IMPERIAL COLLEGE OF SCIENCE & TECHNOLOGY
(University of London)

Department of Electrical Engineering

THE USE OF INTERACTIVE GRAPHICS IN PROGRAMS FOR THE COMPUTER AIDED DESIGN OF ELECTRIC NETWORKS

by

Anthony John Drew, B.Sc.

A thesis submitted for the Degree of Master of Philosophy in the Faculty of Engineering, University of London.

September, 1971.

ABSTRACT

The design of an electric circuit is a process usually involving much trial and error. For some time the process has been expedited by means of digital computers but the designer finds that in order to use them he is faced with the preparation of input data in an unfamiliar form, a tedious procedure and very liable to error.

Lately the use of interactive graphics has enabled the designer to provide his data in that most familiar form, the circuit diagram, thus greatly reducing the chances of mistakes.

If the designer is to be effective he must be able to concentrate on the design problem rather than operating the computer and this thesis considers interactive graphical computer aided design with that point in mind.

It starts with an examination of some of the methods used in computer aided design, and discusses techniques for interactive graphical input and output, with particular reference to the use of the light pen. It continues with some aspects of the analysis programs required and goes on to consider the overall system requirements.

A proto-type interactive graphical program was written for a PDP 9 computer and its operation is described. An attempt to evaluate its performance was made, using professional engineers and academic staff.

Three main conclusions were arrived at, the first was that in

programming interactive graphics for creative designs every effort must be made to consider the convenience of the user. The ergonomics of the program must be correct. The second was that all input by humans to computers should be by interactive graphics to reduce the occurrence of mistakes and fatigue, and the last conclusion was that there is a great need for a universal high level language for graphic programming.

TABLE OF CONTENTS

Page:

1. INTRODUCTION

Electric networks occur in all branches of electrical engineering and their design and analysis have always been tedious but very necessary tasks. As electrical engineering has advanced the number and complexity of these networks have grown, while it has become vital to understand their operation in ever increasing detail.

Fortunately the use of the electronic digital computer has enabled the engineer, to a considerable extent, to meet and overcome these problems. Recent work in network theory gives the promise of facilities for the study of ever larger and more elaborate networks. However the existing methods of communicating the problem to the computer are by no means ideal and this is especially true for these larger, more complex networks.

At present the engineer communicates with the computer in a manner akin to the writing and reading of letters, that is information is sent to the machine and some considerable time elapses before a reply is received. To make full use of the machine he should be able to use a method similar to the interaction in a conversation. The use of speech seems unlikely, at least in the near future, but there are two other techniques which are being used. These are conversational typewriter interaction and interactive graphics.

The first, which is already in wide use employs a typewriter for the exchange of instructions and questions between the man and the machine.

The second, which appears more natural, involves the use of pictures in the conversation. These pictures are, for example, circuit diagrams and graphs. Despite the fact that graphical arrangements of this type were demonstrated as long ago as 1963-4 there has been a reluctance to use them which it is felt is not fully explained by their increased cost.

This thesis will examine these interactive graphical methods of communicating with a computer and try to determine some of the conditions necessary for success in their use.

2. TECHNIQUES OF CIRCUIT DESIGN WITH COMPUTERS

2.1. Design Methods

There are two methods used in the design of electric networks, one is synthesis, the other is the iterative or trial and error method. Design by synthesis is usually applied to filters and sometimes equalisers which are required to meet certain figures of voltage ratio over a range of frequency. The first step is to find a mathematical function whose value meets the voltage ratio requirements, and which is of a form suitable for realising a physical network. The second step, which is usually simpler, at least in theory, is to use the function to generate mathematically both the element types and values and their interconnections. However, there are many cases where either one or other or both of these steps cannot be handled mathematically; the second step can seldom be relied on to give networks with elements that can be satisfactorily made.

The iterative, or trial and error, method is used for the majority of network design because no methods are known of synthesising the required networks. The reason for this is simply that the problem, treated generally, is too complex; one often has difficulty in deciding mathematically whether or not a given circuit meets the requirements.

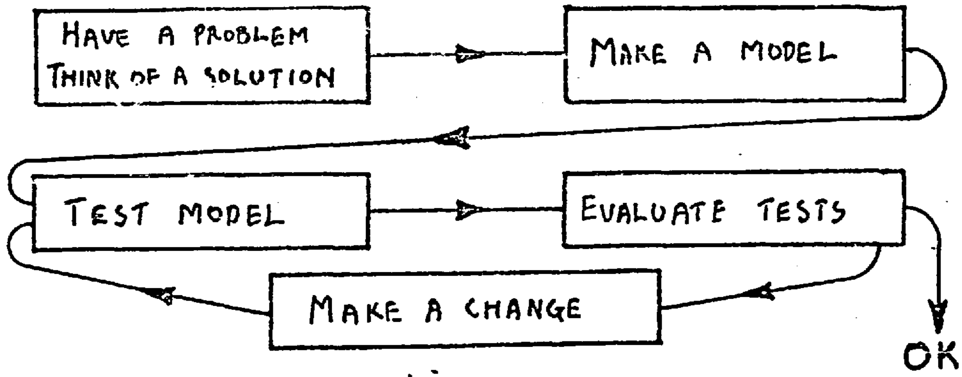
The method involves a loop which the designer enters with a proposed circuit or solution to his problem. He usually expects this first solution to be unsatisfactory, but he doesn't know from which aspect. He determines its performance by calculation or actual measurement, evaluates the performance, and modifies the solution and returns to the beginning of the loop. This loop may be traversed many times before a satisfactory design is achieved and it may have subsidary loops in it. Some representative loops are given in Fig. 2A, these are considerably simplified.

2.2. Application of Computers

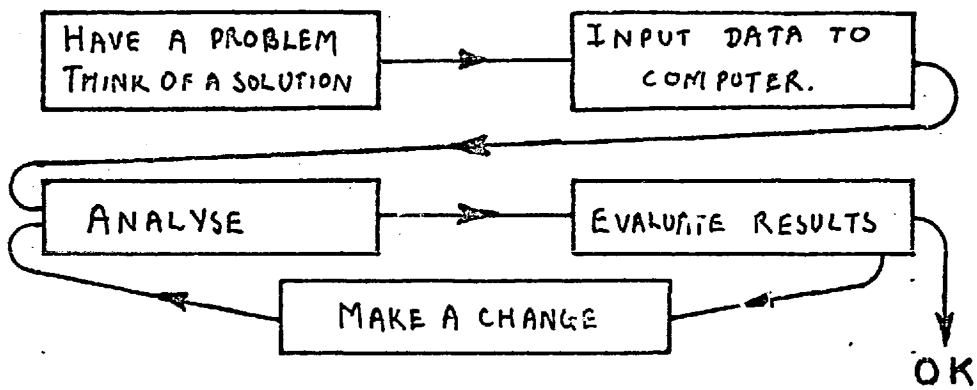
This usually means the use of computers for the analysis of networks, for example to check that the arithmetic is correct, or to study the effect of element changes on the performance. Methods are being developed for using computers in design by synthesis[1] and by the iterative method[2] but so far only limited success has been achieved, even in design by synthesis the experience of the designer counts for a lot.

In order to use a computer, means must be found for the input of data to the machine, and the output of the results. The usual method is to use punched cards or paper tape for the input and to have typewritten output. In addition since the reading and writing from and to these devices is slow magnetic tape is often used as an intermediary to allow fuller utilisation of the expensive machine.

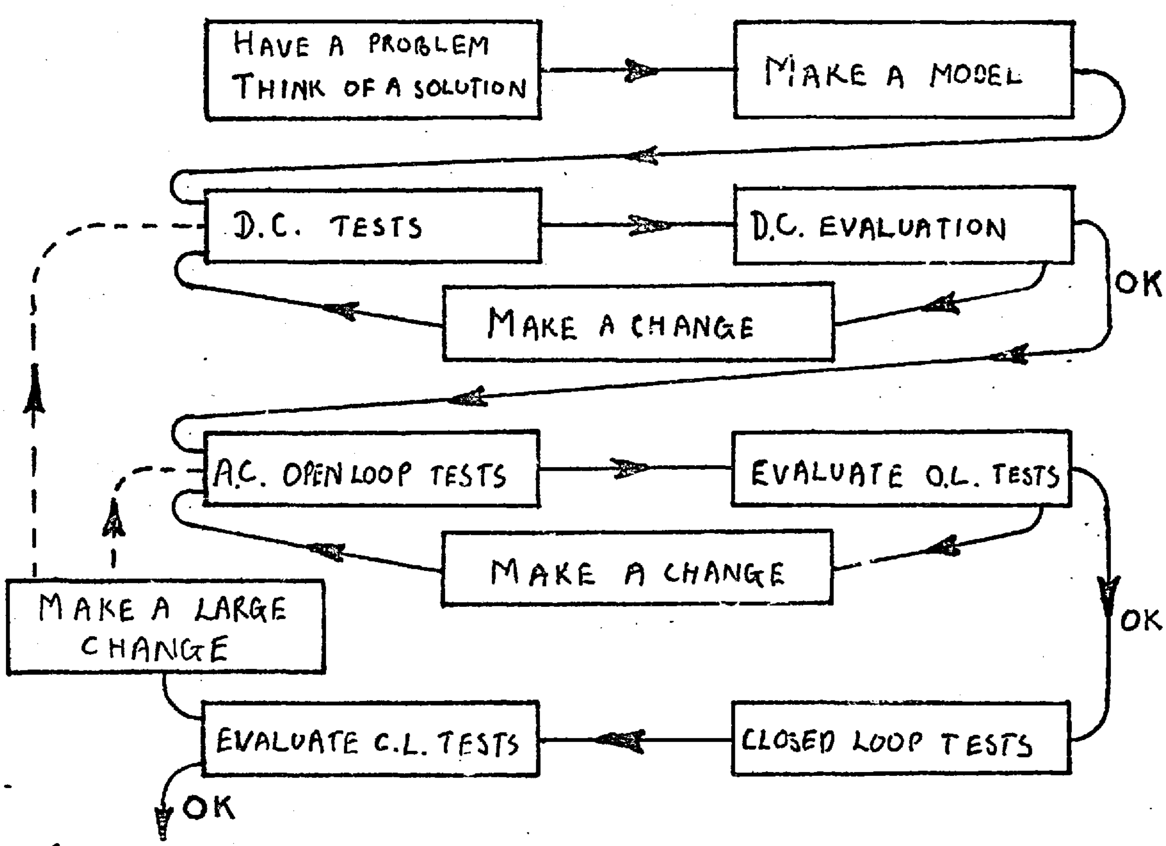
However the result of this is that the engineer loses contact with the machine, he prepares his data and sends it off but waits for from half an hour to several days to receive the results. This is known as "batch mode" and the turn-round time is sufficiently long to allow the engineer to lose his train of thought regarding the problem.



SIMPLE LOOP FOR 'BREADBOARD' DESIGN.



EQUIVALENT LOOP FOR COMPUTER AIDED DESIGN.



SIMPLIFIED LOOPS FOR DESIGN OF FEEDBACK AMPLIFIER

FIGURE 2A SOME SIMPLE DESIGN LOOPS

Another difficulty which compounds the first is to do with the preparation of data. This is almost always in the form of a (long) string of numbers with no indication of what they mean. For example you may be required to start with an integer to indicate the number of resistors, then to give for each one its node numbers followed by its value and then give an integer for the number of inductors and so on. It will be appreciated that there is plenty of room for mistakes which are far from obvious to the casual glance, in addition many computer languages require that numbers be punched in the correct form, (e.g. 1.0E2 not 100) and right justified on the card. Some programs such as ECAP(4) and NET1(5) do try to simplify the situation but they still require some tabulation and have many unnecessary characters. Also the "type, node numbers and value" description of a circuit is itself obscure. When the output of a batch run is returned (half an hour to some days later) it is often found that there are no useful results due to an error in the input data. This is very frustrating to the user who is concerned with his circuit and who regards the preparation of the data and the use of the computer as a necessary (sometimes unnecessary) chore.

In an attempt to overcome these difficulties the idea of on-line interactive use of the computer has been developed(6). Here there are two important improvements, one is that the machine checks the input data, as fast as possible, as it is input. Therefore many trivial errors are corrected within seconds of their occurrence. The other is that the turn-round is so shortened that the engineer gets his results in a time varying from seconds to minutes. Thus he can correct errors while the data input is fresh in his mind and can modify the data in the light of the previous results. Other variants of the idea enable him to alter the computer program on line and also to watch the progress of the computation and modify it as it proceeds.

One possible disadvantage of these interactive techniques is that they may lead to a loss of understanding. It is usually taken for granted that if rapid analyses may be easily performed this will lead to a deeper understanding but it could be nothing more than the "try a .1 across here" method brought up to date. If the engineer is faced with a lot of hard work to derive wanted results he will consider very carefully just what he requires and the best way of finding it out before he starts. This will lead to a better understanding of the problem which may not happen if interactive computer aided design is used too much, or carelessly.

2.3. Graphics

In most engineering the engineer thinks in terms of pictures such as circuit diagrams and he finds these much easier to check and modify than any other description of the problem. This has led to the use of graphics for the input and output of the data, that is the circuit is drawn on some sort of screen and the result appears as a graph on the same or another screen[7]. The principal advantage claimed for these techniques is that they are easy to use due to the engineer's familiarity with circuit diagrams. However it seems that at present considerable difficulty is experienced in getting people to use such schemes when available[8]. This is believed to be due to deficiencies in the graphic programs themselves as well as the natural conservatism of people. A graphical representation of an electric circuit for example is less subject to mistakes than the tabular representation, however the drawing of the circuit for the graphical input can be more difficult than a card input. This applies to the graphical version of ECAP[9] where the diagram is converted to card form which the user is expected to check and remember. The graphical output can be very similar to the graph drawn by the engineer himself except that it is not likely to have the fine graduations of scale seen as graph paper and it is not possible (easily) to keep a copy.

2.4. Cost

There seems little doubt that the use of batch programs for aiding circuit design is well worthwhile even when the circuits considered are conventional discrete component ones. In the case of integrated circuits the saving is even greater.

However, the economics of graphics are much more debatable, largely due to the "inefficient" use of the computing facilities. It is at the moment necessary to involve the computer even before the use has put pen to screen and while he is thinking. Further the time scales of the man and machine are so different that even when the man is experienced the machine does nothing for most of the time and it is fully used for only a minute proportion of the time. These difficulties can be got over to some extent by using a smaller, slower and therefore cheaper machine to control the graphics and a large fast expensive machine only when required for analysis. Even so there is no doubt that graphics are a good deal more expensive to use than normal batch programs with card input and typed output.

However, the price of hardware is falling all the time, and the cost of mistakes, especially in the case of integrated circuits, is rising. Graphics are likely, eventually, to become economic and indeed essential.

3. DESCRIPTION OF AN INTERACTIVE SYSTEM

During the development of the graphical interactive system which will be described later, a simpler system using typewriter input and graphical output was developed. This will be briefly described in order that the reader may have a better idea of the type of computer facilities being discussed.

The machine used was a small PDP 9 made by the Digital Equipment Corporation and intended to support the 340 graphical display supplied with it. The machine has 16K of 18 bit core store and two small magnetic tape units. The makers software included a method of loading the programs from a previously built file on magnetic tape, one section of the file, in core, calling another which overlaid it.

The program actually comprises three sections, changes from one to another being made as infrequently as possible since a delay of about 30 seconds occurs while the new section is read in from the tape. The first section sets the machine up for display purposes and gives the user the chance of getting typewritten instructions if he requires them. After this section, which is never used again, the second is entered and this contains the input, the analysis and the display. Thus the design loop is all within one section for maximum efficiency. The third section contains error diagnostics and facilities for printing out a description of the network and its performance.

The input is by means of the typewriter and is in the form of blocks of data, each block being started by a code letter. The blocks are self contained, need follow no order and are quite short, up to five items. The circuit first has its nodes numbered, '1' being applied to the input node and '2' to the output node. The rest are numbered consecutively up to the highest which is applied to the node common to the input and output. To describe the circuit to the machine each element is considered in turn, the code letter of the block being R, L or C for resistors, inductors or capacitors respectively. Then come the two nodes between which it is connected and lastly its value. Each item, code, node number and value must be delimited by a space or return. By using spaces within the block and a return at the end the circuit of Figure 3A gives the layout of Figure 3B. Note that no unnecessary characters are required, neither is the position on the line important, only the order in the block is important. Node numbers use digits only but values may be in integer form, use a sign or a decimal point or E format or be a combination. The machine continually checks the input and if any nonsense values are given it complains immediately, thus the user can correct the mistake easily and with the minimum loss of time.

The order of the elements is unimportant, if an element already input is repeated the new value overwrites the old, except that if the value is zero the element is deleted. This provides a simple method of correcting and altering the data. The single item F is used to end the circuit data and then two frequencies are entered. These are the lower and upper values of the frequency range over which the circuit is to be analysed.

The program units are basic, ohms, henries, farads, mhos and hertz. When all the data has been input the machine leaves the input routines and proceeds with the analysis. The circuit is first checked for errors and

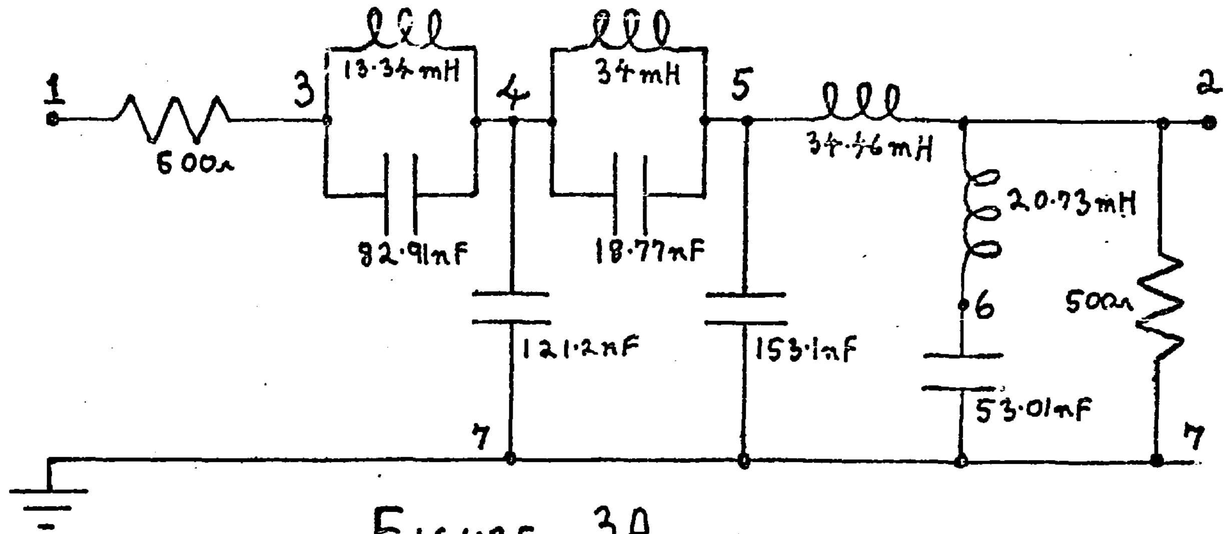


FIGURE 3A

```
$E SLAP

     FREQUENCY ANALYSIS OF LINEAR RLC NETWORKS WITH UP TO TEN NODES
     *********  TWO-PORT VOLTAGE LOSS IN DB AND DEGREES  *********

                    DO YOU WANT INSTRUCTION?  TYPE Y OR N.


N

    TITLE

FILTER

  ELEMENTS. DESCRIPTOR, NODES AND VALUE

R 1 3 500
C 3 4 82.91E-9
R 2 7 500
L 3 4 13.24E-3
C 4 7 121.2E-9
L 4 5 34E-3
C 4 5 18.77E-9
C 5 7 153.1E-9
L 5 2 34.46E-3
L 2 6 20.73E-3
C 7 6 53.01E-9
F

  FREQUENCIES(HERTZ).

1000
3E4
```

FIGURE 3B

if any are found a diagnostic is printed and a return made to the input for corrections. However, if the circuit is satisfactory the analysis is started. Since this may take from 10 seconds to several minutes to complete a progress marker is displayed on the screen and is moved about to show the progress of the calculations.

