ANALYTICAL AUDIT FLOWCHARTING GRAPHICS

ANALYTICAL AUDIT FLOWCHARTING GRAPHICS

By

SUMITRA PADHI, B.Sc., M.Sc.

A Project

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

March, 1980

MASTER OF SCIENCE (1980) MCMASTER UNIVERSITY
(Computation)

TITLE: Analytical Audit Flowcharting Graphics

AUTHOR: Sumitra Padhi,

B.Sc. (Honours), M.Sc. Berhampua University, India

SUPERVISOR: Professor J. Masterson

NUMBER OF PAGES: vi, 113.

ABSTRACT

This project is a contribution towards the development of one component of an analytical audit flowcharting system using computer graphics. The component to be developed will be concerned with producing graphical displays on an intelligent graphics terminal; the configuration of these displays is determined by parametric data supplied through the terminal keyboard. The software is designed in such a way that input options may be exercised to experiment with the shape and size of the component parts of the flowchart.

ACKNOWLEDGEMENTS

I wish to thank my supervisor Professor J. Masterson for his guidance and assistance during the preparation of this project.

I would like to thank all the faculty members, graduate students and staff of the Unit for Computer Science for their help in making my stay at McMaster enjoyable.

Finally, I would like to thank my husband, Hara, for his sacrifices during this year.

TABLE OF CONTENTS

TABLE OF CONTENTS Page

CHAPTER 1
INTRODUCTION

STATEMENT OF PROBLEM

The administrative data processing section of the McMaster University computing centre is presently developing an experimental system for the application of computer graphics to audit flowcharting at McMaster. This project is directed towards a part of the above system. It produces graphical displays on an intelligent graphics terminal; the configurations of these displays are determined by a parametric database stored on a remote computer.

Analytical auditing is a system-oriented auditing technique based on flow-chart analysis. This makes use of:

(i) a combination of flowcharting and limited tests of transactions as review techniques and

(ii) a combination of flowchart analysis and internal control evaluation guides as evaluation techniques.

This technique is presently used widely throughout the accouting profession in North America. Since there is no absolute set of standard symbols for constructing these flowcharts there are many variations in the notations. In order to facilitate the work between the university's Internal Auditing department and it's external auditors, McMaster University adheres to the flowchart conventions which are compatible with those of its external auditors. Thus, any staff member can understand another's analysis. The flowcharts can be carried forward for use in the future year's audits.

The processing of the datasets, such as documents, books, ledgers and files, is usually indicated by brief comments, rather than by flowchart symbols. By using a documentary flowchart it makes it easier to visualise the flow of information between different parts of an integrated system. With a minimum of narrative explanation we can also show the physical aspects of these datasets and their physical flows from one point to another.

HISTORY OF THE AUDITING PROFESSION

The process of investigation of a financial report or statement to determine its correspondence to established criteria is called auditing. For example a tax auditor checks a tax return to determine whether it reflects the taxpayer's tax liability in accordance with legislative rules. In this case the report is the tax return of the taxpayer and established criteria are the legislative rules. There are different types of audits, such as external, internal, governmental, etc. Through the study and evaluation of the organization's system of internal control, and by inspection of documents and by other auditing procedures, the auditors will gather enough information to determine whether the financial statements provide a fair and reasonably complete picture of the organization's financial position and it's activities during the period being audited.

The role of auditing is to add credibility to financial statements and thus to enhance the effectiveness of accountingcommunication needs by the economic system.

In Latin, the word "audit" means "he hears". Over two thousand years ago Egyptians, Greeks and Romans all utilized systems of check and counter-check on the accounting officials to whose care public funds had been entrusted.

Indeed the ancient records of auditing are chiefly confined to public accounts. A responsible official known as the auditor listened to the accounting of those who were required to account for their handling of the public funds. But nowadays there is no such oral presentation and oral examination of accounts. Prior to the industrial revolution, auditing was concerned principally with the detection of frauds. The importance of internal control was generally not recognized. Since the industrial revolution brought the beginning of large scale commercial and industrial enterprises, and the replacement of the one-time joint venture by the continuing corporation, the need for non-public auditing became even more apparent. Throughout the latter part of the eighteenth century and early part of the nineteenth century, it became increasingly common to call upon outsiders, skilled in record keeping, to investigate bankruptcies. In the beginning of the twentieth century the direction of audit work generally tended to move away from fraud detection and towards the new goal of determining whether the financial statements give a fair picture of the financial position, operating results and changes in financial position. The chief objectives of an audit were:

(1) Detection and prevention of fraud.

(2) Detection and prevention of errors.

In present days the main objective of an audit is to verify the actual financial condition and earnings of an enterprise. The minor objective is detection of fraud and errors. Within the last ten years, the detection of large scale management fraud has assumed a large role in audit philosophy.

THE AUDITING PROCESS

When the auditing profession was in its early days, a normal audit included a complete review of all transactions. Around the year 1900 auditors adopted a sampling technique becacuse of the development of large scale business enterprises. In 1930 in Britain and Canada the statutory audit had led to rapid growth of the profession. In the United States the development was comparatively slow, and the pattern of voluntary audits to meet creditor requests had been the testing of internal evidence as to assets and liabilities, rather than transaction testing.

In the new auditing technique, selected transactions are tested instead of verifying all transactions. Auditors gradually came to the conclusion that careful examination of relatively few transactions selected at random will give a reliable indication of the accuracy of other similar transactions. The advent of computers in the 1950's and statistical sampling techniques in the 1960's had significant effects on auditing methods. Auditing procedures have shifted from 100% checking to testing based on reliance on a study of internal control with the following objectives:-

(1) To safeguard assets from waste, fraud and inefficient use.

(2) To promote accuracy and reliability in the accounting

records and timely preparation of reliable financial information.

(3) To encourage and measure compliance with company policies.

(4) To evaluate the efficiency of operations. Evaluation of internal control became recognized as essential for the successful use of sampling techniques. The development of electronic data processing systems have changed the form and nature of accounting records and of the "information sytem" as a whole. Auditors are now trying to modify auditing procedures to fit the computer age.

FLOWCHARTING

Flowcharting is a simple and rapid way of achieving and communicating system comprehension. Generally, the flowchart method is well-known for the presentation of electronic data processing systems. Advantages of flowcharting include:-

1. Its compact form enables a more rapid and objective analysis.
2. Since intelligence is conveyed graphically, its assimilation is easier than from text.
3. A more logical and objective approach to the selection of procedural audit tests is enabled.
4. Audit staff can learn the techniques in a short space of time.
5. Amendments and updating in the systems description are readily accomplished.
6. Evidence of criteria used in evaluating the system is accurate and complete.
7. Chart production is rapid and inexpensive.

If flowcharts are to be used to document the auditor's review of systems, it is desirable for audit staff on the same engagement to use a standardized method of flowcharting. This standardization will enable any staff member to understand another's analysis and will permit flowcharts to be carried forward for use in future years' audits. Various styles of flowcharting have been in use for various purposes for many years.

MANUAL SYSTEMS FLOWCHARTING

EXAMPLE OF A MANUFACTURING COMPANY

The following areas will involve significant accountingprocedures from the beginning of a transaction up to entry in the general ledger or posting source, and the flow of information from posting source to the general ledger.

(1) Sales/receivables/receipts

This whole system can be divided into several parts such as:

(a) Sales

(b) Receivables

(c) Cash receipts

(d) Credit notes.

Since a number of different types of sales may be encountered, each with a different method of processing, one has to select the best way of showing the alternatives.

(2) Purchases/payables/payments

This system can be divided into two parts:

(a) Accounts payable disbursements involving purchase orders.

(b) Miscellaneous disbursements not involving purchase orders.

(3) Wage and salary payrolls

Generally this system can be divided into two parts:

(a) Plant wages

(b) Office Personnel salaries.

(4) Inventory records and cost records

This can also be divided into two parts:

(a) A schedule of the outline cost entries showing at a glance the accounting structure of the cost system.

(b) An outline chart and flowchart showing the flow of records and documents, their summarization, etc. to provide the various cost entries.

(5) Books of account and general

This system should summarize the flow of information from budgets and from all posting sources in the general ledger and from the budgets, general ledger and other sources into the client's monthly statements and other management reports.

CHARTING OF COMPUTER SYSTEMS

Unlike manual systems, the computer system is divided into two parts, the external system and the internal system. The processing that takes place outside the computer is called the external system and the processing that is performed internally by computer programs is called the internal system. The auditor has to prepare a summary flowchart in order to obtain an overall view of the entire system in simplified terms to indicate the following:

(i) The logical sub-systems and their relationship to one another.

(ii) The main system inputs.

(iii) The principal processing functions.

(iv) The main system outputs.

(v) The source of any general ledger entries.

The external system consists of two phases:

(i) Input phase: It consists of the processing of transactions from the point of transaction origination to the point of data conversion into machine readable form.

(ii) Output phase: The flow of data from first visible output from the computer to it's ultimate use.

The internal system consists of the following phases:

(i) The updating of masterfiles.

(ii) The summarization of data into a condensed form.

(iii) Arithmetic calucations such as the pricing and extension of sales invoice data.

(iv) The sorting and merging of data.

(v) The extraction of data from one or more data files based on predetermined criteria.

(vi) The printing of reports.

COMPUTER GRAPHICS

The advances in mini and micro computer lowcost graphic terminals and time sharing have extended the use of computer graphics. Computer graphics is the use of a computer to define, store, manipulate, interrogate and present pictorial output. The computer prepares and presents stored information to an observer in the form of pictures. The observer has no direct control over actual construction of the picture being presented. The application may be as simple as the presentation of a graph of a simple function using a high speed line printer or a time sharing teletype terminal, to something as complex as the simulation of the automatic reentry and landing of a space capsule. This project will employ an HP2647A Graphic Terminal.

The computer graphics can be divided into the following sections.

(1) Representing pictures to be presented.

The picture represented in computer graphics can be considered as a collection of lines, points, and textual material. A point can be represented by a single co-ordinate, a line can be represented by co-ordinates of its end points and textual material can be represented by a collection of lines or points. Almost all graphic devices have built-in "hardware" or software character generators. The representation of curved lines is usually accomplished by approximating them by short straight-line segments. Pictures ultimately consist of points.

(2) Preparing pictures for presentation.

The co-ordinates of the points representing a picture are generally stored explicitly or implicitly in a file (database) prior to being used. There are three basic methods for treating a point as a graphic geometric entity:

(i) move the beam or pen on cursor to the point.

(ii) draw a straight line to that point from the previous point.

(iii) draw a dot at that point.

The position of a point can be specified in either incremental, relative or absolute co-ordinates. For incremental co-ordinates the position of a point is defined by giving the displacement of the point with respect to the previous point. In relative co-ordinates, the position of a point is defined relative to a relocatable origin. In absolute coordinates, the position is defined relative to the lower left corner of the screen.

(3) Presenting previously prepared pictures

The database represents the total picture while the display file represents some portion of it. The display file is created by transforming the database. The picture contained in the database may be resized, rotated, translated, etc. to detail necessary perspective before being displayed.

(4) Interacting with the picture

Interacting with the picture requires some type of interactive device to communicate with the program while it is running. Generally a device like the alphanumeric keyboard is used for this purpose.

GRAPHIC FLOWCHARTING SYSTEM PROPOSED

The proposed flowcharting system of McMaster consists of the following components.

(i) Computer graphics terminal

It serves as the physical interface between each concurrent user and the system, through its keyboard entry and screen display. The display provides for both alphanumeric and graphics output. Form mode and editing capability are used for data entry and some form of graphic input would be desirable. Tape cassettes are included in the terminal for forms layout, data backup, etc.

(ii) A supporting computer

A supporting computer and communications system drives the terminal in response to user commands although some functions are accomplished through the local intelligence of the terminal itself.

(iii) Database

The database stores information for the generation of alphanumeric and graphic displays. It contains all the parameters necessary to define the administrative system which is the subject of analytical audit. It is structured to support the relationships which will be implicit in user queries.

The processing of the database mainly consists of two parts:

(a) Maintenance

(b) Retrieval.

This processing will be performed by means of a dialogue between the user and the flowchart system at the terminal.

(iv) Hardcopy facility

This facility is a device which is local to the terminal. It will produce a matrix plot of the flowcharts and the supporting text in a form corresponding to the pages displayable on the terminal screen.

(v) User's instructions for accessing the flowcharts

This project is based on that part of the above system which will draw symbols to represent datasets such as documents, books, ledgers and files on the screen of the graphics terminal, and draw flowlines to show their physical flows.

OBJECTIVES

The objectives of this project are:-

- To construct a page of graphics flowchart from given paramtric data representing the configuration of an information processing system.

- To provide a software tool enabling experimentation with the form and dimensions of symbols and flowlines and their arrangement on the flowchart page.

CHAPTER 2
PROBLEM ANALYSIS

EXAMPLE OF A FLOWCHART PAGE

The terminal screen is divided into two pages. Different flowcharts can be drawn on each page. The diagram on page 21 illustrates an example of a flowchart page.

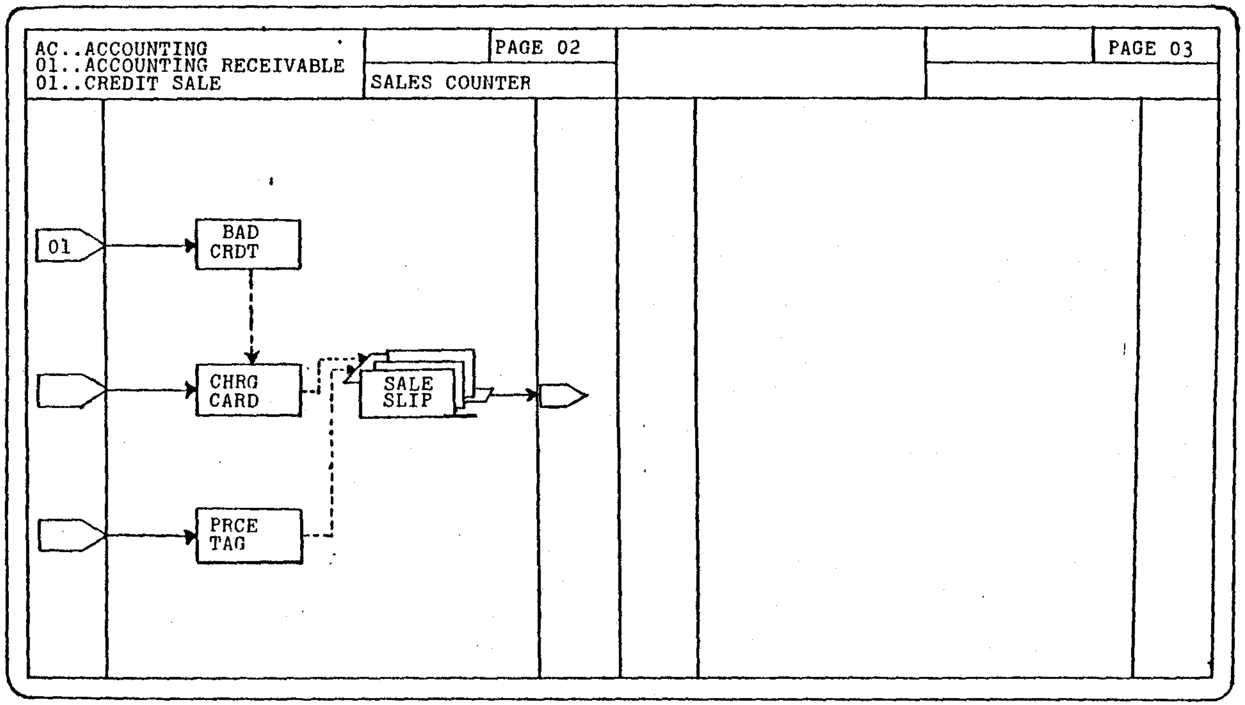
The terminal imposes certain limitations:-

(i) The size of each page is limited to 5" by 5".

(ii) The minimum text size is 10 x 7 dots per character cell where one inch is equal to seventy-two dots. So considering these limitation it has been considered that the capacity of each page is twelve symbols. These twelve symbols should be arranged in a rectangular array of four rows and three columns. So it can be considered that a page is divided into twelve boxes, three in each row and four in each column. A symbol can be drawn in any one of these boxes. Such a rectangular array is chosen so as to enable pathways for flowlines around each symbol "box".

The figure on the next page shows the example of a flowchart page drawn on the screen of the graphics terminal. It consists of four symbols and flowlines between these symbols. Here some symbols are connected to external sources which are indicated by drawing arrowsin the left

HP2647A


AC..ACCOUNTING
01..ACCOUNTING RECEIVABLE
01..CREDIT SALE
PAGE 02
SALES COUNTER
PAGE 03
01
BAD CRDT
CHRG CARD
SALE SLIP
PRCE TAG

margin. Also one symbol is connected to an external destination which is indicated by drawing an arrow in the right hand margin.

The input data for this example will be as follows.

(i) Symbol code

Symbol code indicates the shape of the symbol. In this example, the shape of each symbol is indicated by 1, i.e. the document symbol. In this example it will be 1 1 1 1 .

(ii) Flowline list

This represents how these symbols are connected from source to destination. Zero means an external source and 99 means an external destination.

In this example it will be 0 → 1
0 → 2
0 → 3
1 → 2
2 → 4
3 → 4
4 → 99

Here 4 symbols are used, and these are identified by the ordinal numbers 1,2,3, and 4, as applied to the symbol list.

(iii) Number of Copies

This gives the number of copies of each symbol. In this example all other symbols except the fourth symbol have no copies. The fourth symbol has three copies. When a multiple copy symbol appears in the left (source) side of the flowline list a group number is needed. This group number indicates that this group of copies has been connected to the corresponding symbol in the right (destination) side of the list. Here this input will be 0,0,0,3.

(iv) Group

This gives the group number. For this example the input will be 0,0,0,0,0,0,2. The last line of the flowline list is the only line which uses a multiple copy symbol in the left side. So the group number is 0 for rest of the flowlines and 2 for the last flowline. This indicates that group 2 of symbol 4 has been connected to an external destination.

(v) Type of flowline

This indicates the flowline between two symbols will be active flow or flow of information. If it is 1 then the flow is active flow and is represented by a solid line. If it is 2 then there is flow of information and is represented by a dotted line.

In addition to the above inputs there is also provision for supplying the dimensions of the symbols, so that the shapes of the symbols can be changed.

All these input data have been explained in the latter part of this chapter.

DESCRIPTION OF THE PROBLEM ANALYSIS

The problem can be divided into the following parts:-

(i) Subroutines to construct symbol shapes.

(ii) Subroutine to analyze symbol list and flowlist.

(iii) Draw page outline.

(iv) Draw page heading.

(v) Draw symbols as allocated by (ii).

(vi) Draw flowlines as per flowline list.

ENUMERATION OF FLOWCHART SYMBOLS

Symbol code

NF(I) for I = 1,2,3,... represents a particular shape of flowchart symobl. I is called the symbol index number and NF(I) is called the symbol code. There is a maximum of twelve symbol index numbers, i.e. $1 \leq I \leq 12$. NF(I) may be equal to NF(J) for $1 \leq I \leq 12$ and $1 \leq J \leq 12$. That is, the same symbol shape may appear more than once on a page. Here eight symbol shapes are considered. Each denoted by one of the numbers from 1 to 8 as follows:-

(1) Document

(2) Temporary file

(3) Permanent file

(4) Book

(5) Posting source

(6) Adding machine tape

(7) Computer printout

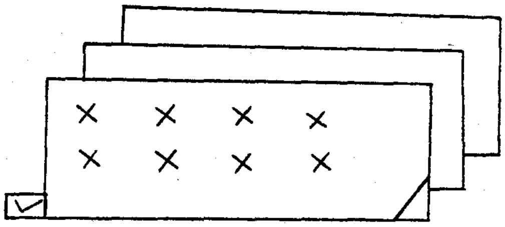
(8) Process symbol

Example

Symbol index #	Symbol Code	Symbol
1	3	Permanent file
2	2	Temporary file
3	4	Book
4	2	Temporary file

Decsription of the symbols

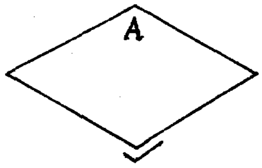
1. Document



It is a multiple copy symbol. The first copy is represented by a rectangle and the other copies are represented by drawing lines as shown in the figure. The copies are numbered as they are originally in the form. But some copies might be separated while processing. So copy #3 may be the first copy at this stage of processing, copy #5 may represent the second copy and so on. A maximum of nine copies are allowed for this symbol. Deposit slips, refund vouchers, credit sales memos are examples of documents. If the number of copies is '0' then the document has no multiple copies originally. If the number of copies is '1' then the document has originally more copies but all other copies have been separated during processing except the 'one' copy shown.

