 host computer. In addition, the file transfer provides the capability to create and access Unigraphics part and CL-Source Files at the Files-11 level.

Part files consist of various entities such as points, lines, arcs, splines, etc. Each of these entities are stored in the Unigraphics data base in a specific format. To transfer these files and retrieve the co-ordinate information of each entity, the subroutines in the UFIP package were utilised. These routines transfer files between the two systems and enable one to read the desired information into a new data file, in a user-defined format, which can be accessed and displayed subsequently.

CL-Source Files contain information of tool position, tooldimensions, speed, and feed-rates. Figure 3-2 shows a sample unedited CL-Source File containing the usual commands. Not all the information in the CL-Source File is required for tool-path display on the VS11 system. To give the reader a better view of the requisites desired in the transfer program, TOOLTRANSFER, the structure of the CL-Source File is discussed below along with some of the important commands.

Usually, the first instruction in a CL-Source File is a 'FROM' statement. This is a post-processor statement which directs the machine tool and informs it the position from which tool motion is to begin. Other instructions, such as, post-processor and macro commands, follow this statement. The program, TOOLTRANSFER,

```
FROM/0,0,0
TOOL PATH/0.125,0.000,1.000,P1
INDEX/10
GOTO/5.000,-0.125,1.500
CIRCLE/5.000,1.000,1.500,0.000,0.000,-1.000,1.125,
       0.002,0.500,0.250,0.000
GOTO/6.125,1.000,1.500
GOTO/6.125,3.000,1.500
CIRCLE/5.000,3.000,1.5000,0.000,0.000,-1.000,1.125,
       0.002,0.500,0.250,0.000
GOTO/5.000,4.125,1.500
GOTO/1.000,4.125,1.500
CIRCLE/1.000,3.000,1.500,0.000,0.000,-1.000,1.125,
       0.002,0.500,0.250,0.000
GOTO/-0.125,3.000,1.500
GOTO/-0.125,-0.125,1.500
GOTO/5.000,-0.125,1.500
INDEX/10,NOMORE
COPY/10,SAME,2
END-OF-PATH
TOOL PATH/0.125,0.000,1.250,P2
CYCLE/DRILL
GOTO/0.500,0.500,2.000
GOTO/3.000,0.500,2.000
GOTO/5.000,0.500,2.000
GOTO/5.000,3.000,2.000
GOTO/3.000,3.000,2.000
GOTO/0.500,3.000,2.000
CYCLE/OFF
END-OF-PATH
FINI
```

Figure 3.2: Example CL-Source File

ignores these, until the next significant statement "TOOL PATH" is encountered. This contains information about the tool -- its diameter, corner and length -- and the name of the path. Since this information is required later for display, it is stored. Next is the "CYCLE" statement which informs the machine tool that the commands which follow are drilling commands. The program, TOOL TRANSFER, sets a flag for the display program, TOOL DISPLAY, and proceeds to the next record.

The "GOTO" and "CIRCLE" commands contain information of the tool path and the orientation of the tool. For the case of 2 and 2&1/2 axis machining the the "GOTO" statement has three arguments, which are the location co-ordinates of the centre of the tool. The tool axis in this case is always along the positive Z-direction. In 5-axis machining the "GOTO" command has six arguments which are the location co-ordinates of the tool center and the unit vectors of the tool axis describing the orientation of the tool. All the information in the "GOTO" command is stored because it is required later for display.

The "CIRCLE" command occurs whenever the tool is going along a circular arc. This command always has eleven arguments of which we require only the first seven. These are the X,Y,Z co-ordinates of the center of the arc which the tool is traversing, the unit vectors of the tool axis, and the path radius.

The next commands are "INDEX" and "COPY" which are for repetitive tool paths. With the help of these commands the user can translate, rotate, or copy the same tool path at any specified location. The last two commands are "END" and "FINI". "END" indicates the end of a tool path. The "FINI" command is not inserted by the system. This command is the last statement in a CL-Source file and the user should ensure that before the file is transferred from Unigraphics to VS11 this command is inserted. For a more detailed discussion of the structure and format of the CL-Source File refer to the Unigraphics Manuals [3, 4, 2].

Various methods have been attempted to access the CL-Source File from the Unigraphics library for eventual transfer to the VS11 system. The first approach has been to access the CL-Source File from the library for immediate transfer. The second option was to transfer CL-Source Files like Part Data Files, through the use of UFIP subroutines. Yet another method was to first convert the CL-Source File into a Grip Source File, by a Grip program. A program then would transfer this "Grip File" into Files-11 format, where it can be manipulated with ease for display.

CL-Source Files could not be accessed directly from the library because the memory address where these are stored are not available to the user. The second approach was not successful because of lack of availability of the data structure of Unigraphics CL-Source Files. The last approach, although it involves one more step, is quite straight-forward and is supported by the vendor. Files can be transferred from the CL-Source file library, through the GRIP Source library, to the VS11 directory without any change in format or any data translation taking place. The process is reasonably fast and does not take too much computer time. The flow of files between the modules is shown explicitly in Figure 3-3.

The CL-Source File of Figure 3-2 was transferred physically (without any change in format) from the CL-Source library in Unigraphics, via Grip, to the VS11 system. Once this is obtained it has to be edited to remove commands not required for display, such as the post-processor and machine commands. This is done by the transfer program, TOOL-TRANSFER, which then transposes the remaining commands into a neutral format, from where they can be accessed and subsequently displayed on the VS11 terminals, along with an animated display of the tool-path, using the TOOL-DIDPLAY routine. Several data storage methods -- both formatted and unformatted -- sequential, direct access, indexed, and relative were tried. The method found best was to store data in unformatted files and sequential organization. Unformatted files were preferred because the stored data is to be used by another program which then would manipulate it before displaying. Unformatted I/O saves

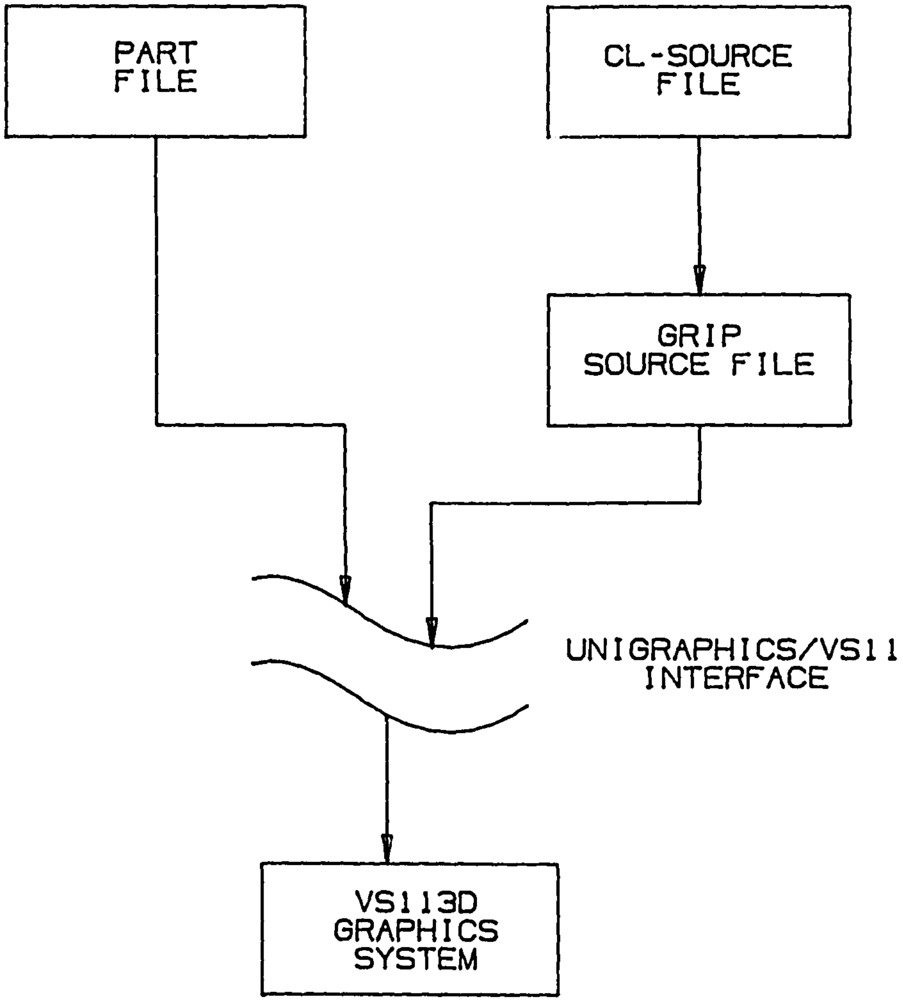


Figure 3.3: File Transfer

execution time by eliminating the data translation process, preserves greater precision in the external data and usually conserves file storage space.

The transfer program, TOOL TRANSFER, which was designed to pick out the required data from the CL-Source File and store it in another data file is described by the flow-chart of Figure 3-4. It is an interactive program which asks the user for the name of the file to be transferred and the new file in which this data is to be stored. Once the user gives this information the program informs the number of commands transferred. The next step for the user is to run the display program, TOOL-DISPLAY, which retrieves data from this file and displays it on the VS11 screen. The source code of the transfer program is available in the Unigraphics/VS11 Mill Module Interface [17].

3.3 Data Retrieval and Display

Once the transfer of data to the neutral format has taken place, it has to be retrieved, manipulated, and displayed in the best way possible. The manipulations involve scaling and transformation of the original picture to suit the present display monitor. Also, since originally the picture was represented in one color only but now it is propitious to make use of the 16-color terminal capability, suitable changes have to be made to the display file after it is retrieved. Moreover, as