When the analysis is complete the marker is replaced by a graph of the circuit loss in decibels against a logarithmic scale of frequency as shown in Figure 3C. The corresponding phase graph seen in Figure 3D is obtained by a typewriter command. There are other typewriter commands which enable the user to obtain a record of the circuit and its performance or to return to the input for circuit changes. The input data may be modified in the same manner as when it was first typed and a new analysis performed.

Thus the design loop, which in batch mode may take from half an hour to several days, may be completed in from 30 seconds to a few minutes at most, any mistakes being indicated in from a $\frac{1}{4}$ second to a minute or so after their making.

The input of commands (or codes) and numbers is done in such a way as to minimise difficulties with errors. At any given time there are only certain codes (commands) permissable and if some other character is typed the machine will respond with a question mark. The user must acknowledge that he has seen the error by typing RUBOUT and any other input will be ignored until this has been done. When an acceptable character has been typed the user must "close" it with a space or RETURN. This serves two purposes, it gives him a chance to change his mind or to correct the

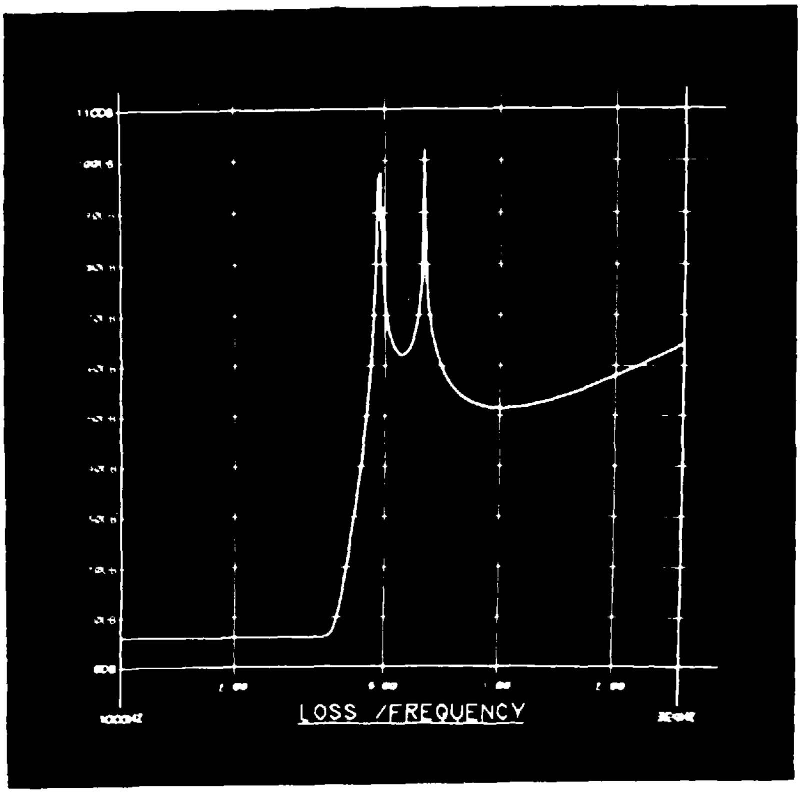


Figure 3C

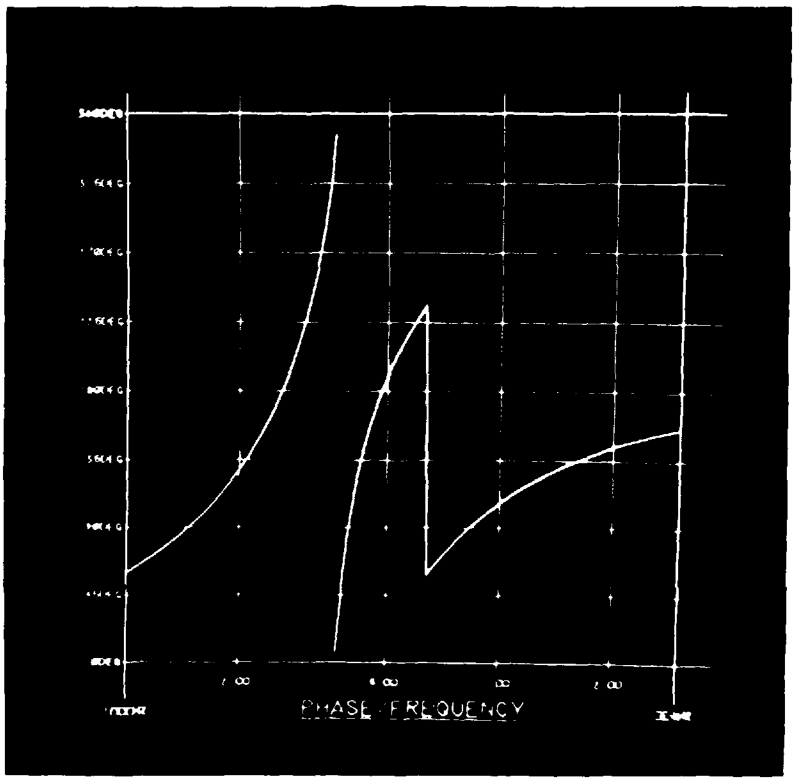


Figure 3D

typing of an acceptable but unintended character, and it serves to separate the input items on the print-out. A similar scheme is used for numerical input of node numbers, the acceptable characters being digits. A check is kept on the size of the number to ensure that it can be stored in the machine. The check on the acceptability of the number as a node is made immediately before the analysis, even so the user will be informed of any mistake within minutes of making it rather than hours or days. The input of numerical values is similar except that more characters are acceptable and they vary as progress is made through the number. A sign is only acceptable as the first character (or the first after an E for E format) and a decimal point is acceptable until one is input and then becomes unacceptable, while a digit is always acceptable. The E is not acceptable until at least one digit has been input and once used may not be used again. The input cannot be closed until a digit has been input and, after an E, is again illegal until another digit has been input. There is no limit (except the size of the paper) to the number of digits, after the first six or seven their value ceases to have meaning but each one causes the number to be increased ten times. If it is certain that a number larger than that which the machine can hold has been input a suitable message is given and the number is restarted.

4. INPUT GRAPHICS

4.1. Requirements

The more important of these may be listed as

a. Easy to use.

b. Comprehensive.

c. Cheap to use.

It is clear that these are to a large extent mutually exclusive, however the problem is to find a compromise, if one exists, that is sufficiently easily to use, sufficiently comprehensive and not uneconomic. As time goes by this is likely to become easier, as the problems both in software and hardware are understood and solved.

Above all, for a system that is to be used in commerce, the balance between the three items must be correct, in a programming research department cheapness is not so important since it is the purpose of such a department to investigate problems without much interest in the cost, also by the time the cost becomes important it is likely to be much less than when the work was commenced.

For the graphics to be easy to use it is necessary that each action should be simple and obvious and easily reversible, thus removing the user's fear of making mistakes. The screen should give clues to any action that the user may wish to perform so that he can find his way to his goal from the indications along the way. Just as one finds one's way across a town without being able to list in order all the places one passes through.

The graphics should be comprehensive so that the labour of describing a circuit does not have to be repeated for various operations such as DC, AC and transient analysis, component layout, heat analysis, stock list and costing etc. This reduces the chance of mistakes creeping in.

4.2. Input Device.

By this, is meant the device which is used to point and/or draw on the screen. There are several such devices which have been used, the more important are listed here

a. Light pen.

b. Tracking ball.

c. Joystick.

d. Mouse.

e. Rand tablet.[10] The Sylvania tablet is similar[11].

The light pen is, as its name implies, similar to a pen in use. It has at its end a light shutter, worked by a finger of the hand holding it and there is a light guide to a photo sensitive device. If, when the pen is pointed at the screen with its shutter open, light from the display enters it, the computer is interrupted. The program may then determine what was being displayed at the moment of interruption. Using this simple action the machine may be programmed so that the pen may be used to point at objects, to operate light buttons * or to draw free or constrained lines.

The tracking ball is a ball about 2" in diameter, seated in a cup with its top protruding. By rolling the ball in the cup a marker may be moved

* A symbol displayed on the screen which causes some action to occur when pointed at by the pen.

about the screen.

The joystick works a marker on the screen in a similar manner, movement of the stick moving the marker up and down and left and right.

The mouse is a small carriage moved by hand over an area corresponding to the screen, a marker copies its movement onto the screen.

The Rand tablet has an area corresponding to the screen over which a stylus is moved, the point on the screen equivalent to the stylus tip is indicated by a marker.

All these devices require two actions in their use, first the alignment of the pointer (end of pen, marker) with the required position and then the indication that it has been aligned (open a shutter, press button, press on surface). The light pen I feel is far the best for the first action while the Rand tablet is best for the second.

I find the discussion of the relative merits of these and similar devices difficult since all my experience, except for half an hour using a tracking ball, is confined to the use of the light pen. I find it difficult to believe that any of these devices can be better than the light pen, but I have met people who have both pen and ball available who think that the latter is better, also there are those who prefer the Rand tablet to the pen. However, we only have a light pen available and this device will be assumed from now on. Some input devices are compared in Ref. No. 12.

4.3. Equipment Layout

The most important item is the position of the screen and its angle. All the screens I have seen have been either vertical or very nearly so, and this of course is excellent for good visibility; however it is not so good as a drawing surface, leading to arm strain. It would seem that a screen, horizontal or nearly so, might be better, however a keyboard or button box will almost certainly be required and this must be near. These and other features are discussed by Michael and Cralle(13). Also it is essential to provide a flat writing surface nearby for the user to put papers on. A good layout might be as shown in Fig. 4A where the screen is at an angle of about 30° to the horizontal and is at the back, the keyboard is at the right and is horizontal and flush and level with the writing and paper surface which is on the left. The typewriter part is separate from the keyboard and is behind and above the screen. Any function buttons are with the screen. Units for paper and magnetic tape are mounted on either side of the screen and there are drawers under the writing surface for storing the tapes.

4.4. Drawing Method

There are two aspects here, the placing of elements on the screen, and the interconnecting of them with wires. When placing an element it is necessary to indicate where it is to go, what type of element it is to be and its orientation. Its position may be indicated in a number of ways; among these is the provision of an array of grid points between which elements may be placed, the desired pair being chosen by a simple pointing action. Alternatively gridpoints may be positioned anywhere by pointing at

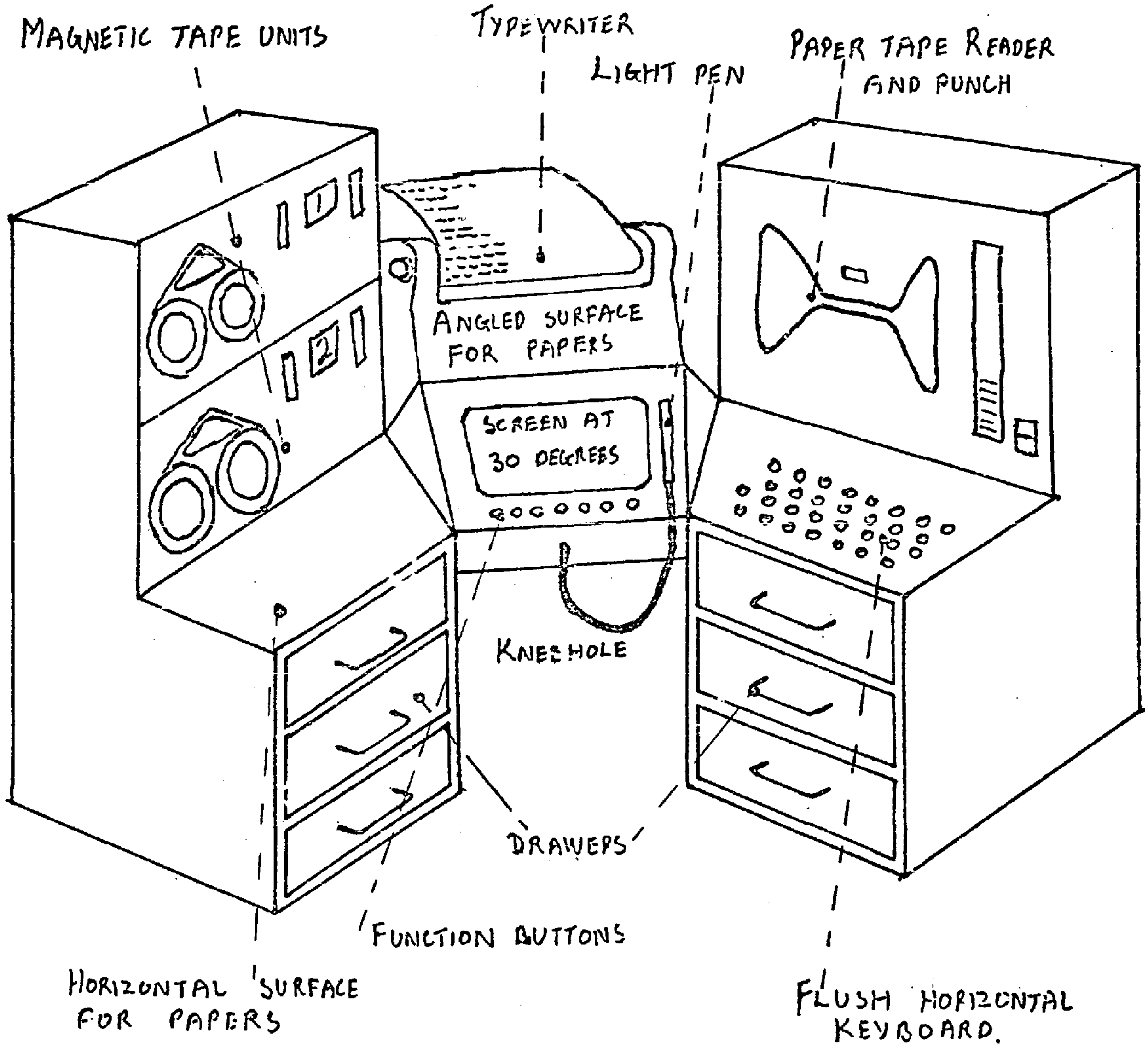


FIGURE 4A CONSOLE LAYOUT

the screen and then placing the elements between them. This requires some sort of searching action on the part of the machine since generally there will be nothing at the required points for the pen to see. If this search action has to be initiated by the user, as on Graphical ECAP[9], it is not very satisfactory except possibly if it is done by using the movement of the shutter to do the initiation. It is of course not necessary to define two points for the element, one will do provided there is some simple way of controlling the orientation. Also the point (or points) need not be identified always in the same manner, for example a pointing and searching action might be one way, another might be to use the end of a line as it is finished, or the end of another element. A method which has been used is to "drag" the element symbol from a menu of symbols with the pen.

There are two methods of indicating the element type, the usual way is to have a menu of buttons, one for each possibility, from which the user must choose. The other method is to program the machine to recognise the element symbol from the pen movements when the user sketches it on the screen.

The buttons can be either normal buttons arranged beside the screen* or can be illuminated symbols displayed on the screen and known as light buttons. The user indicates his choice by pressing the appropriate button or pointing the light pen at the light button. A comparison of the two types of button shows that while the function buttons are easier to program the light buttons enable the user to confine his attention to the screen. The light buttons are usually displayed in fixed positions, say at the bottom of the screen, and there have been attempts to reduce the

* Known as function buttons.

resulting arm movements. Wiseman[14] for example, uses single letters for certain light buttons and displays them around the last position of the pen. Unfortunately he has to have two alternative sets of buttons which reducesthe effectiveness of the idea.

Character recognition has been tried[13] but so far the advantages have not proved worth the penalties. The recognition problem is not so difficult as those usually met with in character recognition because the number of elements to be recognised is fairly small, say 10 to 15 at the most. Also, since on recognition a dictionary symbol will be displayed, incorrect identification will be seen at once and it will be corrected before causing any errors; the immediate feedback is likely to influence the user towards drawing more acceptable symbols. Tests have shown* that the time taken to draw a short wire and insert an element is likely to be about two seconds regardless of whether light buttons or character recognition is used.

Most electric circuits use some types of elements more than others. For example filters and equalisers and amplifiers use mainly resistors, inductors and capacitors while silicon integrated circuits use mainly resistors, diodes and transistors. The drawing process might be speeded up by taking advantage of this. For example the display would have a full complement of light buttons in the usual menu but in addition three or four buttons would be duplicated around the last pen position. Single letter mnemonics would be necessary as used by Wiseman. The actual buttons duplicated should be easily changeable according to requirements.

* See chapter 10.

If the position of the element is specified by means of two points, one at each end, then the orientation is also specified. If the element position is at the end of a line the element may take the same orientation as the line, if no line has been drawn then a fixed choice say horizontal may be made. This could be altered by "flicking" the symbol with the pen in the required direction or by requesting it a second time. This last method can only give rotation in one direction.

There are also various possibilities for the drawing of the wires linking the elements. One approach is to treat the wires as elements and fit them between previously chosen grid points which must coincide with those for elements. This method will not be successful unless provision is made for long wires to be inserted in one operation. Another method is to start a wire either at a newly chosen point or on the end of another wire or element and then extend it with the pen, to some desired spot and finish it there. The wire drawn may either be a "rubber band" line, that is appearing like a piece of rubber between the start and the pen, or it may be constrained to be either horizontal or vertical. The combination or these ways of inputting elements and wires gives a great variety of possible drawing modes.

It is necessary to consider the basic approach to the problem of drawing on the screen. Should the drawing action be as close as possible to the normal action of drawing on paper or should some quite different technique be used? e.g. putting elements from a menu into locations in a fixed grid, and then joining them up with direct point to point wires. I believe that the technique should be close to that with which we are familiar in normal use of pencil and paper, but there are almost certain to be some differences. Even if it were possible to imitate exactly would it be wise? After all the pencil and paper approach has been used presumably because it is cheap and practical and within the limits of available technology, not because it is the best possible. One disadvantage of diagram sketching on paper that springs to mind is the difficulty of deletion, so severe that, for example, in the extreme case of getting the diagram so far to one side that there is not room for it all, it is better to start again with a clean sheet rather than rub-out. The same problem will occur with the computer screen but surely we can find a better solution than rubbing it all out and starting again? Perhaps we could arrange to move items about on the screen while still retaining all their connections, we should have to abandon any ideas of forcing wires to be either horizontal or vertical, but if we included wire junctions among the moveable items the user could adjust his diagram to look nice if he wished.

The rotation of elements or their deletion by appropriate movements of the light pen has been implemented(15) but requires a rapidly following tracking cross and this is also desireable for the drawing of wires. It was previously explained that the light pen interrupts the computer when light from the display falls on it. The light from the display is of two types, an initial blue flash lasting less than 100µS and a long afterglow, it is only the blue flash which operates the pen. Now the pen has a field of view which is quite large, seldom less than about $\frac{1}{4}$" which is much too large an area of uncertainty. To get over the problem and that of pointing to areas where there is no item to cause operation of the pen, a tracking cross is displayed. This is often actually a cross, sometimes other patterns are used but the purpose is to determine the centre of the field of view of the pen. When a cross used it consists of four arms arranged horizontally and vertically and drawn from their outer ends towards the centre. As each one enters the field of view an interrupt occurs and the beam co-ordinates may then be determined. From the four pairs of co-ordinates the centre of the field of view is determined and this is taken as the pen position. The cross is displayed at short intervals of time and so the motion of the pen may be followed, provided that it does not move too much between one display of the cross and the next. To improve the following action the cross should be repeated at fixed short intervals and instead of displaying it at the previous position of the pen, which is often done, it may be displayed at the predicted position of the pen, calculated on the assumption that the pen has constant velocity*. This could be extended to assuming constant accelerations but the increased calculation time may cause difficulty.