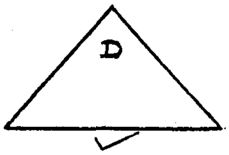
The flag in the lower right hand corner of the first copy is known as the corner flag. If the corner flag is solid then it indicates that the form is created at this stage. If it is hollow then data is added at this stage. If there is no cornerflag then no data is added. The cornerflag is designed according to the value of the variable SICF. There will be no cornerflag, hollow cornerflag or solid cornerflag according as SICF is '0', '1' or '2'. The annotation box is attached to the left hand corner of the first copy. This is controlled by a variable SIAB. If it is '0' then there will be no annotation box which means no checking. If it is one then a "✓" mark is inscribed in the box which means document is just checked but nothing is written. An "i", or "s" is inscribed in the box according as the value of SIAB is 2 or 3. If "i" is written then the record is just initiated and if "s" is written then it is signed. A digit is inscribed in the upper right hand corner of each copy which shows the number of that copy. If the number of copies is 0 then no digit is inscribed. The name of the record is indicated in the middle of the first copy by brief text comprising two lines of four characters each.

2. Temporary file



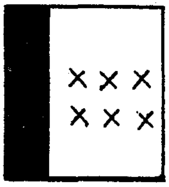
A diamond represents a temporary file of documents. The filing order is represented by one of the letters A, N or D inscribed in the upper part of the symbol. A is used for alphabetical ordering, N is used for numerical ordering and D is used for ordering by date. A mark exscribed near the lower apex indicates the type of checking. This is controlled by the variable Sk2. If Sk2 is 0 then no mark will be exscribed which indicates that the checking of continuity is unimportant and not done. If it is 1 then an "x" mark will be exscribed showing that the checking of continuity would be important but in fact not done. If Sk2 is 2 then the mark "✓" will be exscribed which indicates that the checking of continuity is important and also done. A temporary file is merely a file of documents waiting for further processing after some other event occurs such as the arrival of goods on receipt of a supplier's invoice.

3. Permanent file



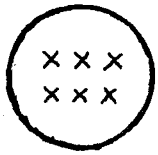
A triangle represents a permanent file. Like the temporary file here also the filing order is represented by one of the letters A, N and D inscribed in the upper part of the symbol. Also the type of checking in a file is indicated by a check mark exscribed directly below the file symbol. The marks have the same meaning as in the case of the temporary file.

4. Book



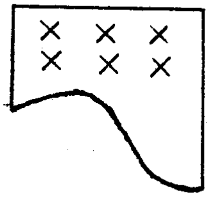
A rectangle represents an accounting book. It is differentiated from documents by shading at the left hand edge. An accounts receivable ledger, a price catalogue, an inventory record can be represented by the symbol. The name of the record would be indicated by abbreviations on the middle of the symbol. The shaded left hand edge has been indicated by drawing some close lines.

5. Posting Source



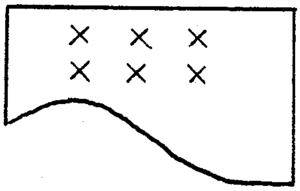
A circle represents a posting source. This symbol represents the books which will be posting sources to the general ledger. The name of the record is indicated in the middle of the symbol.

6. Adding Machine Tape



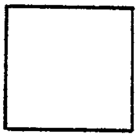
This symbol indicates an adding machine tape. The labelling is done in the middle of the symbol.

7. Computer Printout



This symbol represents a computer printout. Reports, control total listings, pay cheques are examples of this symbol. The labelling is done in the middle of the symbol.

8. Process Symbol



A square represents a process symbol which indicates that action is being taken or some operation is being performed, e.g. checking, matching, agreeing, etc.

Physical movement of documents is shown by a solid line with an arrow at the end of the line.

A dotted line with an arrow on the end represents information flow from the record. This indicates that a record or document is being used for some purpose such as initiating preparation of another document or posting source to a subsidiary ledger, without an actual physical flow of the first document. This is also known as flow of information. For example we can use this when information is taken from one document and is used to start another document set, i.e. the transfer of prices from a price book to an invoice.

The general direction of flow is from left to right and from top to bottom of a chart. But since it is not possible to arrange all the symbols such that all the lines will go as the direction indicated above many lines in fact go in reverse direction. So the arrow heads are included on the end of the line, i.e. near the destination symbol.

GRAPHIC TERMINAL CAPABILITIES

It has been already mentioned in the introduction part that the HP2647A Graphics Terminal is used here. It is capable of executing BASIC programs directly without the need of a host computer system. The BASIC language that is used there is local BASIC which is slightly different from other versions of BASIC. The Graphics Terminal uses a microprocessor under firmware control. It can be used for data entry and preparation, information display and editing, interactive programming, data communications and time sharing operation. The terminal provides interactive graphics features available under user or program control. The program control can be either done by loading the local BASIC language into the terminal from cartridge tape or a program running in a remote computer. Additional data up to 220 kilobytes can be stored using two tape cartridges which allow the terminal to be used for either stand-alone or on-line operation. For example forms designed at the terminal using the line drawing or other character sets can be stored on a tape cartridge and selectively retrieved from the keyboard or commands from BASIC running in the terminal or from a remote computer. The terminal has many powerful features such as self-test, dynamic memory allocation, transparent control codes and off-screen storage.

The terminal has three major and mechanically indepedent sections:

(1) Mainframe

(2) CRT monitor

(3) Keyboard

(1) Mainframe

The mainframe consists of the following modules which are functionally, mechanically and electrically independent giving a high degree of flexibility and reducing service time.

(i) Microprocessor

The terminal uses an eight bit microprocessor to control most of the terminal operation such as alphanumeric display, programmable functions and I/O devices.

(ii) Display

The terminal uses two separate display memories, one for alphanumeric data and one for graphics data.

(a) Alphanumeric Memory

It can hold up to 5,000 ASCII characters. It is maintained by the display control module.

(b) Graphic Memory

Graphic data is stored as a dot pattern containing 259,200 (720 x 360) points. The display is maintained by the Graphics Microcontroller and Graphics Display memory.

Graphic data received from the keyboard, data communication interface, or cartridge tape is processed by the Graphics Microcontroller before being stored in the Graphics memory. Normally the input data are in the form of vector end points. The microcontroller uses the previous vector endpoint together with the new endpoint to generate a line of display dots that approximate the line. These dots are stored in the graphic memory.

The graphic and alphanumeric data are displayed in the same area on the screen but are stored in separate RAM memories. Hence, one can read and modify graphic and alphanumeric data separately.

(iii) Terminal bus

It is a printed circuit board with connector which is attached to the bottom of the main frame and the power supply. The terminal bus distributes power to the individual modules and provides data, address and control lines for communication between the various logic functions. It also provides communication paths between processor, memory, input/output and display refresh on a shared basis.

(iv) Terminal Memory

The terminal memory is of two types.

(a) Random access memory (RAM)

It stores display characters and data. The BASIC work space resides in RAM.

(b) Read only memory (ROM)

It stores terminal programs and makes them more permanent than software but less permanent than hardware. So the terminal programs are called firmware. In the terminal one half of the memory is ROM and the other half is RAM. All of the terminal memory is MOS semi conductor memory. A separate RAM memory is used to store the 720 x 360 graphics data.

(v) Input/Output Modules

The keyboard interface, graphics microcontroller, graphics memory, the data communications, etc. are the Input/Output modules. These modules must respond to commands from the microcomputer and its programs.

(2) CRT MONITOR

This contains sweep and high voltage circuits, the high resolution, low-profile cathode-ray tube and fan.

(3) KEYBOARD

The processor scans the keyboard at discrete intervals for a depressed key. Each key is assigned a position in a matrix of fourteen columns and eight rows. The matrix provides a reference to a look-up table that the firmwire uses to display the character and/or send the character code over the data communication line.

The screen can contain 720 dots horizontally and 360 dots vertically. So graphics data can be displayed by addressing points in a 720 by 360 array. All the graphics function commands can be entered from the terminal keyboard by the operator. The graphic function can be controlled through AGL (A graphic language). AGL programs are comparatively long, and since the terminal memory is small economy of space is of prime concern, hence AGL is not being used here. I have done the programming of graphic functions by escape sequences. Graphics functions are controlled by parameterized escape sequence. All graphic escape sequences begin with ESC*. The third character always lower case selects the type of graphic sequence. Subsequent characters in the control sequence are read as either parameters or commands depending on the location of the character in the ASCII table.

The programmable graphics functions are organized into the following five major groups.

(i) Graphic display control

This is made up of the following functions.

(a) Graphics Cursor Control (To control the graphic cursor).

(b) ZOOM (To control the portion of the graphic memory that has been currently displayed).

(c) Graphics memory control (To control the state of the graphic memory).

(a) Graphic Cursor Control

A separate graphics cursor is available for use in locating points in the graphics display. The graphics cursor is initially off. It can be turned on or turned off using corresponding ASCII functions, or pressing the corresponding key. Turning the cursor on or off does not affect the data in graphic memory. The graphics cursor is initially at position (0,0). The cursor can be positioned using either absolute or relative co-ordinates.

(b) ZOOM

A portion of the graphic memory data can be displayed at increased size without any change of data in graphics memory. The magnification settings are from 1 to 16 lines and is initially set to 1. The magnification data is not diaplyed until the zoom function is turned on. The zoom position is initially the graphics cursor position. It can be set to any position in graphics memory using ASCII absolute co-ordinates. The selected data is not displayed until the zoom function is turned on. Once a zoom size and position are selected the data is displayed by turning on the zoom function.

(c) Graphics Memory Control

The graphics display can be turned on or off or the entire memory can be set to all ones (dots on) or all zeros (dots off). Without affecting the data in the graphics memory the graphics cursor and graphics display can be turned on or off. The graphics data currently displayed can be set on clear. When used together with the zoom function this function can be used to set or clear blocks of the graphics memory.

(ii) PLOTTING

All vector plotting sequences are initiated by ESC*p. Then the drawing format and data are written. For example ESC*pa means lift the pen. If no format is specified then ASCII absolute is assumed. The terminal usesthe concept of a "pen" in drawing vector data. The pen can be lifted or lowered as well as be positioned using absolute or relative co-ordinates. If the pen is down a line will be drawn. If the pen is up then it is moved to a new point without drawing a line and lowered. Absolute co-ordinates, relocatable co-ordinates or incremental co-ordinates can be used by giving corresponding parameters. The pen is initially in the up state and positioned at absolute co-ordinates 0,0.

(iii) Graphic Text

Text strings can be written directly into the graphics image memory. An internal character generator converts the ASCII codes into a dot matrix representation which is drawn as a vector. In this case characters can be drawn in any dot position, they can be rotated in multiples of 90 degrees and can be scaled in size from one to eight times, etc.

(iv) Multiplot

This allows you to quickly and easily plot numeric data in the form of Pie charts, Bar charts and Linear charts.

(v) Compatibility Mode

This enables some degree of compatibility with Tektronix graphics displays.

DESCRIPTION OF THE FLOWCHART PAGE

The entire screen is considered as two pages, each page having a margin width of 33 dots on both sides. The co-ordinates of the four corners of the first and second page are (0,0), (359,0), (359,359), (0,359) and (359,0), (719,0), (719,359), (359,359) respectively. So each page can be considered as a square of 360 by 360 dots. Since each page has right hand margin and left hand margin each of width 33 dots the actual area of a page used for drawing picture is a rectangle of 294 x 326 dots. The top of each page is used for the heading. For the first page the rectangle whose co-ordinates of the corner are (33,325), (198,325), (198,359), (33,350) is used for writing the following:

1 - ID. of System

2 - ID. of Subsystem

3 - ID. of Procedure.

The headings are written in a similar manner in the second page.

Each page is divided into twelve boxes which are arranged in four rows and three columns. Each box is a rectangle with seventy-two dots width and sixty dots height. The horizontal distances between any two boxes is fifteen dots. The vertical distance between any two boxes is eighteen dots. The horizontal distances of the boxes in the side from the margin of that side is fifteen dots. The vertical distances between the boxes on the top and bottom of the page from the page heading and the lower edge respectively are eighteen dots. A symbol is drawn in any one of these boxes as determined by the allocation module. Since the relocatable origin feature is being used the contents of one page can be transferred to the other without any difficulty. Again since variables are used to draw the symbols, the shape and size of the symbols can be changed easily by changing the values of the variables.

ALLOCATION ALGORITHM

As previously mentioned, a maximum of twelve symbols can be drawn in a page. Given a list of symbols and a list of their connections we have to decide which symbol should be placed in which box. The symbols should be arranged in such a way that the flowlines should be clear and easy to understand. To produce pleasing results in the form of shortest possible flowlines, the transportation model of linear programming was considered, but found not to be applicable. Also considered was zero-one integer programming which was found applicable, but computationally expensive and not suitable for the purpose. The problem was eventually discovered to fit the form of a quadratic assignment problem, but a heuristic approach was finally adopted.

The following strategies were considered for allocation of symbols to boxes on the page, consistent with the objective of flowlines which are pleasing to the eye:-

(1) Symbol number increases from left to right and top to bottom.

(2) Process symbols are placed in the centre column.

(3) Input symbols are placed in the left column.

(4) Output symbols are placed in the right column.

(5) Symbols which are connected to the external sources are placed in the left column.

(6) Symbols which are connected to the external destination are placed in the right column.

(7) Strongly connected symbols are placed in the centre.

(8) Doubly or Multiply connected symbols are placed adjacent to each other.

(9) Aggregation of symbols is well spaced and centered.

Since here any list of symbols and all types of connections between them are considered, it is not possible to write an algorithm which will allocate symbols simultaneously satisfying all the nine points as discussed above. So in some cases it violates some of the above points. In the allocation algorithm the points (2), (5), (6) and (7) are given more importance than the other five points. The symbols are arranged in such a way that most of the flowlines are lines joining the neighbouring boxes. It is assumed that in the input symbol list the strongly connected symbols should appear first.

Special Problems of the document symbol:-

The document symbol is the only multiple copy symbol used here. This symbol has some special problems to be considered. It is assumed that a document symbol cannot have more than nine copies. If the number of copies is not less than or equal to five then two boxes are allocated. Since there can be more than one document symbol with different numbers of copies in a page with the input symbol list, an input list for the number of copies of each symbol is needed. If the symbol is not a document symbol then the number of copies should be zero. When the number of copies of the document symbol is not more than five then only one box is allocated.

The symbols can be drawn either in the left or right page of the screen, as the user likes. After allocation is complete, if the bottom two rows of the page are empty then the row of each symbol is shifted down by one. Hence, if there are few symbols they all appear in the middle of the page.

FLOWLINE/ROAD UTILISATION ALGORITHM

Each symbol has four doors. The coordinates of the doors vary depending on the shape of the symbol. A flowline can enter or leave a symbol only at the doors. Flowlines for the document symbol have to enter and leave through the fixed doors. Flowlines for other symbols can enter or leave through any door. There can be a maximum three flowlines entering or leaving a door.

The following cases have been considered for the flowlines.

(1) Adjacent case

This case occurs when source and destination symbols are adjacent on the page. The symbols are either in the same row (or column) and the difference between the column (or row) is one, or the symbols are diagonally placed such that the difference between rows and columns of the symbols are one. The flowline is just a straight line joining the door of the source symbol to the door of the destination symbol. There is an arrow on the end of the line i.e. near the door of the destination symbol. Arrows on both ends of the line implies that two symbols are connected to each other in both directions.

(2) Remote Case

In this case source and destination symbols are not adjacent i.e. the difference between the rows or columns of the source and destination symbol is greater than one. Here the flowline is not a single straight line. It consists of the following parts.

(1) Line from source symbol door to road.

(2) Horizontal or Vertical leg.

(3) Vertical or Horizontal leg.

(4) Road to door of destination symbol.

For each flowline we have two choices.

(a) (i) Leaves the source horizontally.

(ii) Vertical road

(iii) Horizontal road

(iv) Arrives at the destination vertically.

(b) (i) Leaves the source vertically.

(ii) Horizontal road

(iii) Vertical road

(iv) Arrives at the destination horizontally.

There are five horizontal roads and four vertical roads. If the symbol box is i,j, where i = 1,2,3,4 and j = 1,2,3 denote the row and column of the symbol respectively, then it can have two horizontal roads H_i and H_{i+1} and two vertical roads V_j and V_{j+1}. If the source is the left margin then it can have only one vertical road V1 and the same horizontal roads as the destination. Similarly if the destination is the right margin then it can have only one vertical road V4 and the same horizontal roads as the source. There cannot be more than six lines in each of the horizontal or vertical roads. If the road is used for the first time then the line is drawn in the centre of the road. After a road has been used once, a shift within that road must occur so as not to overlap when a second flowline appears in that road. This has been done by using the variables G1 and G2 for horizontal and vertical roads respectively. Since each symbol has four doors and each door can have a maximum three flowlines, there can be 4 x 3 = 12 flowlines enter or leave a symbol.

Special problems of the document symbol:-

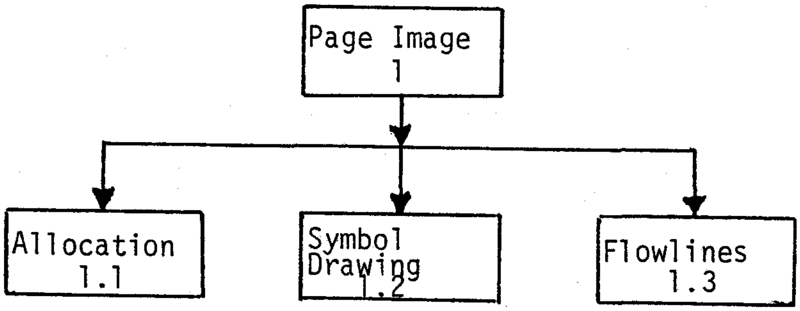
The document symbol will arrive as one entity, but each copy can go to different places. When it is used as a destination the doors are one and the same for all copies, and when it is used as a source symbol the doors may be distinct for different copies. For this symbol, doors 1 and 2 are used when it is used as a source symbol, and doors 3 and 4 are used when it is used as a destination symbol. If it is used as a source, a group number must be given to indicate that copies belonging to this group only are connected to the destination symbol. A bracket is drawn for each group. The destination doors 3 and 4 are at the centre of this bracket.

It is assumed that the copies belonging to each group are sequentially arranged such that no copy exists, between any two conjugate copies of a group, which belongs to another group. It is already mentioned that a solid line indicates active flow and a dotted line indicates flow of information. With the connection list, a code list is also supplied in the input list indicating which line will be solid and which will be dotted. If two symbols are connected to each other it is assumed that the type of flowline in both cases must be the same; i.e. either solid or dotted. So one line with arrows in both the ends can be used to indicate the flow. The flowline should not cross any symbol.

CHAPTER 3
SYSTEM DEVELOPMENT

LOGICAL ORGANIZATION

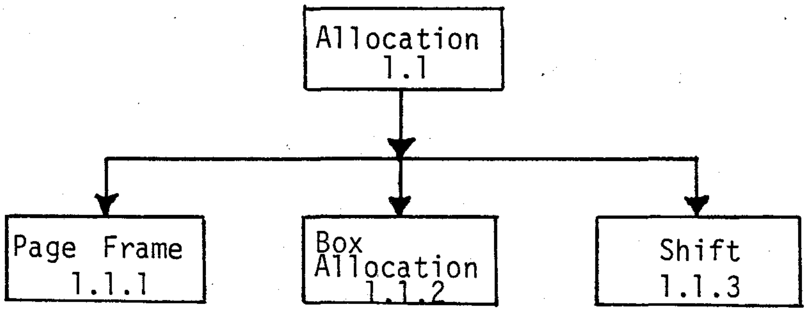
The entire program can be pictured as follows:



The shape and dimensions of the symbol and the flowline lists are given as input, and the output will be the page image consisting of symbols and flowlines between them.