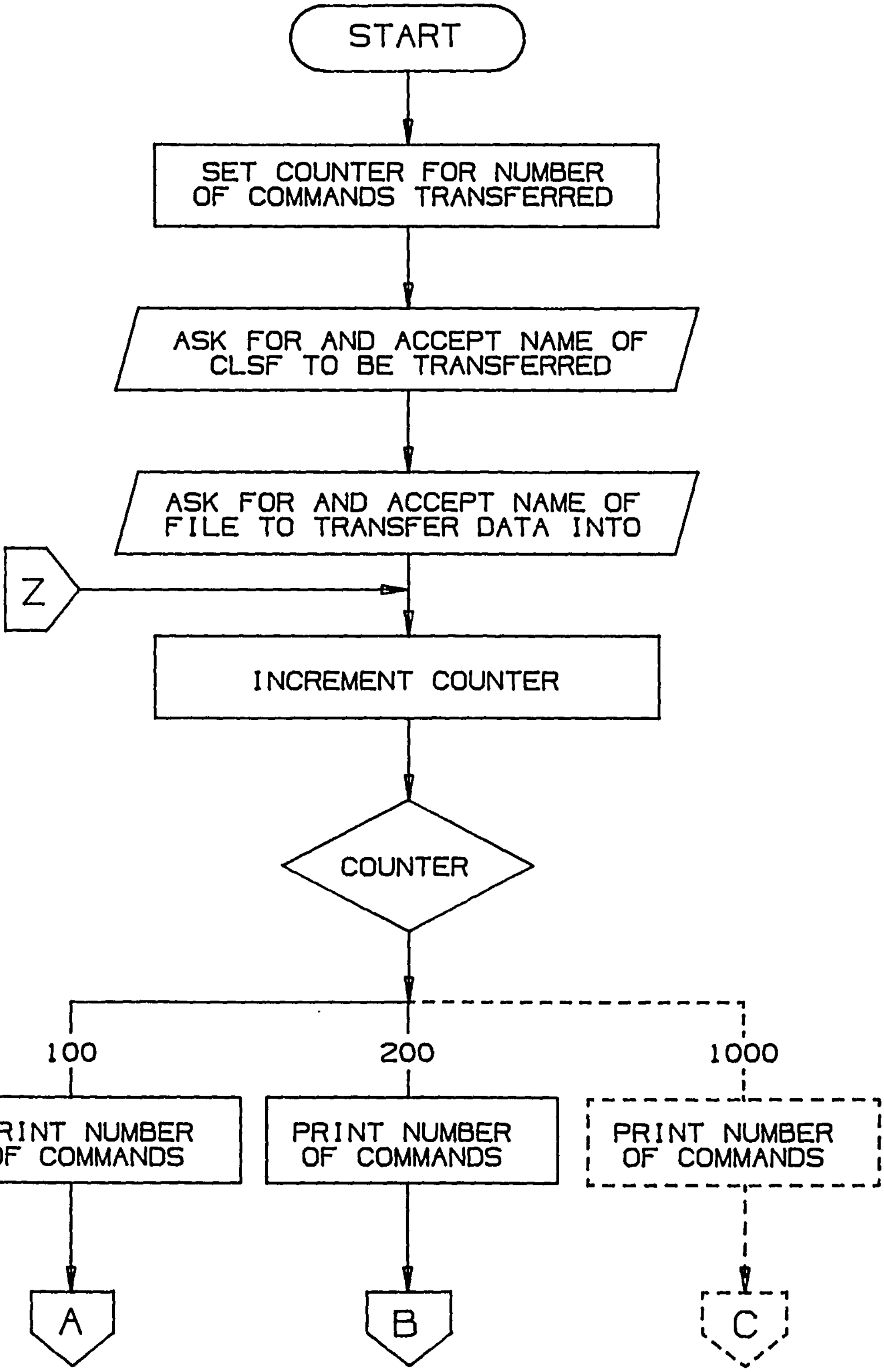


Figure 3.4: Transfer Program Flow-Chart

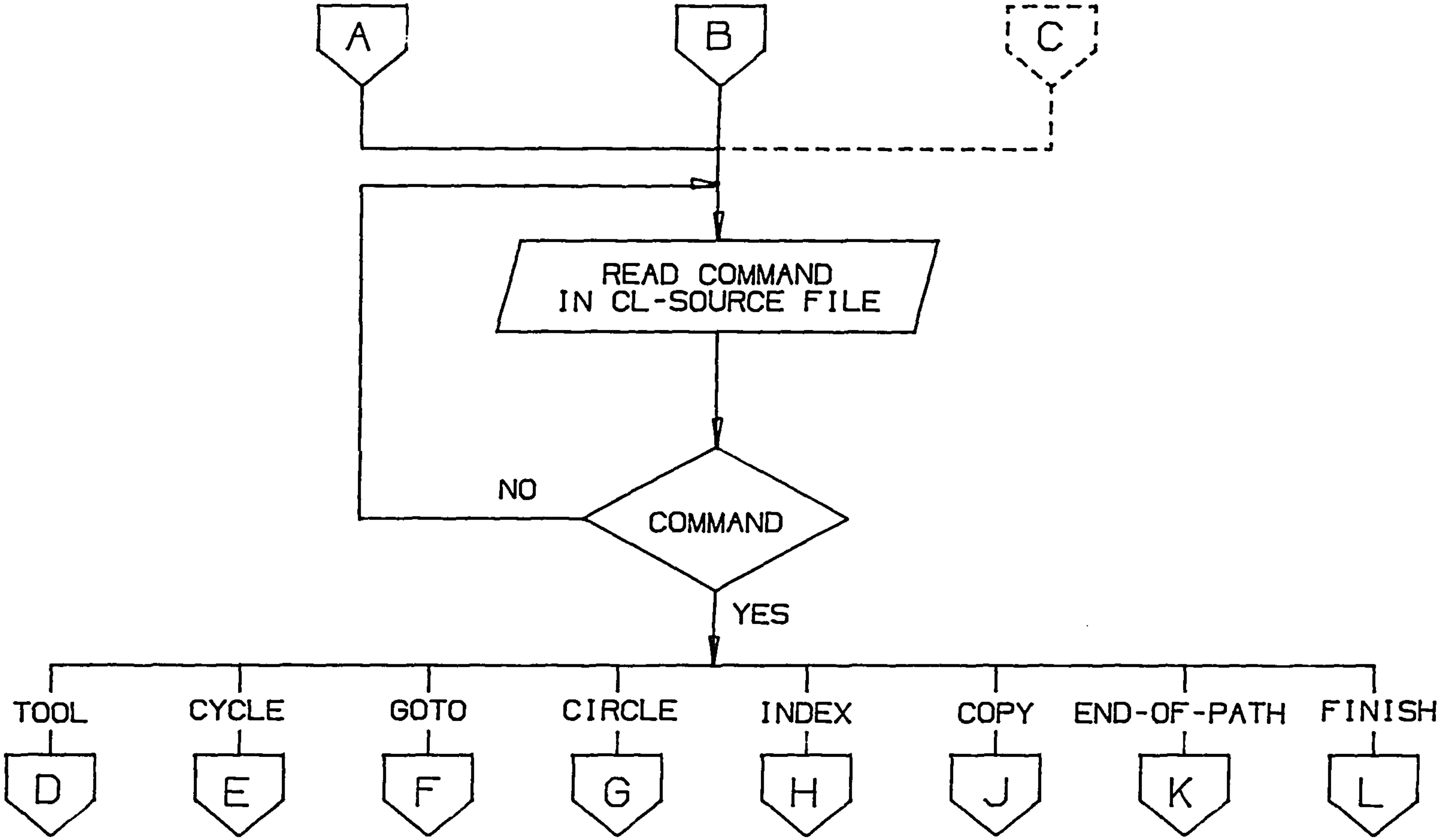


Figure 3.4 (contd.): Transfer Program Flow-Chart

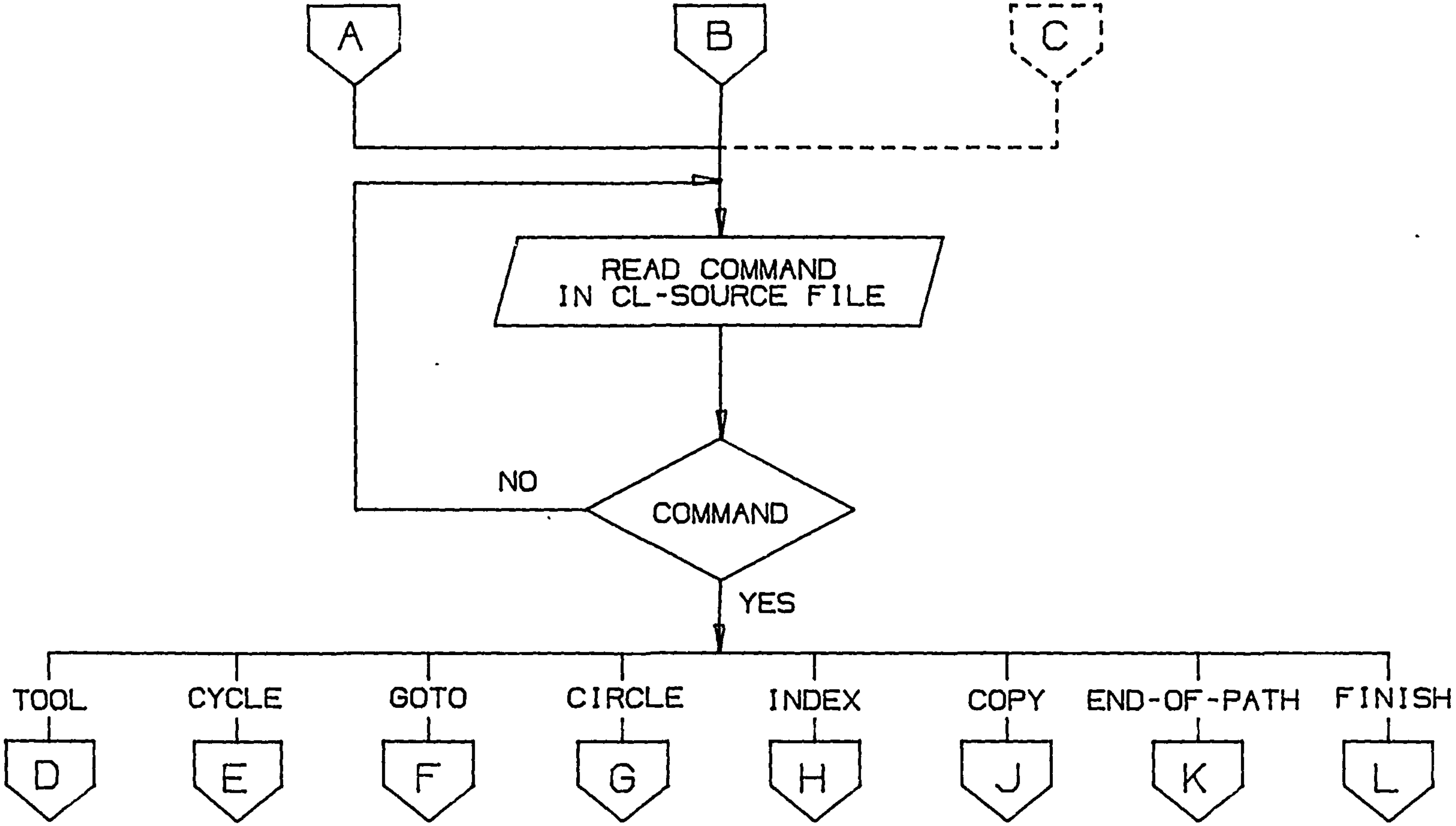


Figure 3.4 (contd.): Transfer Program Flow-Chart

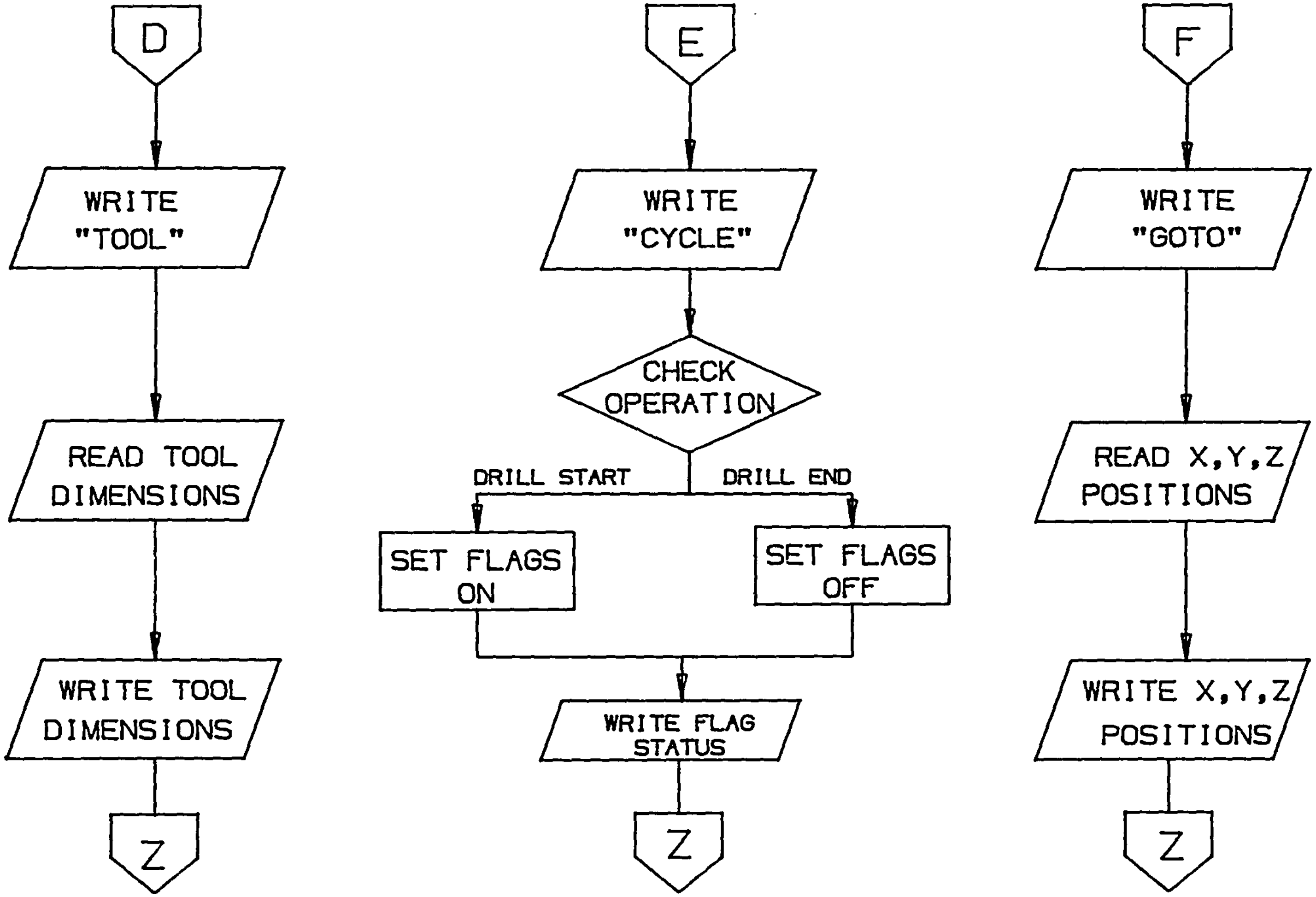


Figure 3.4 (contd.): Transfer Program Flow-Chart

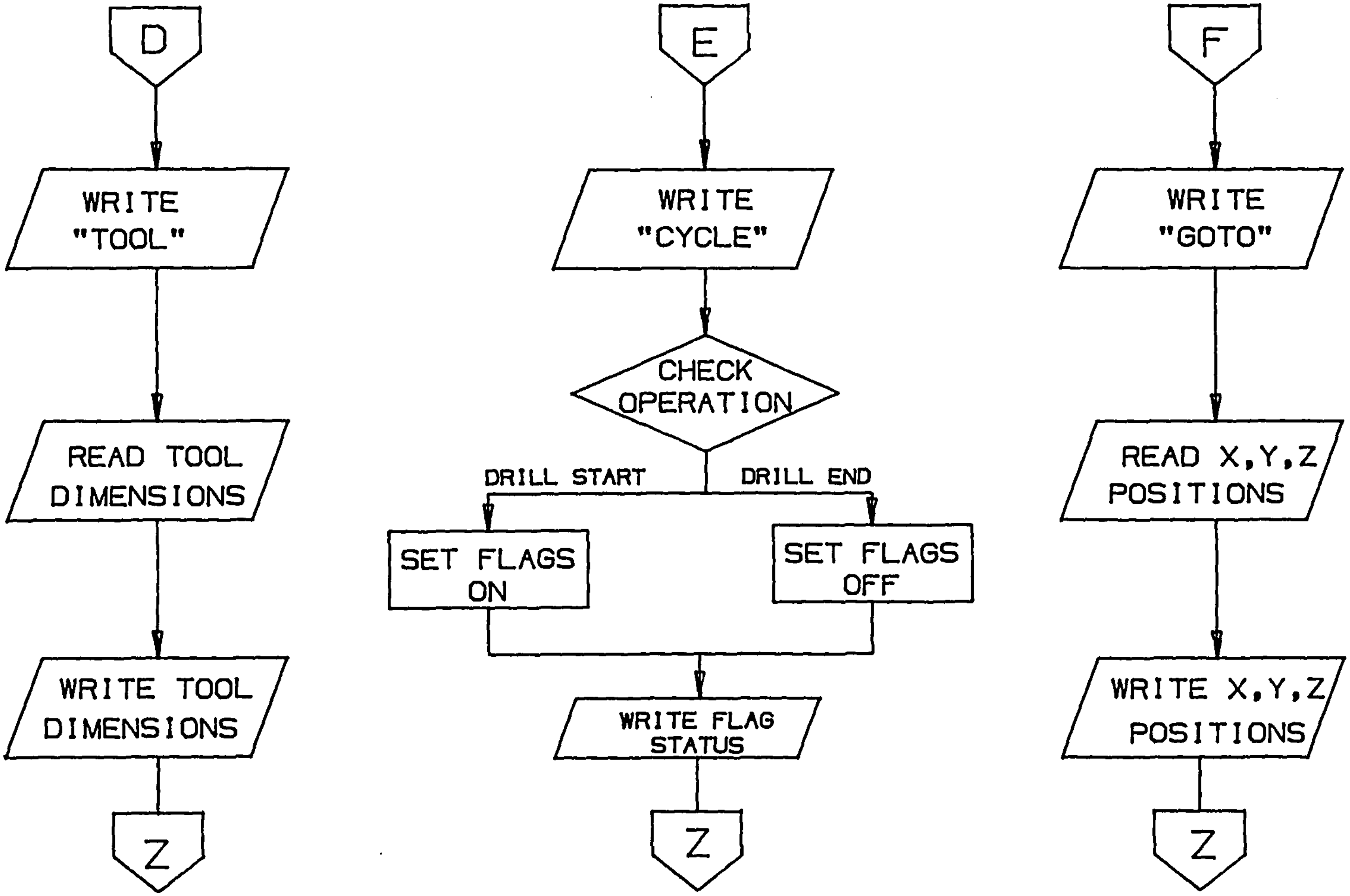