The inability of the light pen to "see" a point on a blank screen causes complications when it is desired to draw lines or wires without references to other items. It is necessary to move the cross to the proposed starting point of a line, without drawing a line and then to move the cross to draw the line. Thus two modes are necessary, one for just moving the cross, and another for moving the cross and drawing a line. Another method is to arrange a search pattern over the whole screen to find the pen and, when found, to position a cross opposite the pen and remove the search pattern. When the cross moves a line is drawn.

Fortunately, while the cross may not be sufficiently fast to enable the user to "scribble", he goes a good deal slower when he is drawing a useful diagram.

* Known as a predictive cross.

4.5. Problems Of Small Screen

Many graphic displays have a rather small screen size and this may cause difficulties when large circuits are to be drawn. (On a screen $9\frac{1}{2}$" x 8" a circuit of 16 modes and 28 elements may easily be drawn). It is not expected that this will be a very serious problem since large circuits are nearly always broken down into smaller portions for analysis. However, it will probably be necessary for small screen displays to increase the effective screen size and this may be done by the "windowing" technique. By this technique the screen is regarded as a movable window looking onto a much larger surface containing the diagram. The surface may be from two or three times the size of the screen upwards.

4.6. Device Library

One way in which the drawing of circuits may be considerably speeded and the chance of mistakes reduced is by having library facilities available. There should be two libraries available, a permanent one and a short term one. The short term library would hold portions of circuit that the user has defined for a particular job; this might, for example, be a section of a long ladder that he wished to repeat in his circuit without the labour of drawing it many times. Each occurrence of the library item would be represented by a rectangle with the appropriate terminals and labels. The permanent library would have devices, probably semiconductors, stored under their type names together with their equivalent circuits and possibly draughting and ordering information. When called, the diagram would show a suitably descriptive symbol but the circuit as analysed would include an equivalent circuit or other representation of the particular device. It is probable that this representation would vary according to the type of analysis to be carried out.

4.7. Numerical Input

There is bound to be a considerable amount of numerical input data required, for example the element values, and there are various ways in which this can be inserted. Some of these are -

Typewriter

Button Box

Light pen with light buttons or light handle.

The usual procedure is to use the typewriter and this is quick and quite convenient, but has the disadvantage of requiring the user to put down the light pen and turn away from the screen, also it is very easy to type nonsense* and this requires preventative programming calling for corrective action by the user. On the other hand the typewriter does give a record of the input which is often useful. The button box is very similar to the typewriter except that the button (key) positions are not fixed or labelled and the echo to the typewriter must be programmed if required. One advantage is that the button box, if suitably designed, may be fitted much closer to the screen than the typewriter, as in Figure 4A.

To use the light pen a series of numerical light buttons may be provided which are pointed at in sequence to give the required number. This technique has the advantage that the user can keep his attention confined to the screen, also he can be prevented from making certain mistakes by only displaying legal buttons, however it is possible that the

* For example, two decimal points or two signs etc.

pen used like this is slower than the typewriter.

For all these devices a free format input should be adopted, the ideal being one in which any input that the user can recognise as a number is accepted by the machine. Even a "close" character at the end of the number is undesirable but is usually very difficult to dispense with.

The light handle(16) is a rectangle on the screen within which the pen is rotated. It is arranged that motion up and down in the rectangle increases or decreases a number and the amount of the change is more on the left than the right. Thus by means of a clockwise rotation the number is increased and the opposite rotation decreases the number. Provision is made for the speed of rotation to give an increased effect and also for a linear region for a final fine adjustment to be made. The method is rather tedious compared with any of the other methods and is quite unsuitable for covering wide floating point ranges such as are used in electric circuits. It probably uses little more screen area than light buttons however, and may be useful for giving a continuous variation of a parameter, when the variation is more important than the exact value.

All this discussion has dealt with the case of elements described by one (or a few) numbers. For the case of non-linear elements, however, a table of values or a graph is needed. It is likely that the item is already in the library, but if not then the input should come from a faster input device than the ones described and the paper tape reader is probably best. This is because the data can be prepared "off line" to save computer time but the format should still be "free" to reduce the chances

chances of mistakes in the tape. Schemes have been used[17] where a sketch of the non-linear function is made on the screen, but I think that the tabular input allied to a good interpolation routine is likely to be more satisfactory. Another possibility is to define the interdependance by means of an algebraic function and in some cases this will be far and away the best. But while the light pen would be used for indicating the element and possibly the variables I cannot imagine any other method of input for the function except the typewriter or paper-tape.

4.8. Input Captions

The circuit diagram will not consist only of wires and elements; at the very least the elements must be labelled with their values. However, it will be necessary in all but the simplest systems to attach names or numbers to the elements and also to the circuit nodes. There is a problem associated with the location of these captions. In a pencil diagram they are not placed in any fixed position relative to the element, as a computer will do, but in a clear space that is sufficiently close. The computer must be responsible for displaying these captions but there will certainly be occasions when they are obscured by other elements or are not very obviously associated with their element or node; this particularly applies to node numbers. The difficulty may be overcome by allowing the user to move the captions, if he wishes, to more suitable positions. While the caption is being moved the associated element or node should be marked in some way, e.g. by making it flash.

It is possible that the screen will appear cluttered if each element has a three character name and an eight character value (-1.2E-12 has eight characters) attached. In this case it may be arranged that the name and/or the value should only be displayed on demand and should substitute for the element itself.

4.9. Screen Organisation

Even a quite small graphical input program is likely to have quite a lot of light buttons, and it is not possible, without wasting space, to display them all at once. It is not desirable either since certain actions are not possible (e.g. analyse before the circuit is complete) or are dangerous (e.g. deletion of the whole circuit). Therefore it must be decided how the sequenceing of the button displays is to be arranged. One general principle, which is probably best, is to offer at any given time all the actions which are legal, i.e. sensible, but it may sometimes be better to offer illegal actions and give an error diagnostic if they are used rather than frustrate the user by refusing to offer them for some reason which he cannot fathom.

It has been suggested[18] that some sort of hierarchical structure be used in which all actions are put into groups and these groups into larger groups and so on through several levels. Thus to get any given option it would be necessary to move up the hierarchy until the path to it was intercepted, and then descend to it. A scheme like this will probably be necessary when very large and comprehensive programs are written but it will suffer from the difficulty that the user and the programmer will often disagree as to what group a given action should belong. The problem might be overcome by arranging, on demand, for the complete structure to be displayed, removing all else from the screen to make room[19].

5. OUTPUT GRAPHICS

This section covers the display of all the possible forms of output, some of which are listed here:-

1. Graph of one variable against another, such as potential or current against time, potential against current, potential or current ratios against frequency.
2. Numbers attached to points on the input diagram, for example D.C. conditions in a circuit, D.C. sensitivity.
3. Graphs attached to points on the input diagram.
4. Layout of circuit components.
5. Display of text such as stock lists or function dependencies for non-linear elements.

The main purpose of the graphical output is to allow a rapid appreciation of the network performance. Any detailed evaluation is best done using a more permanent output.

5.1. Single Graph

When the output is a single graph (or a few graphs) which shows a response of major importance then it must be drawn sufficiently large and in sufficient detail to allow the user to determine fairly quickly, if necessary, if the response is satisfactory. The graph should be titled and the axes should be labelled with their units. There should be a sufficiency of grid lines which should cross the screen and which should be labelled with their values. These values should be in rounded form, 2 or 2.0

not 1.999, and absolute like 3.0E7, not 3.0 on the line and x 10^7 along the axis. The mantissa should be normalised into the range 1 to 10 since this avoids one and sometimes two unnecessary characters and it is almost always easier to comprehend, compare 3E2 with 0.3E3. The E notation for the power of ten is easier to write and read than the older form*. The number of grid lines should be as large as possible, but it is likely to be limited by the display characteristics, either the flicker rate or the quantity of store becoming unacceptable. It is not likely that the number of lines will approach that of a sheet of normal graph paper. The grid lines are usually spaced evenly, or nearly so, across the screen. This, in fact, is easiest to program and assists the quick understanding of the graph. It is possible that better results might be obtained if there were more lines at points where the graph had greatest slope or curvature but this depends very much on the type of network; in any case large areas without grid lines are undesirable.

The fact that the data to be displayed must be scaled to fit the screen raises a number of points. If the magnitude range of the data is very large then a logarithmic scaling may be required. If the data also includes negative or zero values which are of importance, then some special scale, part linear, part logarithmic will be required. For smaller ranges, such as for degrees or decibels, a linear scale is more suitable. In the case of the logarithmic scale the actual values for the intermediate grid lines can present problems. The lines should be spaced across the screen according to some devised scheme, such as evenly or concentrated at points of high slope, yet the values should be rounded. It is possible to store sets of

* Compare 1E-6 with 1 x 10^{-6}

"preferred" values for the grid lines from which suitable values, or multiples are taken.

If the graph is scaled to fit the screen exactly, then a curve showing a large change can only be distinguished from one showing a small change by examining the captions on the axes. It may in some cases be an advantage to have the axis scales fixed even if this results in graphs that go off the screen or appear quite flat.

5.2. Numbers

There will be occasions when the output is in the form of numbers, such as voltages or currents associated with nodes or elements of the circuit. These numbers could be displayed in the form of tables, assuming that both elements and nodes have reference names or numbers. It would be much better if they could be added to the input diagram, but this presents considerable problems as discussed in the previous sections. Much the same solutions are applicable; that is, either movable labels or labels substituting for something else already displayed. Node potentials could substitute for node numbers, but difficulties will occur since the potential will usually take at least three characters to specify and the node numbers seldom more than two. In the case of elements, the numbers could substitute for the label, which would be rather easier than for the nodes, or for the symbol which should be easy, since the symbol can be made sufficiently large. Problems may occur with vertical symbols. It has been suggested that numbers might be shown in analogue form(17) by the length of a line or area of a rectangle or circle but this suffers from the same location problem and is probably not suitable for cases where there is a large range in values.

5.3. Multiple Graphs

There are occasions when the output is in the form of a number of graphs. For example, the display of the sensitivity to element changes of a network response over a frequency band requires one graph for each element. In this example it would not be useful to provide a full screen display for each graph since detail is seldom required in such a case. A rapid comparison element against element, and at frequency against frequency is required. Thus all the graphs should be displayed simultaneously and this could be done by dividing the screen up into enough suitable rectangles, one for each graph. There would be quite complicated scaling, positioning and captioning problems for circuits involving from only two or three elements up to fifty or a hundred or more. It might be possible to use the substitution technique again, drawing very small graphs in place of the elements. These would be scaled with say 1-2mM per decade and all to the same scale. If the output required a graph for each node, then the display of output on the input diagram would present serious difficulties in positioning.

5.4. Layout

If the output were the layout of the components for an electrical circuit, it is unlikely that position difficulties would arise, but it might easily be found that the number of elements to be displayed would make the use of windowing necessary. Very complicated line drawings, with captions especially can also lead to bad flicker due to the unavoidably slow display repetition.

5.5. Textual Output

The display of text is likely to be quite satisfactory, provided that not too much is displayed at a time. This requires windowing but the text must be in "scroll" form, of a width equal to that of the screen. Any need for moving the window sideways across text is likely to make the output unusable.

Sometimes it may be required to display tabular data that was previously input, say from a paper tape. This might, for example, be the volt/amp curve for a diode or non-linear resistor which is being used in the circuit and which it is required to check. If a mistake is found, or if it is wished to change the data it would be very convenient to be able to edit the output. There is little doubt that a more comprehensive program will have to provide this facility since the output of text is so much quicker on the screen than by means of the typewriter. The use of the light pen for pointing is so much quicker than the usual typewriter controlled searches for identifying a point of correction that this feature will be especially desirable.

A number of considerations are associated with the display of output. When an analysis is requested there will be a delay while the work is done and this can vary from a few seconds to several minutes. It is desirable that the user is not left with a blank screen and no idea of how his calculation is proceeding[20]. Probably the best answer is to display the results piecemeal as they are generated. However this may be impossible due to the difficulty of accommodating both display and display generating program, as well as the analysis program in memory at the same time; there may not be room, or this might be too expensive. The user should not be left without any idea as to the progress of his job as he will soon become frustrated and believe that either it is not being done or that something has gone wrong. If he is given some indication of progress he will be able to organise his wait much better.

When an analysis is being done it usually requires very little extra programming to obtain quite extensive information from it, for example, the voltages at all the nodes not just at one. The user should be given the option of requesting output other than that initially requested. It is clear that the benefit from this output may be quite large while the increase in computer and waiting time is negligible. The method of selection may be either light buttons or typewriter commands, and here the typewriter has the advantage of saving screen area at a time when fine detail may be needed on the output graphs.

6. HARD COPY

The (potential) permanent output from the machine is of two types: that which the user knows he will want when it is available, and that which at some later date he wishes that he had kept.

In the first case he may want the permanent copy for several reasons. He may, for example, wish to store data so that he can restart, at a later date, at the point where he left off, he may require a reoord of some successfully concluded piece of work, or he may want a record that he can take away and think about.

Data to be used for restarting the program need not be in a form intelligible to the user, but must be capable of being rapidly read by the machine; paper tape or magnetic tapes are indicated. In the case of the magnetic tape it is likely that one tape will contain several sets of data, so to allow their identification the user must be prepared to give a file name.

Data intelligible to the user must be in printed or drawn form on paper and is likely to take a long time to produce, wasting both the user's and the machine's time. Some form of off-line facilities are required, such as a line printer and Calcomp plotter. It might be reasonable to drive a line printer directly from the machine, especially as the user could then see his permanent record before he destroyed the data producing it.

A serious disadvantage of an all-graphical input and output facility is that after using it for some hours there will be absolutely no output to examine unless the user has had the foresight and patience to obtain some during the course of the work. A great advantage of the on-line interactive programs using the typewriter for input and output is the complete record that they give of what was attempted, the results, and even the mistakes.

Perhaps a nearly ideal solution would be for a line printer to record each circuit analysed and the results as they were generated. The user should be unconscious of it.

Graphical output may be preserved by photography but this usually means that the user must leave the consol to get the camera, put out the room lights etc. This may or may not be acceptable to the user, in any case if worthwhile results are to be obtained the photography must be carefully done.

7. GENERAL ANALYSIS

7.1. Requirements

The most important requirements for the analysis portion of the system are reliability, accuracy and short execution time. Under the requirement of reliability comes all those features which ensure that no matter what the input data the analysis program will either give an accurate answer or will give a straightforward error diagnostic. It is often found when using existing large analysis programs that several attempts are necessary before a satisfactory set of results is obtained, due not to a mistake in the input data, but due to the use of a "queer" circuit[4, 5, 21]. For example if a capacitor has a value say a million times larger than intended, the circuit incorporating it will still have a quite definite response to a given excitation but the program may either break down or go into an almost unending series of calculations. In such and similar cases the circuit must be checked initially to ensure that a reasonable answer will be generated in a reasonable time or, if that is not possible, then a diagnostic saying why not must be given. If a design is being carried out interactively it is very difficult to avoid the kind of circuit that can cause these difficulties, and it cannot be too strongly stressed that the program must give a sensible answer. Otherwise the user will waste valuable time suspecting the machine or program faults, or looking for his own mistake and will lose confidence in the program (and the usefulness of computers). This is probably among the most important requirements, not just for the circuit analysis but for the program as a whole.

The requirement for accuracy is obvious, but it is not so easy to say what degree of accuracy is necessary. Most display screens have a resolution of from 0.1% to 0.01% but this accuracy will not always be sufficient, especially where changes in response are of interest or where the output can be on the typewriter. On the other hand there are many cases when limitations in the input data such as manufacturing spreads on element values or, more important, deficiencies in device models make the achievement of 1% or even 5% accuracy difficult. In linear AC analysis the problem has been artificially simplified and therefore we may expect the closest agreement between theory and practice. However for transient analysis since few concessions to ease the problem are made the agreement is likely to be worst.

Rapidity of response is clearly highly desirable in an on-line interactive system; however it is probably less important than reliability and sufficient accuracy. There have been several estimates and determinations of the acceptable delays in response and these have ranged from several minutes downwards. I believe that the acceptable response time depends on the state of mind of the user; if he is awaiting the proof of some clever modification that he has just thought up any delay is too much. On the other hand, if he has an unsatisfactory circuit and has run out of ideas for improving it, then the longer the wait the better[20].

7.2. Analysis Types

There are three main types of analysis, those for DC circuits, linear AC circuits, and circuits for which the transient or time response is required. In most cases the output is an array of node voltages or branch currents or their ratios as a function of frequency or time. The excitation is one or more sources of either direct current, sinusoidal oscillating current or current with specified wave shapes.

D.C. analysis is used for a variety of purposes, among which are the determinations of the bias conditions in an amplifier, the initial conditions in a circuit for which a transient analysis is to be done, and quite often the solution of equivalent circuits derived from problems quite unrelated to electric networks (e.g. flow of blood through the lungs, heat flow). There will be a need for the use of non-linear elements such as resistors (diodes) and controlled sources (transistors) and these will be obtained either from a library of devices, called for, possibly, by their manufacturer's code, or will be defined by the user at the time of use. In the case of the semi-conductor devices their non-linearities are a function of their temperature which can sometimes be specified. However, very seldom is a change made in this temperature when the device is found to be dissipating appreciable power, it is up to the user to watch for points like this.

A.C. analysis is used mainly for the study of linear amplifiers, equalizers and filters. The analysis method is usually to first generate the indefinite admittance matrix, and then to invert the reduced matrix to find all the node voltages. This method has the advantage that it can be applied to any network regardless of configuration, but it can be a lengthy process if a number of results at different frequencies are required. It might be worth incorporating special routines for dealing with special types of circuit, for example ladder circuits are often used in filter networks and their analysis is much easier and quicker if advantage is taken of the ladder structure.

There are other methods of analysis which make a symbolic reduction from the network to give the network function from which the response is evaluated. An example is NASAP[22], which also uses the function to determine the network sensitivity. Other techniques exist for the determination of sensitivity[23].

Transient analysis, using numerical integration is a much more general analysis method and could, in principle, replace both D.C. and A.C. analyses. It is the more difficult type to carry out satisfactorily and it used for non-linear circuits such as oscillators and semi-conductor switches. It is apt to be time consuming, and for this type of analysis the user should not just be told how long it will take but should be able to terminate it when necessary. The semiconductors models required for this type of analysis are the most demanding since all the electrical parameters of the device must be modelled and may be used at frequencies ranging from zero (D.C.) up to many times the device cut-off frequency.

Both A.C. and Transient analyses may depend on the results of the D.C. analysis for their initial conditions. The small signal incremental parameters used by the A.C. analysis are a function of the large signal or bias conditions while the steady state of the charge on the circuit capacitors or the current through the inductors give the initial conditions for the transient analysis. There should be interconnections between the various available analysis routines so that conditions obtained from one analysis may be applied to a different one without requiring the user to carry any data over. This will save a great deal of time on redrawing a circuit several times for different analyses.

7.3. Models

For each device there will need to be a range of semiconductor models in the device library. For example, for a transistor it might include a simple piecewise linear approximation for D.C. analysis, and, for A.C. analysis, either a 2 port model, such as the hybrid matrix or a physical model, such as the hybrid pi equivalent circuit.