Allocation Module

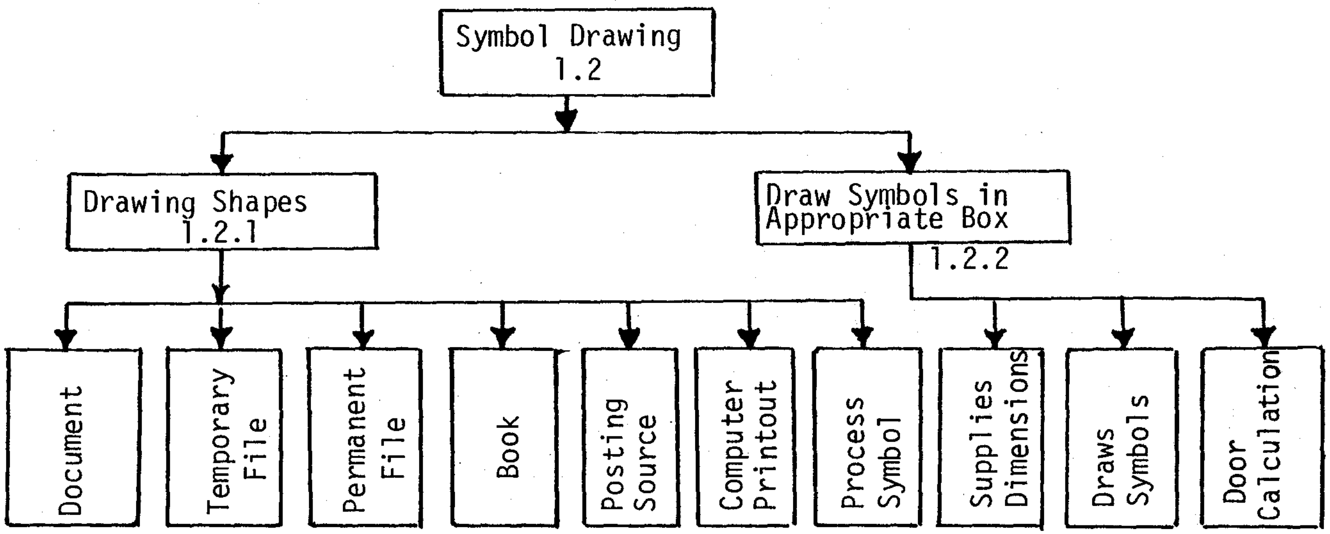
It can be divided as follows:



In this module, a page frame is drawn, and an appropriate box is allocated to each symbol. If the bottom two rows of the page are empty, then each symbol is shifted down by one row.

Symbol Drawing Module

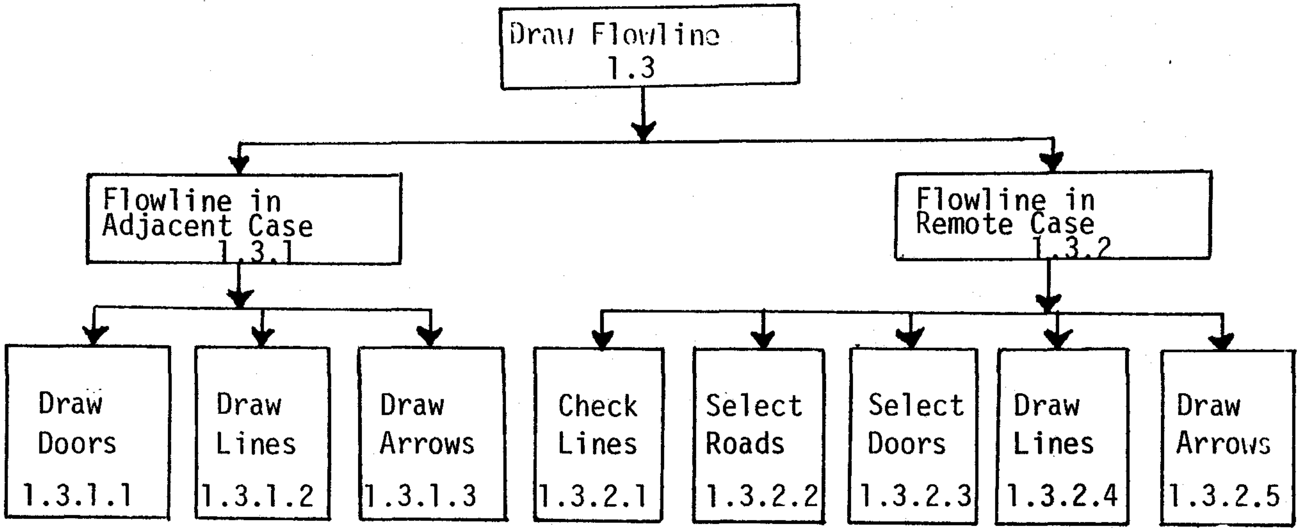
This module consists of the following sections:



Here the symbol is drawn in the boxes as directed by the allocation module. The dimensions of the symbols are the values stored in the data statements, or given as input. After the symbol is drawn the co-ordinates of the doors are calculated. This module has subprograms to draw different shaped symbols. A particular shape of symbol is drawn by referring to the subprogram for that symbol. Since variables are used in each of these subprograms the shape and size of a particular symbol can be changed by giving different inputs.

Draw Flowline Module

This module consists of the following parts:



The flowline list is scanned; when the flowline for an adjacent case is found, the doors for the source and destination are selected. The line is drawn. An arrowhead is drawn near the door of the destination symbol. After the whole list is scanned for adjacent cases the remote case is considered. In the remote case again the flowline list is read from the beginning. Each line is checked, whether it has already been drawn or not. If it has not been drawn then the horizontal and vertical roads as well as doors are selected. The line is drawn in segments. An arrowhead is drawn near the destination symbol.

The above gives a broad overview of the logical organization of the program. But due to the shortage of storage space in the Microprocessor the actual coding of the program is slightly changed.

PROGRAM ORGANIZATION

The basic program consists of four modules.

(i) Allocation Module

(ii) Symbol Drawing Module

(iii) Flowline (1) Module

(iv) Flowline (2) Module

After one module is executed the data that will be used in the next module is stored in a cartridge tape and it goes to the next module automatically. When the next module is executed the required data is read from the cartridge tape. After the flowline (2) module is executed, which is the last module, the program can be terminated. If the user wants to continue he has to again start from the beginning.

Allocation Module

(1) Page frame

The page frame will be drawn on the left or right side of the screen according as the value of the variable PAGE is one or zero. The user has to set the value of PAGE by putting a one or zero when it will ask for input. After the value of PAGE is set, the subprogram "FRAME" is executed. The values of the variables used in "FRAME" are taken from a data statement. After the page frame is drawn the subprogram "SUBROUTINE FOR ALLOCATION" is executed.

(2) Subroutine for allocation

(i) Get Input

The default values for the symbol list and flowline list are taken, or input will be asked, to supply accordingly as the user sets the value of the variable YES 0 or 1. If YES is 1 then the user has to supply inputs for symbol codes i.e. NF(I), I = 1,2,...12, left flowline list i.e. L(I), I = 1,2,...20, right flowline list i.e. R(I) = 1,2,...20, and number of copies of the symbol i.e. SIK(I), I = 1,2,...12. If the number of symbols used is less than twelve then the input is '50' for the rest of the non-existent symbols. Suppose seven symbols are used then symbol codes are given for these seven symbols and fifty is entered for the value of each of the variables NF(8), NF(9), NF(10), NF(11) and NF(12).

For example

NF(1)	NF(2)	NF(3)	NF(4)	NF(5)	NF(6)	NF(7)	NF(8)	NF(9)	NF(10)	NF(11)	NF(12)
2	1	3	4	2	5	6	50	50	50	50	50

Similarly if the number of lines in the flowline list is less than twenty, then '40' is entered for the values of the rest of the L(I) and R(I). If NF(I) is not equal to '1' i.e. if the symbol is not a document symbol, then the corresponding SIK(I) will be zero.

(ii) Priority for columns and rows

After getting all inputs, the number of times each symbol appears in the flowline list is calculated and is stored in the array C(I), I = 1,2,...12. The two dimensional arrays V(I,J), I = 1,...12, J = 1,...4, P(I,J), I = 1,...12, J = 1,...3 are used to store the priority of each symbol for each row and column respectively. The symbol I for which C(I) is maximum i.e. most strongly connected symbol, has more priority for the box in the second row and second column than any other position. The symbols which are connected to zero i.e. external destination or right margin have more priority for the third column.

(iii) Selection of box.

(a) Process Symbol

After values have been given to V(I,J) and P(I,J) the list is scanned for process symbols. The boxes for these symbols are fixed. The process symbols should be placed in the second column.

(b) Other Symbols

The elements of the two dimensional array Sc(I,J), I = 1,2,...12, J = 1,2,...12 stores values which indicate the number of times the symbol I is connected to the symbol J. The elements of the two dimensional array B(X,Y), X = 1,...4, Y = 1,...3 indicate a box in the Xth row and Yth column. The value of B(X,Y) is zero or one according as the box is occupied by any symbol or not. The lists P(I,Y) and V(I,X) are scanned to find X,Y for which P(I,Y) and V(I,X) are maximum. The values of row and column for the symbol are X and Y respectively. The flowline list is also checked if the symbols which are connected to the symbol I have already been assigned to a box. Then the row for the symbol I is set as the value of the row of the symbol with which it is strongly connected. If B(ROW, COL) is zero then that box is assigned to the symbol. If the box is already occupied then the next priority column is considered and B(Row,Col) is checked. If all the columns are exhausted then the row having next priority is checked and B(Row, Col) is tested and so on. When a box B(X,Y) is assigned to a symbol I the values of X and Y are stored in two arrays row(I) and col(I). Row(I) and Col(I) respectively represent the row and column assigned to the symbol I.

(c) Special consideration for document symbol

If the number of copies of the document symbol is less than or equal to five then it is treated as above. But if the number of copies is more than five then a box B(X,Y) is chosen as above. The box B(X-1,Y) is also checked, if it is empty then the two boxes B(X,Y) and B(X-1,Y) are assigned to the symbol. If B(X-1,Y) is not empty then another B(X,Y) is taken for which B(X-1,Y) is empty.

When the value of I is greater than twelve or NF(I) is '50' then the part of the program for selecting a box is no more executed.

(iv) Shifting of rows

After all the symbols have been assigned to boxes, rows are tested. If the bottom two rows are empty then one is added to each row (I), so that all symbols appear in the middle of the page.

After this module is executed each symbol I has been assigned a box shich are stored in the arrays Row(I) and Col(I). At the end of this module the values of NF(I), ROW(I), COL(I), and PAGE are stored in a cartridge tape.

SYMBOL DRAWING MODULE

In this module first the subprogram "Draw Symbol" is executed. In this subprogram, the subprogram to draw a particular symbol is referred and that symbol is drawn on the screen.

(i) Draw Symbols

This part of the program first reads the values of NF(I), ROW(I), COL(I) and page from the corresponding tape. From ROW(I), COL(I) and PAGE it calculates the co-ordinates of the lower left hand corner of the box I. The default values for the dimension of the symbol are taken, or supply of the input will be asked, according as the user sets the value of the variable YES to zero or one. If YES is zero, then for the dimensions of the symbol, the default values stored in the data statement are taken and the subprogram for drawing that symbol is executed. If the value of YES is one then the user will be asked to supply his own input for dimensions of the symbol. After all the inputs for a particular symbol have been supplied the subprogram for drawing that symbol is executed. The symbol is drawn on the screen. After the symbol is drawn the co-ordinates of the doors of that symbol are calculated. They are stored in the three dimensional arrays Doorx(I,J,K) and

Doory(I,J,K). Here I takes any value from one to twelve representing the symbol, J takes any value from zero to five representing the group, K takes any value from one to four representing the door. The group is zero for any symbol other than the document symbol. If the symbol is a document symbol then the group number is used to distinguish the co-ordinates of the doors for copies belonging to different groups as the document copies are separated for distribution.

If I is greater than twelve or NF(I) is '50' then the execution of the draw symbol phase is completed. The values of the elements of the arrays Doorx(I,J,K) and Doory(I,J,K) are stored in the corresponding tape. The symbols are drawn by positioning the pen and using escape sequences.

(ii) Document Symbol

This subprogram draws the document symbol. In order to distinguish different groups, it draws brackets on each group. A bracket is drawn on the left side of the symbol to indicate that the document symbol always comes as a whole. The doors one and two which are used as destination doors are the same for each group. These doors are near the left hand brackets. They are four dots away from a point on the bracket. Doors three and four which are source doors are different for different groups. Doors three and four are near the mid-pointof the bracket of a group. They are eight dots away from the mid-point. The label of the symbol can be written in the place where "XXXX" is written i.e. in the first copy of this symbol. To see the doors clearly, straight lines are drawn from the doors to the corresponding points on the brackets.

(iii) Posting Source Symbol

This part of the program draws a posting source i.e. a circle. "XXX" is written in the middle of the circle where the label of the symbol can be written. Four points on the left, right, top and bottom of the circumfurence of the circle are the doors of the symbol.

(iv) Temporary File Symbol

This part draws a temporary file symbol. It looks like a diamond. The sequential organization and sequential checking of the file are indicated by the marks on the top and bottom of the file respectively. The four corners of the diamond represents the doors of this symbol.

(v) Permanent File Symbol

The permanent file symbol, a triangle, is drawn here. Like a temporary file the sequential organization and the sequential checking of the file are indicated by the marks on the top and bottom of the file respectively. The three corner points of the triangle represent the first, third, and fourth door. Four dots down from the mid-point of the base is the second door.

(vi) Book Symbol

This part of the program draws the book which looks like a rectangle with one shaded edge. "XXX" is written on the middle where the label of the symbol can be written. Four doors are near the mid-point of each side. In order to have distinct flowlines, ie. flowlines without touching the symbol, the doors are not exactly at the mid-point of the sides. They are slightly above or below the mid-points. To see these doors clearly straight lines are drawn from the doors to the mid-points of the corresponding sides.

(vii) Computer Printout Symbol

This part of the program draws the computer printout symbol which is a closed area with straight lines as three sides and curve as one side. "XXX" is written inside where the label of the symbol can be written. Like the book symbol here the first, third and fourth doors are near the mid-points of the three sides which are straight lines. The second door is just four dots below a point on the curved side. Here also, like the book symbol, straight lines are drawn to join doors and mid-points of the corresponding sides. A line is also drawn from the second door to the corresponding point on the curved side.

(viii) Process Symbol

A process symbol which is a square is drawn here. Just like the book symbol in this case the four doors are near the mid-points of the sides and lines are drawn from the door to the corresponding mid-points.

Due to the shortage of memory, symbols other than these seven symbols cannot be drawn in this module. The position of the symbol inside the box can be changed by giving different values to the variables XSI, YSI.

After the execution of the symbol drawing module, symbols are drawn in the required page and boxes as directed by the Allocation module.

FLOWLINE (1) MODULE

(i) Get Inputs

When this module is executed the values of the arrays NF(I), ROW(I), COL(I), Doorx(I,J,K), Doory(I,J,K) and the value of the variable PAGE are read from the corresponding cartridge tape. Again the user will be asked whether he wants to give his own input or to use the default values. If the value of the variable YES is one then the user has to give his own inputs, if it is zero then the default values stored in the corresponding data statements are taken. The user has to supply the values of L(I) and R(I) again. The user has also to supply the values of Grl(I), I = 1,...20 and Code(I), I = 1,...20. The value of Grl(I) is a group of the symbol. It represents which group of L(I) is connected to the corresponding R(I). If L(I) represents any symbol other than the document symbol then Grl(I) is zero. Code(I) represents the type of flowline. If Code(I) is one then the corresponding flowline from L(I) to R(I) is a solid line which indicates a physical flow. If the Code(I) is two then the line from L(I) to R(I) is a dotted line which indicates information flow.

(ii) Scanned List

(a) Special consideration for document symbol.

It has been already mentioned in Chapter 2 that excepting the document symbol, for any other symbol the flowline can leave or enter from any door. But for a document symbol it can only enter in door one and two and leave from door three and four. Two variables "BLOCK 1" and "BLOCK 2" are used to indicate a document symbol in a flowline list. When the flowline list is scanned, if the left side of the symbol represents a document symbol then block 1 is set to 1, if the right side of the symbol is 1 then block 2 is set to 1.

(b) Check List

If I is greater than 1 then while considering the flowline for L(I), R(I) all the I-1 flowlines are checked. If the flowline has already been drawn for R(I), L(I) then just an arrow head is drawn near the corresponding door of the symbol R(I). Then the next L(I), R(I) is considered. The value of an element of the array Ocp(I) is set to be 1 when the corresponding line L(I), R(I) is drawn. Since doors are restricted for a document symbol, if any of the L(I), R(I) represents a document then the list is not checked. Next L(I), R(I) is considered.

(c) Numbering of source and destination symbols.

The variables SX, SY, DX, DY are used to number the source and destination symbols.

SX = ROW(L(I)), SY = COL(L(I))

DX = ROW(R(I)), DY = COL(R(I))

If the external source is used, SX = ROW(R(I)), SY = 0.

If the external destination is used, DX = ROW(L(I)), DY = 4.

(d) Checking for adjacent symbols.

The symbols are said to be adjacent if either of the following cases occur:-

(i) SX = DX, i.e. rows are same and Abs(SY-DY) = 1 i.e. the difference between columns is one.

(ii) SY = DY, i.e. columns are same and Abs(SX-DX) = 1 i.e. the difference between rows is one.

(iii) Both the difference between rows and columns are one.

(e) Draw Lines

If the symbol is adjacent then suitable doors are chosen. The doors will be such that the line joining them will not cross any part of the symbol. Adjacent doors are chosen. Then the doors are tested if they are empty or not. If Block 1 = 1 then doors three and four are used. If Block 2 = 1 then only doors one and two are used. Three lines can be drawn from each door. The four dimensional array Door(I,J,K,L) where I = 1,...12, J = 0,1,...5, K = 1,...4, L = 1,...3 is used to check whether the path l of door K of group J of symbol I is empty or occupied. It is empty or occupied according as the value of Door(I,J,K,L) is 0 or 1. If the doors and paths of both source and destination are empty, then the value of Code(I) is checked to find whether the line will be solid or dotted; then the required line is drawn.

(f) Draw Arrow

After the line is drawn the pen is positioned near the destination symbol and an arrow head is drawn. Then the next L(I), R(I) is considered. If I is greater than twenty or L(I) is forty then the execution of the Scan List part is completed. The values of the elements of the arrays Ocp(I), L(I), R(I), Grl(I), Code(I), Door(I,J,K,L) are stored in the corresponding cartridge tape. Then the next module is executed.

FLOWLINE (2) MODULE

(i) Get Input

Like the other modules this module will read the values of the elements of the arrays NF(I), ROW(I), COL(I), Doorx(I,J,K), Doory(I,J,K), OCP(I), L(I), R(I), GRL(I), CODE(I), DOOR(I,J,k,l) and the value of the variable page are read from the corresponding cartridge tape.

(ii) Scanned List

(a) Special consideration for document symbol:

As in the Flowline (1) Module the variables Block 1 and Block 2 are used to check if L(I) or R(I) represents the document symbol.

(b) Check List

The value of OCP(I) is checked, if it is one then L(I), R(I) is considered. Like the flowline 1 case, here also the I-1 flowlines are checked. If flowline for R(I), L(I) is found then just an arrow is drawn near the destination door. Next L(I), R(I) is considered.

(c) Numbering of the source and destination symbol.

The source and destination symbols are numbered exactly in the same way as in the flowline (1) case.

(d) Check for roads and doors.

The flowline can leave the source horizontally and enter the source vertically. It also can leave the source vertically and enter horizontally. The source symbol has two horizontal roads H(SX) and H(SX+1), two vertical roads V(SY) and V(SY+1). The destination symbol has two horizontal roads H(DX), H(DX+1) and two vertical roads V(DY), V(DY+1). But if the source is zero i.e. external source then it has only two horizontal roads the same as the destination. H(DY) and H(DY+1). Similarly if the destination is ninety-nine i.e. external destination is used it has only two horizontal roads H(SX) and H(SX+1). Each road cannot be used more than six times. If the road is used for the first time then the line is drawn on the centre of the road. After that when the road is used again the line is shifted from the centre by G1 or G2 according as the road is horizontal or vertical.

G1 = INT(LOG(HROAD(H)+1)/LOG(2))

G2 = INT(LOG(VROAD(V)+1)/LOG(2))

After the roads are chosen, suitable doors are chosen. If the doors are occupied then another set of doors are chosen, and so on.

(e) Draw Lines

After roads and doors are chosen, lines are drawn. Generally the line joining the source and destination symbol consist of the following four parts.

1. Source door to horizontal (or vertical) road.
2. Horizontal (or Vertical) road.

3. Vertical (or Horizontal) road.

4. Vertical (or Horizontal) road to destination.

But if external source or destination is used or the source and destination are in the same row or same column then the line degenerates three parts. In this case the horizontal (or vertical) road, or the vertical (horizontal) road become null.

(f) Draw Arrow

After the lines are drawn an arrow is drawn near the destination symbol.

Then next L(I), R(I) is considered. If I is greater than twenty or L(I) is '40' then the execution of the Scan List part is completed. All the flowlines must be drawn by this time. If the user wants to continue, he can start again from the beginning.

CHAPTER 4
RESULTS AND CONCLUSIONS

In the previous chapters the complete design of the project and the implemented portion have been described in detail. In this chapter the developed system has been evaluated and hints are given for future improvement.