Figure 3.4 (contd.): Transfer Program Flow-Chart

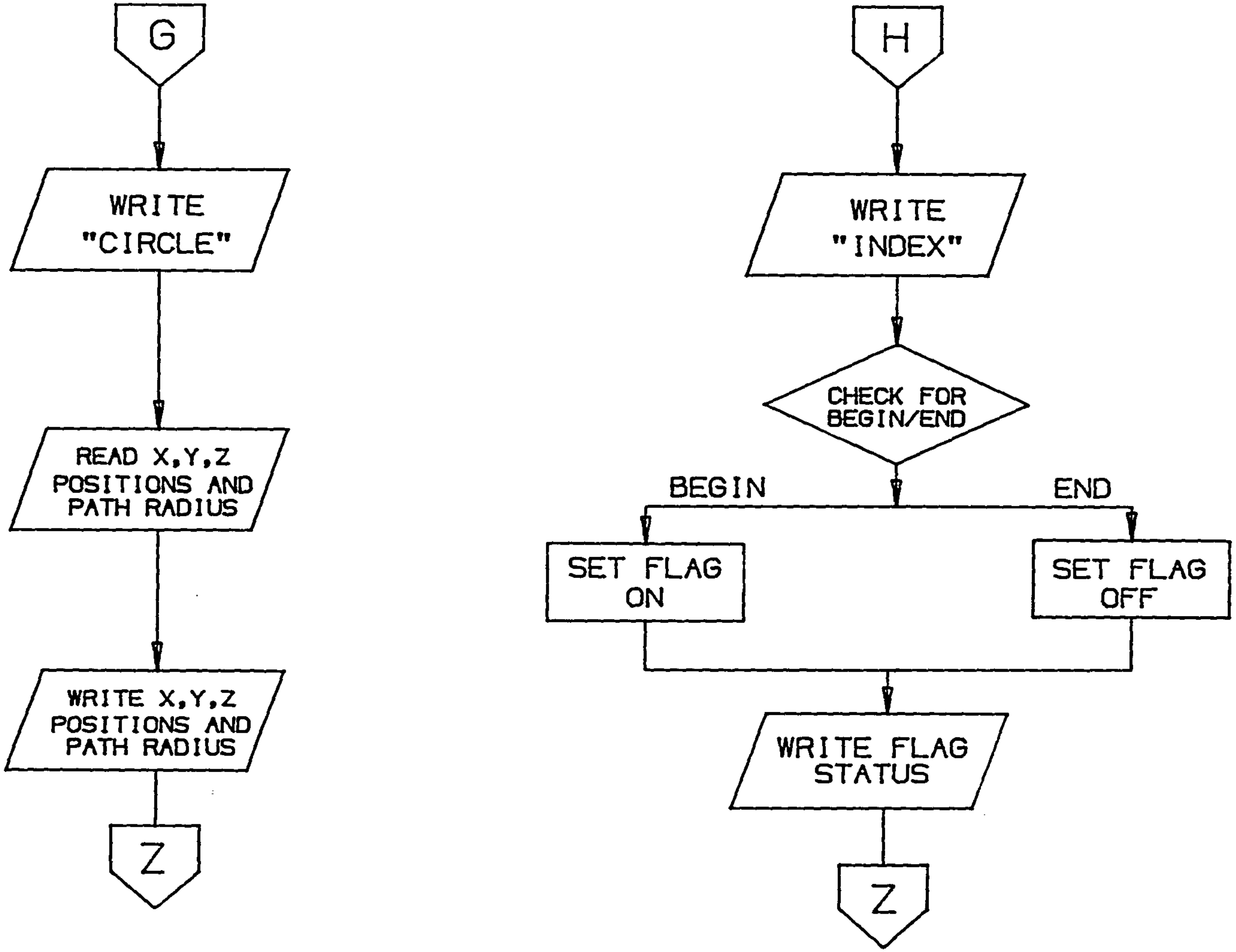


Figure 3.4 (contd.): Transfer Program Flow-Chart

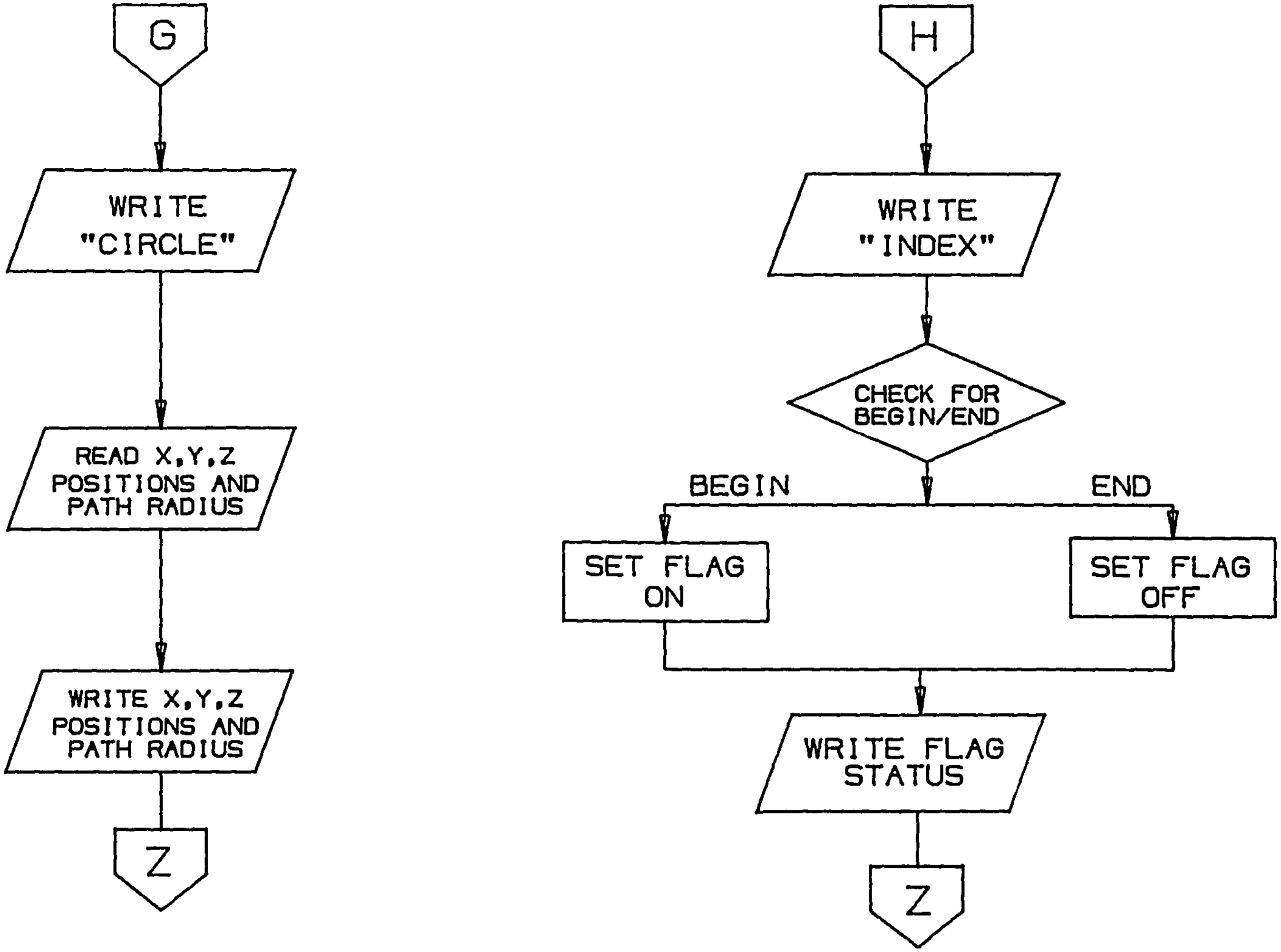


Figure 3.4 (contd.): Transfer Program Flow-Chart

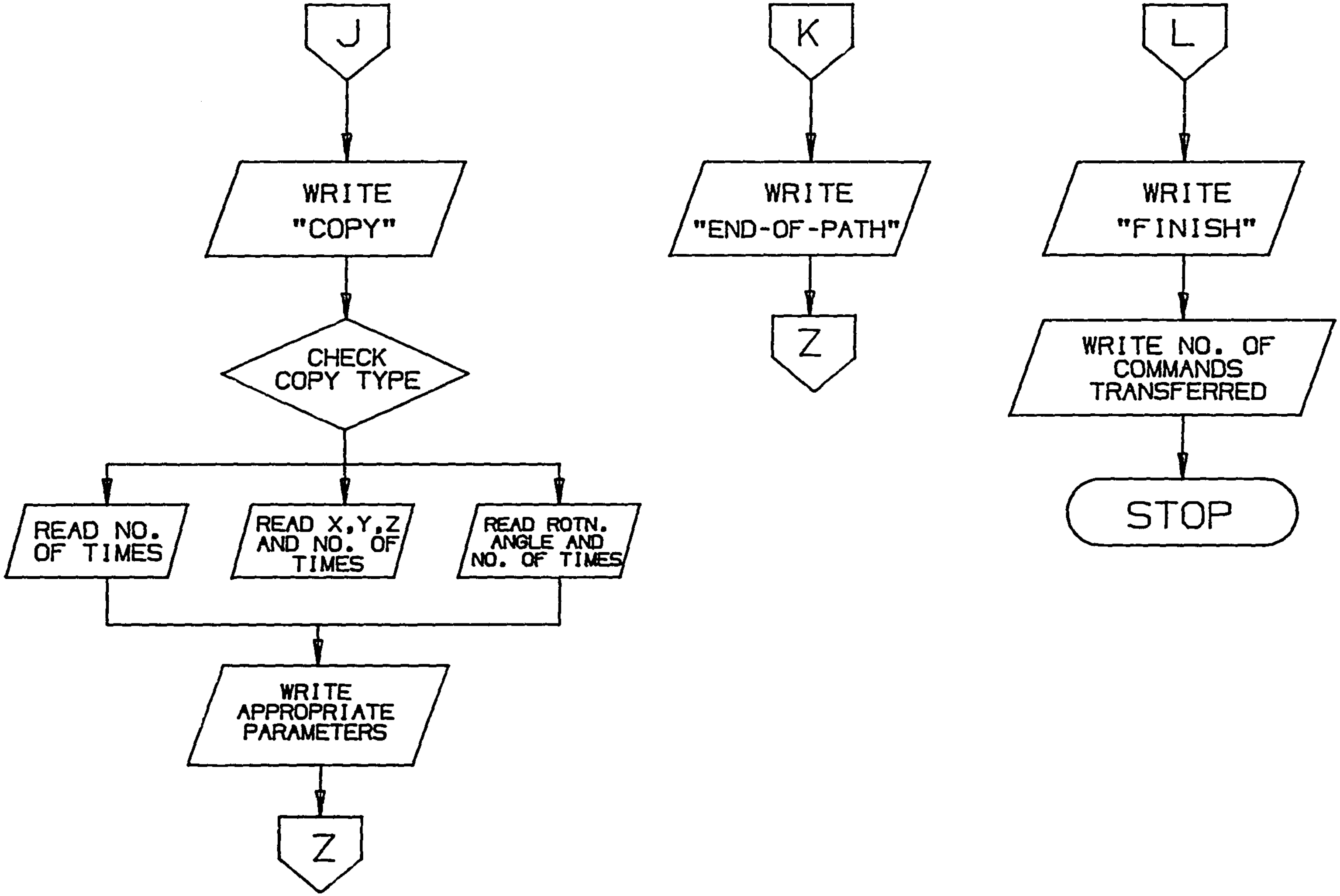


Figure 3.4 (concluded): Transfer Program Flow-Chart

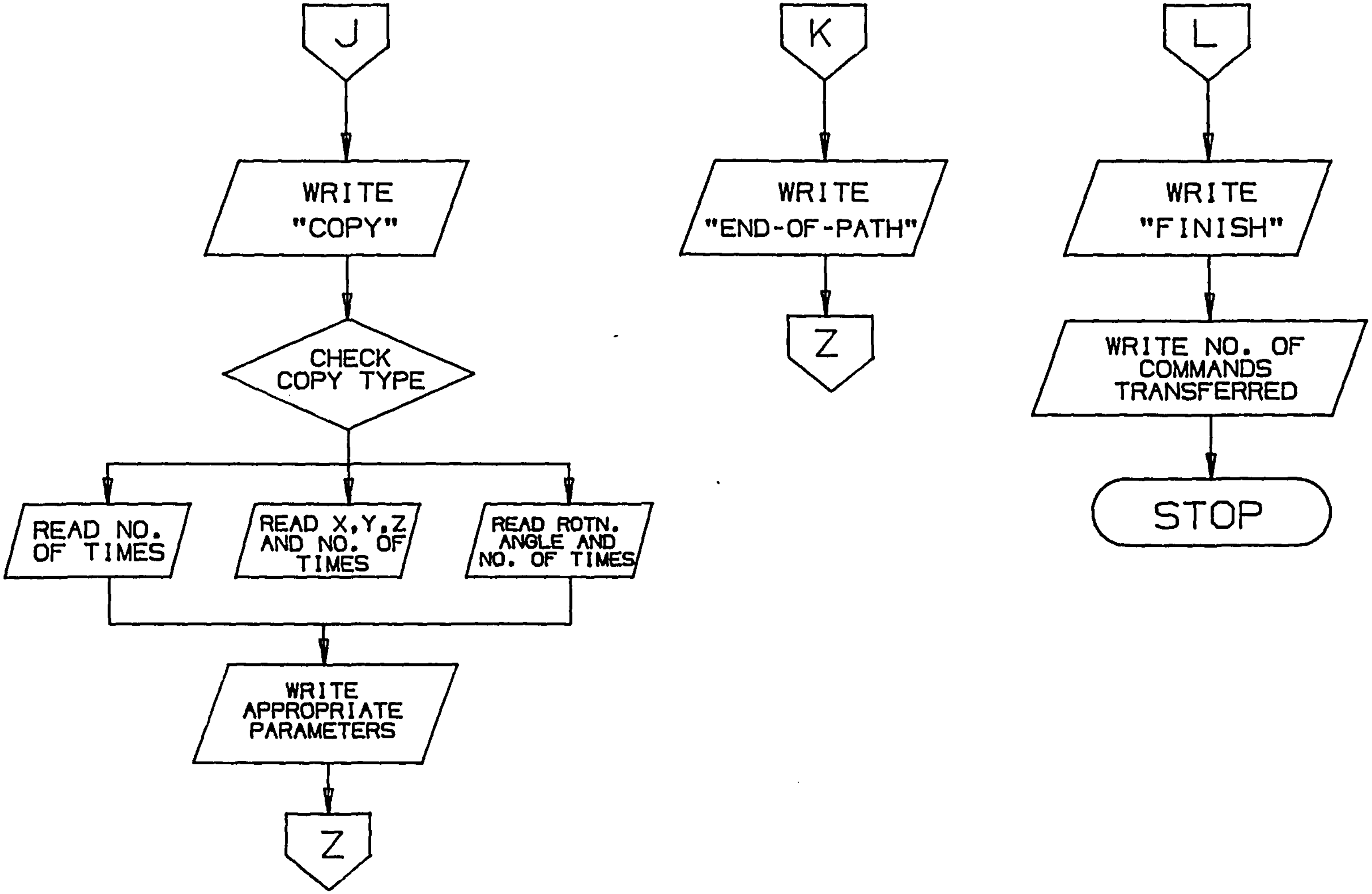
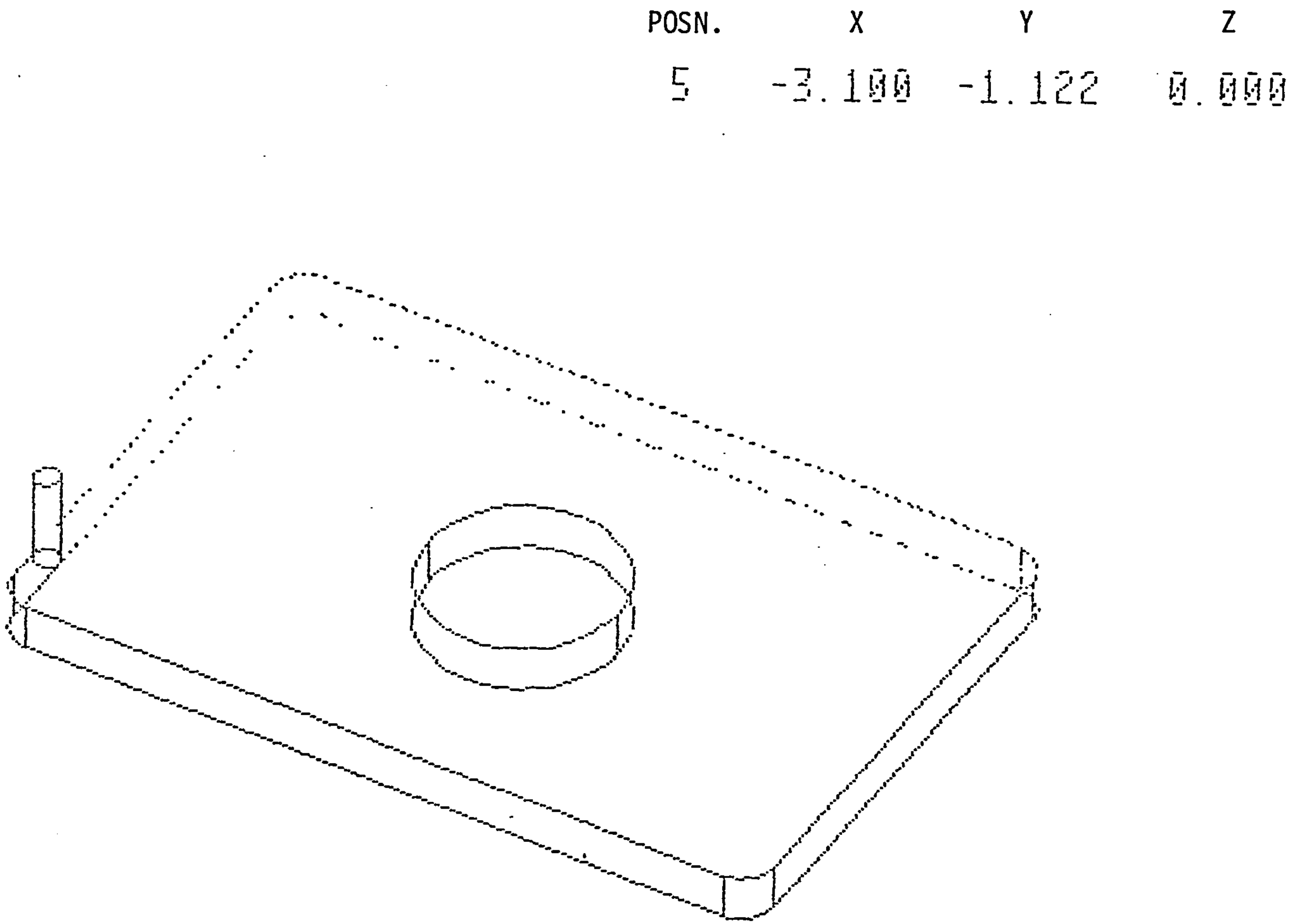