For transient analysis usually the most elaborate model will be required, and this would probably be the Ebers Noll or the charge controlled model. It will be highly desirable for the user to be able to use his own model; he may wish to model a special device, or a standard device in a special way. He may not be able to get sufficient data for the full model of the program library, but may have sufficient for his own particular problem. The user's special model might be in the form of an equivalent circuit using normal components in which case he will have no difficulty, it might be just a set of matrix parameters between which the program would have to interpolate, or it might be a mathematical function connecting the variables,

7.4. User Processing of Data

The output of the analysis will be in a form that the writer of the program thought to be most generally useful, but even if he is right there will be cases where a simple modification of the output will give enormous advantage. For example for A.C. analysis, at the very least a set of two-port matrix parameters will be available, but the program may give as output the voltage ratio between the ports. However the user may prefer the matrix parameters themselves, or he may be interested in something obscure like the return loss of a reactive network where only the angle is significant. The calculation of these special outputs is a trivial problem, but may take an immense time if it must be done by hand. The program should have a facility whereby the user may program, preferably on line, simple modifications to the data display without the need of loading the whole program again.

8. THE OPERATING SYSTEM

The most important requirements for the overall system are that it will cope with all reasonable, and most unreasonable, problems, that it should have a rapid and reliable response to all commands and that it should be sufficiently cheap.

Now in order to solve large network problems a large, fast computer is necessary, and to give a rapid response the user must not be too far away from the machine, which must be waiting ready to execute the appropriate part of the program. The obvious arrangement is to monopolise a large, fast computer with the interactive program, but this is quite uneconomic since such a machine will cost several hundred pounds an hour to operate. There are a number of ways out of this dilemma, but they all result in a worse service to the user.

When a graphical input program is being used a large number of fairly simple commands will be given by the user, perhaps as many as 1000, and occurring as close as 100mS apart. On the other hand each command will perhaps take an average of 500 machine operations to execute and this will take only about 1mS. Thus the machine must respond in less than 100mS to each command but it likely to spend at least 99% of the time waiting for the user. The input program will probably be quite large for example GINA[24] uses 7500 words, SLAP[25] about the same and PIXIE[14] rather less, however a small word size of 16-18 bits will be satisfactory.

The execution time of the analysis program will be many seconds or even minutes and so a rapid response is not so necessary. That is, if it takes a few seconds before the analysis starts, it will not matter. This program will be large, even after segmentation 16K - 32K of memory being required. To cope with the precision real arithmetic needed a large word is desirable, say 30-60 bits.

These features indicate that two computers should be used, one small slow machine to operate the graphics and a larger faster machine for the analysis. This will only result in cheaper operation if someone else can be persuaded to use the large machine when no analysis is being done, that is for most of the time.

Another method is to use a large machine in a time shared mode, where a number of people use it apparently simultaneously. This is done by storing their input data and commands in individual buffers and arranging the machine to attend to each in turn, Such an operating system may give different users different priorities depending on the type of program being operated and its importance. When a user is found to need machine action his program, or part of it, will be taken from a bulk storage device, such as a drum, read into memory (core) and then executed for a time perhaps as short as a few milliseconds. It will then be replaced by the program of another user. These methods of saving core space can result in the machine spending more than half its time swapping programs in and out of core[26]. A lengthy task such as the actual analysis will be done in small sections, the machine being used by other people in between. This method increases the response time which is the penalty for the cheaper operation, but while this may be acceptable for tasks like analysis it can be very disturbing for small tasks repeated many times such as servicing the light pen.

Very often several graphics consoles will be required and these will usually be widely separated. The signalling delays over any transmission links will generally rule out their use when very rapid response is required. These facts provide good reasons for the use of a large machine for the analysis and a small for the graphics. The necessity of getting two computers and a link between to operate simultaneously can cause problems, in fact it has been suggested that it should not be attempted until the system with the machine adjacent has been made to work[37, 38]. The larger machine which will be expensive may be used in either a time-shared mode, or batch mode. The one which is likely to be most reliable should be chosen if a choice is given.

It is very difficult to generalise as to the type of system that should be used as this will depend on what type of equipment is available for purchase or installed, and how much money may be spent. However, it is very likely that a data link os some type will be required and here transmission will be one of the limiting factors. Suppose we have a small computer dedicated to the graphics while the analysis is to be done by a large machine accessed over a link. If the circuit comprises say, up to 100 items each of which takes five words of 18 bits to describe there are 9,000 bits to be sent, but this will probably be increased to 10,000 with descriptions of non-linear items, parity bits etc.. At 4.8K bauds this will take 2 seconds to transmit assuming no difficulties. If we expect to receive back two graphs for a 10" screen we need 200 points on each, or say 15,000 bits allowing for grid lines and captions etc. This will take 3 seconds to transmit. We therefore have a link turn-round time of 5 seconds which is quite satisfactory. The execution time may vary from fractions of a second up to many minutes, depending on the problem and machine, while the queuing time is quite indeterminate, depending on the number of users and their relative priorities. If it were desired to dispense with the small machine and get the larger to service the display, the amount of data to be sent for this would be small, but the allowable turn-round time is about 0.1 seconds and queuing and loading will very likely exceed this.

It is often said that the small machine will not need very much core store and much will depend of how the display is driven, from the core or direct from a bulk storage device like a disc*. However if rapid response to the light pen is to be given, all the data that can be affected must be rapidly accessible and this means either in the local core or on a local disc or drum. In any case a fair portion of core will be necessary to deal with it. Also a large and comprehensive program will need a comparable graphic program to run it and this will need a good deal of storage space. It seems likely that at least 8,000 words and probably more will be needed. It is possible to use overlay techniques to reduce the amount of core required, that is, to read a required piece of program from, say a disc store into core, overlaying an existing part of the program. This needs careful handling if it is not to lead to unacceptable delays.

It is most important that the user is not kept waiting for a response from the machine, and this response can be an error indication on the screen, or the changing of the screen display in recognition of what choice the user has just made. The practice of insisting on a series of inputs (e.g. a character string) being given before the machine takes any action is not suitable for the unskilled operator.

* Compare MACLEAN[29] and PIXIE[14].

DISPLAY	INPUT	OUTPUT	LOCAL MACHINE	REMOTE MACHINE	COMMENTS
Interactive with light pen	Display	Display	Small, short access time	Large, long access time	Probably best
Interactive with light pen	Display	Display	One large machine with Short access time		Probably Uneconomic
Interactive with light pen	Typewriter	Display	One large machine with long access time		Interactive needs short access time
Storage	Typewriter	Display	Small, short access time	Large, long access time	Interactive needs short access time
Storage	Typewriter	Display	None	Large, long access time	Interactive needs short access time

One must remember that when an engineer is using a graphic system to input data such as a circuit he does not feel that he is doing anything very useful, he is acting as a clerk in transcribing data, and so to be satisfactory the graphic input must be very "slick" in operation. It is certainly possible, for example, to input a circuit by means of a storage tube and a joystick, coupled to a remote computer by means of a slow link, but it is doubtful if such, or similar, tedious arrangements will ever be anything more than amusing toys. If such a system is all that can be provided then the idea of graphical input should be abandoned and the typewriter and paper tape reader should be used instead. The display should be used to display the output and to give operating instructions and error messages. An interactive system must be so arranged that the majority of commands are acted on with imperceptible delay. The table gives a tentative listing of several possible systems in their order of merit.

It should be realised that the human operator is very adaptable and will change his mode of operation to get the best out of any given system. This is an advantage to the user when coping with a set of given circumstances but the system designer must beware when he is evaluating a system. He is likely to get a good report on what is really a badly designed arrangement because the user has adapted himself and because he (the user) cannot visualise a better system.

9. DESCRIPTION OF A GRAPHICAL INTERACTIVE FACILITY

A program has been written to test some of the ideas discussed above and to find the reactions of various users. The program is a development of that described in Section 3, it has a graphical input, and uses the same manufacturers' hardware and software, except that a small disc store with an average access time of less than 50 mS became available. The program is segmented as before, but since it is kept on the disc during use there is no longer any need to try and keep the main design loop in one segment. The program has facilities for linear AC analysis of circuits of up to 16 nodes, comprised of resistors, inductors, capacitors and voltage controlled current sources. It includes operating instructions and very full error checking and error diagnostics. There are facilities for storing the input diagram on paper tape, to be read in later, and for giving a typewritten description of the circuit and its performance.

Suppose that it is desired to study the performance of the circuit of Fig. 9A which is a very simple band limited amplifier. For input to the computer it must be drawn as in Fig. 9B and an experienced user would make this change as he drew on the screen.

To run the graphic program it must first be transferred from the magnetic tape where it is kept to the machine's disc store. This is done by means of the maker's system software which also may require loading. It seems inevitable that some such procedure will face the prospective user unless a permanent machine operator is provided, which is unlikely for the small satellite computer envisaged. If the disc storage were large the program could be kept there, but it might still be necessary to load

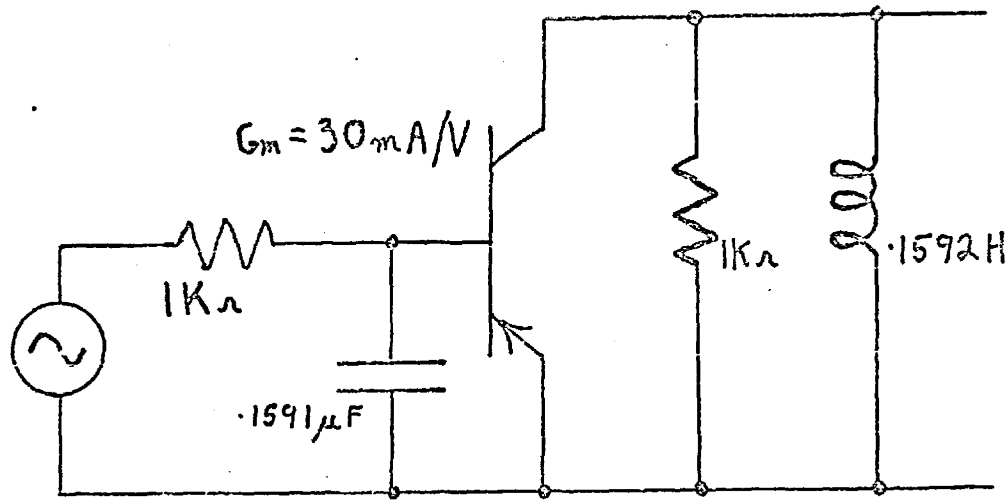
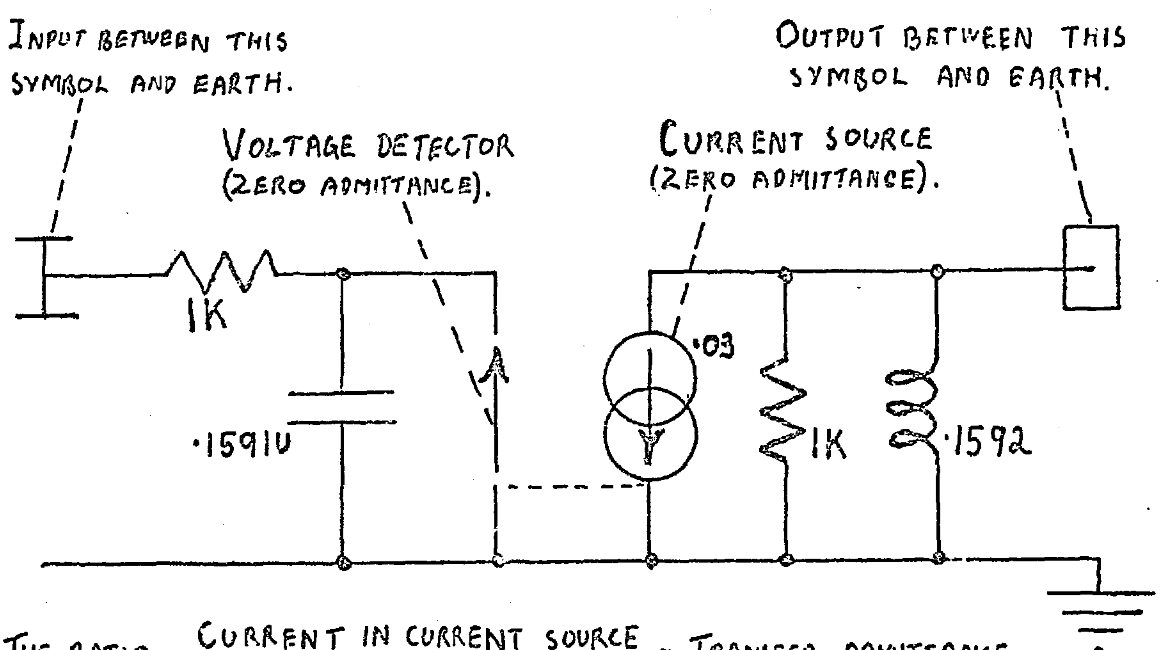


FIGURE 9A SIMPLE BAND-PASS AMPLIFIER



THE RATIO $\frac{\text{CURRENT IN CURRENT SOURCE}}{\text{VOLTAGE ACROSS VOLTAGE DETECTOR}}$ = TRANSFER ADMITTANCE

FIGURE 9B EQUIVALENT CIRCUIT FOR COMPUTER.

the system software. In the instructions for operating the program this part takes two pages to describe but it only takes a minute to do.

When the program is on the disc the simple command E SLAP on the typewriter will cause theprogram to start. The user will first see a title page on the screen which will remain for 30 seconds, unless removed by command, and is replaced by the first page of the operating instructions, shown in Fig. 9C. There are 12 pages of these instructions and they give a complete and concise account of how to use the program.

On leaving the instructions a display as shown in Fig. 9D is seen. A prominent feature of this is the search pattern which covers the majority of the area available for drawing. It will be remembered that the light pen must be pointed at a displayed point for it to be detected*. By using this search pattern the pen may be pointed and detected, and the drawing commenced, anywhere on the screen. When the activated pen is placed on the pattern, the pattern is replaced by a tracking cross and the pen co-ordinates are marked by the current position marker (CPM). The CPM is a small square and it marks the start of the line or wire. On moving the pen, say to the right, nothing will happen until a short distance has been traversed, the cross merely follows the pen but the CPM stays still; then the CPM disappears and a line appears running from the CPM position to the cross. As the movement of the pen continues the line extends until the time comes for the first resistor to be inserted. The pen shutter is

* The pen is fitted with a shutter which may be opened by the forefinger. When the pen has been placed in any desired position on the screen this shutter is opened to start the desired action and closed when it is to be terminated; that is when a line has been drawn or a light button hit.

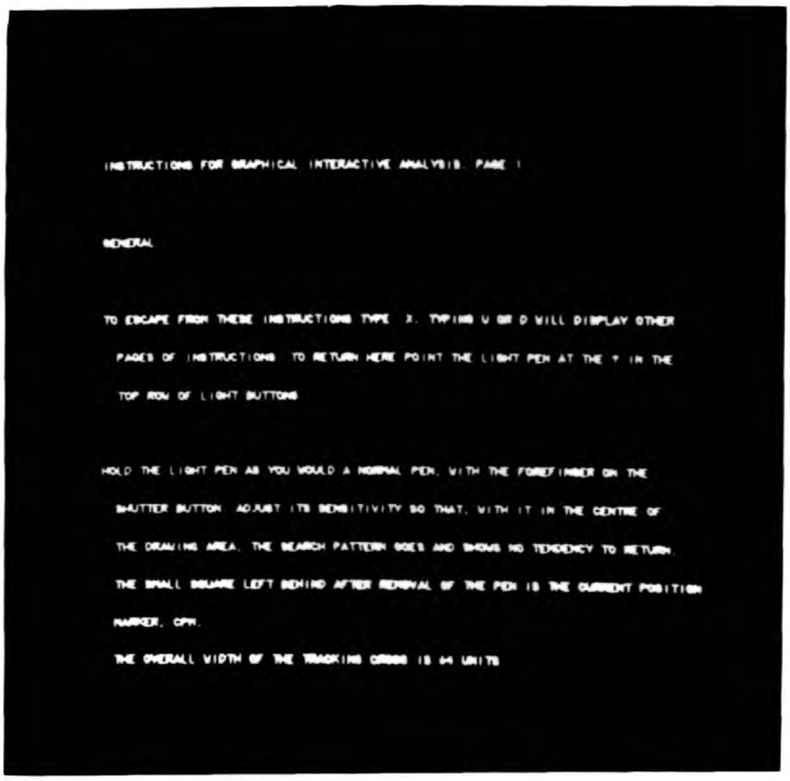


Figure 9C Operating Instructions

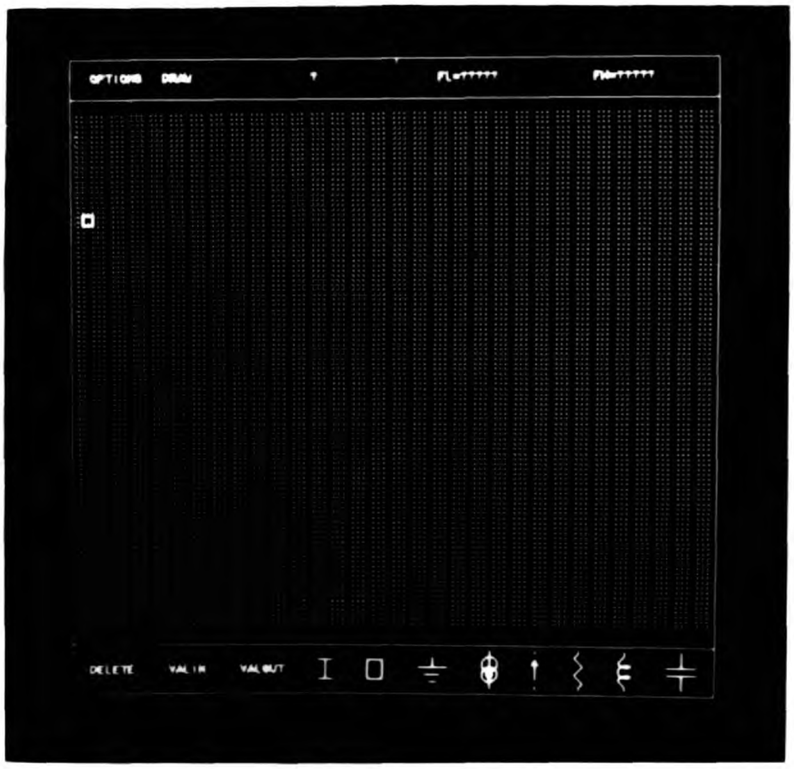
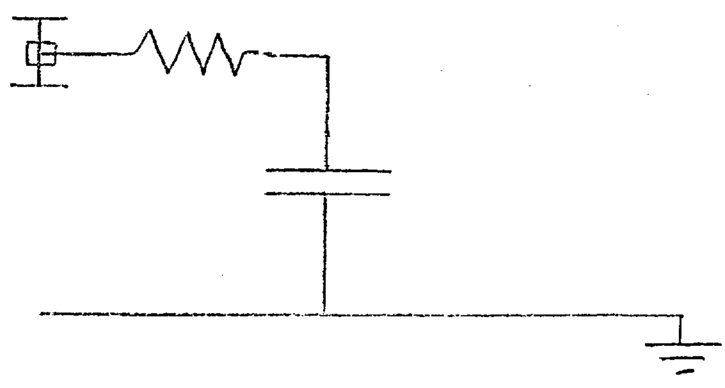


Figure 9D Initial view of Drawing Area

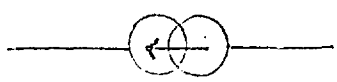
closed (i.e. the pen is deactivated) and the pen removed, leaving the line with the CPM on its new end to indicate that that is where the element will appear if requested.