CRITICAL EVALUATION

The developed system can be evaluated in terms of the fulfillment of the stated objectives and the observed performance of the system and supporting software. The page of graphics flowchart has been constructed from the given parametric data representing the configuration of an information processing system. The following examples have been considered to observe the performance of the system.

Example 1

INPUTS

No. of copies	NF(I)	L(J)	R(J)	GRL(J)	CODE(J)
5	1	0	1	0	2
0	2	1	2	2	1
0	3	2	3	0	1
0	4	2	4	0	1
0	5	4	2	0	1
		4	5	0	1
		5	99	0	1
		4	99	0	1

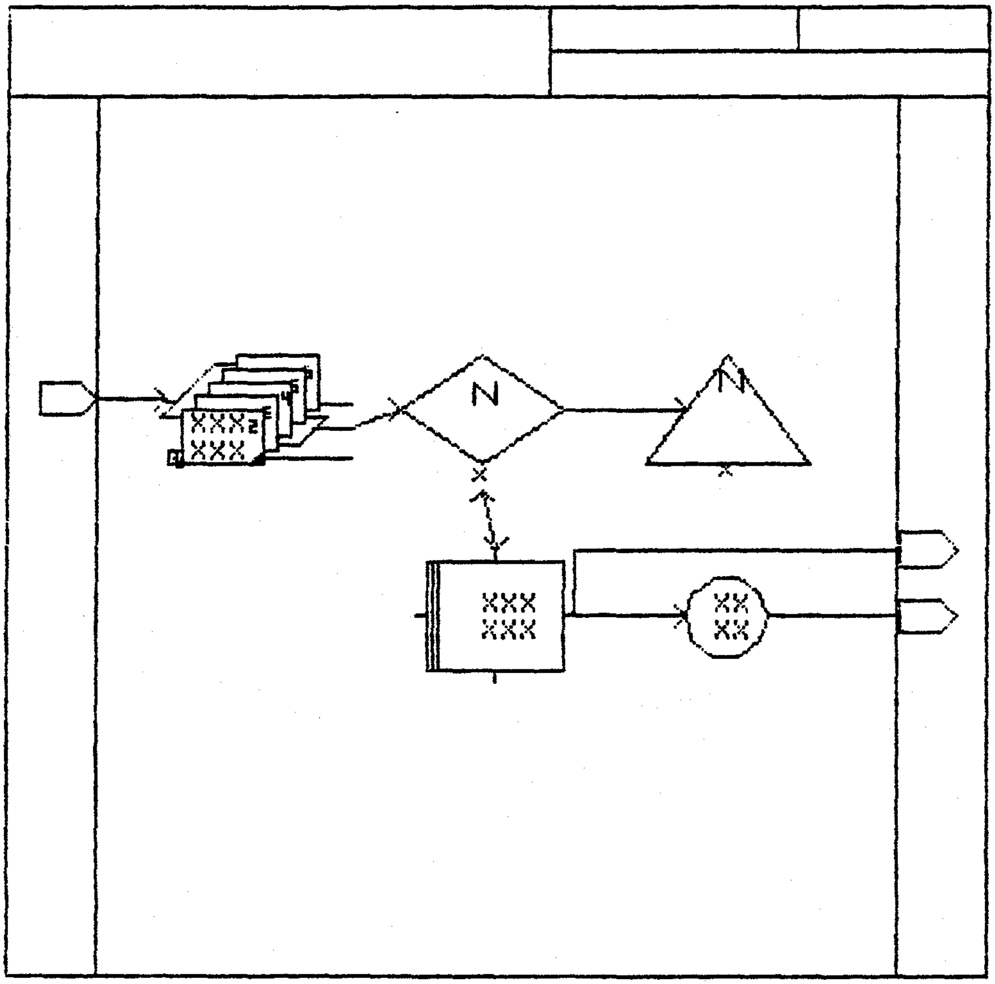


Figure 1

Figure 1 represents the flowchart of the above example. This example consists of five symbols. Most of the flowlines are adjacent here. The allocation of symbols is perfect and all the eight flowlines are clear and distinct. This picture has been drawn on the right page of the screen.

Example 2

INPUTS

No. of copies	NF(I)	L(J)	R(J)	GRL(J)	CODE(J)
9	1	1	2	4	1
0	4	2	4	0	1
0	5	4	1	0	1
0	7	4	8	0	1
0	8	1	5	3	1
0	3	7	5	2	1
5	1	3	5	0	1
0	2	5	3	0	1
		4	5	0	1
		5	1	0	1
		8	2	0	1
		2	8	0	1
		8	6	0	1

The above example consists of eight symbols and thirteen flowlines. Here two types of document symbols have been considered. One is with nine copies and the other is with five copies. The other six symbols are temporary file, permanent file, book, posting source, computer printout, process symbol. Figure 2 represents this example.

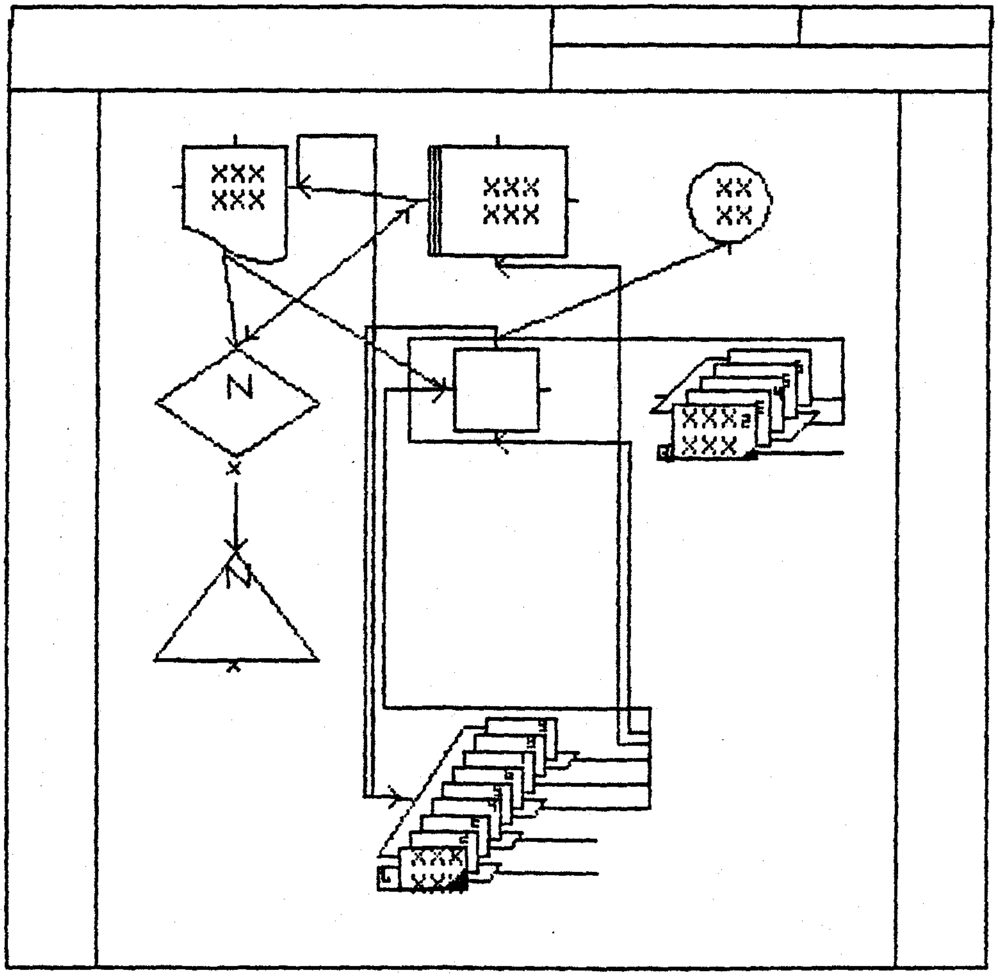


Figure 2

In this example, default values are taken for the dimensions of the shape of different symbols. Here the allocation of the symbols is good. But the flowlines in some cases are not very clear. The flowlines which are coming out from the document symbol with nine copies are touching the symbol and are not distinct. When the source and destination symbols are placed diagonally the arrow heads are not very clear.

Example 3

The same inputs have been used as in example 2. But the dimensions of the shapes of the symbols have been changed. The symbol #1 is with seven copies and symbol #7 with four copies. Here the temporary file and permanent file have been ordered alphabetically. When the circle representing the posting source is made bigger, the resulting arrow heads are not clear.

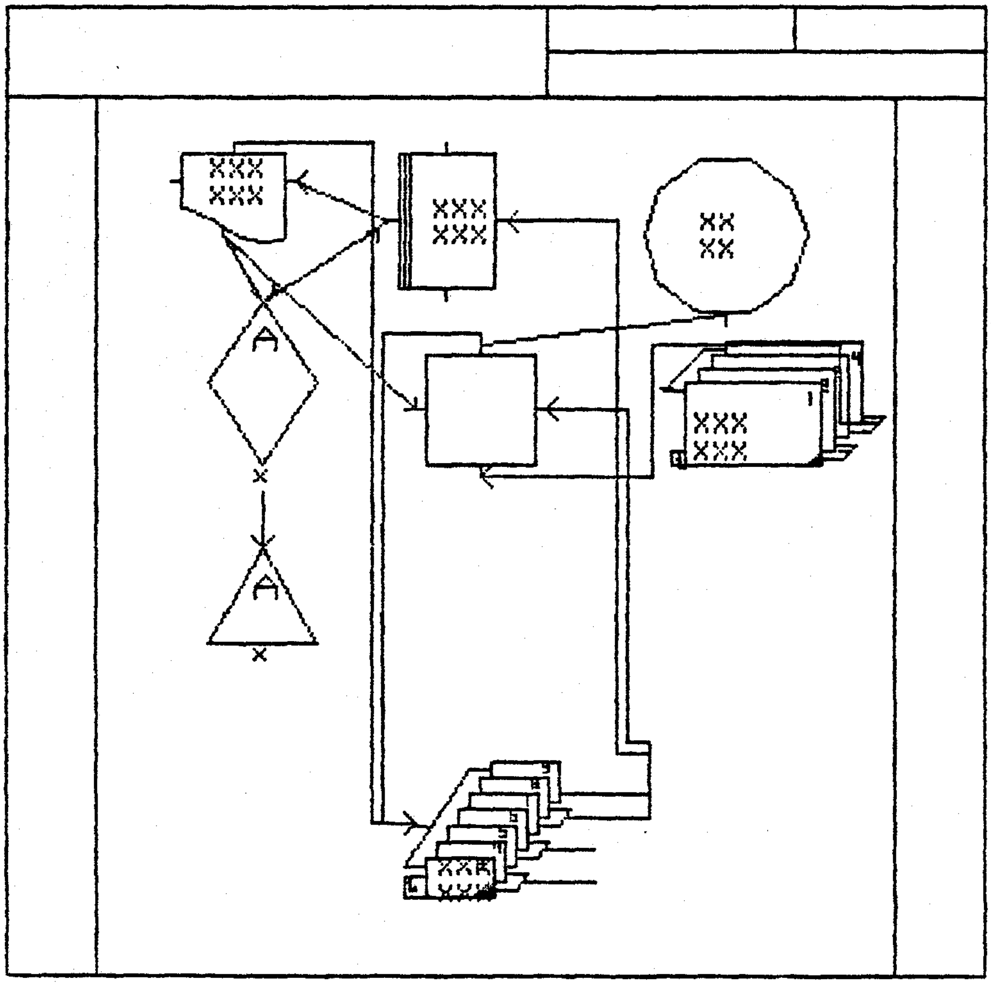


Figure 3

Example 4

INPUTS

No. of copies	NF(I)	L(J)	R(J)	GRL(J)	CODE(J)
5	1	0	1	0	1
0	2	1	2	2	1
0	3	2	3	0	1
0	4	3	4	0	1
0	2	3	1	0	1
0	4	3	5	0	1
0	5	5	6	0	1
5	1	6	7	0	1
0	8	5	8	0	1
0	3	9	2	0	1
0	7	9	5	0	1
0	2	9	6	0	1
		9	10	0	1
		2	12	0	1
		12	11	0	1
		11	8	0	1
		11	9	0	1

Figure 4 represents the results of this example. Here Twelve symbols and seventeen flowlines are used. In this case the allocation of symbols is not bad. But flowlines are not distinct. Flowlines are crowded in one road while the other suitable roads are empty. The arrow heads are clear for most of the lines.

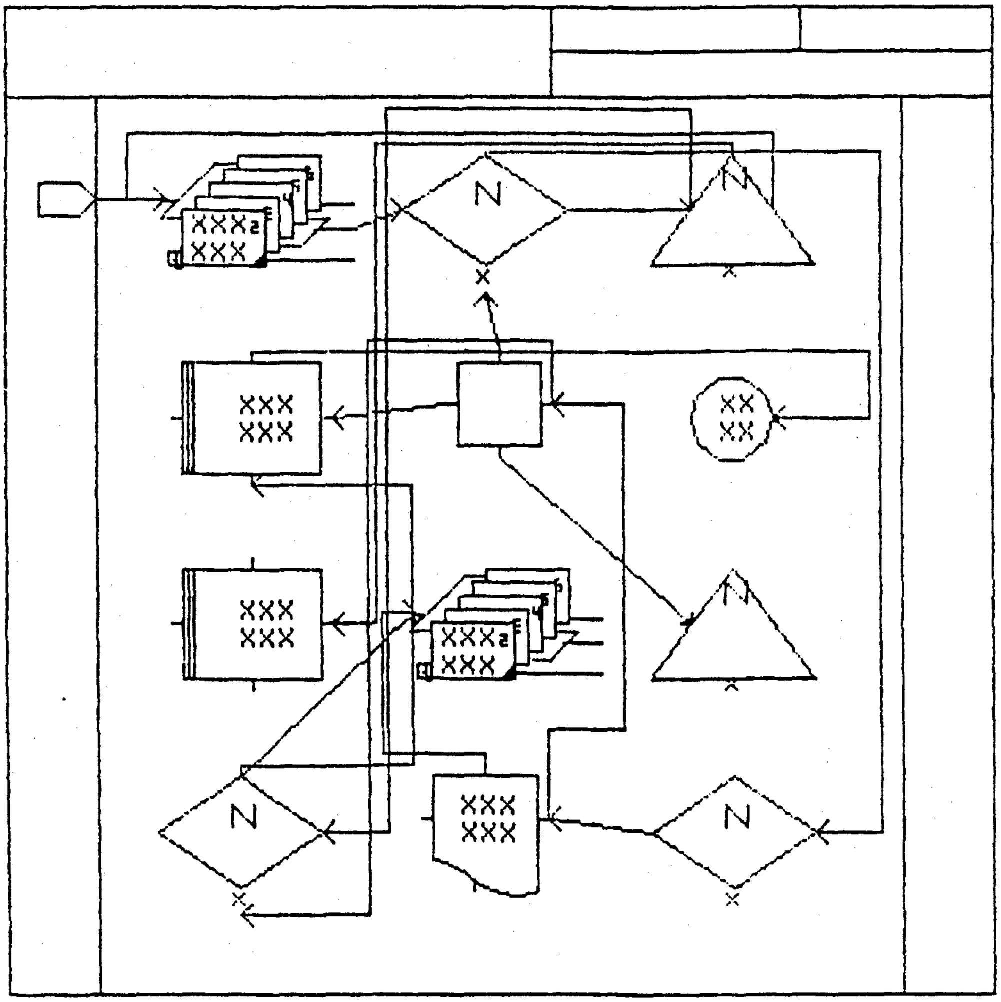


Figure 4

AREAS FOR IMPROVEMENT

(i) Allocation algorithm

The allocation algorithm can be improved and boxes can be allocated to the symbols more effectively.

In this context, effectively means the symbols will be arranged in boxes in such a way that the flowlines will not cross each other to an undue extent, nor will the symbols and flowlines be unduly bunched within a confined area of the page. In other words, the resulting flowchart is legible and balanced.

(ii) Improvement of flowlines

Here different flowlines are crossing each other. In some cases some particular roads are crowded more even if there are other suitable roads for the corresponding flowlines. Improved algorithms can be written to avoid these problems.

(iii) When two symbols are connected to each other in both directions.

Here it has been assumed that when two symbols are connected to each other in both directions the type of flow in both cases is the same: either a physical flow, else a flow of information. Hence, the flowline is drawn once, and in the second case only an arrow head is drawn near the destination symbol. By slightly changing the algorithm the type of flowlines in both cases can be different.

(iv) Document Symbol

It has already been mentioned that there are some problems for the document symbol. It is assumed that the copies belonging to each group are adjacent to each other within the total set of copies; i.e. no copy exists between any two conjugate copies of a group, which belongs to another group. This problem is left open. The copies belonging to a group could in fact be non-contiguous; for example the group #1 consists of copies 2, 4, 6 and the group #2 consists of copies 1 and 3, etc.

The program software is coded in a uniform structure. The structured programming concepts have been used throughout the programs as far as BASIC will allow. So future modification is possible. Hence the problems discussed above can be avoided with further development.

ACHIEVEMENT OF OBJECTIVES

It has already been mentioned in the first chapter that one of the objectives of this project is to construct a page of graphics flowchart from given parametric data representing the configuration of an information processing system. It is clear from the above examples that this has been achieved, although more improvements can be made.

The second objective of this project is to provide a software tool enabling experimentation in the form and dimensions of symbols as well as flowlines and their arrangement on the flowchart page. From example 2 and 3 it is clear that this objective has been achieved.

APPENDIX I

Due to the constraints of the system and not of the author this listing is not in the most satisfactory form. In particular, the overrriding system constraint is the memory available in the BASIC workspace of the HP2647A.

As a consequence, the program is fractured into several components, remarks are used infrequently (because they consume memory), and the functions of the terminal are initiated at the escape sequence (machine) level, rather than by the Advanced Graphics Language (AG.).

This is why the program listing will be hard to read.