Figure 3.4 (concluded): Transfer Program Flow-Chart

explained in the previous chapter, the hardware and software configuration of the VS11 is quite different from that of the Unigraphics workstations. Most of the machining operations have to be displayed on dualimage memory channel terminals so that the tool in its motion, does not erase the part. Figure 3-5 shows a machining operation on a single channel VS11 graphics system, which indicates how the part boundaries over which the tool traverses are erased.

While Unigraphics has a separate Message Monitor, Program Function Keyboard and Terminal Keyboard for user-interaction, the VS11's have only one keyboard through which all the functions have to be performed. This is where we make use of the "hierarchical dynamic menu", which is displayed on the screen, while the keyboard alone is the user interface.

The Display Format on the screen is shown in Figure 3-6. The display work area contains the application output and is the entire displayable area of the display. The Dynamic Menu Area is the portion of the display where the user is presented with, and selects the various menus to control the aplications programs. If the menu obstructs the displayed picture, this menu display area can be cleared by the execution of a single command from the main menu. Also in this area, the system prompts the user for various types of input , such as files to be accepted and displayed, etc. The Message Area contains



POSN.	X	Y	Z
5	-3.100	-1.122	0.000

Figure 3.5: Machining on Single Channel Terminals

POSN.	X	Y	Z
5	-3.100	-1.122	0.000

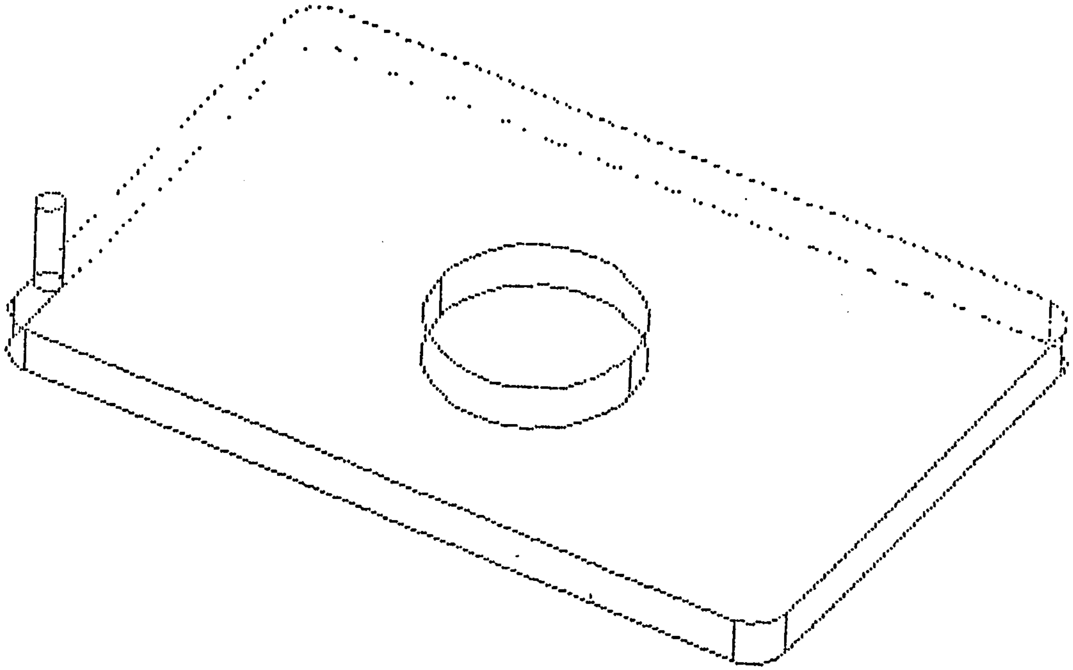


Figure 3.5: Machining on Single Channel Terminals

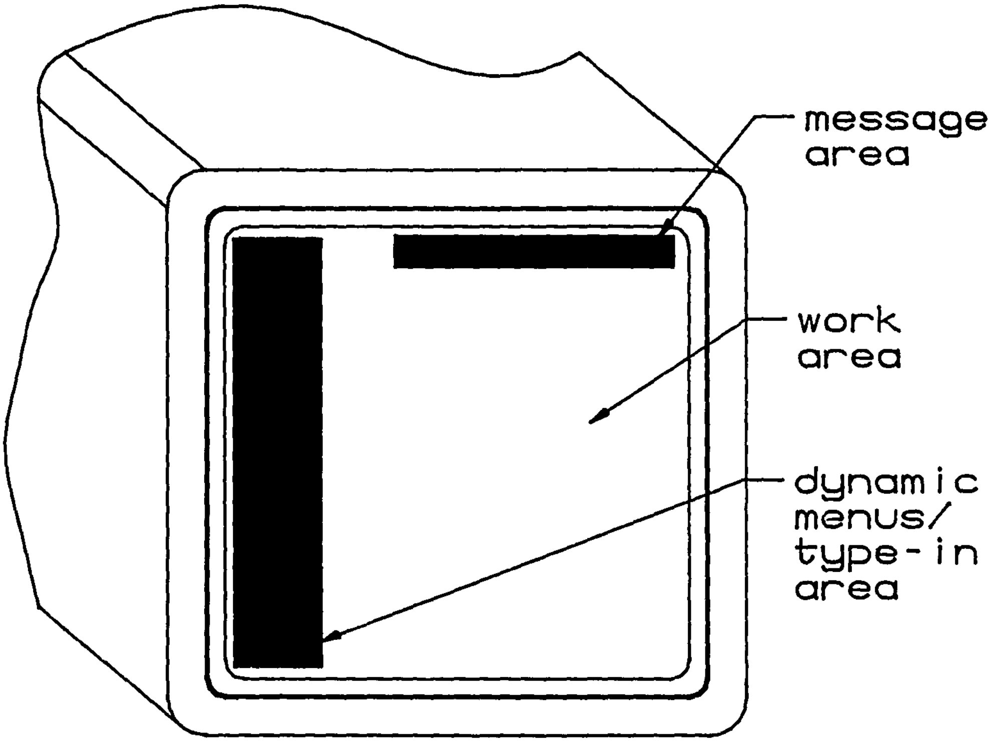


Figure 3.6: Display Format

information of the tool position at any instant during its motion. This is only "ON" during the time that the tool is milling the part. The user can stop the program at any time to check the tool position and alter it if desired. This is accomplished by pressing the interrupt switch on the joystick assembly.

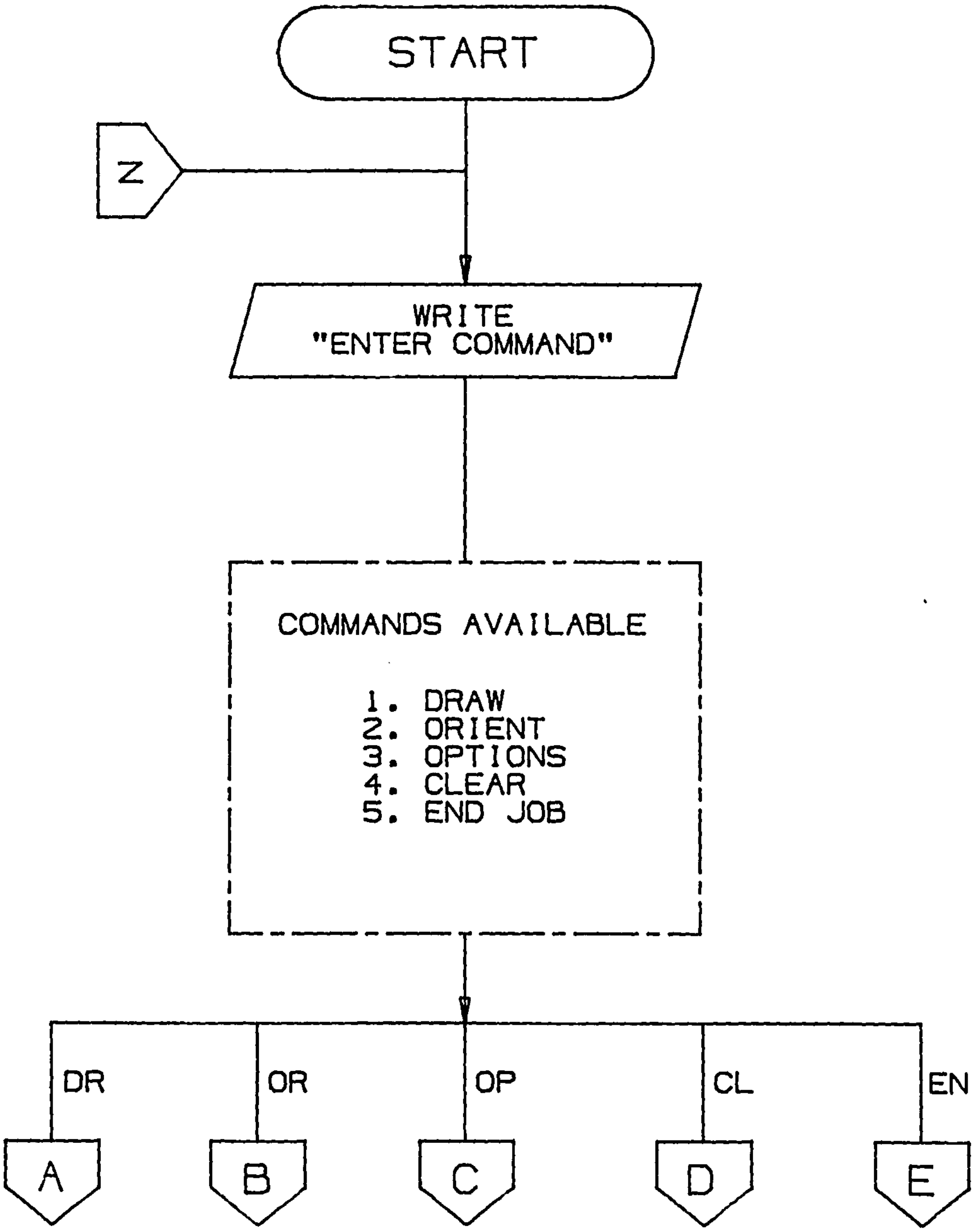
The interactive programs in Fortran, TOOL TRANSFER and TOOL Display, have been designed to recover data from the unformatted files, display the part and tool motions, and provide the user with the options of tool trace, tool erase, tool display, and rapid or slow tool motions. The program TOOL-DISPLAY is described by the flow-chart of Figure 3-7. This program reads the data, one record at a time, from the data file transferred by the TOOL-TRANSFER program, manipulates and displays it. Then it goes again to the file to read the next record. Since the data recovered from the transfer file is not stored in any arrays, the length of the CL-Source File which can be transferred is essentially unlimited -- depending only on the memory space available in the user's directory.

Once the program is executed it enters the interactive phase where the user has to input commands, by responding to the displayed menu, to perform a certain operation. The main or the sub-menus do not appear on the screen immediately, but can be called by hitting 'M' on the terminal keyboard followed by a carriage return. This does not crowd the screen with too many characters.

Another advantage of this is that experienced users can operate such menus very rapidly; with continued use they learn the contents and layout of each sub-menu and can quickly make a selection without having to print the menus every time. The main menu consists of six options and the options have a maximum of three levels.

The commands in the menus consist of mnemonic names and the user responds to these by typing only the first two alphabets of each command on the keyboard, followed by a carriage return. The system accepts this and depending upon the menu-level the user is on, it either goes to a sub-menu, or executes the command and displays the appropriate graphics. If at any time the user wishes to step back to the previous menu-level, it can be done by just typing an exclamation mark, '!', followed by a carriage return. The exact commands in the menus and there operation is explained clearly in the flow-chart of the display program in Figure3-7.