To request an element the pen is pointed at the appropriate symbol in the light button menu at the bottom of the screen whereupon the element will appear at the CPM with the same orientation as the line. To inform the user that he has hit a light button it is extinguished for $\frac{1}{2}$ second afterwards. If the pen is still present, with its shutter open, when the light button returns then the element at the CPM is rotated one quarter of a turn clockwise, an incremental rotation which occurs every half second until the pen is removed. The pen is now pointed at the unconnected end of the resistor and if the aim is fairly good the CPM will appear on the resistor end so that a connected line may be drawn. If after drawing a short horizontal line the pen is moved downwards a short way the line direction will change, a vertical line will be drawn. After a short distance the pen is removed and a capacitor inserted. Now starting below and to the left of the capacitor the earth line is drawn to near the right hand side of the screen and the earth symbol put on the end. The earth symbol does not rotate with successive hits. At this stage a line may be drawn from the lower end of the capacitor to the earth line, get the end of the line very close to the earth line before closing the shutter and a join will be automatically made, the CPM will not appear,



thereby proving that the join has been made. The input symbol may now be added by first placing the CPM at the input of the network and then using the I light button. The sketch shows the progress made with the diagram at this stage.

The controlled source which simulates the transistor may now be inserted. It consists of two parts, the current source symbol



and the voltage detector symbol

The voltage across the detector controls the current through the source, both devices have zero self admittance. The pen is pointed at the angle between the resistor and capacitor and short lines, first horizontal, then vertical, are drawn and then the voltage detector is added and joined to earth. The pen is then pointed at the earth line at a suitable point to the right of the detector and a short vertical wire drawn. Then the source is added and the remainder of the circuit is drawn. The input of the circuit is between the stylised I and earth, and the output is between the stylised O and earth.

It now remains to give the elements their values. This is done by first pointing the pen at the button marked VALIN and then at one of the elements. The element will start to flash and a set of numerical light buttons as shown in Fig. 9E, will replace the original light buttons

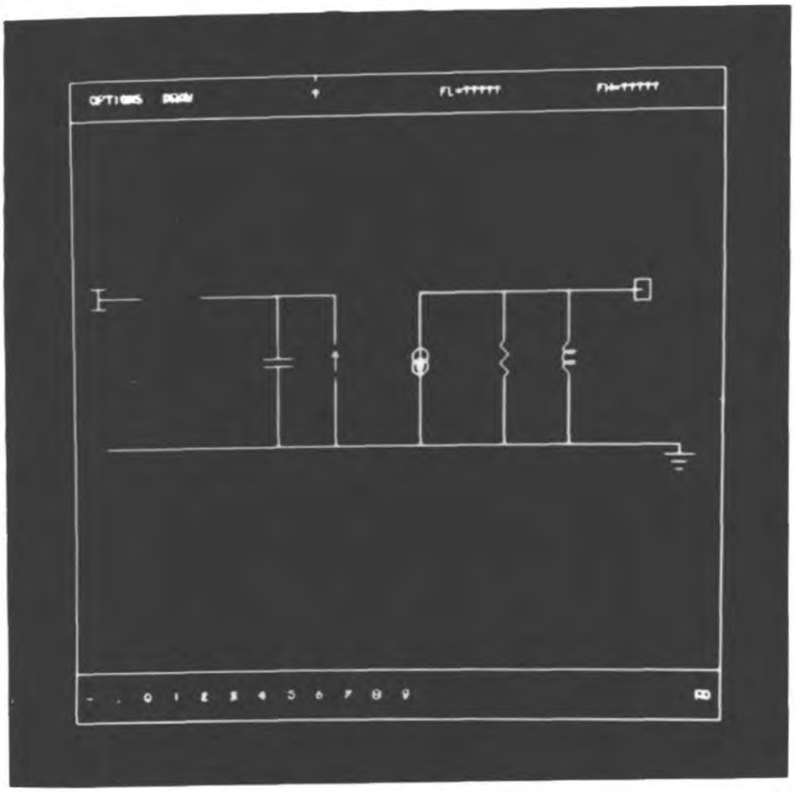


Figure 9E Display of Circuit and Numerical Buttons

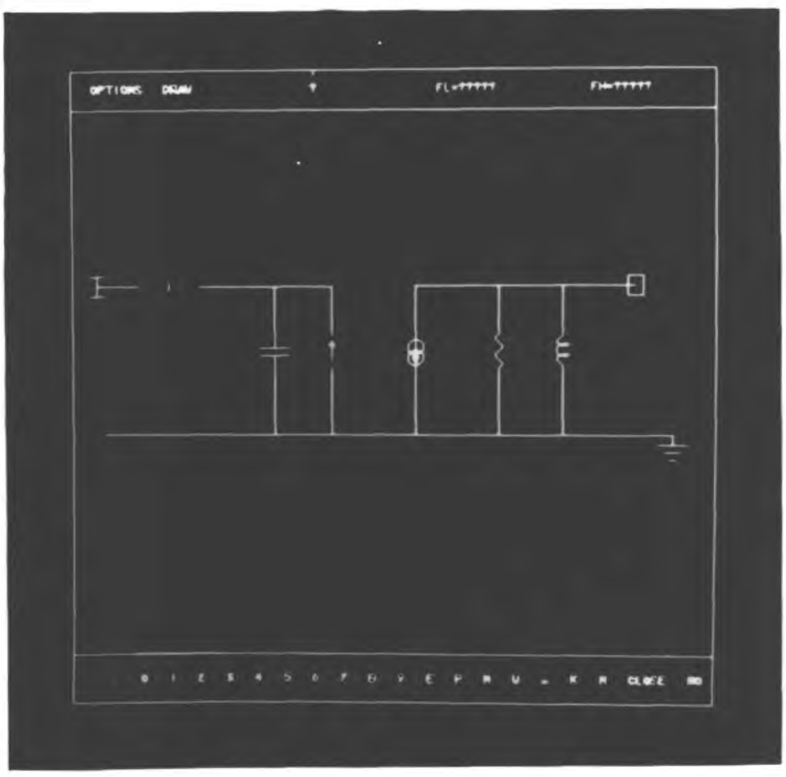


Figure 9F Numerical Buttons after First Digit

at the bottom of the screen. It will be noticed that only legal buttons are displayed. The pen is then pointed at the first character of the value, which will be displayed over the flashing element. The light buttons displayed will then change. The actual change will depend on what was chosen, in any case the minus sign will go, and if the point was chosen, so will that; if a digit was chosen then the buttons will appear as in Fig. 9F. The extra buttons, which are those permissible after a digit, are the E for E format, and various versions of the closure action. There is the simple CLOSE button, and also the letter multiplier buttons which give various powers of ten in addition to closure. If the E is chosen, then all the closing buttons go, as does the point, if it hasn't already, but the minus reappears. Once a digit for the exponent has been chosen all the closure buttons return. The number is tested for size and if a value outside the machine range is found an error diagnostic is given. Only the first eight characters input are displayed, but all characters within the accuracy of storage contribute to the final value. When one value has been closed another element may be chosen and its value input in the same way. The voltage detector and the current source must be associated together and this must be done when the values are being input. The pen is pointed first at the voltage detector and then at the current source to be associated. The result is that they are linked by the dotted association line.

The original light buttons are restored by use of the DRAW button and then the values may be inspected by means of the VALOUT button. A hit on this button causes all the elements to be replaced by their values (Fig. 9G) and a second hit on the button restores the element symbols. In practice the pen may be left pointing at the button when the elements and

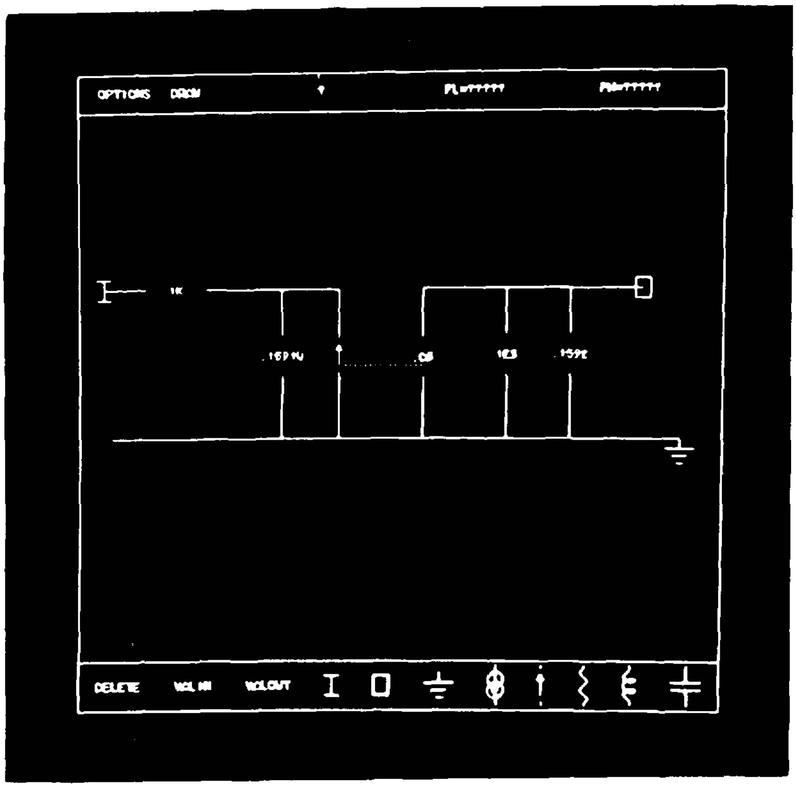


Figure 9G Use of VALOUT Button

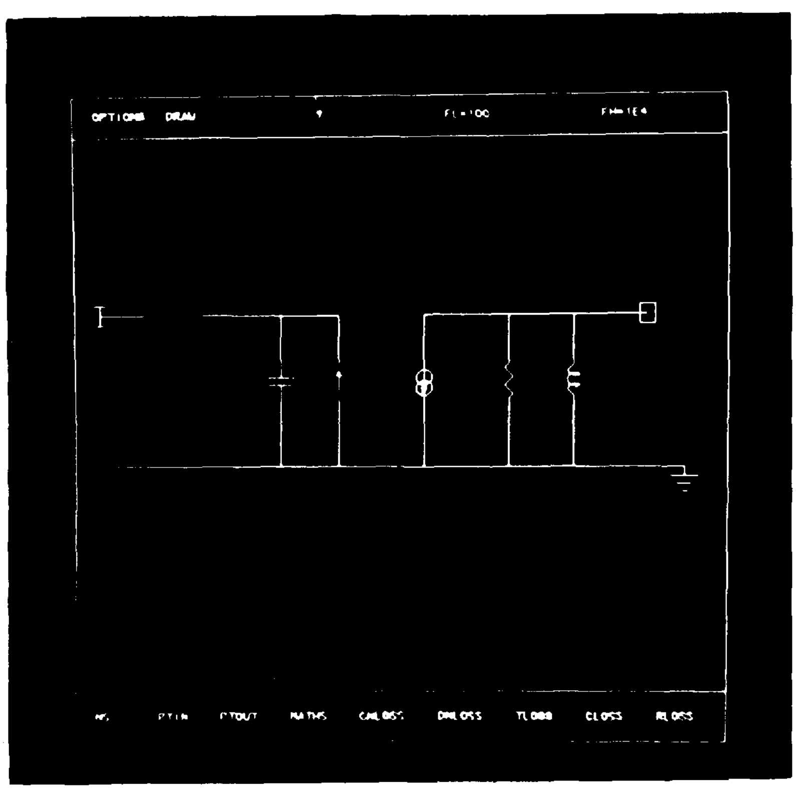


Figure 9H Analysis Buttons and other Options their values will alternate at half second intervals.

The frequency analysis is carried out over a range of frequencies which is specified by using the FL and FH light buttons. These are used in DRAW mode, and the numerical values of the frequencies are inserted just like any other values.

When the circuit, its values and frequency range of interest have been entered correctly an analysis may be made. The analysis buttons, shown in Fig. 9H, are several in number, and we chose GNLOSS which stands for "Generate Nominal Loss"; the circuit will be analysed and the results displayed and stored. As soon as the button is used the whole displayed picture will disappear and be replaced by a small cross in the top left hand corner. The cross will move about the screen as the information for the output display is generated, the most obvious feature being a slow progression from the left of the screen to the right. This may take from six seconds to many minutes to complete and occurs while the actual analysis is being carried out. The frequency range given is divided into 220 geometrically equal intervals and the analysis is made at each of the resulting 221 frequencies. For each one of these frequencies the cross, which starts at the left hand side of the screen, is moved a short distance to the right. The arrival of the cross at the right hand side of the screen coincides with the termination of the analysis, a characteristic which enables the user to judge how long he has to wait. When the analysis is complete the machine types out a short message concerning the circuit and analysis. In this case it will be

PDP9 19S 4N 4E 1CS

which tells that the analysis was done by the PDP9 and that it took 19 seconds, and that the circuit has 4 nodes, 4 elements and 1 controlled source.

After this message has been given the cross makes a few more rapid movements and then the output graph is displayed as shown in Fig. 9I. This has a linear scale of decibels of loss on its vertical axis and a logarithmic scale of frequency on its horizontal axis. The negative values of loss mean that the circuit is in fact giving a gain. The extreme frequencies are the values given for FL and FH and are so labelled. The graph has a caption which in this case is N LOSS/FREQUENCY. The corresponding phase graph, shown in Fig. 9J, may be requested by pointing the lightpen at the word LOSS. It has the same frequency scale but the vertical axis is scaled linearly in degrees. The original loss curve may be retrieved by pointing the light pen at the word PHASE.

At this stage various typewriter commands become available. They use the same scheme as that described in Section 3. Thus to discover the accurate value of the network loss at 1KHz we type*

S 1E3

with the result shown in Fig. 9K. "TLOSS" stands for transmission loss; in this case a gain of 23.525 dB.

To return to the graphical input another typewriter command is used, this is X (for exit) and a space. The diagram returns in a few seconds and may then be modified. Suppose that we wish to decrease the gain by 10dB. We may do this by changing the value of the 1KΩ series resistor

* The line must be closed by a RETURN or space. S means single frequency, which is not restricted to the range FL to FH.

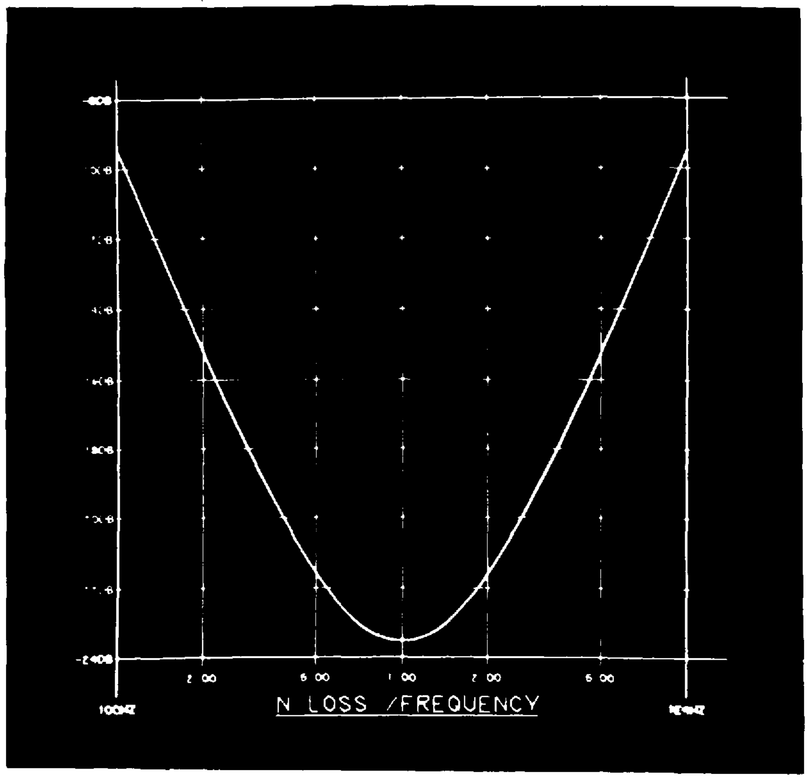


Figure 9I Nominal Loss

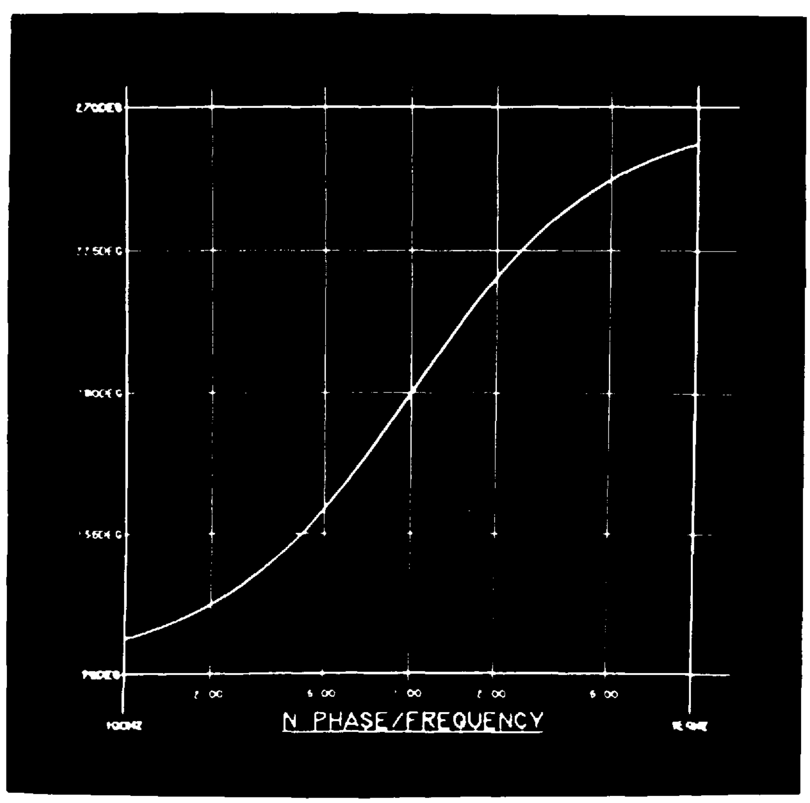


Figure 9J Nominal Phase

```
PDP9   19S   4N   4E 1CS

S 1E3

T LOSS AT  1.00000E+03HZ    -23.525DB   179.998DEG
```

FIGURE 9K SINGLE FREQUENCY INTERROGATION

```
MATHS. TYPE Y OR N.

N
10 F .5 =

  3.162278E+00

* 1E3 =

  3.162278E+03

/ 2.162278 =

  1.462475E+03
```

USER TYPED ?

```
?+-*/IP=RKX
```

MACHINE LISTED ACCEPTABLE COMMANDS

USER CHOSE X

FIGURE 9L EXAMPLE OF DESK CALCULATOR ROUTINE

to a new value of $1000\sqrt{10}\ \Omega$ and adding a new resistor of value $1000\sqrt{10}/(\sqrt{10}-1)\Omega$ in parallel with the capacitor. The calculation of these may be performed by using the desk calculator option provided. Operate the OPTIONS button and then the MATHS button, the circuit diagram will be extinguished and the typewriter will type MATHS, TYPE Y OR N as shown in Fig. 9L. If Y, and a space or return, is typed then the machine will respond with a set of operating instructions and wait for the users input. By typing N and a return we omit the instructions. To evaluate $\sqrt{10}$ and print it out we type

10 P .5 =

The value of $\sqrt{10}$ is still retained by the machine and we multiply it by 1000 and so obtain the value of the series resistor by typing

* 1000 =

To obtain the value of the shunt resistor we now type

/ 2.16 =

The routine accepts an alternation of real numbers and single letter operators. We may find out what it is expecting by typing a ?. If we do this after the last answer we get the list shown in Fig. 9L which is explained in the instructions, although many are obvious. By choosing X we return to the graphic input.