APPENDIX I

```
R
 10 COMMAND "SU C"
 20 COMMAND "DI W#1"
 30 PRINT CHR$(27);"h";CHR$(27);"J"
 40 PRINT "****ALLOCATION MODULE****"
 50 RESTORE 70
 60 READ Px,Py,Ph,Pm,H5,H6
 70 DATA 359,359,17,33,198,91
 80 PRINT "IF YOU WANT TO DRAW SYMBOLS ON LEFT PAGE PUT 1 ELSE 0"
 90 INPUT Page
 100 IF Page=0 THEN 140
 110 PRINT CHR$(27);"*m0,0J"
 120 GOSUB 320
 130 GOTO 160
 140 PRINT CHR$(27);"*m360,0J"
 150 GOSUB 320
 160 GOSUB 430
 170 ASSIGN "RIGHT TAPE" TO #1
 180 PRINT #1 USING 240;Nf(1),Nf(2),Nf(3),Nf(4),Nf(5),Nf(6),Nf(7),Nf(8)
 190 PRINT #1 USING 240;Nf(9),Nf(10),Nf(11),Nf(12),Row(1),Row(2),Row(3),Row(4)
 200 PRINT #1 USING 240;Row(5),Row(6),Row(7),Row(8),Row(9),Row(10),Row(11),Row(12)
 210 PRINT #1 USING 240;Col(1),Col(2),Col(3),Col(4),Col(5),Col(6),Col(7),Col(8)
 220 PRINT #1 USING 230;Col(9),Col(10),Col(11),Col(12),Page
 230 IMAGE 2D,",",2D,",",2D,",",2D,",",2D
 240 IMAGE 2D,",",2D,",",2D,",",2D,",",2D,",",2D,",",2D,",",2D,","
 250 PRINT CHR$(27);"h"
 260 COMMAND "RE R"
 270 COMMAND "DI W#4"
 280 PRINT CHR$(27);"h"
 290 COMMAND "F F 6 L"
 300 COMMAND "RES C"
 310 END
 320 REM   ***FRAME***
 330 PRINT CHR$(27);"*pha 0,0";Px;",0";Px;Py;",0";Py;"0,0Z";
 340 Y=Py-2*Ph
 350 PRINT CHR$(27);"*pha";Pm;",0";Pm;Y;"Z";
 360 PRINT CHR$(27);"*pha";Px-Pm;",0";Px-Pm;Y;"Z";
 370 PRINT CHR$(27);"*pha";",0";Y;Px;Y;"Z";
 380 PRINT CHR$(27);"*pha";H5;Py-2*Ph;H5;Py;"Z";
 390 PRINT CHR$(27);"*pha";H5;Py-Ph;Px;Py-Ph;"Z";
 400 PRINT CHR$(27);"*pha";H5+H6;Py-Ph;H5+H6;Py;"Z";
 410 RETURN
 420 REM ****SUBROUTINE FOR ALLOCATION******
 430 DIM C(12)
 440 DIM S(12)
 450 DIM Empty(4)
 460 DIM Mx(12)
 470 DIM Row(12)
 480 DIM Col(12)
 490 DIM C1(3)
 500 DIM R1(4)
 510 DIM P(12,3)
 520 DIM V(12,4)
 530 DIM Nf(12)
 540 DIM L(20)
 550 DIM R(20)
 560 DIM B(4,3)
```

```
570 DIM Sc(12,12)
580 DIM Sik(12)
590 PRINT "IF YOU WANT TO GIVE YOUR OWN INPUT PUT 1 ELSE PUT 0"
600 INPUT Yes
610 IF Yes=1 THEN 760
620 RESTORE 690
630 READ Nf(1),Nf(2),Nf(3),Nf(4),Nf(5),Nf(6),Nf(7),Nf(8),Nf(9),Nf(10)
640 READ Nf(11).Nf(12),L(1),L(2),L(3),L(4),L(5),L(6),L(7).L(8),L(9),L(10)
650 READ L(11),L(12),L(13),L(14),L(15),L(16),L(17),L(18),L(19),L(20)
660 READ R(1),R(2),R(3),R(4),R(5),R(6),R(7),R(8),R(9),R(10),R(11),R(12)
670 READ R(13),R(14),R(15),R(16),R(17),R(18),R(19),R(20),Sik(1),Sik(2),Sik(3)
680 READ Sik(4),Sik(5).Sik(6),Sik(7),Sik(8),Sik(9),Sik(10),Sik(11),Sik(12)
690 DATA 1,2,3,4,5,50,50,50,50,50
700 DATA 50,50,0,1,2,2,4,4,5.4,40,40
710 DATA 40,40,40,40,40,40,40,40,40,40
720 DATA 1,2,3,4,2,5,99,99,40,40,40,40
730 DATA 40,40,40,40,40,40,40,40,5,0,0
740 DATA 0,0,0,0,0,0,0,0,0
750 GOTO 840
760 INPUT Nf(1),Nf(2),Nf(3),Nf(4),Nf(5).Nf(6),Nf(7),Nf(8),Nf(9),Nf(10)
770 INPUT Nf(11),Nf(12)
780 INPUT L(1),L(2),L(3),L(4),L(5),L(6),L(7),L(8),L(9),L(10),L(11),L(12)
790 INPUT L(13),L(14),L(15),L(16),L(17),L(18),L(19),L(20)
800 INPUT R(1),R(2),R(3),R(4),R(5),R(6),R(7),R(8),R(9),R(10),R(11),R(12)
810 INPUT R(13),R(14),R(15),R(16),R(17),R(18),R(19).R(20)
820 INPUT Sik(1),Sik(2),Sik(3),Sik(4),Sik(5),Sik(6),Sik(7),Sik(8)
830 INPUT Sik(9),Sik(10),Sik(11),Sik(12)
840 FOR I=1 TO 12
850 IF Nf(I)=50 THEN 930
860 FOR J=1 TO 20
870 IF L(J)=40 THEN 920
880 IF L(J)=I THEN 900
890 IF R(J)<>I THEN 910
900 C(I)=C(I)+1
910 NEXT J
920 NEXT I
930 Max=0
940 N=0
950 FOR I=1 TO 12
960 IF C(I)<Max THEN 990
970 Max=C(I)
980 N=I
990 NEXT I
1000 V(N,1)=1
1010 V(N,2)=3
1020 V(N,3)=2
1030 V(N,4)=1
1040 P(N,1)=0
1050 P(N,2)=3
1060 P(N,3)=0
1070 S(1)=N
1080 FOR K=1 TO 11
1090 Max=0
1100 FOR I=1 TO 12
1110 FOR T=1 TO 12
1120 IF I=S(T) THEN 1170
1130 NEXT T
1140 IF C(I)<Max THEN 1170
1150 Max=C(I)
1160 N=I
1170 NEXT I
1180 FOR J=1 TO 20
1190 IF N<>L(J) THEN 1210
1200 IF R(J)=99 THEN 1230
1210 NEXT J
1220 GOTO 1310
```

```
1230 P(N,1)=0
1240 P(N,2)=1
1250 P(N,3)=2
1260 V(N,1)=1
1270 V(N,2)=2
1280 V(N,3)=3
1290 V(N,4)=4
1300 GOTO 1510
1310 FOR J=1 TO 20
1320 IF N<>R(J) THEN 1340
1330 IF L(J)=0 THEN 1360
1340 NEXT J
1350 GOTO 1440
1360 P(N,1)=2
1370 P(N,2)=1
1380 P(N,3)=0
1390 V(N,1)=4
1400 V(N,2)=3
1410 V(N,3)=2
1420 V(N,4)=1
1430 GOTO 1510
1440 V(N,1)=4
1450 V(N,2)=1
1460 V(N,3)=3
1470 V(N,4)=2
1480 P(N,1)=0
1490 P(N,2)=2
1500 P(N,3)=0
1510 S(K+1)=N
1520 NEXT K
1530 FOR I=1 TO 12
1540 IF Nf(I)=50 THEN 1650
1550 IF Nf(I)<>8 THEN 1640
1560 Row(I)=2
1570 Col(I)=2
1580 IF B(Row(I),Col(I))=1 THEN 1610
1590 B(Row(I),Col(I))=1
1600 GOTO 1640
1610 IF Row(I)>3 THEN 1650
1620 Row(I)=Row(I)+1
1630 GOTO 1580
1640 NEXT I
1650 FOR I=1 TO 12
1660 IF Nf(I)=50 THEN 1790
1670 IF I=S(1) THEN 1780
1680 FOR J=1 TO 20
1690 IF L(J)=40 THEN 1780
1700 IF I<>L(J) THEN 1740
1710 IF R(J)=99 THEN 1770
1720 Sc(I,R(J))=Sc(I,R(J))+1
1730 GOTO 1770
1740 IF I<>R(J) THEN 1770
1750 IF L(J)=0 THEN 1770
1760 Sc(I,L(J))=Sc(I,L(J))+1
1770 NEXT J
1780 NEXT I
1790 FOR L=1 TO 12
1800 IF Nf(L)=50 THEN 2550
1810 IF B(Row(L),Col(L))=1 THEN 2540
1820 Max=0
1830 FOR J=1 TO 3
1840 IF P(L,J)<Max THEN 1870
1850 Max=P(L,J)
1860 Y=J
1870 NEXT J
1880 Max=0
```

```
1890 FOR A=1 TO 4
1900 IF V(L,A)<Max THEN 1930
1910 Max=V(L,A)
1920 X=A
1930 NEXT A
1940 Max=0
1950 FOR K=1 TO 12
1960 IF Nf(K)=50 THEN 2010
1970 IF Sc(L,K)<Max THEN 2000
1980 Max=Sc(L,K)
1990 At=K
2000 NEXT K
2010 Mx(L)=At
2020 IF L<At THEN 2040
2030 X=Row(At)
2040 IF B(X,Y)=1 THEN 2140
2050 IF Nf(L)=1 AND S1k(L)>5 THEN 2120
2060 Row(L)=X
2070 Col(L)=Y
2080 B(X,Y)=1
2090 IF Nf(L)<>1 OR S1k(L)<=5 THEN 2110
2100 B(X-1,Y)=1
2110 GOTO 2540
2120 IF Row(L)<=1 THEN 2370
2130 IF B(Row(L)-1,Col(L))<>1 THEN 2060
2140 FOR Q=1 TO 3
2150 C1(Q)=0
2160 NEXT Q
2170 FOR D=1 TO 3
2180 C1(D)=Y
2190 Max=0
2200 FOR J=1 TO 3
2210 FOR T=1 TO 3
2220 IF J=C1(T) THEN 2270
2230 NEXT T
2240 IF P(L,J)<Max THEN 2270
2250 Max=P(L,J)
2260 Y=J
2270 NEXT J
2280 IF B(X,Y)=1 THEN 2320
2290 IF Nf(L)=1 AND S1k(L)>5 THEN 2310
2300 GOTO 2060
2310 IF B(X-1,Y)<>1 THEN 2060
2320 NEXT D
2330 FOR Y=1 TO 3
2340 FOR J=1 TO 4
2350 R1(J)=0
2360 NEXT J
2370 FOR E=1 TO 4
2380 R1(E)=X
2390 Max=0
2400 FOR B=1 TO 4
2410 FOR T=1 TO 4
2420 IF B=R1(T) THEN 2470
2430 NEXT T
2440 IF V(L,B)<Max THEN 2470
2450 Max=V(L,B)
2460 X=B
2470 NEXT B
2480 IF B(X,Y)=1 THEN 2520
2490 IF Nf(L)=1 AND S1k(L)>5 THEN 2510
2500 GOTO 2060
2510 IF B(X-1,Y)<>1 THEN 2060
2520 NEXT E
2530 NEXT Y
2540 NEXT L
```

```
2550 FOR I=1 TO 12
2560 IF Nf(I)=50 THEN 2890
2570 IF I=S(1) THEN 2880
2580 FOR J=1 TO 20
2590 IF L(J)=40 THEN 2880
2600 IF L(J)<>I THEN 2740
2610 IF R(J)<>99 THEN 2870
2620 IF Col(I)<>1 THEN 2880
2630 IF B(Row(I),3)=0 THEN 2880
2640 FOR K=1 TO 12
2650 IF Nf(K)=50 THEN 2880
2660 IF K=I THEN 2720
2670 IF Row(K)<>Row(I) THEN 2720
2680 IF Col(K)<>3 THEN 2720
2690 Col(K)=1
2700 Col(I)=3
2710 GOTO 2880
2720 NEXT K
2730 GOTO 2880
2740 IF R(J)<>I THEN 2870
2750 IF L(J)<>0 THEN 2870
2760 IF Col(J)<>3 THEN 2880
2770 IF B(Row(J),1)=0 THEN 2880
2780 FOR K=1 TO 12
2790 IF Nf(K)=50 THEN 2880
2800 IF Row(K)<>Row(I) THEN 2850
2810 IF Col(K)<>1 THEN 2850
2820 Col(K)=3
2830 Col(I)=1
2840 GOTO 2880
2850 NEXT K
2860 GOTO 2880
2870 NEXT J
2880 NEXT I
2890 FOR I=1 TO 12
2900 IF Nf(I)=50 THEN 3240
2910 IF Nf(I)=8 OR Sik(I)>5 THEN 3225
2920 IF ABS(Row(I)-Row(Mx(I)))<=1 THEN 3225
2940 IF Row(Mx(I))=1 THEN 3080
2950 IF B(Row(Mx(I))-1,Col(Mx(I)))=1 THEN 2990
2955 B(Row(I),Col(I))=0
2960 Row(I)=Row(Mx(I))-1
2970 Col(I)=Col(Mx(I))
2980 GOTO 3220
2990 IF Col(Mx(I))=3 THEN 3040
3000 IF B(Row(Mx(I))-1,Col(Mx(I))+1)=1 THEN 3040
3005 B(Row(I),Col(I))=0
3010 Row(I)=Row(Mx(I))-1
3020 Col(I)=Col(Mx(I))+1
3030 GOTO 3220
3040 IF Col(Mx(I))=1 THEN 3080
3050 IF B(Row(Mx(I))-1,Col(Mx(I))-1)=1 THEN 3080
3055 B(Row(I),Col(I))=0
3060 Row(I)=Row(Mx(I))-1
3070 Col(I)=Col(Mx(I))-1
3080 IF Row(Mx(I))=4 THEN 3225
3090 IF B(Row(Mx(I))+1,Col(Mx(I)))=1 THEN 3130
3095 B(Row(I),Col(I))=0
3100 Row(I)=Row(Mx(I))+1
3110 Col(I)=Col(Mx(I))
3120 GOTO 3220
3130 IF Col(Mx(I))=3 THEN 3180
3140 IF B(Row(Mx(I))+1,Col(Mx(I))+1)=1 THEN 3180
3145 B(Row(I),Col(I))=0
3150 Row(I)=Row(Mx(I))+1
3160 Col(I)=Col(Mx(I))+1
```

```
3170 GOTO 3220
3180 IF Col(Mx(I))=1 THEN 3220
3190 IF B(Row(Mx(I))+1,Col(Mx(I))-1)=1 THEN 3220
3195 B(Row(I),Col(I))=0
3200 Row(I)=Row(Mx(I))+1
3210 Col(I)=Col(Mx(I))-1
3220 B(Row(I),Col(I))=1
3225 NEXT I
3240 FOR I=1 TO 4
3250 FOR J=1 TO 12
3260 IF Row(J)=I THEN 3310
3270 NEXT J
3280 Empty(I)=I
3290 IF Empty(I)<3 THEN 3310
3300 Count=Count+1
3310 NEXT I
3320 IF Count<>2 THEN 3370
3330 FOR J=1 TO 12
3340 IF Nf(J)=50 THEN 3370
3350 Row(J)=Row(J)+1
3360 NEXT J
3370 FOR I=1 TO 12
3380 PRINT I,"ROW",Row(I),"COL",Col(I)
3390 NEXT I
3400 RETURN
```

```
R
 10 COMMAND "SU C"
 20 COMMAND "DI W#1"
 30 PRINT CHR$(27);"h";CHR$(27);"J"
 40 PRINT "****SYMBOL DRAWING MODULE****"
 50 GOSUB 2770
 60 FOR I=1 TO 12
 70 FOR J=1 TO 4
 80 PRINT #1 USING 90;Doorx(I,0,J),Doory(I,0,J)
 90 IMAGE 3D,",",3D,","
 100 NEXT J
 110 NEXT I
 120 FOR I=1 TO 12
 130 FOR J=1 TO 5
 140 FOR K=1 TO 4
 150 PRINT #1 USING 160;Doorx(I,J,K),Doory(I,J,K)
 160 IMAGE 3D,",",3D,","
 170 NEXT K
 180 NEXT J
 190 NEXT I
 200 COMMAND "RE R"
 210 COMMAND "DI W#4"
 220 PRINT CHR$(27);"h"
 230 COMMAND "F F 7 L"
 240 COMMAND "RES C"
 250 END
 260 REM ***DOCUMENT***
 270 A=Xb j+Xs1
 280 B=Yb1+Ys1
 290 PRINT CHR$(27);"*pha";A;B;A+S1x;B;A+S1x;B+S1y;A;B+S1y;A;B;"Z";
 300 IF S1k<=1 THEN 350
 310 FOR I=1 TO S1k-1
 320 PRINT CHR$(27);"*pha";A+I*S1dx;B+S1y+(I-1)*S1dy;"Z";
 330 PRINT CHR$(27);"*pg";"0";S1dy;S1x;"0";"0,0";-S1y;-S1dx;"0Z";
 340 NEXT I
 350 IF S1cf=0 THEN 400
 360 FOR J=1 TO S1c
 370 PRINT CHR$(27);"*pha";A+S1x-S1c+J-1;B;A+S1x;B+S1c-J+1;"Z";
 380 IF S1cf=1 THEN 400
 390 NEXT J
 400 IF S1ab=0 THEN 540
 410 PRINT CHR$(27);"*pha";A-S1a;B;A;B;A;B+S1b;A-S1a;B+S1b;A-S1a;B;"Z";
 420 D=A-S1a+S1tx3
 430 E=B+S1b-S1ty3
 440 IF S1ab()1 THEN 480
 450 PRINT CHR$(27);"*pha";D;E;"Z";
 460 PRINT CHR$(27);"*pga2,0,2,4,-2,-4,-1,1Z";
 470 GOTO 540
 480 IF S1ab()2 THEN 520
 490 PRINT CHR$(27);"*pha";D;E;"Z";
 500 PRINT CHR$(27);"*pga0,4,0,-1,a0,-1,-1,0,1,0,0,-3,2,0,0,1Z";
 510 GOTO 540
 520 PRINT CHR$(27);"*pha";D;E;"Z";
 530 PRINT CHR$(27);"*pg2,0,0,2,-2,0,0,2,2,0,0,-1 Z";
 540 REM  ***DIGIT***
 550 FOR I=1 TO S1k
 560 PRINT CHR$(27);"*pha";A+S1x-S1tx2+(I-1)*S1dx;B+S1y-S1ty2+2+(I-1)*S1dy;"Z";
```

```
570 ON Sin(I) GOTO 580,590,600,610,620,630,640,650,660
580 PRINT CHR$(27);"*pga1,0,0,4 Z"\ GOTO 670
590 PRINT CHR$(27);"*pg2,0,-2,0,0,2,2,0,0,2,-2,0Z"\ GOTO 670
600 PRINT CHR$(27);"*pg2,0,-2,0,0,2,2,0,-2,0,0,2,2,0Z"\ GOTO 670
610 PRINT CHR$(27);"*pga4,0,0,4,0,-2,-2,0,0,2Z"\ GOTO 670
620 PRINT CHR$(27);"*pg2,0,0,2,-2,0,0,2,2,0Z"\ GOTO 670
630 PRINT CHR$(27);"*pg2,0,0,2,-2,0,0,-2,0,4,1,0Z"\ GOTO 670
640 PRINT CHR$(27);"*pga0,4,2,0,0,-4Z"\ GOTO 670
650 PRINT CHR$(27);"*pg2,0,0,4,-2,0,0,-2,2,0,-2,0.0,-2Z"\ GOTO 670
660 PRINT CHR$(27);"*pg2,0,0,4,-2,0,0,-2,2,0Z"\ GOTO 670
670 NEXT I
680 PRINT CHR$(27);"*pha";A+Sitxi;B+Siy-Sityi;"Z";
690 PRINT CHR$(27);"*lXXX"
700 PRINT CHR$(27);"*lXXX"
710 PRINT CHR$(27);"*aB";CHR$(27);"*dT";
720 IF Sik<=1 THEN 1170
730 PRINT CHR$(27);"*pha";A;B+Siy-Sidy/2;A-8;B+Siy-Sidy/2;
740 PRINT A+(Sik-1)*Sidx-8;B+Siy+(Sik-3/2)*Sidy;
750 PRINT A+(Sik-1)*Sidx;B+Siy+(Sik-3/2)*Sidy;"Z";
760 SHORT Xx,Yy
770 Xx=A-8+(Sik-1)*Sidx/2
780 Yy=B+(3/4)*Siy+(Sik-3/2)*Sidy/2
790 PRINT CHR$(27);"*pha";Xx-8;Yy;Xx-4;Yy;"Z";
800 K=1
810 FOR Q=1 TO 8
820 IF Scn(Q)=0 THEN 1190
830 IF Scn(Q)<>Scn(Q+1) THEN 860
840 K=K+1
850 GOTO 1080
860 J=Q
870 IF K<>1 THEN 960
880 PRINT CHR$(27);"*pha";A+(J-1)*Sidx+Six;B+(J-1/2)*Sidy;
890 IF J<=5 THEN 920
900 PRINT Xbj+92;B+(J-1/2)*Sidy;"Z";
910 GOTO 940
920 PRINT Xbj+72;B+(J-1/2)*Sidy;"Z";
940 Y(Scn(J))=(J-1/2)*Sidy
950 GOTO 1120
960 PRINT CHR$(27);"*pha";A+(J-K)*Sidx+Six;B+(J-K+1/2)*Sidy;
970 PRINT A+(J-K)*Sidx+Six+8;B+(J-K+1/2)*Sidy;
980 PRINT A+Six+(J-1)*Sidx+8;B+(J-1/2)*Sidy;
990 PRINT A+Six+(J-1)*Sidx;B+(J-1/2)*Sidy;"Z",
1000 Xj=Six+8+(J-(K+1)/2)*Sidx
1010 Y(Scn(J))=(J-K/2)*Sidy
1020 IF J>5 THEN 1050
1030 PRINT CHR$(27);"*pha";A+Xj;B+Y(Scn(J));Xbj+72;B+Y(Scn(J));"Z";
1040 GOTO 1060
1050 PRINT CHR$(27);"*pha";A+Xj;B+Y(Scn(J));Xbj+92;B+Y(Scn(J));"Z";
1060 K=1
1070 GOTO 1120
1080 IF Q<>8 THEN 1150
1090 IF Scn(8)<>Scn(9) THEN 860
1100 J=Q+1
1110 GOTO 960
1120 IF J<>8 THEN 1150
1130 J=Q+1
1140 GOTO 870
1150 NEXT Q
1160 GOTO 1190
1170 PRINT CHR$(27);"*pha";A;B+Siy/2;A-8;B+Siy/2;"Z";
1180 PRINT CHR$(27);"*pha";A+Six;B+(3/4)*Siy;A+Six+8;B+(3/4)*Siy;"Z";
1190 RETURN
1200 REM ***POSTING SOURCE ****
1210 SHORT Phi,Cx,Cy
1220 PRINT CHR$(27);"*pha";Xbj+Xs5+S5;Ybi+Ys5;"Z";
1230 N=10
```

```
1240 FOR I=1 TO N
1250 Phi=6.28319*(I/N)
1260 Cx=S5*COS(Phi)+Xs5+Xbj
1270 IF ABS(Cx)<.01 THEN Cx=0
1280 Cy=S5*SIN(Phi)+Ys5+Ybi
1290 IF ABS(Cy)<.01 THEN Cy=0
1300 PRINT CHR$(27);"*phb";Cx;Cy;"Z";
1310 NEXT I
1320 PRINT CHR$(27);"*pha";Xbj+Xs5-S5t;Ybi+Ys5;"Z";
1330 PRINT CHR$(27);"*lXX"
1340 PRINT CHR$(27);"*lXX"
1350 PRINT CHR$(27);"*aB";CHR$(27);"*dT";
1360 RETURN
1370 REM   ***TEMPORARY FILE***
1380 A=Xbj+Xs2
1390 B=Ybi+Ys2
1400 PRINT CHR$(27);"*pha";A;B;A-S2x;B+S2y;A;B+2*S2y;A+S2x;B+S2y;A;B;"Z";
1410 IF Sk1=0 THEN 1500
1420 PRINT CHR$(27);"*pha";A-S2tx+2;B+2*S2y-S2ty1+2;"Z";
1430 IF Sk1=1 THEN 1470
1440 IF Sk1=2 THEN 1490
1450 PRINT CHR$(27);"*pg0,8,4,0,4,-2,0,-4,-4,-2,-4,0Z";
1460 GOTO 1500
1470 PRINT CHR$(27);"*pg0,4,4,4,4,-4,0,-4,0,4,-8,0Z";
1480 GOTO 1500
1490 PRINT CHR$(27);"*pg0,8,8,-8,0,8Z";
1500 IF Sk2=0 THEN 1590
1510 IF Sk2=1 THEN 1570
1520 PRINT CHR$(27);"*pha";A-S2tx+2;B-S2ty2+4;"Z";
1530 PRINT CHR$(27);"*pg4,4Z";
1540 PRINT CHR$(27);"*pha";A-S2tx+2;B-S2ty2+8;"Z";
1550 PRINT CHR$(27);"*pg4,-4Z";
1560 GOTO 1590
1570 PRINT CHR$(27);"*pha";A-S2tx+2;B-S2ty2+4;"Z";
1580 PRINT CHR$(27);"*pg0,2,1,-2,3,4Z";
1590 RETURN
1600 REM  ***PERMANENT FILE ***
1610 A=Xbj+Xs3