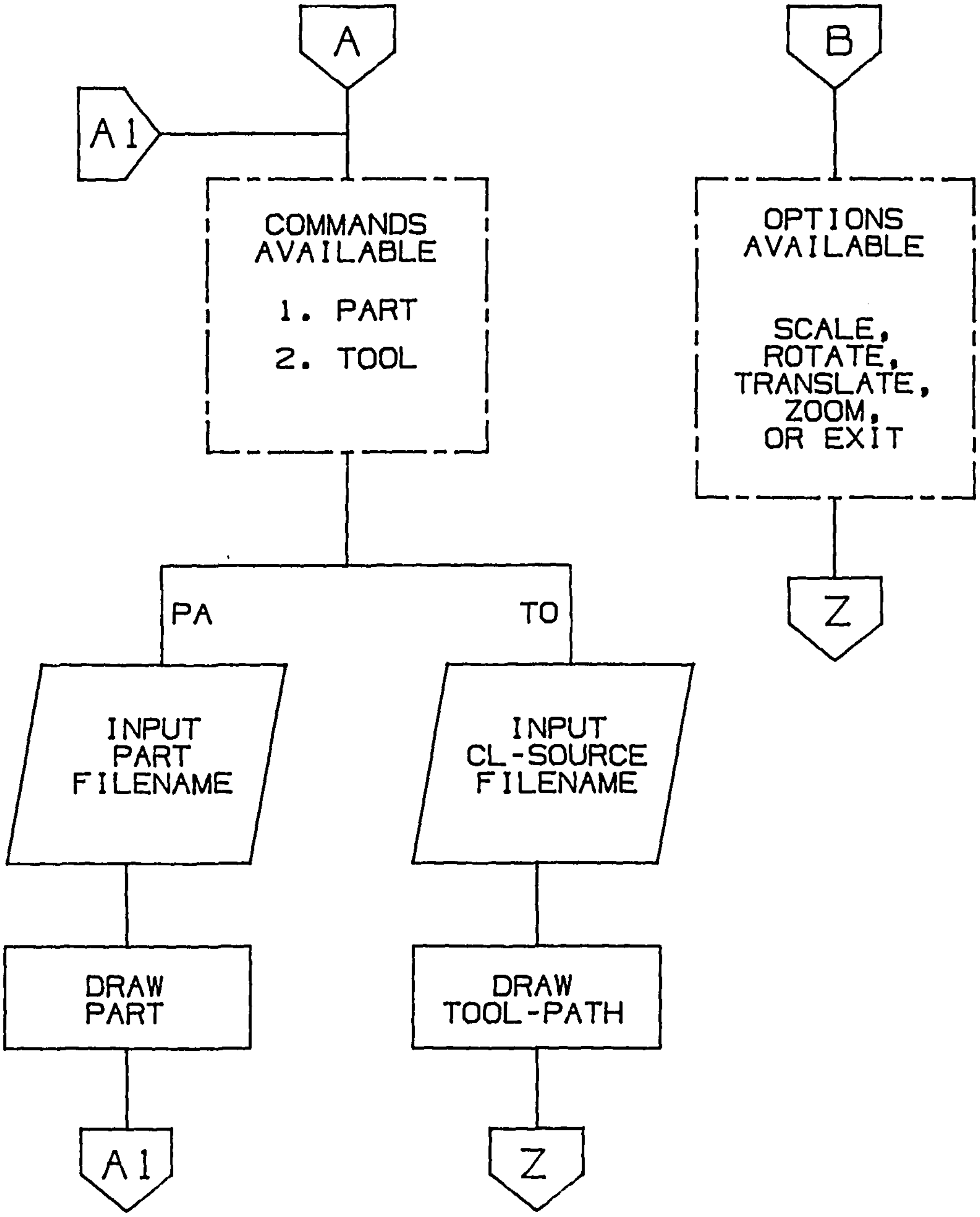
The program, TOOL DISPLAY, gives the user the option to change colors of any entity or the displayed tool. The part can be oriented in any direction and the tool motion displayed. The user also has the option to translate the part anywhere on the screen, scale, rotate, or zoom in on any portion for a closer look. Options available for the tool motion are two speeds -- a rapid or slow -- and are selectable from the menu. These speeds are for display only and have no relation to the



NOTE: At any stage of the program:

- 'M' lists the commands available at that menu-level
- '!' takes the user to the previous menu-level or the main menu

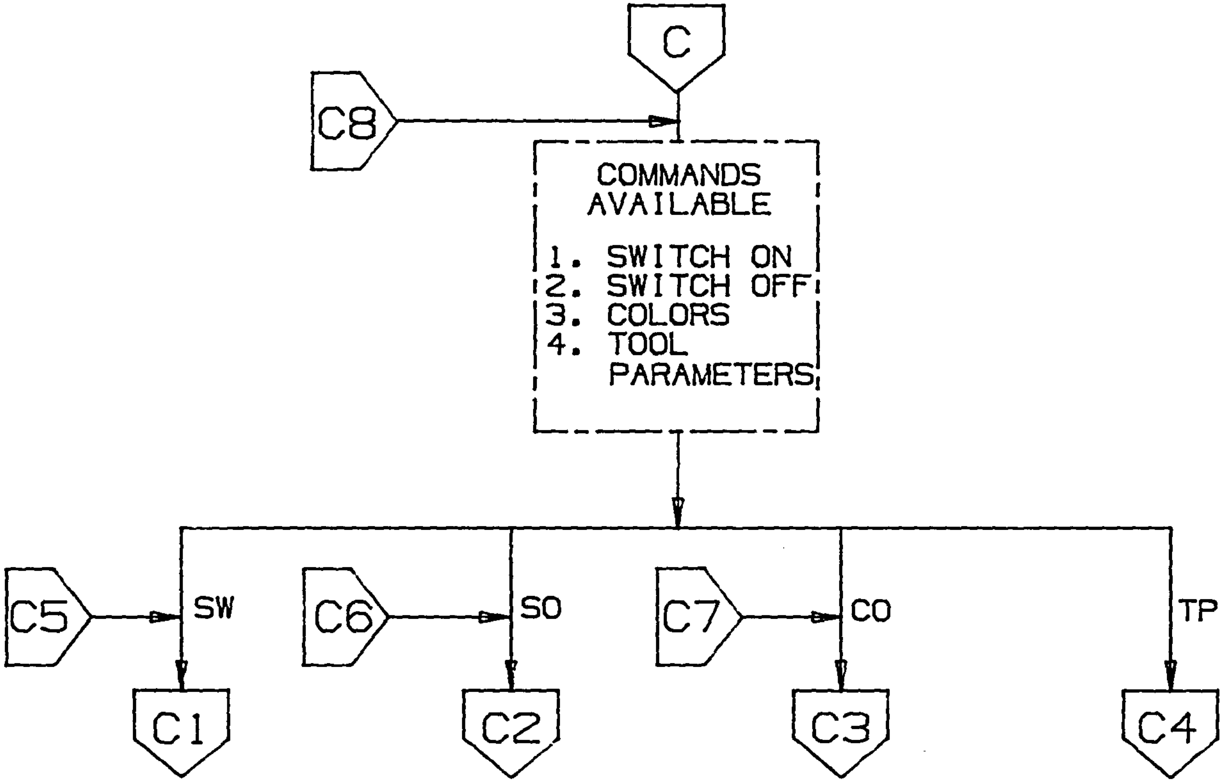
Figure 3.7: Display Program Flow-Chart



NOTE: At any stage of the program:

- 'M' lists the commands available at that menu-level
- '!' takes the user to the previous menu-level or the main menu

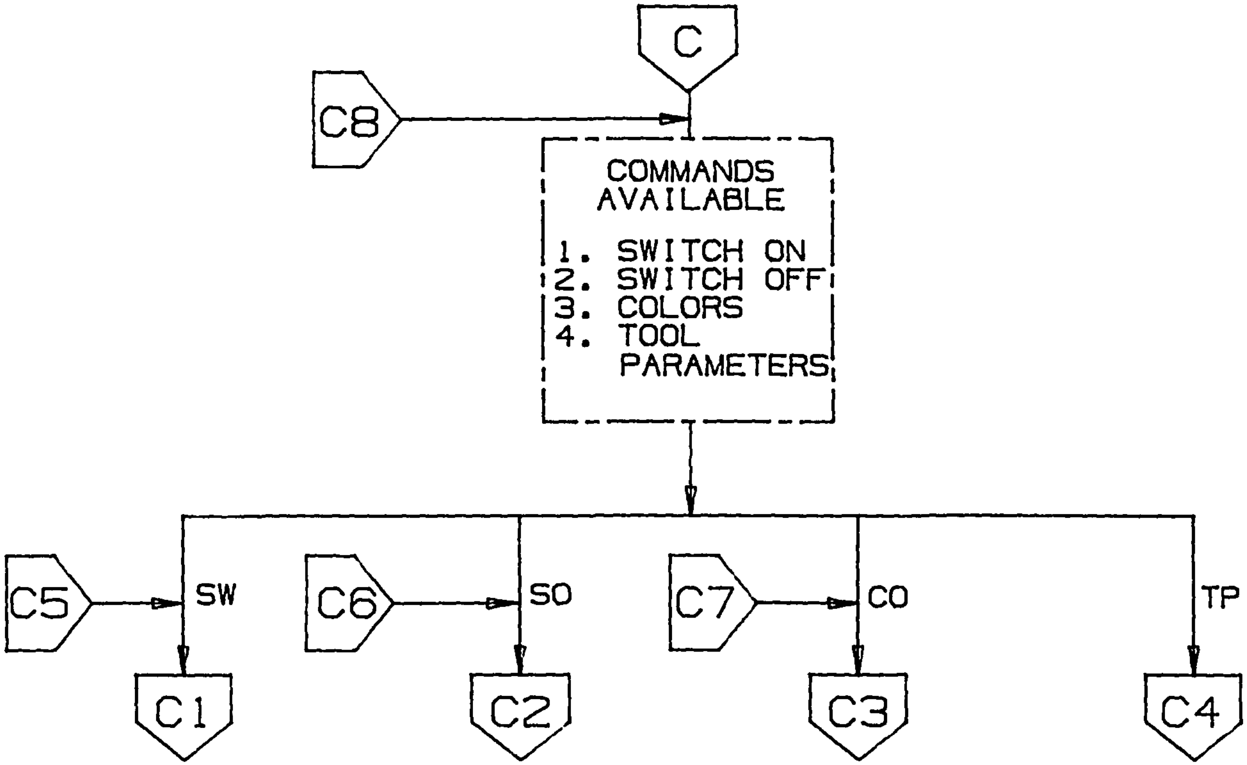
Figure 3.7 (Contd.): Display Program Flow-Chart



NOTE: At any stage of the program:

- 'M' lists the commands available at that menu-level
- 'I' takes the user to the previous menu-level or the main menu

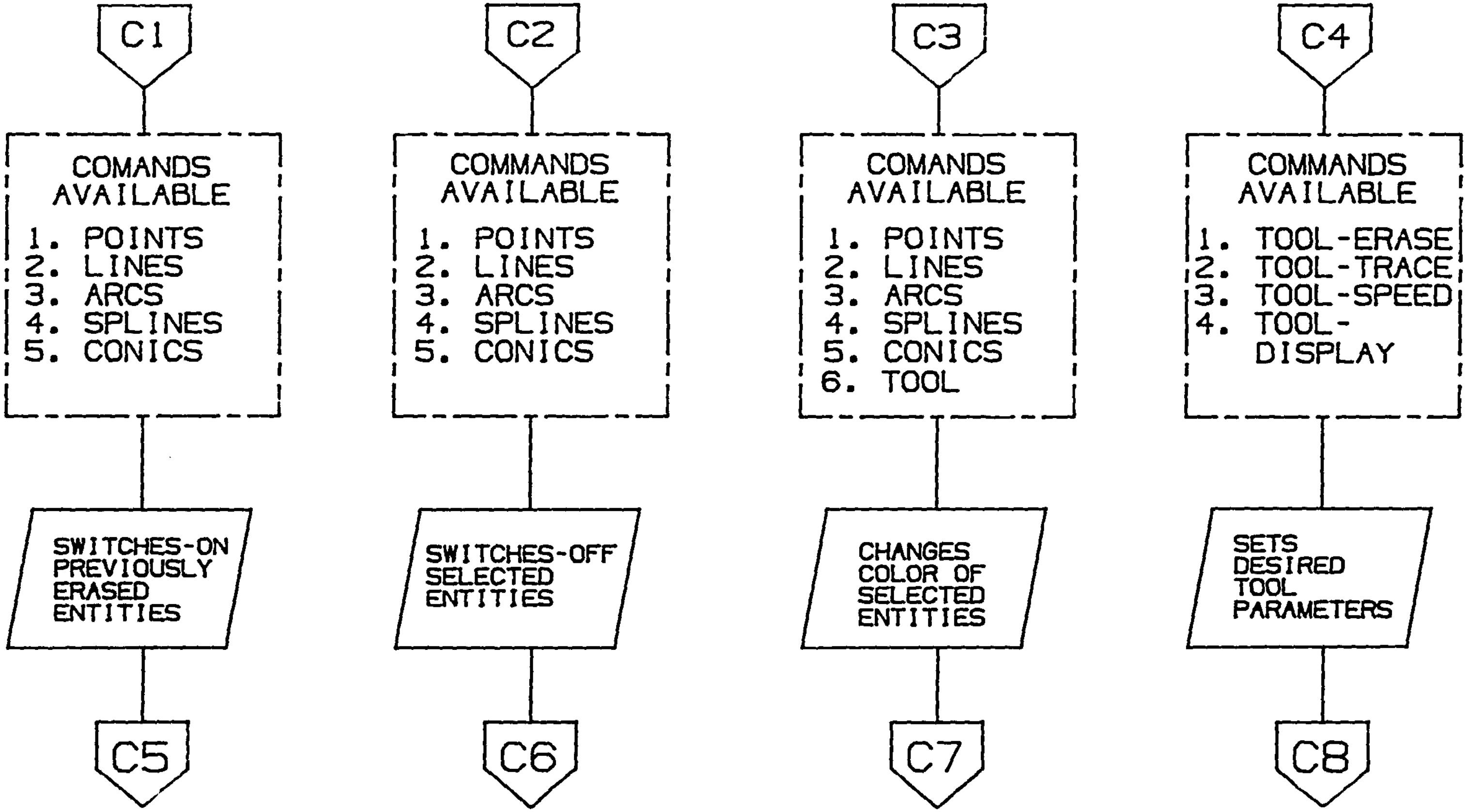
Figure 3.7 (contd.): Display Program Flow-Chart



NOTE: At any stage of the program:

- 'M' lists the commands available at that menu-level
- '!' takes the user to the previous menu-level or the main menu