We may now make the desired changes to the diagram by pointing the pen at the right hand end of the input resistor whereupon the CPM should be seen positioned at the resistor end. If it is in the correct place we draw a short line downwards and put a new resistor on its end and finally join the lower end of the resistor to the earth line. This is shown in Fig. 9M and all that remains is to insert the element values. This is

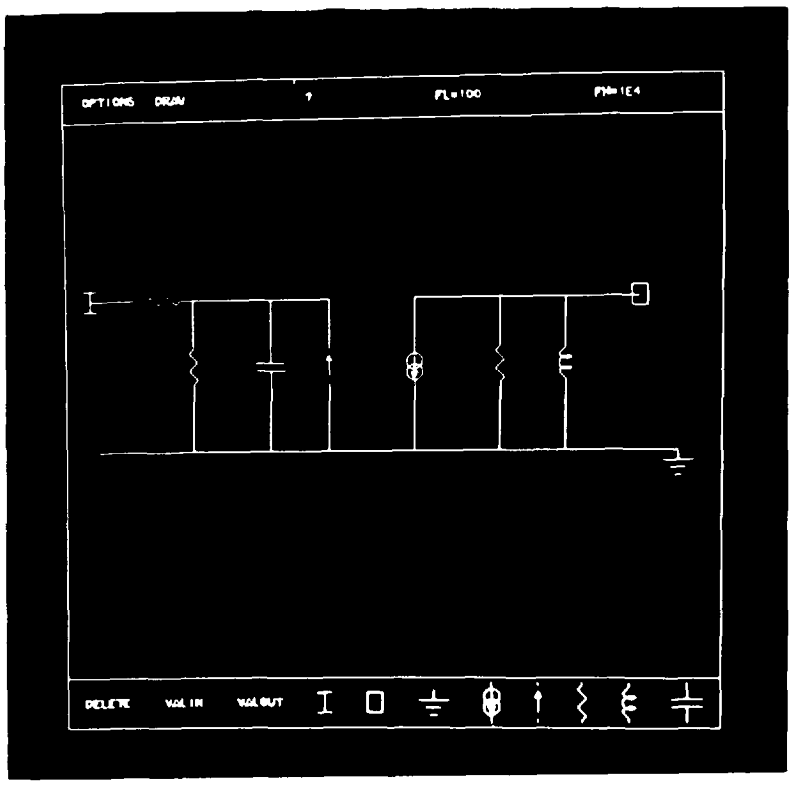


Figure 9M Changed circuit

done, after using the VALIN button, in exactly the same manner as previously, a new value overwriting the old. The new response may be examined exactly as was the old by using the GNLOSS button.

It is possible to preserve the circuit details by two methods, one tabular for perusal by the user, the other by means of paper tape*. To get a tabular output of the circuit the command letter H plus space is typed. This gives the output as shown in Fig. 9N where the Y or N refers to the provision of instructions. After typing N, a list of frequencies and increments is typed. Each frequency is preceded by the letter F and each increment by I. An increment is added to the previous frequency until the succeeding one is reached. When the desired frequencies have been entered E (for execute) is typed to give the output shown. It will be seen that the circuit is included, thus a complete record is available.

On returning to the input diagram by use of the X command the other type of permanent output becomes available. A light pen hit on OPTIONS causes the button PTOUT to be displayed at the lower edge of the screen. On hitting this button a paper tape containing a description of the input diagram will be punched out.

The paper tapes punched out using PTOUT may be read in by using the PTIN button, but first if there has been any previous drawing on the screen, it is necessary to delete it by use of the NS (New Start) button. This is displayed at the extreme left after use of the OPTIONS button and

* One may also photograph the screen of course.

```
H

TYPE Y OR N


N
F 100 F 900 I 50 F 1100 F 1E4 E
```

FREQUENCY ANALYSIS OF LINEAR NETWORKS WITH UP TO 16 NODES
TWO PORT VOLTAGE LOSS IN DB AND DEGREES

NODE 1 IS THE INPUT, NODE 2 IS THE OUTPUT AND NODE 4 IS COMMON.

ELEMENT	NODE1	NODE2	VALUE
RESISTOR	1	3	3.162280E+03 OHMS
CAPACITOR	4	3	1.591000E-07 FARADS
RESISTOR	4	2	1.000000E+03 OHMS
INDUCTOR	4	2	1.592000E-01 HENRIES
RESISTOR	4	3	1.462480E+03 OHMS

VOLTAGE CONTROLLED CURRENT SOURCES

INPUT 1	INPUT 2	OUTPUT 1	OUTPUT 2	GM
4	3	4	2	3.000000E-02 MHOS

FREQUENCY HERTZ	LOSS DECIBELS	PHASE DEGREES
1.0000E+02	0.5415	101.4208
9.0000E+02	-13.4764	173.9727
9.5000E+02	-13.5131	177.0607
1.0000E+03	-13.5246	179.9983
1.0500E+03	-13.5142	182.7926
1.1000E+03	-13.4852	185.4509
1.0000E+04	0.5410	258.5735

FIGURE 9N TABULAR OUTPUT.

it deletes any existing diagram and resets the program storage to zero. It does not delete the frequencies but these can be overwritten by a read-in paper tape, it is not possible however to overwrite any diagram. To read in a tape, insert it in the reader the correct way round and check that it runs properly by means of the button above, then use the PTIN button.

A circuit which has been stored on paper tape is shown in Fig. 9P and by using GNLOSS its transmission response can be displayed as shown in Fig. 9Q. It is also possible to examine its return loss by changing the output symbol from its existing position to the position shown in Fig. 9R and then using the RLOSS button. The result is shown in Fig. 9S and represents the return loss of the network driving point impedance between the output symbol and earth, against the resistor between the input and output symbols. When moving the output (or input) symbol it is only necessary to put the CPM at the new position and then point at the symbol button. Since only one output, or input, is allowed, as the new item is inserted the old will be automatically deleted.

Suppose that it is necessary to determine the effect of the inductor dissipations on the filter transmission response. The dissipation may be simulated by the addition of a resistor in series with each inductor, and in order to do this it is necessary to redraw parts of the circuit. First of all hit the DELETE button and then with great care point the pen at one inductor and open the shutter momentarily. The element will disappear, then delete the other in the same way, and restore the drawing action by pointing at DRAW. Place the CPM at the LHS of one of the gaps and then point at the inductor button, the inductor just deleted will

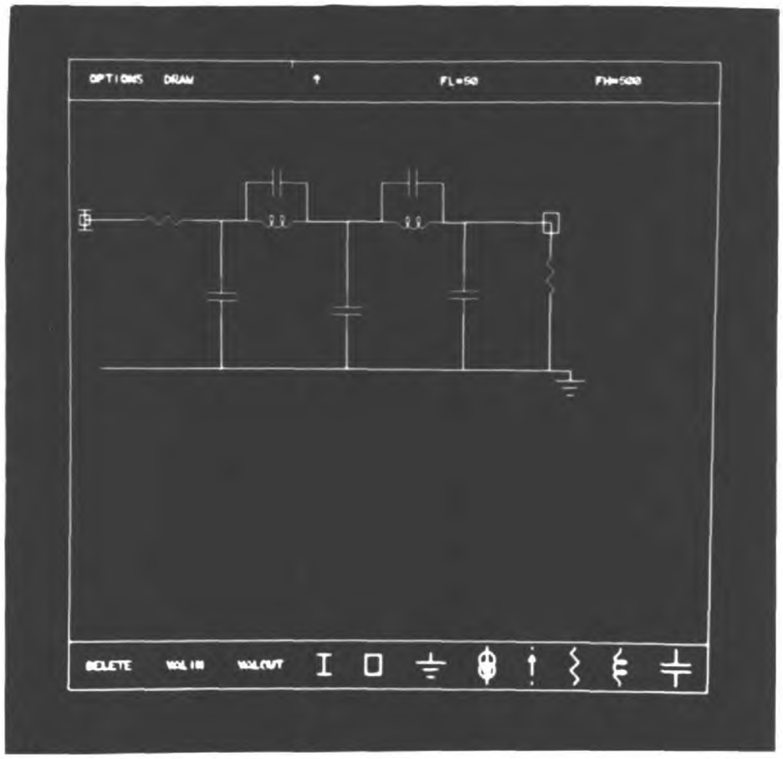


Figure 9P Circuit from paper tape

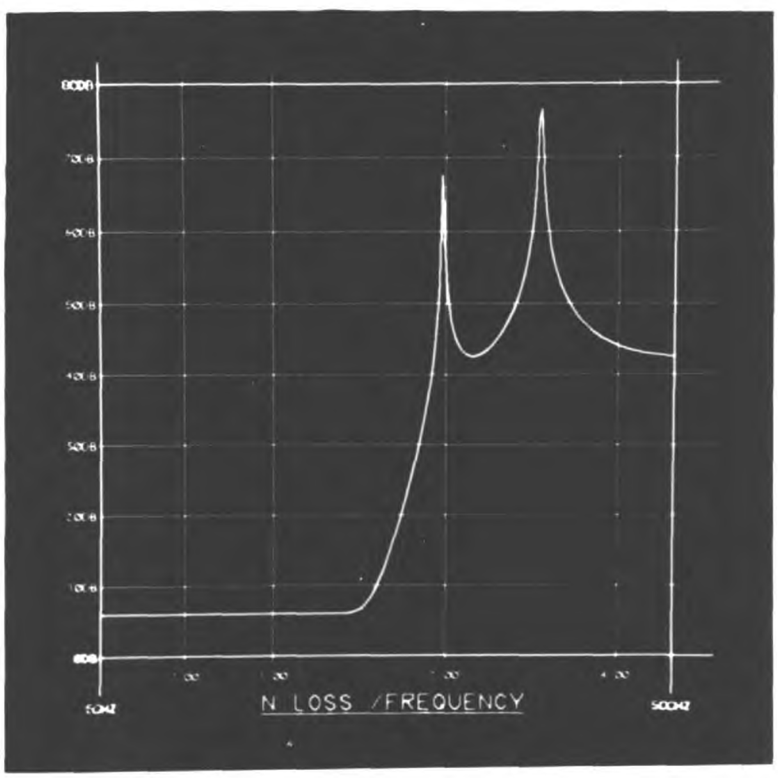


Figure 9Q Response of Figure 9P

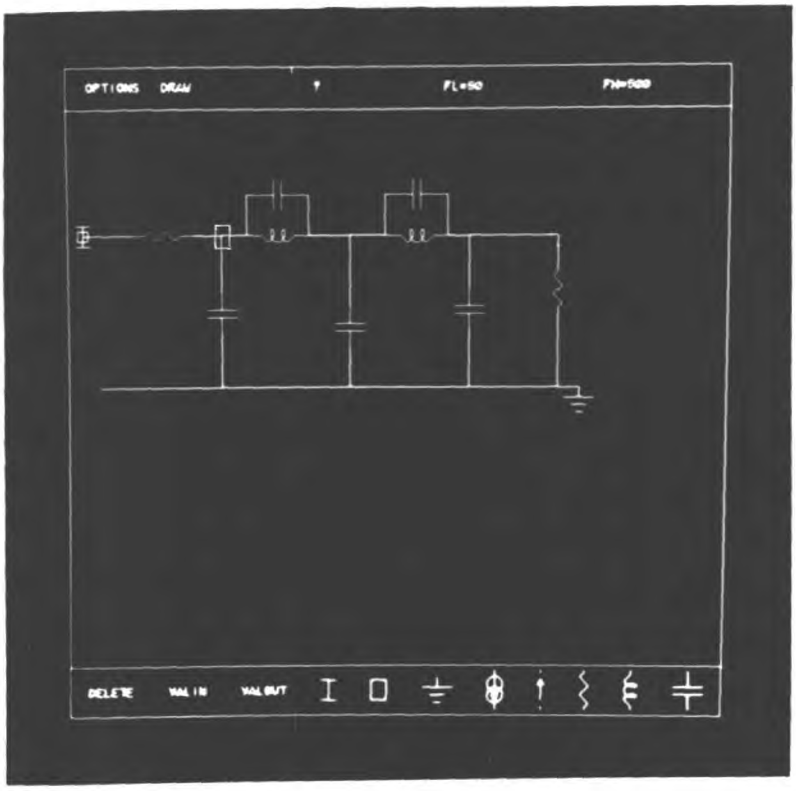


Figure 9R Inputs and Outputs for Return Loss

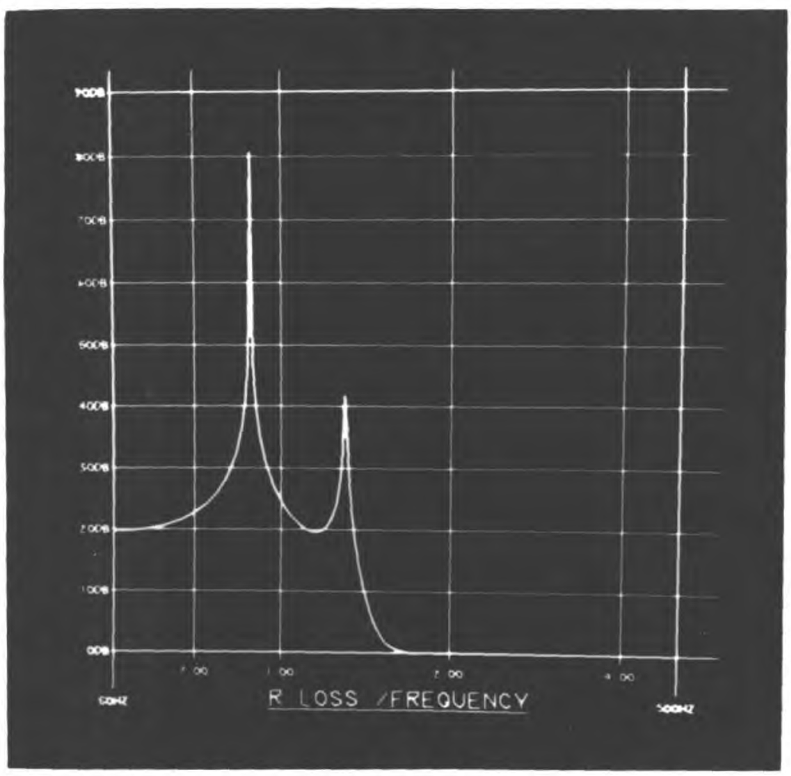


Figure 9S Return Loss

reappear. Point at the inductor button a second time and the element will rotate so that it points downwards. Then place the CPM on the other side of the gap and insert a resistor, drawn downwards in the same way. Then put the CPM on the open end of the inductor and draw a horizontal line to end very close to the free end of the resistor. The line will join to the resistor and the CPM will not appear at the line end as it usually does. Repeating the operation for the other inductor leads to the circuit diagram shown in Fig. 9T, the output symbol having been restored to its previous position. Values are inserted as before, the original values of the inductors being restored and the resistors having suitable values.

We may now do the analysis, using the GNLOSS button if we wish, but it is often desirable in a case like this to know the change of response. To obtain the change of response we use the CLOSS button and we get the curves shown in Fig. 9U and Fig. 9V. The loss and phase axes have been rescaled to give better presentation. We could have seen the actual response by using the TLOSS button, this is similar in action to the GNLOSS except that the stored nominal results are left undisturbed. These stored results may be viewed, no analysis being done, by means of the DNLOSS button (Display Nominal LOSS).

The time taken for the analysis varies from six seconds, for a simple two element L section, upwards. The circuit shown in Fig. 9W which has 18 elements, 16 nodes and 5 controlled sources takes $5\frac{1}{2}$ minutes. The program will accommodate 50 elements, 16 nodes and 7 controlled sources as a maximum.