1620 B=Ybi+Ys3
1630 PRINT CHR$(27);"*pha";A-S3x;B;A+S3x;B;A;B+S3y;A-S3x;B;"Z";
1640 IF Sk1=0 THEN 1730
1650 PRINT CHR$(27);"*pha";A-S3tx+2;B+S3y-S3ty1+2;"Z";
1660 IF Sk1=1 THEN 1700
1670 IF Sk1=2 THEN 1720
1680 PRINT CHR$(27);"*pg0,8,4,0,4,-2,0,-4,-4,-2,-4,0Z";
1690 GOTO 1730
1700 PRINT CHR$(27);"*pg0,4,4,4,4,-4,0,-4,0,4,-8,0Z";
1710 GOTO 1730
1720 PRINT CHR$(27);"*pg0,8,8,-8,0,8Z";
1730 IF Sk2=0 THEN 1810
1740 PRINT CHR$(27);"*pha";A-S3tx+2;B-S3ty2+4;"Z";
1750 IF Sk2=1 THEN 1800
1760 PRINT CHR$(27);"*pg4,4Z";
1770 PRINT CHR$(27);"*pha";A-S3tx+2;B-S3ty2+8;"Z";
1780 PRINT CHR$(27);"*pg4,-4Z";
1790 GOTO 1810
1800 PRINT CHR$(27);"*pga 0,2,1,-2,3,4Z";
1810 RETURN
1820 REM ***BOOK****
1830 A=Xbj+Xs4
1840 B=Ybi+Ys4
1850 PRINT CHR$(27);"*pha";A;B;A+S4x;B;A+S4x;B+S4y;A;B+S4y;A;B;"Z";
1860 I=INT(S4a/2)
1870 FOR J=2 TO I STEP 2
1880 PRINT CHR$(27);"*pha";A+J;B;A+J;B+S4y;"Z";
1890 NEXT J
```

```
1900 PRINT CHR$(27);"*pha";A+S4tx;B+S4ty;"Z";
1910 PRINT CHR$(27);"*1XXX"
1920 PRINT CHR$(27);"*1XXX"
1930 PRINT CHR$(27);"*aB";CHR$(27);"*dT";
1940 PRINT CHR$(27);"*pha";Xbj+Xs4-4;Ybi+Ys4+S4y/2;Xbj+Xs4;Ybi+Ys4+S4y/2;"Z";
1950 PRINT CHR$(27);"*pha";Xbj+Xs4+S4x/2;Ybi+Ys4-4;Xbj+Xs4+S4x/2;Ybi+Ys4;"Z";
1960 PRINT CHR$(27);"*pha";Xbj+Xs4+S4x+4;Ybi+Ys4+S4y/2;
1970 PRINT Xbj+Xs4+S4x;Ybi+Ys4+S4y/2;"Z";
1980 PRINT CHR$(27);"*pha";Xbj+Xs4+S4x/2;Ybi+Ys4+S4y+4;
1990 PRINT Xbj+Xs4+S4x/2;Ybi+Ys4+S4y;"Z";
2000 RETURN
2010 REM ***COMPUTER PRINTOUT********
2020 Pi=3.1415925
2030 R=S7x/(2*SIN(Pi/4))
2040 A=Xbj+Xs7
2050 B=Ybi+Ys7-R
2060 K=7
2070 FOR I=1 TO INT((K*2+1)/5)
2080 Ma=Pi/2-(((I-1)*5*Pi)/(8*K))
2090 Mb=Pi/2-((I*5*Pi)/(8*K))
2100 PRINT CHR$(27);"*pha";A+R*COS(Ma);B+R*SIN(Ma);A+R*COS(Mb);B+R*SIN(Mb);"Z";
2110 NEXT I
2120 C=Xbj+Xs7+S7x
2130 D=B+2*R*COS(Pi/4)
2140 FOR I=1 TO INT((K*3)/5)
2150 Ma=5*(Pi/4)+((I-1)*5*Pi/(8*K))
2160 Mb=5*(Pi/4)+(I*5*Pi/(8*K))
2170 PRINT CHR$(27);"*pha";C+R*COS(Ma);D+R*SIN(Ma);C+R*COS(Mb);D+R*SIN(Mb);"Z";
2180 NEXT I
2190 M1=S7x+R*SIN(Pi/8)
2200 Ma=Pi*(5/4)+(INT(K*3/5)/K)*(Pi/8)*5
2210 PRINT CHR$(27);"*pha";C+R*COS(Ma);D+R*SIN(Ma);A+M1;D-R*COS(Pi/8);"Z";
2220 PRINT CHR$(27);"*pha";A;B+R;A;B+R+S7y;A+M1;B+R+S7y;A+M1;D-R*COS(Pi/8);"Z";
2230 PRINT CHR$(27);"*pha";A+S7tx;B+R+S7ty;"Z";
2240 PRINT CHR$(27);"*1XXX"
2250 PRINT CHR$(27);"*1XXX"
2260 PRINT CHR$(27);"*aB";CHR$(27);"*dT";
2270 PRINT CHR$(27);"*pha";Xbj+Xs7-4;Ybi+Ys7+S7y/2;Xbj+Xs7;Ybi+Ys7+S7y/2;"Z";
2280 PRINT CHR$(27);"*pha";A+R*SIN(Pi/4);B+R*COS(Pi/4)-4;
2290 PRINT A+R*SIN(Pi/4);B+R*COS(Pi/4);"Z";
2300 PRINT CHR$(27);"*pha";A+M1+4;Ybi+Ys7+S7y/2;A+M1;Ybi+Ys7+S7y/2;"Z";
2310 PRINT CHR$(27);"*pha";A+M1/2;Ybi+Ys7+S7y;A+M1/2;Ybi+Ys7+S7y+4;"Z";
2320 RETURN
2330 REM ***PRCESS SYMBOL***
2340 PRINT CHR$(27);"*pha";Xbj+Xs8;Ybi+Ys8;Xbj+Xs8+S8i;Ybi+Ys8;
2350 PRINT Xbj+Xs8+S8i;Ybi+Ys8+S8i;Xbj+Xs8;Ybi+Ys8+S8i;Xbj+Xs8;Ybi+Ys8;"Z";
2360 PRINT CHR$(27);"*pha";Xbj+Xs8+S8i/2;Ybi+Ys8+S8i+4;
2370 PRINT Xbj+Xs8+S8i/2;Ybi+Ys8+S8i;"Z";
2380 PRINT CHR$(27);"*pha";Xbj+Xs8-4;Ybi+Ys8+S8i/2;Xbj+Xs8;Ybi+Ys8+S8i/2;"Z";
2390 PRINT CHR$(27);"*pha";Xbj+Xs8+S8i/2;Ybi+Ys8-4;Xbj+Xs8+S8i/2;Ybi+Ys8;"Z";
2400 PRINT CHR$(27);"*pha";Xbj+Xs8+S8i+4;Ybi+Ys8+S8i/2;
2410 PRINT Xbj+Xs8+S8i;Ybi+Ys8+S8i/2;"Z";
2420 RETURN
2430 RESTORE 2480
2440 READ Xs1,Ys1,S1x,S1y,S1dx,S1dy,S1a,S1b,S1c,S1tx1,S1ty1,S1tx2,S1ty2
2450 READ S1tx3,S1ty3,S1k,S1cf,S1ab
2460 READ S1n(1),S1n(2),S1n(3),S1n(4),S1n(5),S1n(6),S1n(7),S1n(8),S1n(9)
2470 READ Scn(1),Scn(2),Scn(3),Scn(4),Scn(5),Scn(6),Scn(7),Scn(8),Scn(9)
2480 DATA 10,10,30,20,5,5,5,5,5,3,10,5,10
2490 DATA 3,5,5,2,2
2500 DATA 2,3,4,5,6,0,0,0,0
2510 DATA 1,2,2,2,3,0,0,0,0
2520 RETURN
2530 RESTORE 2550
2540 READ Xs2,Ys2,S2x,S2y,S2tx,S2ty1,S2ty2,Sk1,Sk2
2550 DATA 30,10,30,20,5,20,10,2,3
```

```
2560 RETURN
2570 RESTORE 2590
2580 READ Xs3,Ys3,S3x,S3y,S3tx,S3ty1,S3ty2,Sk1,Sk2
2590 DATA 30,10,30,40,5,15,8,2,3
2600 RETURN
2610 RESTORE 2630
2620 READ Xs4,Ys4,S4x,S4y,S4a,S4tx,S4ty
2630 DATA 10,10,50,40,10,20,20
2640 RETURN
2650 RESTORE 2670
2660 READ Xs5,Ys5,S5,S5t
2670 DATA 30,30,15,5
2680 RETURN
2690 RESTORE 2710
2700 READ Xs7,Ys7,S7x,S7y,S7tx,S7ty
2710 DATA 10,20,30,30,10,15
2720 RETURN
2730 RESTORE 2750
2740 READ Xs8,Ys8,S81
2750 DATA 20,20,30
2760 RETURN
2770 REM ****DRAW SYMBOLS*****
2780 DIM Nf(12)
2790 DIM Col(12)
2800 DIM Row(12)
2820 DIM Doorx(12,5,4)
2830 DIM Doory(12,5,4)
2840 DIM Y(9)
2850 ASSIGN "RIGHT TAPE" TO #1
2860 READ #1;Nf(1),Nf(2),Nf(3),Nf(4),Nf(5),Nf(6),Nf(7),Nf(8)
2870 READ #1;Nf(9),Nf(10),Nf(11),Nf(12),Row(1),Row(2),Row(3),Row(4)
2880 READ #1;Row(5),Row(6),Row(7),Row(8),Row(9),Row(10),Row(11),Row(12)
2890 READ #1;Col(1),Col(2),Col(3),Col(4),Col(5),Col(6),Col(7),Col(8)
2900 READ #1;Col(9),Col(10),Col(11),Col(12),Page
2910 SHORT A,B,R,Pi,M1,C,D,Ma,Mb
2920 PRINT "EACH TIME BEFORE DRAWING THE SYMBOL IT WILL ASK FOR INPUT"
2930 PRINT "IF YOU WANT TO GIVE YOUR OWN INPUT FOR THE SYMBOL PUT 1 ELSE 0 "
2940 FOR L=1 TO 12
2950 IF Nf(L)=50 THEN 3970
2960 Xbj=54+90*(Col(L)-1)
2970 Ybi=254-75*(Row(L)-1)
2980 IF Page=0 THEN 3000
2990 GOTO 3010
3000 Xbj=Xbj+360
3010 INPUT Yes
3020 IF Yes=1 THEN 3050
3030 ON Nf(L) GOSUB 2430,2530,2570,2610,2650,2840,2690,2730
3040 ON Nf(L) GOTO 3100,3290,3400,3510,3620,3720,3760,3870
3050 ON Nf(L) GOTO 3060,3280,3390,3500,3610,3720,3750,3860
3060 INPUT Xs1,Ys1,S1x,S1y,S1dx,S1dy,S1a,S1b,S1c,S1tx1,S1ty1,S1tx2,S1tv2
3070 INPUT S1tx3,S1ty3,S1k,S1cf,S1ab
3080 INPUT S1n(1),S1n(2),S1n(3),S1n(4),S1n(5),S1n(6),S1n(7),S1n(8),S1n(9)
3090 INPUT Scn(1),Scn(2),Scn(3),Scn(4),Scn(5),Scn(6),Scn(7),Scn(8),Scn(9)
3100 GOSUB 260
3110 Doorx(L,0,1)=Xbj+Xs1-16+(S1k-1)*S1dx/2
3120 Doory(L,0,1)=Ybi+Ys1+(3/4)*S1y+(S1k-3/2)*S1dy/2
3130 Doorx(L,0,2)=Xbj+Xs1-16+(S1k-1)*S1dx/2
3140 Doory(L,0,2)=Ybi+Ys1+(3/4)*S1y+(S1k-3/2)*S1dy/2
3150 FOR J=1 TO 9
3160 IF Scn(J)=0 THEN 3230
3161 IF J<=5 THEN 3170
3162 Doorx(L,Scn(J),3)=Xbj+92
3163 GOTO 3180
3170 Doorx(L,Scn(J),3)=Xbj+72
3180 Doory(L,Scn(J),3)=Ybi+Ys1+Y(Scn(J))
3181 IF J<=5 THEN 3190
```

```
3182 Doorx(L,Scn(J),4)=Xbj+72
3183 GOTO 3200
3190 Doorx(L,Scn(J),4)=Xbj+72
3200 Doory(L,Scn(J),4)=Ybi+Ys1+Y(Scn(J))
3210 NEXT J
3220 GOTO 3960
3230 Doorx(L,0,3)=Xbj+Xs1+8
3240 Doory(L,0,3)=Ybi+Ys1+(3/4)*S1y
3250 Doorx(L,0,4)=Xbj+Xs1
3260 Doory(L,0,4)=Ybi+Ys1+(3/4)*S1y
3270 GOTO 3960
3280 INPUT Xs2,Ys2,S2x,S2y,S2tx,S2ty1,S2ty2,Sk1,Sk2
3290 GOSUB 1370
3300 Doorx(L,0,1)=Xbj+Xs2-S2x
3310 Doory(L,0,1)=Ybi+Ys2+S2y
3320 Doorx(L,0,2)=Xbj+Xs2
3330 Doory(L,0,2)=Ybi+Ys2-S2ty2
3340 Doorx(L,0,3)=Xbj+Xs2+S2x
3350 Doory(L,0,3)=Ybi+Ys2+S2y
3360 Doorx(L,0,4)=Xbj+Xs2
3370 Doory(L,0,4)=Ybi+Ys2+2*S2y
3380 GOTO 3960
3390 INPUT Xs3,Ys3,S3x,S3y,S3tx,S3ty1,S3ty2,Sk1,Sk2
3400 GOSUB 1600
3410 Doorx(L,0,1)=Xbj+Xs3-S3x/2
3420 Doory(L,0,1)=Ybi+Ys3+S3y/2
3430 Doorx(L,0,2)=Xbj+Xs3
3440 Doory(L,0,2)=Ybi+Ys3-S3ty2
3450 Doorx(L,0,3)=Xbj+Xs3+S3x/2
3460 Doory(L,0,3)=Ybi+Ys3+S3y/2
3470 Doorx(L,0,4)=Xbj+Xs3
3480 Doory(L,0,4)=Ybi+Ys3+S3y
3490 GOTO 3960
3500 INPUT Xs4,Ys4.S4x,S4y,S4a,S4tx,S4ty
3510 GOSUB 1820
3520 Doorx(L,0,1)=Xbj+Xs4-4
3530 Doory(L,0,1)=Ybi+Ys4+S4y/2
3540 Doorx(L,0,2)=Xbj+Xs4+S4x/2
3550 Doory(L,0,2)=Ybi+Ys4-4
3560 Doorx(L,0,3)=Xbj+Xs4+S4x+4
3570 Doory(L,0,3)=Ybi+Ys4+S4y/2
3580 Doorx(L,0,4)=Xbj+Xs4+S4x/2
3590 Doory(L,0,4)=Ybi+Ys4+S4y+4
3600 GOTO 3960
3610 INPUT Xs5,Ys5,S5,S5t
3620 GOSUB 1200
3630 Doorx(L,0,1)=Xbj+Xs5-S5
3640 Doory(L,0,1)=Ybi+Ys5
3650 Doorx(L,0,2)=Xbj+Xs5
3660 Doory(L,0,2)=Ybi+Ys5-S5
3670 Doorx(L,0,3)=Xbj+Xs5+S5
3680 Doory(L,0,3)=Ybi+Ys5
3690 Doorx(L,0,4)=Xbj+Xs5
3700 Doory(L,0,4)=Ybi+Ys5+S5
3710 GOTO 3960
3720 REM  ***ADDING MACHINE TAPE ***
3750 INPUT Xs7,Ys7,S7x,S7y,S7tx,S7ty
3760 GOSUB 2010
3770 Doorx(L,0,1)=Xbj+Xs7-4
3780 Doory(L,0,1)=Ybi+Ys7+(S7y/2)
3790 Doorx(L,0,2)=A+R*SIN(Pi/4)
3800 Doory(L,0,2)=B+R*COS(Pi/4)-4
3810 Doorx(L,0,3)=Xbj+Xs7+M1+4
3820 Doory(L,0,3)=Ybi+Ys7+(S7y/2)
3830 Doorx(L,0,4)=Xbj+Xs7+M1/2
3840 Doory(L,0,4)=Ybi+Ys7+S7y+4
3850 GOTO 3960
3860 INPUT Xs8,Ys8.S8i
3870 GOSUB 2330
3880 Doorx(L,0,1)=Xbj+Xs8-4
3890 Doory(L,0,1)=Ybi+Ys8+(S8i/2)
3900 Doorx(L,0,2)=Xbj+Xs8+(S8i/2)
3910 Doory(L,0,2)=Ybi+Ys8-4
3920 Doorx(L,0,3)=Xbj+Xs8+S8i+4
3930 Doory(L,0,3)=Ybi+Ys8+(S8i/2)
3940 Doorx(L,0,4)=Xbj+Xs8+(S8i/2)
3950 Doory(L,0,4)=Ybi+Ys8+S8i+4
3960 NEXT L
3970 RETURN
```

```
R
 1 COMMAND "SU C"
 2 COMMAND "DI W#1"
 3 PRINT CHR$(27);"h";CHR$(27);"J"
 10 PRINT "****FLOW LINE (1) MODULE***x"
 20 GOSUB 5170
 21 ASSIGN "RIGHT TAPE" TO #1
 22 FOR I=1 TO 20
 23 PRINT #1 USING 24;Ocp(I),L(I),R(I).Grl(I),Code(I)
 24 IMAGE 2D,","
 28 NEXT I
 30 FOR I=1 TO 12
 31 FOR J=1 TO 4
 32 FOR K=1 TO 3
 33 PRINT #1 USING 34;Door(I,0,J,K)
 34 IMAGE D,","
 35 NEXT K
 36 NEXT J
 37 NEXT I
 38 FOR M=1 TO 12
 39 FOR I=1 TO 5
 40 FOR J=1 TO 4
 41 FOR K=1 TO 3
 42 PRINT #1 USING 43;Door(M,I,J,K)
 43 IMAGE D,","
 44 NEXT K
 45 NEXT J
 46 NEXT I
 47 NEXT M
 50 COMMAND "RE R"
 60 COMMAND "DI W#4"
 70 PRINT CHR$(27);"h"
 80 COMMAND "F F 8 L"
 90 COMMAND "RES C"
 100 END
 5170 REM ****DRAW FLOW LINES****
 5180 DIM Door(12,5,4,3)
 5190 DIM Doorx(12,5,4)
 5200 DIM Doory(12,5,4)
 5210 DIM Nf(12)
 5220 DIM L(20)
 5230 DIM R(20)
 5240 DIM Code(20)
 5260 DIM Row(12)
 5270 DIM Col(12)
 5290 DIM Rd(20)
 5300 DIM Ld(20)
 5310 DIM Ocp(20)
 5320 DIM Grl(20)
 5325 DIM P(20)
 5330 ASSIGN "RIGHT TAPE" TO #1
 5331 READ #1;Nf(1),Nf(2),Nf(3),Nf(4),Nf(5),Nf(6),Nf(7).Nf(8)
 5332 READ #1;Nf(9),Nf(10),Nf(11),Nf(12),Row(1).Row(2),Row(3),Row(4)
 5333 READ #1;Row(5),Row(6),Row(7),Row(8).Row(9),Row(10),Row(11),Row(12)
 5334 READ #1;Col(1),Col(2),Col(3),Col(4),Col(5),Col(6),Col(7),Col(8)
 5335 READ #1;Col(9),Col(10),Col(11),Col(12),Page
 5350 FOR I=1 TO 12
```

```
5360 FOR J=1 TO 4
5370 READ #1;Doorx(I,0,J)
5375 READ #1;Doory(I,0,J)
5380 NEXT J
5390 NEXT I
5400 FOR I=1 TO 12
5410 FOR J=1 TO 5
5415 FOR K=1 TO 4
5420 READ #1;Doorx(I,J,K)
5425 READ #1;Doory(I,J,K)
5430 NEXT K
5440 NEXT J
5445 NEXT I
5480 PRINT "IF YOU WANT TO GIVE YOUR OWN INPUT PUT 1 ELSE 0"
5490 INPUT Yes
5500 IF Yes=1 THEN 5730
5510 RESTORE 5560
5520 READ L(1),L(2),L(3),L(4),L(5),L(6),L(7),L(8),L(9),L(10),L(11),L(12)
5530 READ R(1),R(2),R(3),R(4),R(5),R(6),R(7),R(8),R(9),R(10),R(11),R(12)
5540 READ Grl(1),Grl(2),Grl(3),Grl(4),Grl(5),Grl(6),Grl(7),Grl(8)
5550 READ Code(1),Code(2),Code(3),Code(4),Code(5),Code(6),Code(7),Code(8)
5560 DATA 0,1,2,2,4,4,5,4,40,40,40,40
5570 DATA 1,2,3,4,2,5,99,99,40,40,40,40
5580 DATA 0,2,0,0,0,0,0,0
5590 DATA 1,1,1,1,1,1,1,1
5600 GOTO 5830
5730 INPUT L(1),L(2),L(3),L(4),L(5),L(6),L(7),L(8),L(9),L(10),L(11),L(12)
5740 INPUT L(13),L(14),L(15),L(16),L(17),L(18),L(19).L(20)
5750 INPUT R(1),R(2),R(3),R(4),R(5),R(6),R(7),R(8),R(9),R(10),R(11),R(12)
5760 INPUT R(13),R(14),R(15),R(16),R(17),R(18),R(19),R(20)
5770 INPUT Grl(1),Grl(2),Grl(3),Grl(4).Grl(5),Grl(6),Grl(7),Grl(8),Grl(9)
5780 INPUT Grl(10),Grl(11),Grl(12),Grl(13),Grl(14),Grl(15),Grl(16),Grl(17)
5790 INPUT Grl(18),Grl(19),Grl(20)
5800 INPUT Code(1),Code(2),Code(3),Code(4),Code(5),Code(6),Code(7),Code(8)
5810 INPUT Code(9),Code(10),Code(11),Code(12),Code(13),Code(14),Code(15)
5820 INPUT Code(16),Code(17),Code(18),Code(19),Code(20)
5830 FOR I=1 TO 20
5835 Rev=0
5840 Block1=0
5850 Block2=0
5860 IF L(I)=40 THEN 8110
5870 IF L(I)=0 THEN 5910
5880 IF Nf(L(I))<>1 THEN 5900
5890 Block1=1
5900 IF R(I)=99 THEN 5930
5910 IF Nf(R(I))<>1 THEN 5930
5920 Block2=1
5930 IF Block1=1 OR Block2=1 THEN 6080
5950 IF I<2 THEN 6080
5960 FOR J=1 TO I-1
5970 IF L(I)<>R(J) OR L(J)<>R(I) THEN 6000
5990 GOTO 6020
6000 NEXT J
6010 GOTO 6080
6020 IF Ocp(J)=0 THEN 6080
6050 Doorx(R(I),0,Rd(I))=Doorx(L(J),0,Ld(J))
6051 Doory(R(I),0,Rd(I))=Doory(L(J),0,Ld(J))
6052 Rd(I)=Ld(J)
6055 W=P(J)
6060 Rev=1
6080 IF L(I)<>0 THEN 6140
6090 Sx=Row(R(I))
6100 Sy=0
6110 GOTO 6170
6140 Sx=Row(L(I))
6150 Sy=Col(L(I))
```

```
6160 IF R(I)=99 THEN 6200
6170 Dx=Row(R(I))
6180 Dy=Col(R(I))
6185 IF Rev=1 THEN 7810
6190 GOTO 6240
6200 Dx=Row(L(I))
6210 Dy=4
6240 Difr=ABS(Dx-Sx)
6250 Difc=ABS(Dy-Sy)
6260 IF Dx<>Sx THEN 6590
6270 IF Difc<>1 THEN 8100
6280 IF Dy<Sy THEN 6500
6290 IF L(I)<>0 THEN 6350
6300 IF Door(R(I),0,1,2)=1 THEN 8100
6310 Rd(I)=1
6315 IF Page=1 THEN 6320
6316 Doorx(L(I),Grl(I),Ld(I))=33+360
6317 GOTO 6330
6320 Doorx(L(I),Grl(I),Ld(I))=33
6330 Doory(L(I),Grl(I),Ld(I))=Doory(R(I),0,Rd(I))
6331 Door(R(I),0,1,2)=1
6332 PRINT CHR$(27);"*pha";Doorx(L(I),Grl(I),Ld(I));Doory(L(I),Grl(I),Ld(I));
6333 PRINT "Z";
6334 PRINT CHR$(27);"*pg-6,-6,-15,0,0,12,15,0,6,-6Z";
6340 GOTO 7808
6350 IF R(I)<>99 THEN 6430
6360 IF Door(L(I),Grl(I),3,2)=1 THEN 8100
6365 IF Page=1 THEN 6370
6366 Px=327+360
6367 GOTO 6380
6370 Px=327
6380 PRINT CHR$(27);"*pha";Doorx(L(I),Grl(I),3);Doory(L(I),Grl(I),3);
6390 PRINT Px;Doory(L(I),Grl(I),3);"Z";
6391 Door(L(I),Grl(I),3,2)=1
6392 Ocp(I)=1
6393 PRINT CHR$(27);"*pha";Px;Doory(L(I),Grl(I),3);"Z";
6394 PRINT CHR$(27);"*pg0,-6,15,0,6,6,-6,6,-15,0,0,-6Z";
6420 GOTO 8100
6430 IF Door(L(I),Grl(I),3,2)=1 OR Door(R(I),0,1,2)=1 THEN 8100
6450 Door(L(I),Grl(I),3,2)=1
6460 Door(R(I),0,1,2)=1
6470 Rd(I)=1
6480 Ld(I)=3
6490 GOTO 7808
6500 IF Block1=1 OR Block2=1 THEN 8100
6520 IF Door(L(I),Grl(I),1,2)=1 OR Door(R(I),0,3,2)=1 THEN 8100
6540 Ld(I)=1
6550 Rd(I)=3
6560 Door(L(I),Grl(I),1,2)=1
6570 Door(R(I),0,3,2)=1
6580 GOTO 7808
6590 IF Dy<>Sy THEN 6780
6600 IF Difr<>1 THEN 8100
6610 IF Dx<Sx THEN 6710
6620 IF Block1=1 OR Block2=1 THEN 8100
6640 IF Door(L(I),Grl(I),2,2)=1 THEN 8100
6650 IF Door(R(I),0,4,2)=1 THEN 8100
6660 Ld(I)=2
6670 Rd(I)=4
6680 Door(L(I),Grl(I),2,2)=1
6690 Door(R(I),0,4,2)=1
6700 GOTO 7808
6710 IF Door(L(I),Grl(I),4,2)=1 OR Door(R(I),0,2,2)=1 THEN 8100
6720 IF Block1=1 OR Block2=1 THEN 8100
6730 Ld(I)=4
6740 Rd(I)=2
```

```
6750 Door(L(I),Grl(I),4,2)=1