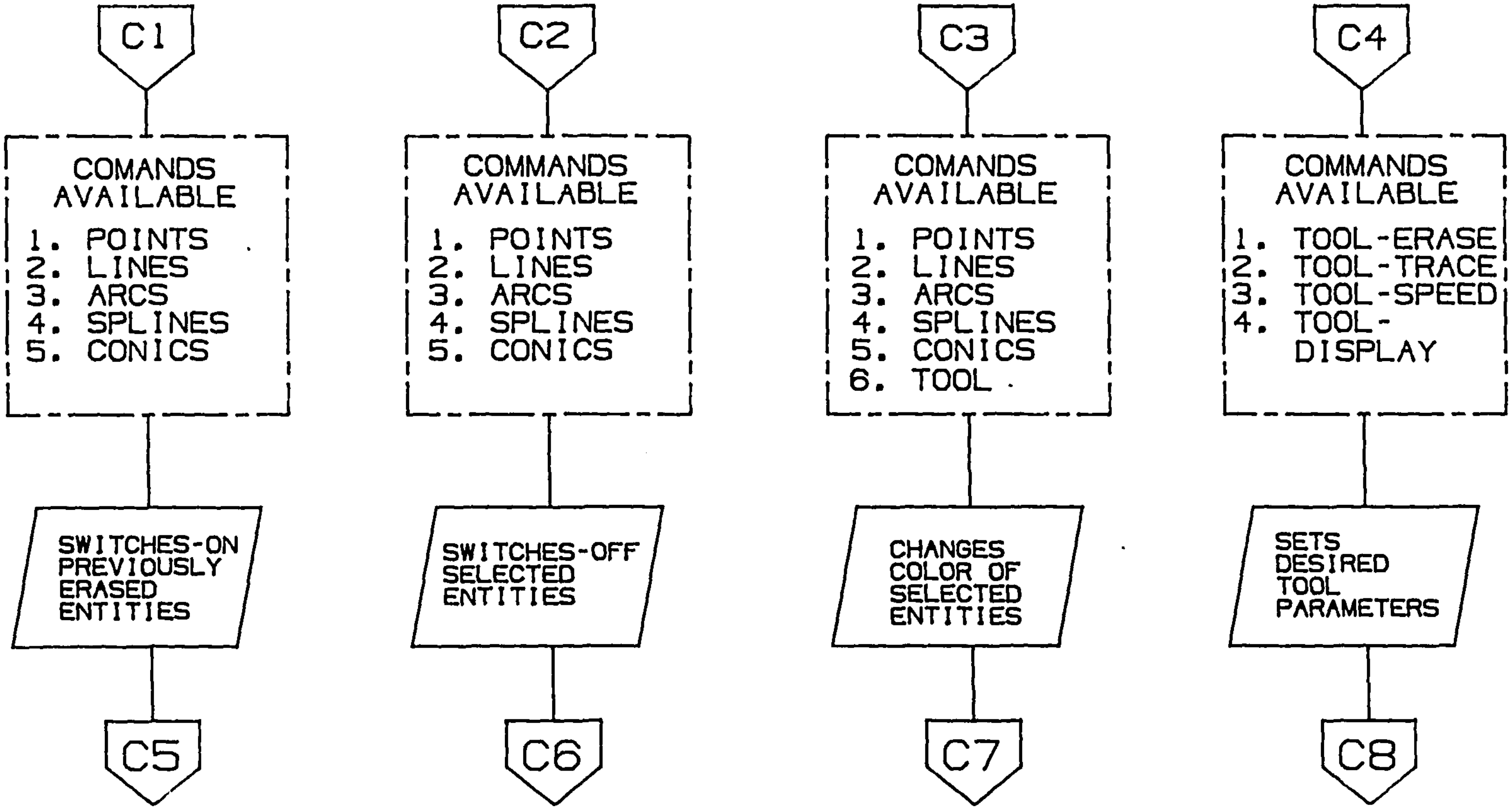
Figure 3.7 (contd.): Display Program Flow-Chart



NOTE: At any stage of the program:

- 'M' lists the commands available at that menu-level
- '!' takes the user to the previous menu-level or the main menu

Figure 3.7 (contd.): Display Program Flow-Chart



NOTE: At any stage of the program:

- 'M' lists the commands available at that menu-level
- '!' takes the user to the previous menu-level or the main menu

Figure 3.7 (contd.): Display Program Flow-Chart

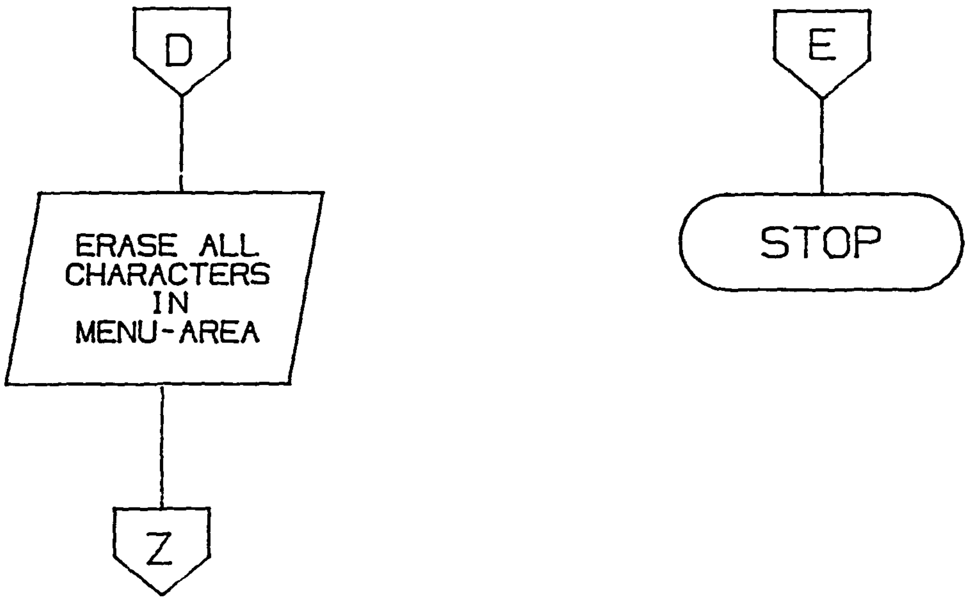


Figure 3.7 (concluded): Display Program Flow-Chart

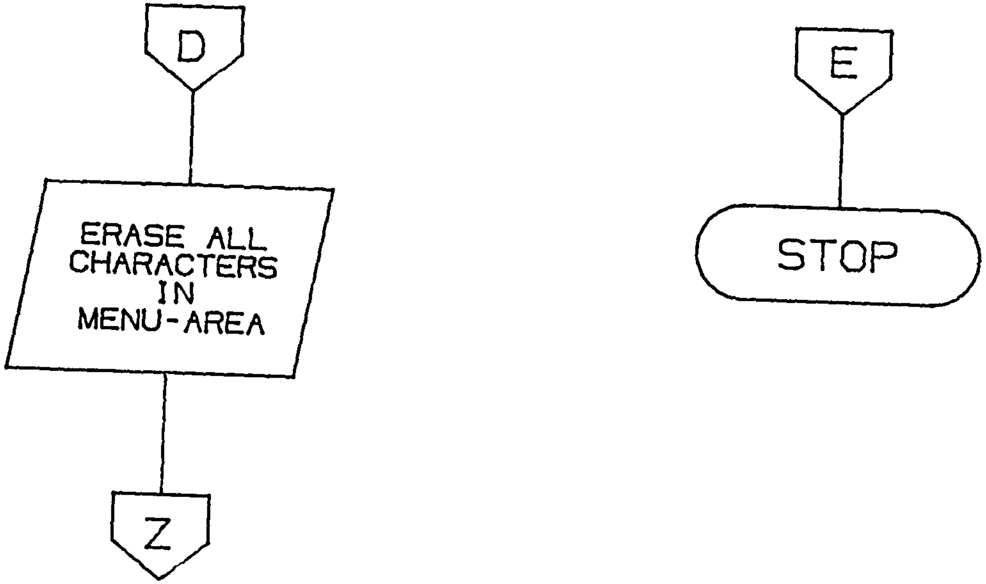


Figure 3.7 (concluded): Display Program Flow-Chart

actual feed and speeds of the tool. It is left to the user to decide if the tool is to be displayed and not erased, or have the tool erased from one position as soon as it moves to the next. Also available is an option which provides a tool-trace along the center-line of the tool. As the tool moves along the part to mill it, and if the "TOOL TRACE" option is set, the path along which the tool travels is traced. For parts on which a great deal of machining work has to be done, the display of the tool would obscure the part, and therefore only the tool path is displayed.

A complete detail of the system operation and its effectiveness is provided by a case study in the following chapter.

CHAPTER 4

CASE STUDY

This chapter illustrates the user-interface and the display advantages achieved by using the VS11 workstation for display of machining operations which have been simulated by the Mill Module in Unigraphics.

A part has been designed on the Unigraphics CAD/CAM system. The manufacturing operations have to be simulated on this part by profiling it from a metal plate and then machining the pockets within it. The plan and isometric view of the part are first drawn, which are viewed on the Unigraphics terminal, as shown in Figures 4-1 and 4-2. Figure 4-1 is the plan and Figure 4-2 the isometric view of the part. This part can be clamped by the standard clamps which have been designed as a part of the system design.

Machining operations have been simulated on this with the help of the Mill Module on Unigraphics. First the profiling operation has been performed and then the nine inner pockets have been milled. Figure 4-3 shows what the display looks like once machining has been simulated by the storage tube based system.The method in which this has been displayed is that, first the system displays the part, and then the tool performing the milling operations, in the same sequence in which they were done by the user. It is not possible to stop the tool during its motion over the part, to examine interferences. The

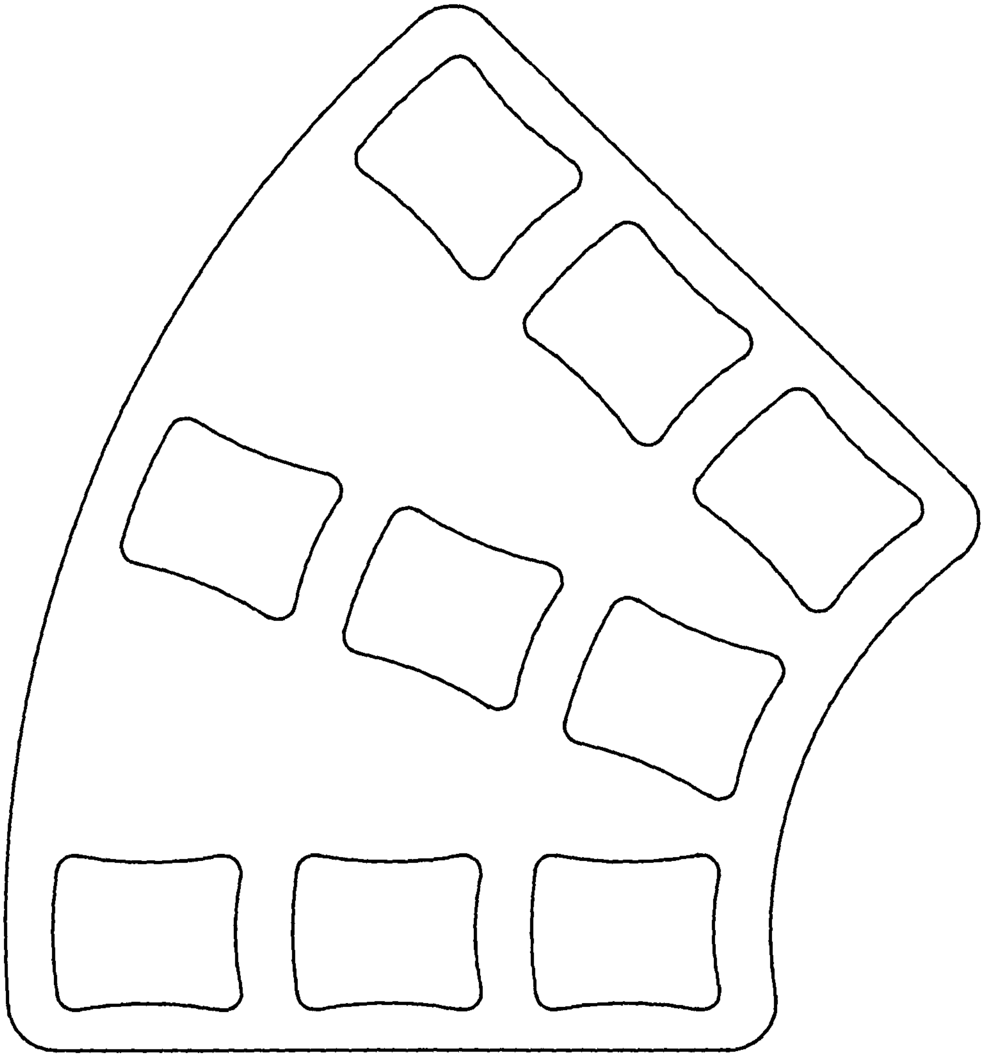


Figure 4.1: Plan View of Part Designed on Unigraphics

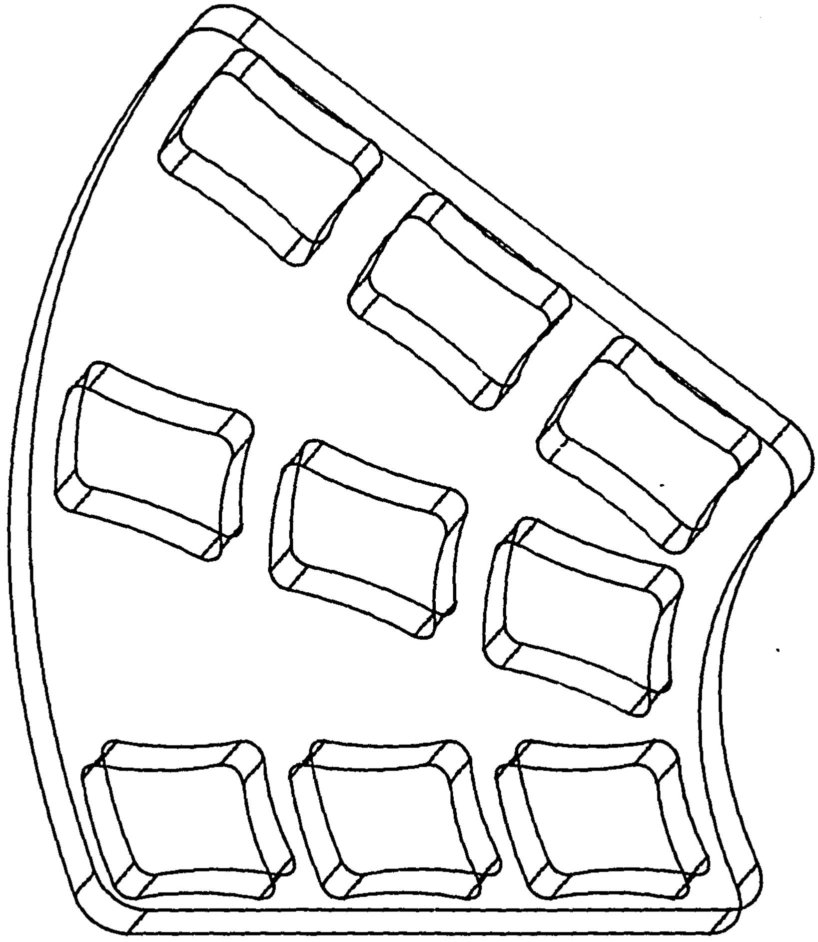


Figure 4.2: Isometric View of Part Designed on Unigraphics

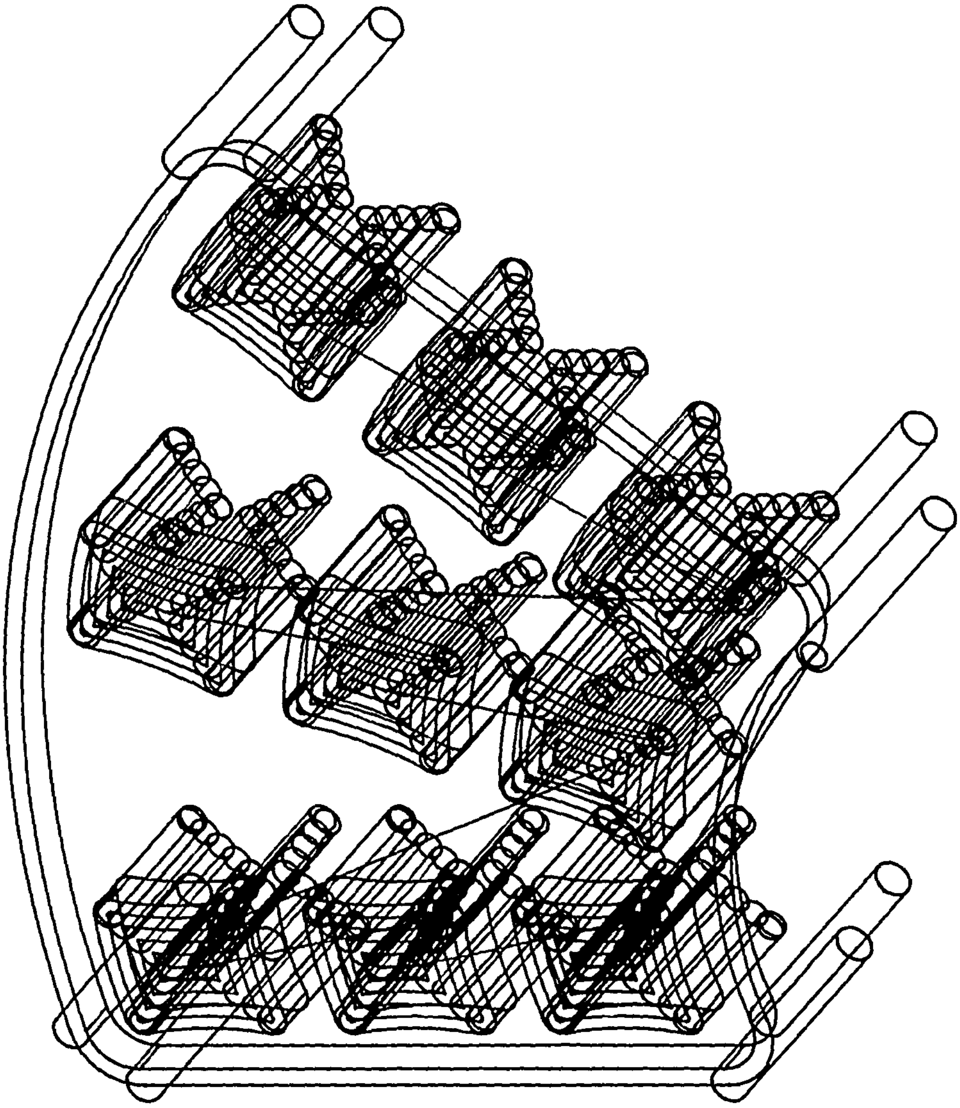


Figure 4.3: Part Machined on Unigraphics

user has to visually keep track of the tool to verify the tool path.