There is another version of the program in which the analysis for

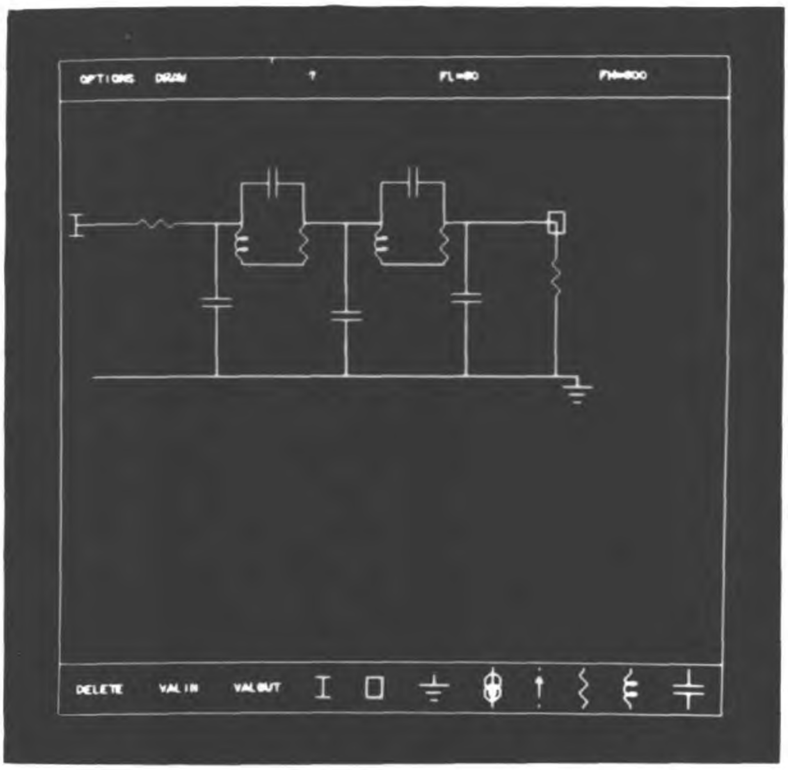


Figure 9T Circuit with dissipation

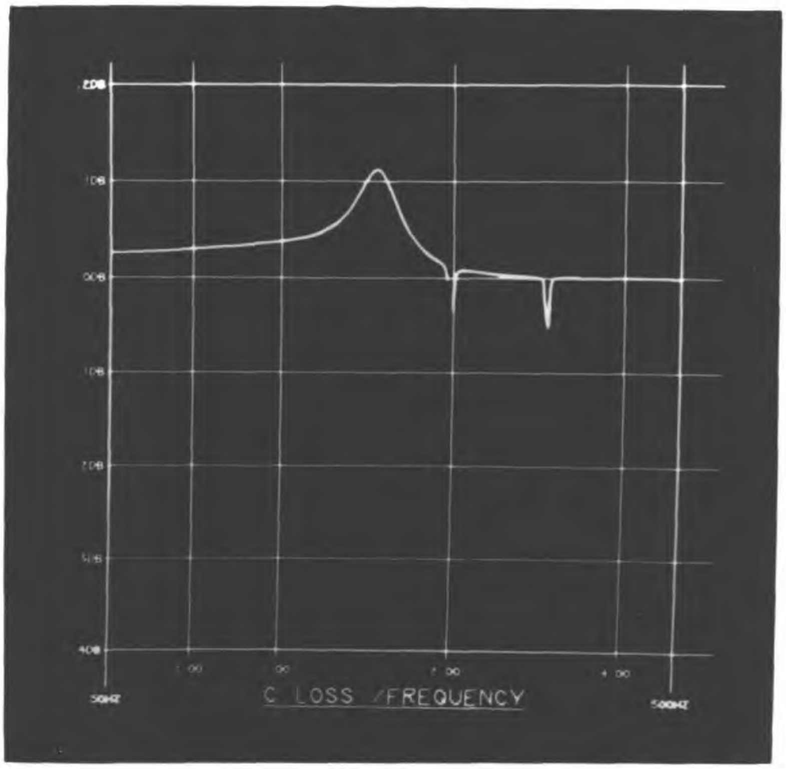


Figure 9U Change due to dissipation

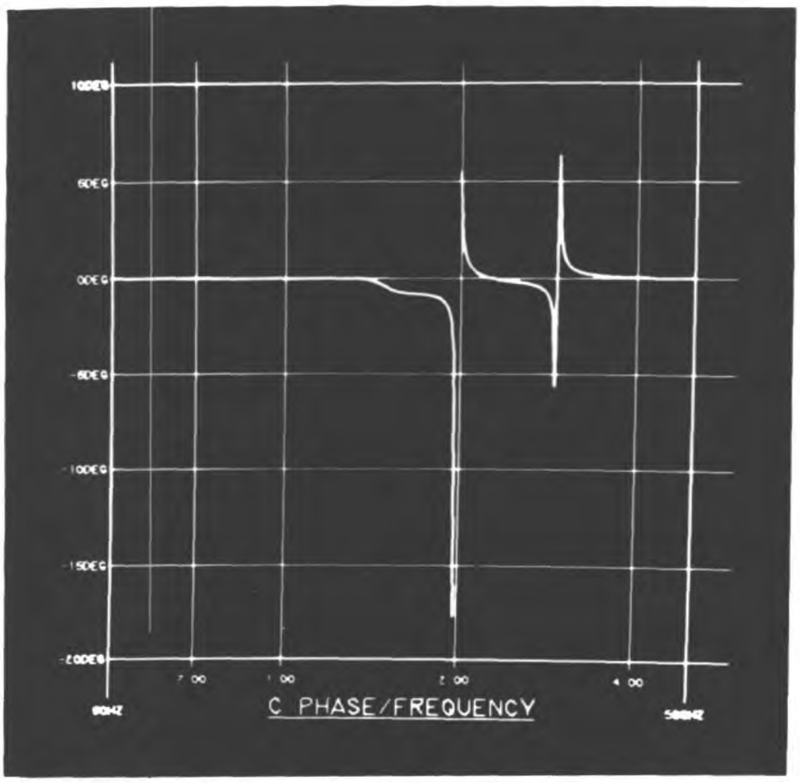


Figure 9V Change due to dissipation

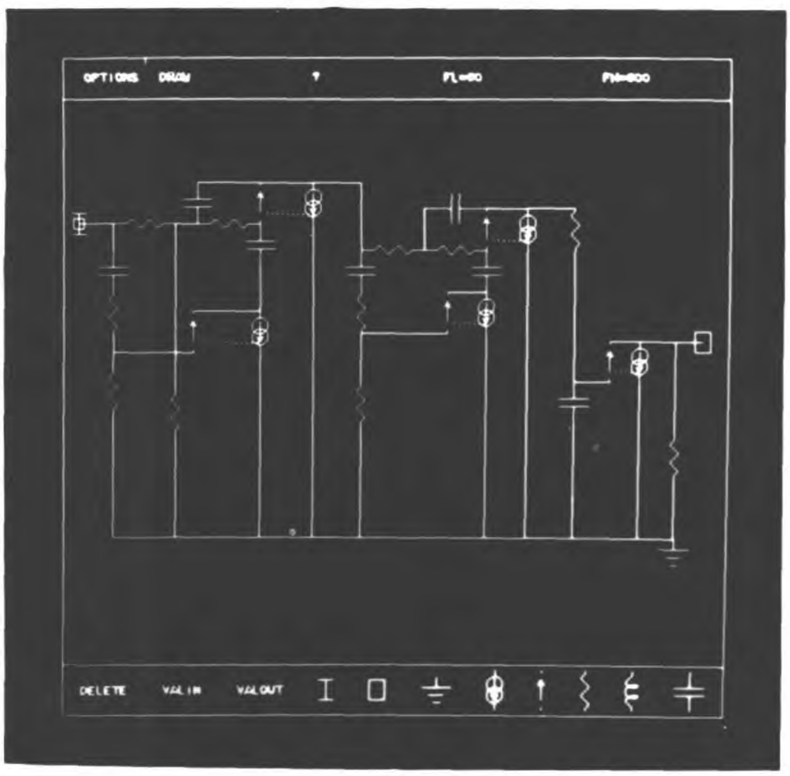


Figure 9W Large circuit

graphical output is carried out on a Univac 1108 machine accessed over a telephone link[31]. No changes in the size of the circuit handled by the program have been made. The analysis delay for this set up varies from 30 seconds to $1\frac{1}{2}$ to 2 minutes and is mainly due to queuing time at the 1108, it is therefore of great potential value for handling the larger circuits. However, it suffers from one overwhelming disadvantage, that of unreliability, and consequentially is seldom used.

To aid the user who wishes to draw a large circuit all versions of the program have a space marker. This is a short vertical line which moves along the top frame line. The distance between it and the frame line on the left is proportional to the unused storage space. Compare Figures 9D, 9E and 9W.

10. EVALUATION OF FACILITY

The program just described (known as SLAP - Small Linear Analysis Program) is discussed item by item and compared with other circuit analysis programs using graphics. These are graphical ECAP[9], graphical CIRCAL[7], PIXIE[14], ISLEND[32], and MACLEAN[29]. The people whose reactions are described range from those who have only seen the program being used to those who have used it for several hours over a period of days or weeks. Among them are college lecturers, professional engineers, and postgraduate students.

10.1. Loading the Program

There has been one failure so far by a user trying for the first time to run the program solo. This was a failure of the handbook, which the user had read. The procedure is complicated by the small disc, there is no doubt that this type of program should be kept in some form of rapid access bulk storage that is permanently attached to the machine. The failure was associated with putting the magnetic tape on the tape unit, similar difficulties occur with paper tape readers, disc packs, etc. The procedure, apart from these mechanical operations, compares quite favourably with the signing on procedures for remote access time shared systems.

10.2. Operating Instructions

Since a duplicated handbook has been provided the instructions incorporated in the program have been little used. The program has been written on the "maximum choice, minimum order, anything legal" principle; that is, the user is allowed to do anything he pleases, when he pleases, and is protected against catastrophic mistakes; light buttons have been given moderately informative labels. With ISLEND and graphical ECAP an operating guide is given by screen captions as the input progresses. PIXIE gives no help and has rather mysteriously labelled light buttons. With ECAP there is the possibility of making a catastrophic mistake.

10.3. General Display

One of the more noticeable features of the display (Fig. 9D) is the search pattern covering the whole drawing area. Despite many doubts regarding its reception there have been no comments at all concerning it. A few peopl· have remarked on the pronounced flicker of the display, especially when a rather large circuit (Fig. 9W) has been drawn, no one seems to have been seriously upset by this flicker when using the program. One left-handed user was most upset by the positioning of the light pen at the right hand side of the screen. The display area and the light button areas are framed within lines; once again most people have made no comment, a few have said they like their presence, and a few have said that they did not. ECAP, MACLEAN and CIRCAL have no framing lines while PIXIE and ISLEND have. There is another version of SLAP (Fig. 10A) which has no framing lines and which concentrates all the light buttons at the bottom of the screen. It still has the search pattern and the flicker; due to lack of use there are few user reactions to it.

10.4. Positioning of Wires and Elements

This is done by simply pointing at the desired position, which will

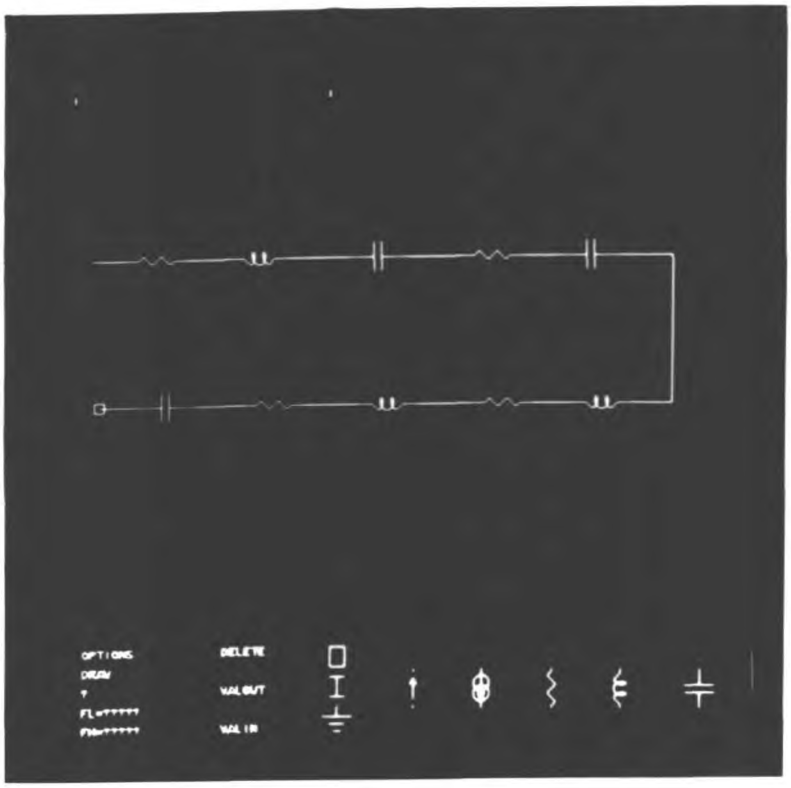


Figure 10A Alternative Display

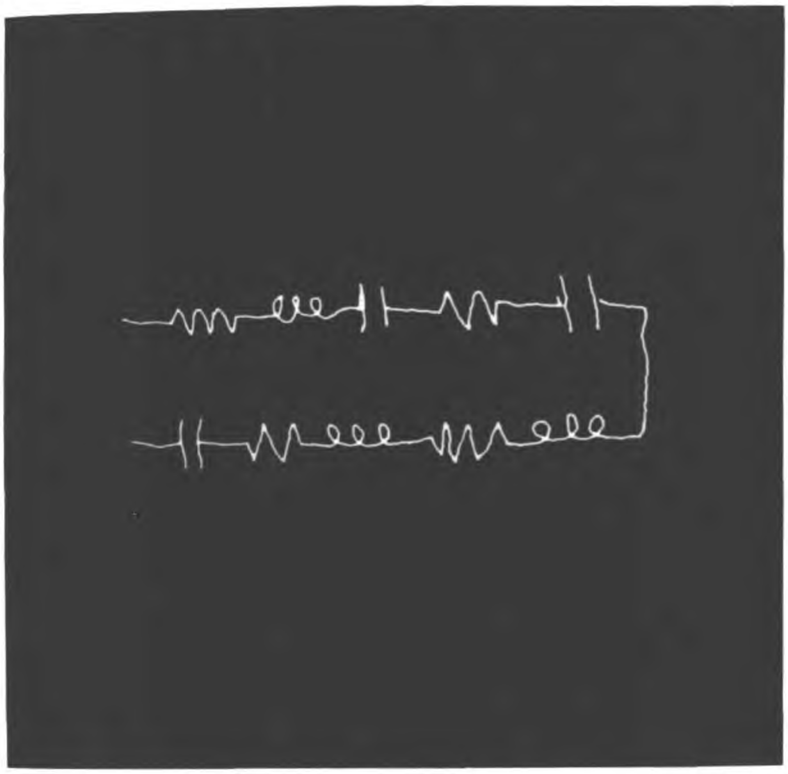


Figure 10B Free drawing program

then be indicated by the current position marker. This simple action is made possible by the search pattern which enables a rough pen position to be established which is then made precise by display of the tracking cross. Some users have shown a tendency to attempt to drag the CPM to the new position and to try and discourage this it was changed from a cross to the existing square. Some programs such as CIRCAL and ISLEND use a coarse grid of points, which are the only permissible starting positions for lines and elements, other programs require elements to be dragged up from a menu to the required starting position. A wire can only be started from an existing wire or element. PIXIE and MACLEAN use a tracking cross which is permanently on the screen and for which there are two modes of operation, one for moving it without drawing a wire, and the other for drawing a wire; ECAP has a search pattern but it has to be initiated by the user operating a function button with one hand while holding the light pen on the screen with its shutter open with the other. If the search pattern had been initiated by the opening of the shutter, the arrangement would I believe have been almost the best possible, obviating the distraction of the SLAP search pattern while preserving its convenience.

10.5. Drawing of Wires

When the search pattern detects the presence of the pen the pattern is replaced by the tracking cross which will then follow any movement of the pen. The cross, which is predictive*, appears adequately fast in

* The cross is not displayed at the last pen position as is usual but at the expected next pen position. This is predicted from the two previous positions of the pen and assuming constant pen velocity.

response, no case of inadvertant loss being noted. The wires drawn are constrained to be either horizontal or vertical which gives a neat diagram but leads to certain difficulties with joins between wires and wires, and wires and elements. Two features in particular in this joining process have caused difficulty, they both occur at the finish end of a wire or element, that is the end not marked by the current position marker when the element or wire position is being determined. When a wire has been drawn and the pen removed the program will attempt to join the wire to an element or another wire. It will do this by adjusting the length of the wire so that its end falls on a suitable point on the other item and a join will only occur when this is possible. When one wire is drawn towards the side of another, since the whole of a wire is made of suitable points the joining is easy but if an attempt is made by the user to "hit" the wire (or element) end and he is just outside it no join will be made even if the two ends are within one increment (0.009") of each other. This sort of thing causes the user to "scribble" over the hoped for junction which does not have the desired effect. The other difficulty occurs with elements, if an element is deleted and another inserted in the same position it will be joined up correctly but if when an element is inserted, its "finish" end falls nearly or even exactly on a wire it will not be joined, it will join only when it falls exactly on the end of a wire or another element. These features of the program have not been changed because a knowledgeable user, i.e. the programmer, has had no difficulties, not an adequate reason! An improved algorithm for this joining is required, or a method for the user to determine which items are joined to a given node, or both.

Most other programs circumvent these problems by restricting circuit junctions to grid points, but PIXIE has a similar drawing scheme. Here the joining action is much better, due to a more elaborate data structure, a superior algorithm and the use of down and up compilers. ECAP is similar to programs using grid-points except that their position is by choice of the user. ISLEND has a special symbol for jumping over a grid-point, here the problem is one of avoiding joins. In the case of ISLEND and CIRCAL wires are inserted in the same way as elements, they are of fixed length. For MACLEAN a tracking cross is used but an initial light button hit is required to set the wire direction and another if it is required to start the wire on an existing item. Another light button hit is needed to finish the wire or to alter its direction.

10.6. Selection and Orientation of Elements

Selection is done by means of a menu of light buttons and this is the method used by most programs. The element appears at the CPM position, initially with the same alignment as the previously drawn wire, or horizontally if the pen was just pointed at the screen. To prevent multiple interruptions the light button is blacked out for $\frac{1}{2}$ second after being used, this also serves to tell the user that he has hit it. If the pen is present after the button is redisplayed the element at the CPM will turn through 90 degrees clockwise, this will repeat for each hit on the button, the element rotating once in 2 seconds. Some users have had difficulties with this, and other buttons, due to keeping the pen on the button too long but after appreciating how it works they have had very little trouble.

A few simple tests on the time taken to input elements have been made.

The diagram shown in Fig. 10A* was used, it consists of 10 elements and 13 wire segments and it took about 20 - 25 seconds to draw. That is about 2 seconds for an element and a short segment of wire. Fig. 10B shows the same diagram drawn using a free drawing program as a simulation of character recognition. The time taken was about the same. It is probable that PIXIE will improve on this for here the "toing and froing" of the hand, between the line and the menu, has been largely eliminated by grouping the menu items round the tracking cross. The buttons are of necessity rather small and in fact consist of single letters and there are ten of them. They are arranged in two sets of six, one being used to get the alternative set. I think that the use of two sets destroys the idea to some extent, it might be better to put the majority of elements in a normal menu and only the four or six most used ones round the pen. In PIXIE, elements always take the orientation of the original line.

A light button menu is used in MACLEAN, an element may be selected from it after a wire has been drawn and the element takes the same orientation as the wire. In CIRCAL elements are chosen by function buttons, a button for one of four orientations first being used, then a button for the element type, the element appears at the tracking cross position. ECAP and ISLEND, both of which use grid-points, have the element orientation determined by the grid-point position and the type is chosen from a menu. In the case of ECAP any orientation is possible.

10.7. Numerical Input.

Light buttons are used for the input of numerical values and

* Also shows the layout of the alternative display.

difficulties of three types have become evident. One type is the getting of unintended digits due to not opening and closing the light pen shutter cleanly. An improvement might be made by spacing the buttons further apart but at the expense of losing some of the drawing area, this would slow down the more experienced user. The alternative display layout* was intended to help investigate this problem but what little evidence is available indicates that it is no better in this respect. Another type of mistake is omission of the CLOSE button. If a value using a letter multiplier is input e.g. 2K, 32m or 1M, then closure is not necessary, but for values like 10, 27.6 or 1E-3 it is, and this is often forgotten. The result is a puzzled state on the part of the user, he has finished the number but the machine will not allow him to continue. The last case of difficulty arises from unnecessary differences between the inputting of frequencies and element values. Before an element is pointed at, the machine must be set for deletion of the element or input of its value and this is the purpose of the DELETE and VALIN buttons. However when the user points at the FL or the FH buttons he can only require to input a frequency, the prior use of the VALIN button is not necessary. However, it should be allowed; programmed as it is, it is prohibited, which has quite often caused confusion. The light button input of values has proved less succeptable to errors than typewriter input although some users have said that they preferred the typewriter, this even includes users who have had to abandon the program due to difficulties with typewriter input!

* See Figures 10A and 10C.

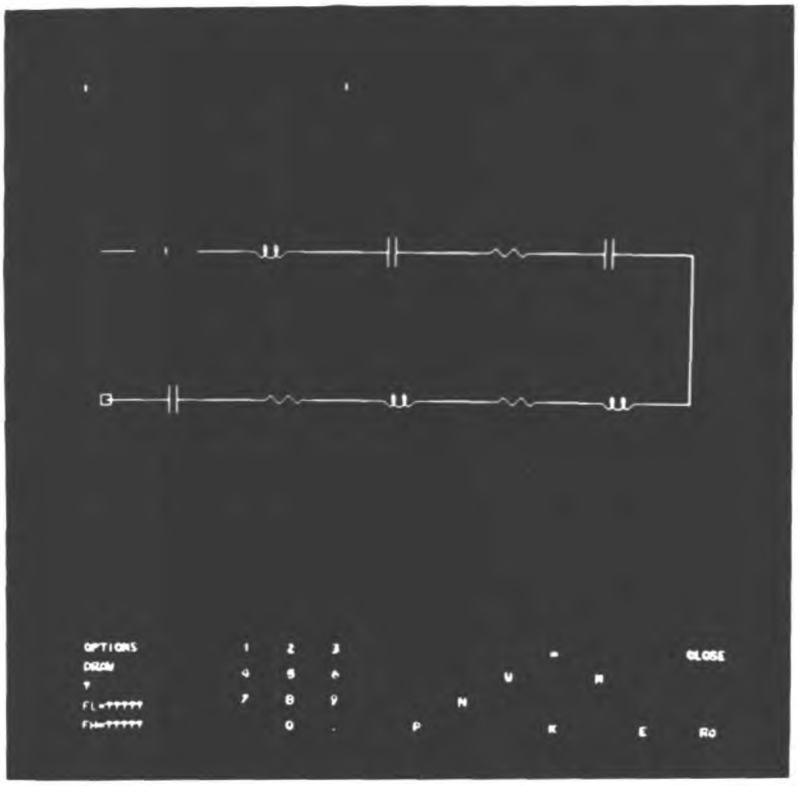


Figure 10C Alternative Display with Numerical Buttons

Most programs use the typewriter for numerical input, very often in the form of character strings the validity of which is not checked until some time later, there are a few cases of numerical input using the light pen(29, 33). MACLEAN arranges that a number is first put into temporary visable storage and then assigned to elements later, this is a useful feature since it saves time if a number of elements have the same value.

10.8. Analysis

The work was not done with the idea of testing analysis programs but there have been a number of useful comments. Perhaps the most common is the desirability of a reduced analysis so that only say 20 points are calculated initially instead of the normal 220. The best solution would be to arrange that the main analysis does not start at one end of the frequency range and work towards the other but rather jumps about so that it can be terminated at any time while still giving a fairly even spread of points over the whole range. Another suggestion is that there should either be some access to the Y parameters calculated for the three terminal network or facilities for plotting functions of them other than those provided, say current gain. One user said that he would like to know all the nodel voltages and element currents at one frequency. This was as an aid to checking that he had the correct input and for testing for possible overloading in the network.