6760 Door(R(I),0,2.2)=1
6770 GOTO 7808
6780 IF Difr<>1 OR Difc<>1 THEN 8100
6800 IF Dx<Sx THEN 7360
6810 IF Dy>Sy THEN 7100
6820 IF Block1=1 OR Block2=1 THEN 8100
6840 IF Door(L(I),Grl(I),2,1)=1 THEN 6970
6850 IF Door(R(I),0,4,3)=1 THEN 6910
6851 W=3
6852 P(I)=1
6860 Ld(I)=2
6870 Rd(I)=4
6880 Door(L(I),Grl(I),2,1)=1
6890 Door(R(I),0,4,3)=1
6900 GOTO 7810
6910 IF Door(R(I),0,3,1)=1 THEN 6970
6911 W=1
6912 P(1)=1
6920 Ld(I)=2
6930 Rd(I)=3
6940 Door(L(I),Grl(I),2,1)=1
6950 Door(R(I),0,3,1)=1
6960 GOTO 7810
6970 IF Door(L(I),Grl(I),1,3)=1 THEN 8100
6980 IF Door(R(I),0,4,3)=1 THEN 7040
6981 W=3
6982 P(I)=3
6990 Ld(I)=1
7000 Rd(I)=4
7010 Door(L(I),Grl(I),1,3)=1
7020 Door(R(I),0,4,3)=1
7030 GOTO 7810
7040 IF Door(R(I),0,3,1)=1 THEN 8100
7041 W=1
7042 P(I)=3
7050 Ld(I)=1
7060 Rd(I)=3
7070 Door(L(I),Grl(I),1,3)=1
7080 Door(R(I),0,3,1)=1
7090 GOTO 7810
7100 IF Door(R(I),0,1,3)=1 THEN 7230
7110 IF Door(L(I),Grl(I),2,3)=1 THEN 7170
7111 W=3
7112 P(I)=3
7120 Ld(I)=2
7130 Rd(I)=1
7140 Door(R(I),0,1,3)=1
7150 Door(L(I),Grl(I),2,3)=1
7160 GOTO 7810
7170 IF Door(L(I),Grl(I),3,1)=1 THEN 8100
7171 W=3
7172 P(I)=1
7180 Ld(I)=3
7190 Rd(I)=1
7200 Door(R(I),0,1,3)=1
7210 Door(L(I),Grl(I),3,1)=1
7220 GOTO 7810
7230 IF Door(R(I),0,4,1)=1 THEN 8100
7240 IF Door(L(I),Grl(I),2,3)=1 THEN 7300
7241 W=1
7242 P(I)=3
7250 Ld(I)=2
7260 Rd(I)=4
7270 Door(R(I),0,4,1)=1
7280 Door(L(I),Grl(I),2,3)=1
```

```
7290 GOTO 7810
7300 IF Door(L(I),Grl(I),3,1)=1 THEN 8100
7301 W=1
7302 P(I)=1
7310 Ld(I)=3
7320 Rd(I)=2
7330 Door(L(I),Grl(I),3,1)=1
7340 Door(R(I),0,2,1)=1
7350 GOTO 7810
7360 IF Dy>Sy THEN 7660
7370 IF Block1=1 THEN 7520
7380 IF Door(L(I),Grl(I),1,3)=1 THEN 7520
7390 IF Door(R(I),0,2,1)=1 THEN 7450
7391 W=1
7392 P(I)=3
7400 Ld(I)=1
7410 Rd(I)=2
7420 Door(L(I),Grl(I),1,3)=1
7430 Door(R(I),0,2,1)=1
7440 GOTO 7810
7450 IF Block2=1 THEN 7520
7460 IF Door(R(I),0,3,3)=1 THEN 7520
7461 W=3
7462 P(I)=3
7470 Ld(I)=1
7480 Rd(I)=3
7490 Door(L(I),Grl(I),1,3)=1
7500 Door(R(I),0,3,3)=1
7510 GOTO 7810
7520 IF Door(L(I),Grl(I),4,1)=1 THEN 8100
7530 IF Door(R(I),0,2,3)=1 THEN 7590
7531 W=3
7532 P(I)=1
7540 Ld(I)=4
7550 Rd(I)=2
7560 Door(L(I),Grl(I),4,1)=1
7570 Door(R(I),0,2,3)=1
7580 GOTO 7810
7590 IF Block2=1 THEN 8100
7600 IF Door(R(I),0,3,3)=1 THEN 8100
7601 W=3
7602 P(I)=1
7610 Ld(I)=4
7620 Rd(I)=3
7630 Door(L(I),Grl(I),4,1)=1
7640 Door(R(I),0,3,3)=1
7650 GOTO 7810
7660 IF Door(L(I),Grl(I),4,3)=1 THEN 7790
7670 IF Door(R(I),0,1,3)=1 THEN 7730
7671 W=3
7672 P(I)=3
7680 Ld(I)=4
7690 Rd(I)=1
7700 Rd(I)=1
7710 Door(R(I),0,1,3)=1
7715 Door(L(I),Grl(I),4,3)=1
7720 GOTO 7810
7730 IF Door(R(I),0,2,1)=1 THEN 7790
7731 W=1
7732 P(I)=3
7740 Ld(I)=4
7750 Rd(I)=2
7760 Door(L(I),Grl(I),4,3)=1
7770 Door(R(I),0,2,1)=1
7780 GOTO 7810
7790 IF Door(L(I),Grl(I),3,1)=1 THEN 8100
```

```
7791 IF Door(R(I),0,2,1)=1 THEN 7800
7792 Ld(I)=3
7793 Rd(I)=2
7794 Door(L(I),Grl(I),3,1)=1
7795 Door(R(I),0,2.1)=1
7796 W=1
7798 P(I)=1
7799 GOTO 7810
7800 IF Door(R(I),0,1,3)=1 THEN 8100
7801 Ld(I)=3
7802 Rd(I)=1
7803 Door(L(I),Grl(I),3,1)=1
7804 Door(R(I),0,1,3)=1
7805 P(I)=1
7806 W=3
7807 GOTO 7810
7808 P(I)=2
7809 W=2
7810 Ocp(I)=1
7811 Ct=0
7812 Kx=0
7813 Ky=0
7817 FOR J=1 TO 3
7818 IF Door(R(I),0,Rd(I),J)<>1 THEN 7820
7819 Ct=Ct+1
7820 NEXT J
7821 IF Ct<=1 THEN 7847
7822 ON Rd(I) GOTO 7823,7829,7834,7839
7823 ON W GOTO 7824,7826,7827
7824 Kx=-4
7825 Ky=4\ GOTO 7847
7826 Kx=-4\ GOTO 7847
7827 Kx=-4
7828 Ky=-4\ GOTO 7847
7829 ON W GOTO 7830,7831,7832
7830 GOTO 7827
7831 Ky=-4\ GOTO 7847
7832 Kx=4
7833 Ky=-4\ GOTO 7847
7834 ON W GOTO 7835,7836,7837
7835 GOTO 7832
7836 Kx=4\ GOTO 7847
7837 Kx=4
7838 Ky=4\ GOTO 7847
7839 ON W GOTO 7840,7841,7842
7840 GOTO 7824
7841 Ky=4\ GOTO 7847
7842 GOTO 7837
7847 IF Rev=1 THEN 7860
7852 IF Code(I)=1 THEN 7854
7853 PRINT CHR$(27);"*m7b";
7854 PRINT CHR$(27);"*pha";Doorx(L(I),Grl(I),Ld(I));Doory(L(I),Grl(I),Ld(I));
7855 PRINT Doorx(R(I),0,Rd(I));Doory(R(I),0,Rd(I));"Z";
7860 PRINT CHR$(27);"*pha";Doorx(R(I),0,Rd(I))+Kx;Doory(R(I),0,Rd(I))+Ky;"Z";
7870 IF Sx<>Dx THEN 7930
7880 IF Sy<Dy THEN 7910
7890 PRINT CHR$(27);"*pg4,4,-4,-4,4,-4Z";
7900 GOTO 8100
7910 PRINT CHR$(27);"*pg-4,4,4,-4,-4,-4Z";
7920 GOTO 8100
7930 IF Sy<>Dy THEN 8000
7940 IF Sx<Dx THEN 7970
7950 PRINT CHR$(27);"*pg-4,-4,4,4,4,-4Z";
7960 GOTO 8100
7970 PRINT CHR$(27);"*pg-4,4,4,-4,4,4Z";
7980 GOTO 8100

 8000 IF Sx>Dx THEN 8060
 8010 IF Sy<Dy THEN 8040
 8020 PRINT CHR$(27);"*pg4,0,-4,0,0,4Z";
 8030 GOTO 8100
 8040 PRINT CHR$(27);"*pg-4,0,4,0,0,4Z";
 8050 GOTO 8100
 8060 IF Sy)Dy THEN 8090
 8070 PRINT CHR$(27);"*pg-4,0,4,0,0,-4Z";
 8080 GOTO 8100
 8090 PRINT CHR$(27);"*pg4,0,-4,0,0,-4Z";
 8100 NEXT I
 8110 RETURN
```

```
R
1 COMMAND "SU C"
2 COMMAND "DI W#1"
3 PRINT CHR$(27);"h";CHR$(27);"J"
10 PRINT "****FLOW LINE(2) MODULE****"
20 GOSUB 80
21 PRINT "THIS IS THE LAST MODULE"
22 PRINT "IF YOU WANT YOU CAN CONTINUE AGAIN"
25 COMMAND "RE R"
30 COMMAND "DI W#4"
40 PRINT CHR$(27);"h"
50 COMMAND "F F 5 L"
60 COMMAND "RES C"
70 END
80 REM ***DRAW FLOW LINES FOR REMOTE CASES***
85 SHORT X,Y,G
90 DIM Door(12,5,4,3)
100 DIM Doorx(12,5,4)
110 DIM Doory(12,5,4)
120 DIM Nf(12)
130 DIM L(20)
140 DIM R(20)
150 DIM Code(20)
160 DIM Hroad(5)
170 DIM Vroad(5)
180 DIM Row(12)
190 DIM Col(12)
200 DIM Rd(20)
210 DIM Ld(20)
220 DIM Ocp(20)
230 DIM Grl(20)
235 DIM Tip(20)
240 ASSIGN "RIGHT TAPE" TO #1
241 READ #1;Nf(1),Nf(2),Nf(3),Nf(4),Nf(5),Nf(6),Nf(7),Nf(8)
242 READ #1;Nf(9),Nf(10),Nf(11),Nf(12),Row(1),Row(2),Row(3),Row(4)
243 READ #1;Row(5),Row(6),Row(7),Row(8),Row(9),Row(10),Row(11),Row(12)
244 READ #1;Col(1),Col(2),Col(3),Col(4),Col(5),Col(6),Col(7),Col(8)
245 READ #1;Col(9),Col(10),Col(11),Col(12),Page
250 FOR I=1 TO 12
260 FOR J=1 TO 4
270 READ #1;Doorx(I,0,J)
275 READ #1;Doory(I,0,J)
280 NEXT J
290 NEXT I
300 FOR I=1 TO 12
310 FOR J=1 TO 5
315 FOR K=1 TO 4
320 READ #1;Doorx(I,J,K)
325 READ #1;Doory(I,J,K)
330 NEXT K
340 NEXT J
345 NEXT I
380 FOR I=1 TO 20
390 READ #1;Ocp(I),L(I),R(I),Grl(I),Code(I)
400 NEXT I
620 FOR I=1 TO 12
621 FOR J=1 TO 4
```

```
622 FOR K=1 TO 3
623 READ #1;Door(I,0,J,K)
624 NEXT K
625 NEXT J
626 NEXT I
627 FOR M=1 TO 12
628 FOR I=1 TO 5
629 FOR J=1 TO 4
630 FOR K=1 TO 3
631 READ #1;Door(M,I,J,K)
632 NEXT K
633 NEXT J
634 NEXT I
635 NEXT M
637 ON END #1 GOTO 740
740 FOR I=1 TO 20
750 Block1=0
760 Block2=0
770 IF L(I)=40 THEN 3610
780 IF Ocp(I)=1 THEN 3600
790 IF Nf(L(I))<>1 THEN 810
800 Block1=1
810 IF R(I)=99 THEN 840
820 IF Nf(R(I))<>1 THEN 840
830 Block2=1
840 IF Block1=1 OR Block2=1 THEN 960
850 IF I<2 THEN 960
860 FOR J=1 TO I-1
870 IF L(I)<>R(J) OR L(J)<>R(I) THEN 890
880 GOTO 910
890 NEXT J
900 GOTO 960
910 Ocp(I)=1
920 PRINT CHR$(27);"*pha";Doorx(L(J),0,Ld(J));Doory(L(J),0,Ld(J));"Z";
925 Rd(I)=Ld(J)
930 ON Tip(J) GOTO 3510,3560
960 IF L(I)<>0 THEN 1000
970 Sx=Row(R(I))
980 Sy=0
990 GOTO 1030
1000 Sx=Row(L(I))
1010 Sy=Col(L(I))
1020 IF R(I)=99 THEN 1060
1030 Dx=Row(R(I))
1040 Dy=Col(R(I))
1050 GOTO 1080
1060 Dx=Row(L(I))
1070 Dy=4
1080 IF Hroad(Sx)>6 THEN 1690
1090 IF R(I)=99 THEN 1110
1100 IF Vroad(Dy+1)>6 THEN 1350
1110 IF Door(L(I),Grl(I),4,2)=1 THEN 1250
1120 T=1
1130 IF R(I)=99 THEN 1310
1131 IF Door(R(I),0,3,2)=1 OR Block2=1 THEN 1210
1140 Rd(I)=3
1150 Door(R(I),0,3,2)=1
1160 GOTO 1310
1210 IF Door(R(I),0,2,3)=1 OR Block2=1 THEN 1350
1220 Rd(I)=2
1230 Door(R(I),0,2,3)=1
1240 GOTO 1310
1250 IF Door(L(I),Grl(I),3,1)=1 THEN 1280
1260 T=2
1270 GOTO 1130
1280 IF Door(L(I),Grl(I),1,1)=1 OR Block1=1 THEN 1690
```

```
1290 T=3
1300 GOTO 1130
1310 IF P=1 THEN 1830
1320 V=Dy+1
1330 GOTO 1580
1350 IF Vroad(Dy)>6 THEN 2200
1360 IF Door(L(I),Grl(I),4,2)=1 THEN 1500
1365 T=1
1370 IF R(I)=99 THEN 1560
1371 IF Door(R(I),0,1,2)=1 THEN 1410
1380 Rd(I)=1
1390 Door(R(I),0,1,2)=1
1400 GOTO 1560
1410 IF Door(R(I),0,2,1)=1 THEN 1500
1420 Rd(I)=2
1430 Door(R(I),0,2,1)=1
1450 GOTO 1560
1500 IF Door(L(I),Grl(I),3,1)=1 THEN 1530
1510 T=2
1520 GOTO 1370
1530 IF Door(L(I),Grl(I),1,1)=1 OR Block1=1 THEN 1690
1540 T=3
1550 GOTO 1370
1560 IF P=2 THEN 1970
1570 V=Dy
1580 H=Sx
1590 ON T GOTO 1600,1630,1660
1600 Ld(I)=4
1610 Door(L(I),Grl(I),4,2)=1
1620 GOTO 2090
1630 Ld(I)=3
1640 Door(L(I),Grl(I),3,1)=1
1650 GOTO 2090
1660 Ld(I)=1
1670 Door(L(I),Grl(I),1,1)=1
1680 GOTO 2090
1690 IF Hroad(Sx+1)>6 THEN 2200
1700 IF R(I)=99 THEN 1720
1710 IF Vroad(Dy+1)>6 THEN 1860
1720 P=1
1730 IF Door(L(I),Grl(I),2,2)=1 OR Block1=1 THEN 1760
1740 T=1
1750 GOTO 1130
1760 IF Door(L(I),Grl(I),1,3)=1 OR Block1=1 THEN 1800
1770 T=2
1790 GOTO 1130
1800 IF Door(L(I),Grl(I),3,3)=1 THEN 2200
1810 T=3
1820 GOTO 1130
1830 V=Dy+1
1840 GOTO 1980
1860 IF R(I)=99 THEN 1870
1861 IF Vroad(Dy)>6 THEN 2200
1870 P=2
1880 IF Door(L(I),Grl(I),2,2)=1 OR Block1=1 THEN 1910
1890 T=1
1900 GOTO 1370
1910 IF Door(L(I),Grl(I),1,3)=1 OR Block1=1 THEN 1930
1920 T=2
1930 GOTO 1370
1940 IF Door(L(I),Grl(I),3,3)=1 THEN 2200
1950 T=3
1960 GOTO 1370
1970 V=Dy
1980 H=Sx+1
1990 ON T GOTO 2010,2040,2070
```

```
2010 Ld(I)=2
2020 Door(L(I),Gr1(I),2,2)=1
2030 GOTO 2090
2040 Ld(I)=1
2050 Door(L(I),Gr1(I),1,3)=1
2060 GOTO 2090
2070 Ld(I)=3
2080 Door(L(I),Gr1(I),3,3)=1
2090 Tip(I)=1
2100 GOTO 3270
2200 IF L(I)=0 THEN 2210
2205 IF Vroad(Sy)>6 THEN 2790
2210 IF Hroad(Dx+1)>6 THEN 2450
2220 IF Door(L(I),Gr1(I),1,2)=1 OR Block1=1 THEN 2360
2230 T=1
2240 IF R(I)=99 THEN 2420
2241 IF Door(R(I),0,2,2)=1 THEN 2280
2250 Rd(I)=2
2260 Door(R(I),0,2,2)=1
2270 GOTO 2420
2280 IF Door(R(I),0,1,3)=1 THEN 2320
2290 Rd(I)=1
2300 Door(R(I),0,1,3)=1
2320 IF Door(R(I),0,3,3)=1 OR Block2=1 THEN 2450
2330 Rd(I)=3
2340 Door(R(I),0,3,3)=1
2350 GOTO 2420
2360 IF Door(L(I),Gr1(I),2,1)=1 OR Block1=1 THEN 2390
2370 T=2
2380 GOTO 2240
2390 IF Door(L(I),Gr1(I),4,1) THEN 2790
2400 T=3
2410 GOTO 2240
2420 IF P=1 THEN 3020
2430 H=Dx+1
2440 GOTO 2680
2450 IF Hroad(Dx)>6 THEN 3600
2460 IF Door(L(I),Gr1(I),1,2)=1 OR Block1=1 THEN 2600
2470 T=1
2480 IF R(I)=99 THEN 2660
2481 IF Door(R(I),0,4,2)=1 OR Block2=1 THEN 2520
2490 Rd(I)=4
2500 Door(R(I),0,4,2)=1
2510 GOTO 2660
2520 IF Door(R(I),0,3,1)=1 OR Block2=1 THEN 2560
2530 Rd(I)=3
2540 Door(R(I),0,3,1)=1
2550 GOTO 2660
2560 IF Door(R(I),0,1,1)=1 THEN 3600
2570 Rd(I)=1
2580 Door(R(I),0,1,1)=1
2590 GOTO 2660
2600 IF Door(L(I),Gr1(I),2,1)=1 OR Block1=1 THEN 2630
2610 T=2
2620 GOTO 2480
2630 IF Door(L(I),Gr1(I),4,1)=1 THEN 2790
2640 T=3
2650 GOTO 2480
2660 IF P=2 THEN 3150
2670 H=Dx
2680 V=Sy
2690 ON T GOTO 2700,2730,2760
2700 Ld(I)=1
2710 Door(L(I),Gr1(I),1,2)=1
2720 GOTO 3260
2730 Ld(I)=2
```

```
2740 Door(L(I),Grl(I),2,1)=1
2750 GOTO 3260
2760 Ld(I)=4
2770 Door(L(I),Grl(I),4,1)=1
2780 GOTO 3260
2790 IF L(I)=0 THEN 2810
2800 IF Vroad(Sy+1))6 THEN 3600
2810 IF Hroad(Dx+1))6 THEN 3040
2820 P=1
2830 IF Door(L(I),Grl(I),3,2)=1 THEN 2860
2840 T=1
2850 GOTO 2240
2860 IF Door(L(I),Grl(I),4,3)=1 THEN 2890
2870 T=2
2880 GOTO 2240
2890 IF Door(L(I),Grl(I),2,3)=1 OR Block1=1 THEN 3600
3000 T=3
3010 GOTO 2240
3020 H=Dx+1
3030 GOTO 3160
3040 IF Hroad(Dx))6 THEN 3600
3050 P=2
3060 IF Door(L(I),Grl(I),3,2)=1 THEN 3090
3070 T=1
3080 GOTO 2480
3090 IF Door(L(I),Grl(I),4,3)=1 THEN 3120
3100 T=2
3110 GOTO 2480
3120 IF Door(L(I),Grl(I),2,3)=1 OR Block1=1 THEN 3600
3130 T=3
3140 GOTO 2480
3150 H=Dx
3160 V=Sy+1
3170 ON T GOTO 3180,3210,3240
3180 Ld(I)=3
3190 Door(L(I),Grl(I),3,2)=1
3200 GOTO 3260
3210 Ld(I)=4
3220 Door(L(I),Grl(I),4,3)=1
3230 GOTO 3260
3240 Ld(I)=2
3250 Door(L(I),Grl(I),2,3)=1
3260 Tip(I)=2
3270 Ocp(I)=1
3280 G1=INT(LOG(Hroad(H)+1)/LOG(2))
3285 G2=INT(LOG(Vroad(V)+1)/LOG(2))
3290 V1=8+(5-H)*75
3291 IF Page=1 THEN 3300
3292 H1=360+V*90-46
3293 GOTO 3310
3300 H1=V*90-46
3310 Hroad(H)=Hroad(H)+1
3320 Vroad(V)=Vroad(V)+1
3330 IF Hroad(H)=1 THEN 3360
3340 Y=V1+18*(((Hroad(H)+(3/2))/2^G1)-(3/2))
3350 GOTO 3370
3360 Y=V1
3370 IF Vroad(V)=1 THEN 3400
3380 X=H1+18*(((Vroad(V)+(3/2))/2^G2)-(3/2))
3390 GOTO 3401
3400 X=H1
3401 IF Page=0 THEN 3405
3402 Q3=33
3403 Q4=327
3404 GOTO 3410
3405 Q3=33+360
```

```
3406 Q4=327+360
3410 IF Tip(I)=2 THEN 3450
3411 IF L(I)=0 THEN 3441
3420 PRINT CHR$(27);"*pha";Doorx(L(I),Grl(I),Ld(I));Doory(L(I),Grl(I),Ld(I));
3421 IF R(I)<>99 THEN 3430
3422 PRINT Doorx(L(I),Grl(I),Ld(I));Y;Q4;Y;"Z";
3423 GOTO 3595
3430 PRINT Doorx(L(I),Grl(I),Ld(I));Y;X;Y;X;Doory(R(I),0,Rd(I));
3435 PRINT Doorx(R(I),0,Rd(I));Doory(R(I),0,Rd(I));"Z";
3440 GOTO 3470
3441 PRINT CHR$(27);"*pha";Q3;Doory(R(I),0,Rd(I));Doorx(R(I),0,Rd(I));Y;
3442 PRINT Doorx(R(I),0,Rd(I));Doory(R(I),0,Rd(I));"Z";
3443 GOTO 3592
3450 IF L(I)=0 THEN 3462
3451 PRINT CHR$(27);"*pha";Doorx(L(I),Grl(I),Ld(I));Doory(L(I),Grl(I),Ld(I));
3452 IF R(I)<>99 THEN 3455
3453 PRINT X;Doory(L(I),Grl(I),Ld(I));X;Y;Q4;Y;"Z";
3454 GOTO 3595
3455 PRINT X;Doory(L(I),Grl(I),Ld(I));X;Y;Doorx(R(I),0,Rd(I));Y;
3460 PRINT Doorx(R(I),0,Rd(I));Doory(R(I),0,Rd(I));"Z";
3461 GOTO 3470
3462 PRINT CHR$(27);"*pha";Q3;Doory(Rd(I),0,Rd(I));X;Doory(R(I),0,Rd(I));
3463 PRINT X;Y;Doorx(R(I),0,Rd(I));Y;Doorx(R(I),0,Rd(I));Doory(R(I),0,Rd(I);"Z"
3464 GOTO 3592
3470 PRINT CHR$(27);"*pha";Doorx(R(I),0,Rd(I));Doory(R(I),0,Rd(I));"Z";
3480 IF Tip(I)=1 THEN 3560
3510 IF Rd(I)=2 THEN 3540
3520 PRINT CHR$(27);"*pg-4,4,4,-4,4,4Z";
3530 GOTO 3600
3540 PRINT CHR$(27);"*pg-4,-4,4,4,4,-4Z";
3550 GOTO 3600
3560 IF Rd(I)=1 THEN 3590
3561 IF Rd(I)=2 AND Block2=1 THEN 3590
3570 PRINT CHR$(27);"*pg4,-4,-4,4,4,4Z";
3580 GOTO 3600
3590 PRINT CHR$(27);"*pg-4,-4,4,4,-4,4Z";
3591 GOTO 3600
3592 PRINT CHR$(27);"*pha";Q3;Doory(R(I),0,Rd(I));"Z";
3593 PRINT CHR$(27);"*pg-6,-6,-15,0,0,12,15,0,6,-6Z";
3594 GOTO 3470
3595 PRINT CHR$(27);"*pha";Q4;Y;"Z";
3596 PRINT CHR$(27);"*pg0,-6,15,0,6,6,-6,6,-15,0,0,-6Z";
3600 NEXT I
3610 RETURN
```