The commands and the procedure to transfer this part to the VS11 system are now discussed. First, the user has to work on the Unigraphics system, to design and draw the part and create a Cutter-Location (CL) Source File by using the N/C Machining function, and file both of these. Assume that a part and CL-Source File, named SAMPLE, have been created and filed (parts and CL-Source Files can have the same name because these are filed in separate libraries) in the Unigraphics system and it is desired to transfer these for display on the VS11 terminals.

As stated previously, this is done in two steps. First the part, and then the CL-Source File is transferred. The commands required to transfer the part are listed and explained below first:

1. `RUN [UGRAF]UGU06`

2. `RUN UGTOVS`

Working on the VS11 system, the user has to type in the first command to run the program, [UGRAF]UGU06. This program transfers the part data files from Unigraphics to the VS11 system. It prompts the user for the name of the part to be transferred, and the new name under which it has to be stored in the VS11 directory, say SAMPLE and SAMPLE.DAT respectively, and performs the desired transfer of the part data file to the VS11. Then the second program, UGTOVS, is run, which reads the part data file and organizes it into a format which is to be read by the display program, TOOL-DISPLAY, and displayed on the VS11. It prompts the user for the name of the part data file in the VS11 directory, that is, SAMPLE.DAT, and also for the name of the new file, say SAMPLE.PAT. This completes the transfer of the part.

The commands required for the transfer and display of the CL-Source File are now listed and explained:

1. RUN the Grip program, CLSF TO GRIP

2. RUN UGXFR

3. RUN TLTRANSFER

Working on the Unigraphics system, the user runs the first program, CLSF TO GRIP, which transfers the CL-Source File to the Grip library. This program prompts the user for the name of the CL-Source File to be transferred, that is, SAMPLE, to the Grip library under the same name, SAMPLE. Now the user has to come on the VS11 workstation and run the second program, UGXFR. The transfer of the GRIP files to the VS11 system is done by this program. It prompts the user for the name of the Grip file to be transferred, that is, SAMPLE, and the name under which it has to be stored in the VS11 directory, say SAMPLE.GRP. The third program, TLTRANSFER, reads the Grip, SAMPLE.GRP, and organises it into a format which is to be read by the display program, TOOL-DISPLAY, for eventual display of the tool. This program asks the user for name of the file to be transferred, that is, SAMPLE.GRP, and the name of the new file created, say SAMPLE.TOL. This completes the transfer of the CL-Source File also.

It should be noted that the actual names of the part and the CL-Source File have not been changed during the process of transfer. Only the extensions to the name have been changed, for example, the Grip file is called SAMPLE.GRP, the tool file is called SAMPLE.TOL, the part file SAMPLE.PAT, etc.

Finally, the program, TOOL-DISPLAY, is run on the VS11 system. This program prompts the user for the names of the part and tool files, that is, SAMPLE.PAT and SAMPLE.TOL, and interacts with the user through menu-driven commands to draw the desired picture. The commands and menus available in this program have been explained in Chapter 3.

Figure 4-4 shows the isometric view of the part discussed before, displayed on the VS11 terminal. This can be oriented in any direction and viewed. It is possible to do such operations as scaling and zooming also.

Figures 4-5 through 4-8 illustrate the machining operation being simulated with different options set. These figures also show the "command input" area on the left and the "message area" in the top right corner. The

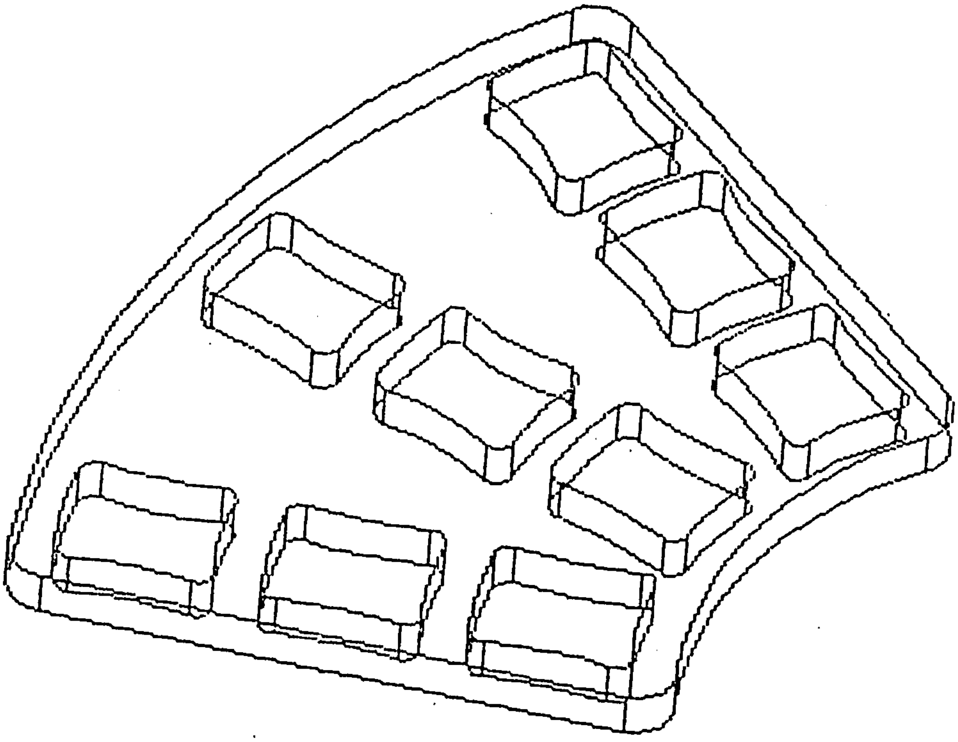


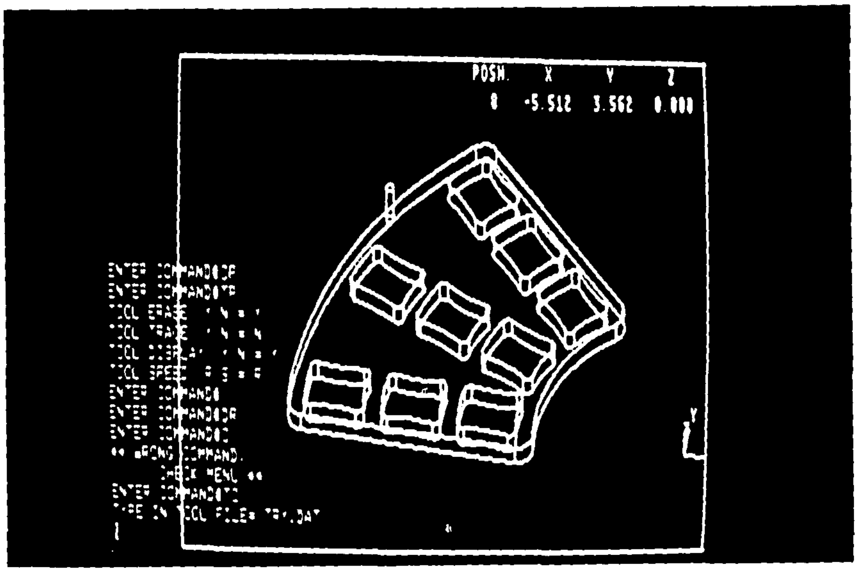
Figure 4-4: Part Displayed on VS11

"message area" contains the position of the tool in the CL-Source File and its location coordinates. The "command area" shows the commands the user has to give to display a particular tool-path.

Figure 4-5 shows the display with the options "TOOL DISPLAY" and "TOOL ERASE" set. Here the tool machines the part and does not leave a tracing line as it performs the operation. Figure 4-6 displays the part and tool-path with only the "TOOL TRACE" option set. In this case the part is oriented differently, and since the tool is not displayed, the user only sees a line drawn by the center-line of the tool as it simulates the machining.

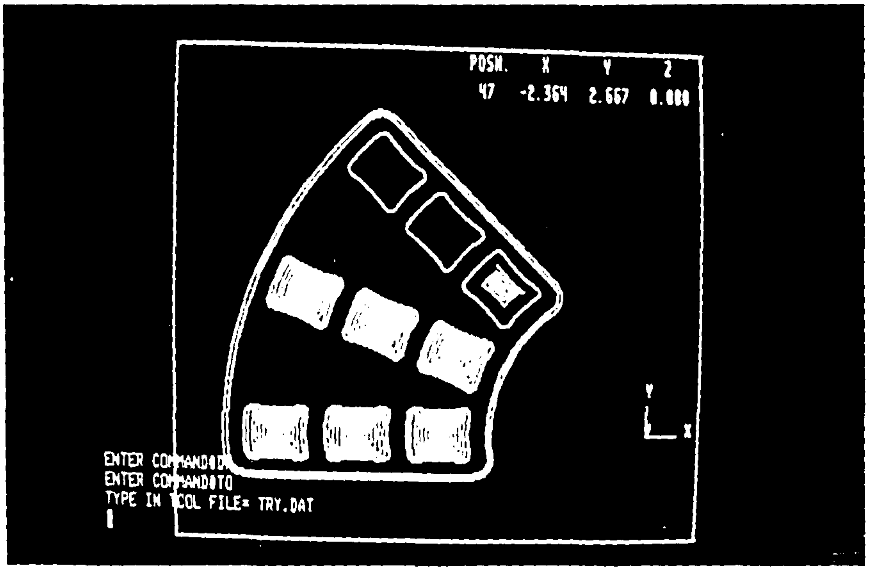
Figure 4-7 displays the tool and the part without the "TOOL ERASE" option being set. As the tool moves around the part, it does not erase its previous positions, resulting in it being displayed at all positions. Figure 4-8 has the same options set as Figure 4-5, but the part has been zoomed in on one specific area for a "closer-look" of the simulated machining operation.

If an error were detected or if a modification were required based on the VS11 display, the X, Y, Z tool co-ordinates can be modified directly in the Grip file in the VS11 directory, and the programs TOOL-TRANSFER and TOOL-DISPLAY run again to display the new tool-path. It should be emphasized that the color capability of the VS11 system could not be shown in this case study because of the inability to present color pictures in the thesis



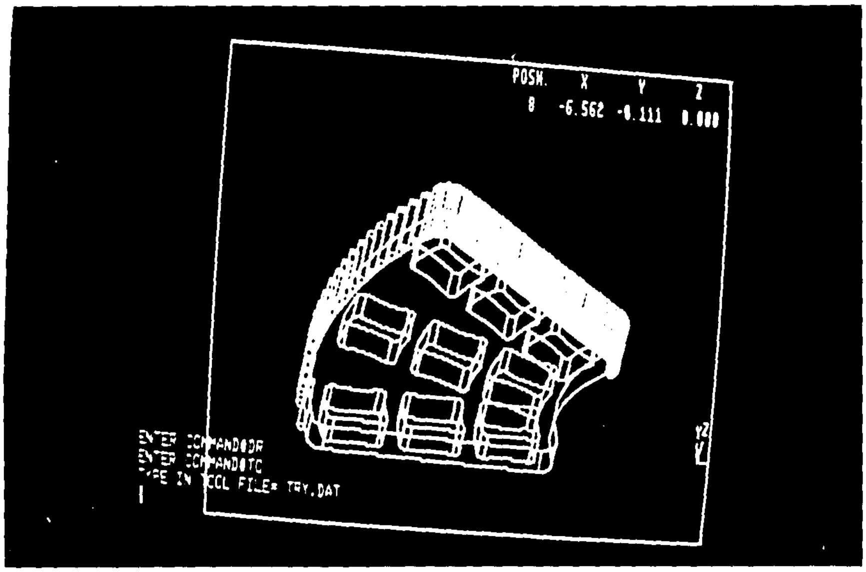
```
POSN.    X       Y      Z
  0   -5.512   3.562  0.000
ENTER COMMAND@DP
ENTER COMMAND@TP
TOOL ERASE ? N = Y
TOOL TRACE ? N = N
TOOL DISPLAY ? N = Y
TOOL SPEED ? S = 5
ENTER COMMAND@
ENTER COMMAND@DR
ENTER COMMAND@D
** WRONG COMMAND.
      CHECK MENU **
ENTER COMMAND@TO
TYPE IN TOOL FILE= TRY.DAT
```

Figure 4-5: Tool-Path with options 'TOOL-DISPLAY' and 'TOOL ERASE'



```
POSN.    X       Y      Z
 47   -2.364   2.667  0.000
ENTER COMMAND@D
ENTER COMMAND@TO
TYPE IN TOOL FILE= TRY.DAT
```

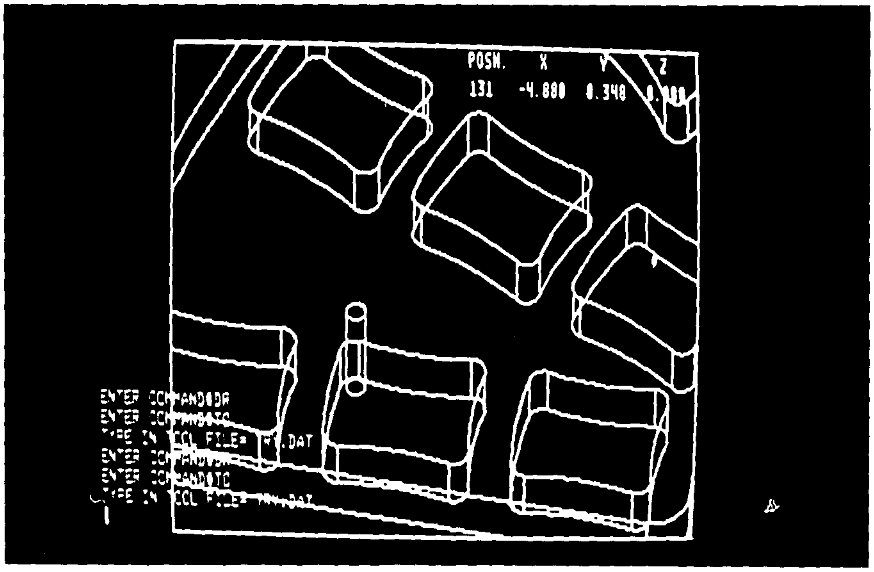
Figure 4-6: Tool-Path with 'TOOL-TRACE' option



```
POSN.    X       Y      Z
  8   -6.562  -0.111  0.000
```

```
ENTER COMMAND@DR
ENTER COMMAND@TC
TYPE IN CCL FILE= TRY.DAT
```

Figure 4-7: Tool-Path without 'TOOL-ERASE' option



```
POSN.    X       Y      Z
 131  -4.880   0.348  0.000
```

```
ENTER COMMAND@DR
ENTER COMMAND@TC
TYPE IN CCL FILE= TRY.DAT
ENTER COMMAND@DR
ENTER COMMAND@TC
TYPE IN CCL FILE= TRY.DAT
```

Figure 4-8: Tool-Path with 'TOOL-TRACE' option and ZOOM text. There is a significant loss of impact in the figures because color and motion cannot be seen.

CHAPTER 5

CONCLUSIONS

5.1 Summary and Conclusions

The first part of this thesis discussed the current approach to integration of CAD/CAM systems. The different functions of a CAD/CAM system have been described. It has been stressed that as separate functions are becoming more numerous in an integrated CAD/CAM environment, users become aware of growing inefficiencies: namely, it is troublesome to learn the peculiarities of different systems, and the manual extraction of data from one system to another is a source of errors. The existing problem for this study has been defined as the integration of two systems to provide dynamic, real-time, interactive display of tool-motion on numerically-controlled milling machines. The approach taken to solve this problem has considered factors such as: system interfacing, display structure, and user interaction.

In the second part, the existing system has been modelled. The hardware, software, and the two graphics systems available have been described. The Unigraphics CAD/CAM system is first detailed, and the different modules available in Unigraphics and their capabilities have been explained. The current method of designing a part and machining it with the Mill Module is described.

Although the resolution on the storage-tube type of terminals, on which Unigraphics currently runs, is very good, there are several disadvantages to it. Pictures of a single color and line intensity only can be displayed. Selective erasure is not possible and dynamic graphics cannot be run. During machining of complex parts, the lines drawn by the tool obscure the part, so that by the time the machining operation is complete, it is almost impossible to distinguish the part from among the maze of lines on the screen.

The VS11 terminals are of the raster-scan type. The basic components of the VS11 graphics system are: a high-speed display processor, image memory modules and a sync generator/cursor control. This system is capable of displaying 16 different colors. Because of the system hardware and software characteristics, pictures can be displayed dynamically, and features such as rotation, translation, and scaling can be performed. During the machining of a part it is simpler to visually inspect the display to detect interferences of the machined part with the tool, clamps, or fixtures.

The characteristics of the designed system are discussed in section three of this thesis. The system works in an interactive mode. The man/machine interface is accomplished by using a user-selectable menu, which appears on the graphics screen and prompts the user to interact in real-time. Cutter-Location Source Files are transferred from Unigraphics, via the Grip source library, to the VS11 directory. The part to be machined is transferred by the use of UFIP sub-routines directly to the VS11 terminals.

An interactive program, TOOL-TRANSFER, then takes the CL-Source File and stores the required information in this file in a neutral format. Another program, TOOL-DISPLAY, then retrieves this data, manipulates it, and displays it on the VS11 monitor.

The VS11 screen is used for messages, command input, and display purposes. The messages displayed are the location co-ordinates of the tool and its position in the CL-Source File at that instant. The method of command display and input is by the use of dynamic menus. The user is provided with options to display the tool, tool trace, tool erase, and rapid or slow tool motions. Also, the user can change the colors of any entity or the tool, orient the part in any direction and then machine it, or do operations like translation, rotation, and scaling of the part.

The final section consists of a case study to support the technical feasibility of the desired system. A part is first designed and machined on the Unigraphics system. The same part is then transferred to the VS11 system and displayed along with the tool motions. Various options available to the user are illustrated by the use of figures. The advantages of the display on the VS11 system are clearly evident. It should be noted that the color capability of the VS11 could not be shown in this case study because of the inability to produce color pictures in the thesis text. However, the advantage of color pictures over black-and-white ones is an obvious fact.

The technical feasibility of using real-time animation techniques in conjunction with interactive graphics technology has been demonstrated for the display of tool motion on numerically-controlled milling machines.

Software that displayed graphics on storage tube terminals has been interfaced to terminals of the raster-scan type, which have the capability for real time graphics, 16-color representation, and multi-channel operation. In doing so, it has been shown how the capabilities of the display monitor can improve the understanding of what is being presented.

Parts designed and manufactured on Unigraphics can now be transferred to and displayed on the VS11 terminals, making use of all the advantages which the VS11 posseses. It is evident that the tool path displayed on the dynamic, real-time color display is much easier to follow. The displayed picture is not obscured by the lines traced by the tool, as on the storage tube. The part can be oriented in any direction and the machining operation performed to get a better view. The software available on VS11 also permits operations such as zooming, translation, scaling, and clipping -- all of which contribute to more effective display.

Since VS11 and Unigraphics are two separate software packages, to interface them data transfer techniques were required. This has been accomplished by first transferring data to a neutral format and then its retrieval by the VS11 system for display. The inherent differences between the two systems necessitated the design of a different display format. This has been achieved by using the screen of the VS11 workstation both for display and messages. The screen has been divided into a message area, command input area, and picture display area. User interaction, the part of the program which determines how the user and the computer communicate, has been given special attention. The user operates the designed system by the use of dynamic hierarchical menus. Menu-selection forms a sort of dialog between the user and the computer and has been found to be a friendly method of system operation.

5.2 Recommendations For Further Study

There are several areas that should be studied in more depth so that such integrations are successful.

The interfaced programs designed to transfer parts and Cutter-Location (CL) Source Files between the Unigraphics and VS11 systems require further study. This interface is a key factor in the success of the system because if users do not like the way they have to interact with the system, they will not use it efficiently. All the different programs -- to transfer the part, and to transfer the CL-Source Files, through Grip, to the VS11 directory, and the transfer and display programs, TOOL-TRANSFER and TOOL-DISPLAY, should be combined into one complete program. This shall make the system compact, simpler, and more user-friendly.

The integrated data-base and its structure needs to be investigated in more depth. The efficiency of the system depends in part on the ease with which data can be accessed and transferred from one format to another. It should be very helpful to have a common, standardized data base, like the IGES format, into which one system should translate its data and another system should retrieve it. This shall ensure that no duplication of effort by different users takes place.

The ultimate system would be to use the VS11 as a Unigraphics workstation also. This would eliminate any need for the transfer or display programs to be designed and for any shuttling of files between the two systems. A great deal of effort is required to achieve this, though, in rewriting all the application modules to take advantage of the features of raster-scan displays that support real-time dynamic motion.

REFERENCES

[1] Carroll, M.P., Gingerich, J.Z., et al.
A Hybrid CAD/CAM Syatem for Mechanical Applications.
Proceedings of the 19th Design Automation Conference , 1982.

[2] *Graphics Multi-Axis Module*
MCAUTO, St. Louis, Mo, 1980.

[3] *Graphics Machining Editor*
MCAUTO, St. Louis, Mo, 1980.

[4] *Graphics Mill Module*
MCAUTO, ST. Louis, Mo, 1980.

[5] *Graphics Interactive Programming*
MCAUTO, St. Louis, Mo, 1979.

[6] *Digital Representation for Communication of Product definition Data*
American Society of Mechanical Engineers, 1982.

[7] Liewald, Michael H. and Kennicott, Philip R.
Intersystem Data Transfer via IGES.
Computer Graphics and Applications , 1982.

[8] *Unigraphics User's Manual*
MCAUTO, St. Louis, Mo, 1980.

[9] Meyers, Ware.
CAD/CAM: The need for a broader focus.
Computer , 1982.

[10] Newman, William M. and Sproull, Robert F.
Principles of Interactive Computer Graphics.
McGraw Hill, 1979.

[11] Ozsoy, Tulga.
VS113D Graphics Package.
Technical Report, Lehigh University, Computer-Aided Design Lab., 1982.

[12] Summer, Randi and Ozsoy, Tulga.
VS113D Example Manual.
Technical Report, Lehigh University, Computer-Aided Design Lab., 1982.

[13] Lynne A. Price.
Design of Command Menus for CAD Systems.
Proceeding of the 19th Design Automation Conference , 1982.

[14] Donald Robbins.
Making the wire frame solid.
Proceedings of the 19th Design Automation Conference , 1982.

[15] George Schaffer.

Computer Graphics goes to Work.
American Machinist , 1980.

[16] George Schaffer.
Computers in Manufacturing.
American Machinist , 1978.

[17] Bhalla, Sunil.
Unigraphics/VS11 Mill Module Interface.
Technical Report, Lehigh university, Computer-Aided Design Lab., 1982.

[18] Unigraphics/Files-11 Interface Package
MCAUTO, St. Louis, Mo, 1980.

[19] Stig Ulfsby, Steiner Meen, et al.
Tornado: A Data Base Management System for Graphics applications.
Computer Graphics and Applications , 1982.

[20] Unigraphics Programming Manual
MCAUTO, St. Louis, Mo, 1979.

[21] VAX Technical Summary
Digital Equipment Corporation, Maynard, MA, 1981.

[22] VS11 Installation Guide
Digital Equipment Corporation, Maynard, MA, 1979.

APPENDIX A
GLOSSARY

ALPHANUMERIC KEYBOARD: The typewriter keyboard on the console on which the user keys in data to the system.

ASCII: AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

AXIS: A general direction of relative motion between the cutting tool and the workpiece.

BUFFER STORAGE: A place for storing information in either a computer or control unit, so that it is immediately available for action once the previous instructions have been completed.

CAD/CAM: Computer Aided Design/Computer Aided Manufacturing. CAD is the process which uses the computer to assist in the creation or modification of a design. CAM is the process employing computer technology to manage and control the operations of a manufacturing facility.

CATHODE RAY TUBE: CRT; an electron tube whose face is covered with a phosphor that emits light when energized by its electron beam.

CENTRAL PROCESSING UNIT: CPU; A unit of the computer that includes the circuits controlling the interpretation and execution of instructions.

CLAMP: A device used to secure a workpiece to the machine tool table.

CL FILE: Cutter-Location File.

CLIPPING: Removing parts of display elements that lie outside physical bounds (e.g., a viewport or a display surface) or logical bounds (e.g., a window).

CUTTER PATH: The ordered combination of vectors, typically a combination of entry, cut, return, stepover, and retract trajectories.

DATA BASE: A collection of related data attributes that makes the same data acceptable by more than one program, providing for integration or sharing of common data and unleashing the program from the constraints of hardware or data.

DEVICE DRIVER: Software which converts device-independent graphic commands into device-specific display orders.

DIRECT VIEW STORAGE TUBE: A storage tube whose display is directly visible on the display surface.

DISK STORAGE: Storage on direct-access devices that record data magnetically on rotating disks.

DISPLAY PROCESSOR: A microprogrammed processor that interprets and processes operator commands and generates and manipulates the graphics displays.

FEEDBACK: The return of part of the output of a machine, process, or system to the computer as input for another phase, especially for self-correcting purposes.

FEED RATE: Tool rate of movement; At a safe distance away from the part (RAPID), while engaging the part (FEED), while cutting (CUT), and while retracting from the part (RETRCT).

FLOW CHART: Used to symbolically state and diagram the steps to be followed in a program.

FUNCTION KEYBOARD: An input device for an interactive display console consisting of a number of function keys.

GRAPHICS LANGUAGE: Any language used to program a display device.

HOST COMPUTER: The primary or mainframe computer.

INTERACTIVE GRAPHICS: The use of a display console in the interactive mode, usually involving CRT displays.

JOYSTICK: A lever that can be used in at least two degrees of freedom to control the movement of a display item. Used to provide coordinate input to the display device.

MENU: A list of options, typically represented on a display device, that allows an operator to select the next program action by indicating one or more choices with an input device.

MENU AREA: The area of the screen reserved for user function menus.

MESSAGE AREA: The area of the screen reserved for user messages.

MESSAGE MONITOR: A small monitor which displays instructions and menus to which the user responds.

PIXEL: Picture Element. The smallest displayables are on the display surface whose characteristics (intensity, color, etc.) can differ from those of its nearest neighbours. Applies to raster scan.

POSTPROCESSOR: The software module that takes the centerline output of the tool and adapts it to the particular machine control unit/machine tool combination that will machine the part.

PROGRAM FUNCTION KEYBOARD: A "button box" for making selections from menus.

RASTER DISPLAY: A display device that uses raster scan to generate a display.

RASTER SCAN: A technique for generating or recording a display with an intensity-controlled, line-by-line sweep across the display surface. This technique is used to generate a picture on a TV set.

RESOLUTION: The smallest distance between two points which can be percieved as distinct by the user.

SCALING: Resizing a graphics display to accomplish a specific operation or to clarify or enlarge specific areas of the display.

SUBMENU: A displayed list of options available in a specific operating routine that is subordinate to a higher-level menu.

TOOL AXIS: A vector defining the tool axis, stationed at tool end and pointing into the tool.

TOOL PATH: The path that the end of the cutting tool follows while performing a machining function.

TURNKEY SYSTEM: A computer system including hardware and software that has been assembled, installed, tested, and implemented, and is therefore available for immediate use.

WINDOW: An area on the display screen, specified by the user, in which an image or view is displayed.

ZOOMING: Scaling all elements of a viewport to give the apperance of having moved toward or away from a point or object of interest.

VITA

The author was born in Assam, India, on October 25, 1956.

He attended the Institute of Technology, Banaras Hindu University, from July 1973 to May 1978, when he graduated with honors as a B.S. Mechanical Engineer. He was active in sports, captained the University boxing team, and was awarded a "college color" for excellence in sports.

From July 1978 to July 1980 he worked with Union Carbide India Limited, at their Bhopal location, as a Mechanical Engineer. He left Union Carbide to pursue further studies.

He began graduate school in January 1981, working towards a Master of Science degree in Mechanical Engineering at Lehigh University. During his stay at Lehigh, he worked as a Research Assistant in the Computer-Aided Design Laboratory.

His parents are Mr. and Mrs. R.S. Bhalla. He is married to Bindiya, and they have a daughter named Sunindia.