APPENDIX II

Glossary of Variables

Variable Name	Description
B	Array of dimension 4x3 which indicates that the box is occupied or not. B(I,J) corresponds to the box in I^{th} row and J^{th} column.
Block1	A variable used to indicate whether there is a document symbol in the left side of a line in the flowline list.
Block2	A variable used to indicate whether there is a document symbol in the right side of a line in the flowline list.
C	Array of dimension 12 which indicates the number of times a symbol appears in the flowline list. The element C(I) corresponds to the number of times the symbol I appears in the flowline list.
C1	Array of dimension 3 which is used to find the column for the symbol.

Variable Name	Description
COL	Array of dimension 12 which indicates the column in which the symbol should be drawn. Col(I) corresponds to the column in which symbol I should be drawn.
Code	Array of dimension 20 which indicates if the flowline should be solid or dotted. Code(I) corresponds to the type in which I^{th} flowline should be drawn.
Door	Array of dimension 12x5x4x3 which indicates that a flowline is drawn from the door or not. Door(I,J,K,L) corresponds to the path L of door K of group J of symbol I is occupied or not.
Doorx(12,5,4)	Array of dimension 12x5x4 which indicates the x-co-ordinate of the door. Door(I,J,K) corresponds to the x-co-ordinate of the door K of group J of symbol I.

Variable Name	Description
Empty	Array of dimension 4 which indicates the row that is empty. Empty(I) corresponds to the row in which no symbols have been drawn.
Grl	Array of dimension 20 which indicates the group of the symbol that is connected. Grl(I) corresponds to the group of the left symbol in flowline I is connected to the right symbol of that flowline.
L	Array of dimension 20 which indicates the symbol present in the left side of the flowline list. The element L(I) corresponds to the symbol presented in the left side of the Ith flowline.
Ld	Array of dimension 20 which indicates the door of the left symbol of the flowline list from which line should be drawn. The element Ld(I) corresponds to the door of the left symbol in the

Variable Name — Description

Ld — I^{th} flowline list from which line should be drawn.

Mx — Array of dimension 12 which is used to find the row for the symbol. Mx(I) indicates that the row for the symbol I is the same as that for the symbol Mx(I).

NF — Array of dimension 12 which indicates the shape of the symbol. The element NF(I) corresponds to the shape of the symbol I.

OCP — Array of dimension 20 which indicates the flowline is drawn or not.
The element OCP(I) corresponds to the I^{th} flowline is drawn or not.

P — Array of dimension 12x3 which indicates the score of the symbol in the column.
P(I,J) corresponds to the score which I^{th} symbol has in J^{th} column.

Variable Name	Description
Page	A variable used to indicate which page will be used.
R	Array of dimension 20 which indicates the symbol present in the right side of the flowline list. The element R(I) corresponds to the symbol presented in the right side of the I^{th} flowline.
R1	Array of dimension 4 which is used to find the row for the symbol.
Row	Array of dimension 4 which is used to find the row for the symbol.
Rd	Array of dimension 20 which indicates the door of the right symbol of the flowline list to which line should be drawn. The element Rd(I) corresponds to the door of the right symbol in the flowline list to which line should be drawn.

Variable Name	Description
Row	Array of dimension 12 which indicates the row in which the symbol should be drawn. Row(I) corresponds to the row in which symbol I should be drawn.
S	Array of dimension 12 which gives elements that have already been given points for different row and columns.
Sc	Array of dimension 12x12 which indicates how many lines two symbols are connected to each other. Sc(I,J) corresponds to the number of times the symbol I is connected to symbol J.
S1X S1Y	Dimension of the prime document symbol
S1DX S1DY	Displacement of secondary document symbols.
S1A S1B	Dimension of the annotation box for the document symbol.
S1C	Dimension of corner flag of document symbol.

Variable Name	Description
S1TX1	Origin of symbol name for document symbol.
S1TY1	
S1TX2	Origin of copy number for document symbol.
S1TY2	
S1TX3	Origin of annotation box in document symbol.
S1TY3	
SCN	Array of dimension 9 indicating the copy number of the document symbol. The element SCN(I) corresponds to the copy number of I^{th} copy of the document symbol.
SIN	Array of dimension 9 indicating the group number of the copy of the document symbol. The element SIN(I) corresponds to the group number of I^{th} copy of the symbol.
S2X	Half of the width of the temporary file.
S2Y	Half of the height of the temporary file.
S2TX	Origin of the text in temporary file.

Variable Name	Description
S2TY1	Sequence of file ["A","D","N"].
S2TY2	Sequence check on file [" ","X"].
S3X	Half of the width of the permanent file.
S3Y	Half of the height of the permanent file.
S3TX	Origin of text in permanent file.
S3TY1	Sequence of file in case of permanent file ["A","N","D"].
S3TY2	Sequence check on file in case of permanent file [" ","X"].
S4X	Dimension for book symbol.
S4Y	
S4A	Dimension of left margin for book symbol.
S4TX	Origin of the symbol name for book symbol.
S4TY	
S5	Radius of the posting source.
S5T	Origin of the symbol name for posting source.
S7X	Dimension of computer printout.
S7Y	
S7X	Origin of the symbol name for computer printout.
S7Y	

Variable Name	Description
S8I	Length of the process symbol.
V	Array of dimension 12x4 which indicates the score of the symbol in the row. V(I,J) corresponds to the score of I^{th} symbol in J^{th} row.
XS1 YS1	Displacement of the document symbol within the box.
XS2 YS2	Displacement of the temporary file within the box.
XS3 YS3	Displacement of the permanent file within the box.
XS4 YS4	Displacment of the book within the box.
XS5 YS5	Displacement of the posting source within the box.
XS7 YS7	Displacment of the computer printout within the box.
XS8 YS8	Displacement of the process symbol within the box.

The Basic Language used here does not make any distinction between lower case and upper case characters. In the report, references to variables sometimes are inconsistent in the use of upper and lower case letters, but in fact, the variable is uniquely defined by spelling alone.

BIBLIOGRAPHY

[1] Alanadams, J. and Rogers, David F., Mathematical Elements for Computer Graphics, McGraw-Hill Book Company.

[2] Anderson, R.J., The External Audit 1, Pitman Publishing, Toronto, 1977.

[3] Anderson, R.J., The External Audit 2, Pitman Publishing, Toronto, 1977.

[4] HP 2647A Reference Manual, Hewlett-Packard Company, California, 1979.

[5] Terminal BASIC Reference Manual #02647-90005, Hewlett-Packard Company, California, 1979.

[6] Anderson and Skinner, Analytical Auditing, Sir Isaac Pitman (Canada) Ltd., Toronto.

[7] Lee, Norman and Tseng, Joseph, "Flowchart Construction by Computer Graphics", Undergraduate Project Report, Department of Applied Mathematics, McMaster University, 1979.

[8] Taha, H.A., Operations Research, MacMillan Company, New York, 1971.