10.9. Output

The output graphs which are of either transmission or return loss and phase against frequency have caused a few difficulties. They are scaled logarithmically in frequency along the abscissa and linearly in loss along the ordinate, either in decibels or degrees. People, such as amplifier designers, more used to graphs of gain have been disturbed to see points of high gain at the bottom of the screen. The phase curves have caused a little difficulty due to the fact that they can be discontinuous. A change from say 140 degrees to 320 degrees should probably be represented by a gap, while since phase is cyclic a change from 350 degrees to 10 degrees is probably continuous. The program has been written so that if the dependant variable changes by more than half the screen height between two adjacent values of frequency a gap is put in the graph line. This will very often give a good representation of a phase curve and sometimes for a loss curve also, often however the effect is not really satisfactory*. One reason for the large (220) number of points plotted was that this was the easiest way of solving some of these problems, an analysis with fewer points would make the problem more difficult.

Some users have said that a means of examining features such as troughs on the output graphs would be useful. The single frequency interrogation facility can be a bit slow, especially if the feature is between two widely spaced grid lines. A tracking cross which can be positioned on the feature and a command to calculate the frequency equivalent to its position has been suggested.

10.10. Hardcopy

This is available in two forms≠, a paper tape record of the picture and

* See Figure 9U.

≠ Apart from direct photography of the screen.

a tabular description on the typewriter of the circuit and its performance. The problems with the paper tape record have been due to inadequate hardware, especially the folds in the fanfold paper tape catching in the reader.

The tabular output on the typewriter is slow and several users have said that a line printer is necessary. As programmed the user may have either a circuit description, or a circuit description and the circuit performance at selected frequencies. One user said that he would like to have just the circuit performance as a third alternative, I believe that the provision of a line printer would remove any desire for this. The frequencies for the output may be given either singly or in arithmetic increment form or a mixture, and it has been suggested that the possibility of logarithmic increments should be given, even if this gives "non rounded" values. This would be specially useful for wide frequency range devices such as amplifiers.

10.11. Typewriter Commands

The typewriter has been used for the input of single letter commands and multi-character numbers. The single letter commands used in the operating instructions (U, D and X) need no close character. Typing U, for example, causes an immediate change to the next page. This also applied to the commands used for photography of the display. Now, when the output display is being shown, there are certain commands allowed, which are S, H and X, and the one chosen must be followed by a closing character, that is a space or carriage return. This was to prevent the accidental use of X (say) leading to the loss of a display which might take 5 or more minutes to recover. It also makes the command consistent with the numerical input which needs a closing character to indicate the end of the number. If a mistake is made with one of the inputs the input may be restarted by first using the rubout key. If the machine detects a mistake it will give a ? and refuse any input until a rubout has been given. If the user is unsure what the machine is waiting for he may type a ? and the machine will reply with either a single R if it needs a real number or a string of acceptable command characters if it requires a command.

Many (most?) users have run into difficulties with this input, and an example is shown in Fig. 10D, The E SLAP is the initiation of the program and the D is an example of a single letter command moving the instructions down one page. The X takes the program into the graphical input. Presumably the user forgot to input a value for FL leading to the diagnostic but he corrected this and got a successful analysis taking 105 seconds. This circuit had 10 nodes, 22 elements and no controlled sources. He then typed H and closed it to enter the hard copy tabular output routine, which asked him to type Y or N for instructions. He typed N and closed it with a space. This routine requires a list of frequencies as input, each frequency is prefixed by an F which must be closed and is itself closed. It is permissible to input increments, which are added to the previous frequency until the following frequency is reached. They must of course occur between two frequencies and are preceded by an I and closed. The last command is E which causes the machine to give the tabular description of the circuit and its performance.

Now he typed the N signifying that he did not require instructions and went on to the first frequency which was successfully input. He then typed I for an increment and closed it with a space and then typed an

```
MONITOR V4E

$PIP

PIP V9A

>N DK1

>T DK1 (H) ←DT1 SLAP XCT

>↑C

MONITOR V4E

$E SLAP
```

TRANSFER OF PROGRAM FILE FROM TAPE TO DISC.

STARTS PROGRAM RUNNING

```
DX
```

THE USER'S TYPING IS UNDERLINED

```
ERROR DIAGNOSTICS.
LOWEST FREQUENCY TOO LOW.

PDP9 105S 10N 22E 0CS

H

TYPE Y OR N

N F 0.5 I ??
?.#.#.#
N? ??
?0?
?#0.5 F 0?#0.5
                 ?#
E?_?#E?#?RR?_?#?R_?RR? ?#
#
D?
?_?E?_?##
?R_H?_?
?##
_R?_?0?
?#0.2E1 E
```

FIGURE 10D SOME USER DIFFICULTIES.

unacceptable non-printing character, possibly an ALT MODE. The machine gave a ?, he (probably) repeated it and got another ?, he then input a carriage return and got another ?. To correct these three errors he then typed three rubouts, although one is sufficient. After proceeding to a new line he tried to retype the previous line, starting with the N. These errors indicate confusion between this input style and the style that waits for the user to type a complete line before commenting on it.

After several tries he succeeded in completing the increment at 0.5 and went on to a frequency. He typed letter O instead of zero but corrected this to 0.5 and then tried to close it with a line feed instead of a space or return. It took him some time and many errors to finish the frequency input with 0,2E1[*].

On several occasions he interrogated the machine with a ? and got R as the reply but did not understand its significance.

There have been other examples of similar difficulties with these inputs, some users had so much trouble with the necessity for closing the letter commands that the program was modified to give the message 'RETYPE AND CLOSE' when this was omitted. Many fewer comparable difficulties have occurred with the equivalent light button inputs, and it seems that the feature of displaying all the legal possibilities and no others is responsible for this.

* This long way of writing 2 indicates the "brainwashing" effect of using normal Fortran input and output.

10.12. General Observations

Apart from the version of the program described previously three variations were written. They all operated in the same manner but incorporated certain changes. One version differed in the layout of the light buttons, etc. for the input display, it is shown in Figures 10A and 10C. It was hoped to determine the influence of the screen framing lines by omitting them and to see the effect of grouping the light buttons differently. No very clear effects were noticed, probably due to lack of time for sufficient tests. It was thought that perhaps the light buttons were a little too close, accounting for more errors in this version but this might have been due to unfamiliarity.

Another version arranged that each time a user selected a light button a record was kept of the time and the button selected, this was done on magnetic tape without disturbance to the user. Later a typewritten copy was obtained from the tape, a specimen of which is given in Fig. 10E. The most obvious features of these records were that many users sometimes spent well over a minute just thinking about their next move and the very large number of operations needed just to input a simple circuit e.g. 300-400 for a two stage amplifier.

A third version arranged that the main analysis for the output graphs were done on a Univac 1108 machine accessed over a 4.8K baud link. It is clear that almost any worthwhile analysis will probably have to be done on a machine other than that supporting the graphics and this will usually need some similar set-up. The arrangement was useful even in this limited form, analyses could take from 30-120 seconds compared with 6 seconds to

```
1718.1S 9            }
1719.6S 4            } VALUE OF 94m.
172Ø.8S SMALL M      }
1735.8S DRAW
1739.1S T.C. ON  —— TRACKING CROSS ON.
1739.3S T.C. OFF
174Ø.9S OUTPUT —— INSERT OUTPUT SYMBOL.
1746.2S VALOUT
1798.4S DRAW
18Ø6.7S DELETE
1815.9S DRAW
1818.2S T.C. ON
1821.8S T.C. OFF
1823.4S RESISTOR —— INSERT RESISTOR.
1825.8S T.C. ON
1826.1S T.C. OFF
1827.7S INPUT
183Ø.5S VALIN        }
1833.6S 2            }
1834.9S 7            } VALUE OF 270.
1836.ØS Ø            }
1837.8S CLOSE        }
1843.8S DRAW
1846.1S VALOUT
1852.4S DRAW
1854.7S T.C. ON
1854.9S T.C. OFF
1856.4S OUTPUT
1859.1S VALOUT —— VIEW VALUES.
186Ø.5S VALOUT
1861.5S VALOUT
1949.4S OPTIONS
1952.8S PTOUT        ]
  24.3S OPTIONS      ] LEAVE GRAPHIC INPUT.
  41.3S GNLOSS       ]
 18Ø.5S DELETE          3 MINUTE CONTEMPLATION
 183.4S DRAW              OF CIRCUIT.
 184.8S T.C. ON
 184.9S T.C. OFF
 186.ØS INPUT
 187.1S OPTIONS
 193.6S DNLOSS
  23.3S DELETE
  86.8S DRAW
  96.1S T.C. ON
  97.6S T.C. OFF
  98.1S T.C. ON
  99.2S T.C. OFF
 1Ø1.2S T.C. ON
 1Ø1.5S T.C. OFF
 1Ø3.3S CURRENT SOURCE }
 1Ø3.9S CURRENT SOURCE } ROTATION OF ELEMENT.
 1Ø5.3S T.C. ON
 1Ø6.1S T.C. OFF
```

FIGURE 10E PART RECORD OF OPERATION OF SLAP

6 minutes or more on the local machine. Unfortunately due to defects in the software philosophy and to the choice of an unsuitable main machine this version only worked infrequently. The software defects were due to too ambitious an implementation leading to complex software and insufficient thought concerning possible faults and how to retain control by the program. The main machine executive was unsuitable for running real time programs. We found that quite often after a few successful analyses the 1108 would simply refuse to take any action for us, presumably because a very large program at high priority was being run, thus we would lose the machine for an hour or more.

10.13. Cost of SLAP

Some rather tentative estimates have been made both of the cost of using SLAP and the cost of writing it.

For the cost of running it, it has been assumed that the machine and associated hardware is to be written off over five years and that the capital cost and about six months' running expenses were borrowed at 12% interest. The machine maintenance is done under contract and it was assumed that the cost of space and heating was £12.5 per square foot, this included electricity.

CAPITAL COST	£	Per Annum Over 5 Years £
Computer	38,000	
Display	14,500	
Air Conditioning	1,000	
	£53,500	10,700
Interest at 12% on £26,750		3,210
Maintenance		2,000
Manager/Technician/Clerk. Staff of one		5,000
Space, heating etc. 20' x 20'		5,000
Papers, tapes, etc.		500
		£25,910
Add £15,000 initial float to capital		3,000
Extra interest at 12%		900
		£30,000

Therefore if the machine is used for between 2,000 and 3,000 hours per year the cost will be from £10 to £15 per hour. It will be difficult to improve on these figures since 2,000 hours represents the working year of the typical engineer who will not usually be prepared to do shift work to improve the machine utilisation.

The cost makes no allowance for the use of the large machine that will be necessary nor any allowance for rent of a telephone line and associated hardware.

Licklider[34] has estimated a cost of $2 per hour for an optimistic system using 10 terminals and a PDP 10 computer while Lunday[8] quotes costs of $50 - $200 per hour for an IBM 2250 terminal.

It took about 4 man years of programming effort to write SLAP and involved about 500 hours of machine time, this gives a cost of about £15,000 for this program.

11. PROGRAM STRUCTURE

The program is comprised of a main part which is written in Fortran and which calls a number of subroutines or subprograms. The main part or program is in fact divided into seven sections each of which calls its share of the subroutines and which also calls the other sections. Where possible the subroutines are written in Fortran, all those concerning the display had to be written in assembler language. The complete program is held on the disc and only one section is in core memory at any given time. When a new section is called it is taken from disc into core and completely overwrites the calling section. Communication of data from one section to another is through the Fortran "blank common"* which is never overwritten by program. Figure 11A shows the layout of the sections.

The program starts with section C1 which effects a permanent transfer into blank common of certain files for character generation and display layout. The operating instructions may also be displayed if required, the file transfers occur once only for any run of the program. On leaving C1 an entry to C2 is made.

C2 contains the input graphics and is shown in more detail in Figure 11B. It commences by an entry to each of five subroutines, this is to set them ready for operation. Afterwards DPINIT is re-entered which actually starts the graphic display. The display is generated from a list of items, such as light buttons and circuit elements, each of which is to be shown in turn. One item is taken from the list, or data structure, and used to cause

* A collection of Fortran arrays and variables which are common to all sections of the main program and all subroutines.

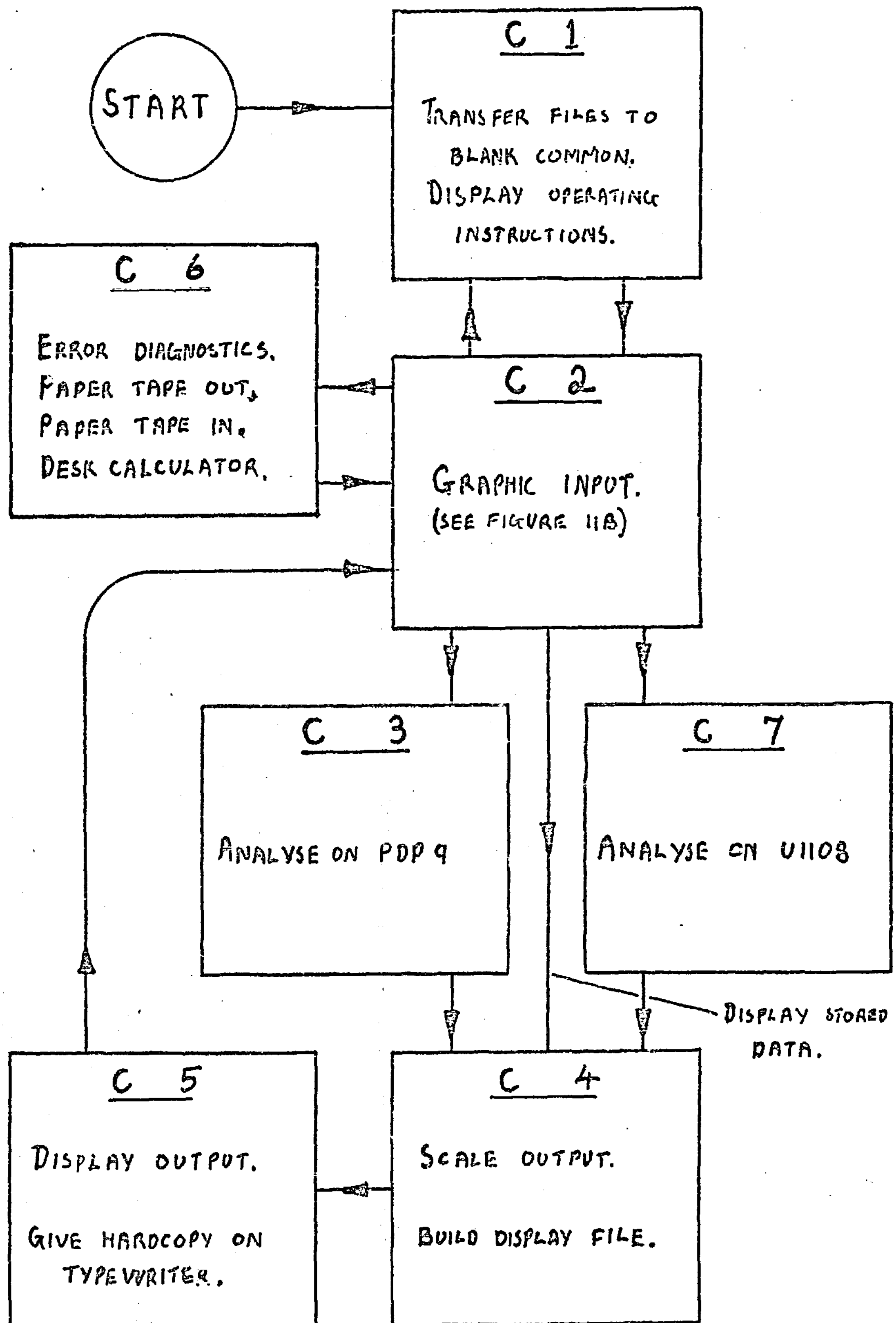


FIGURE 11A OVERLAYS USED IN SLAP.

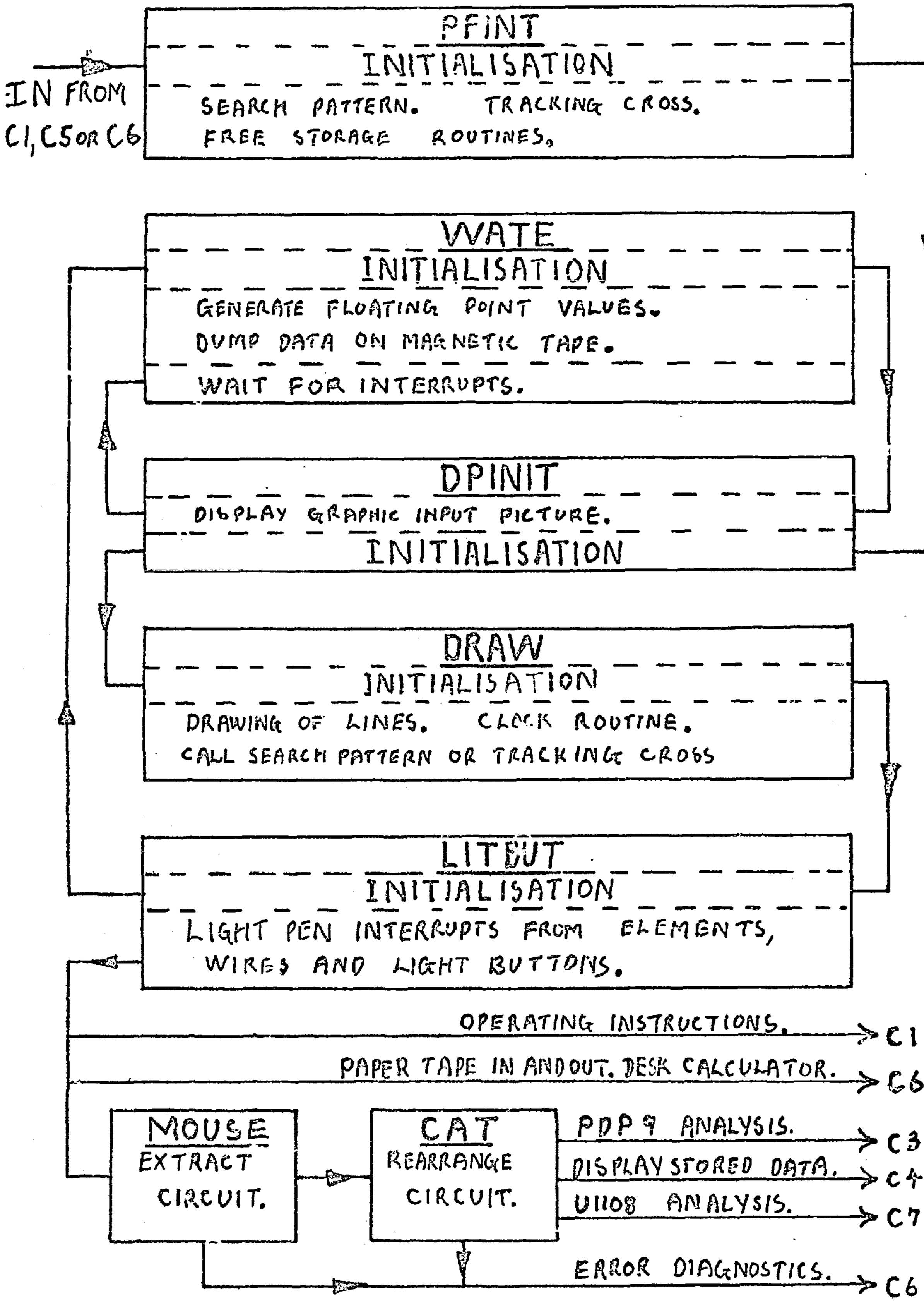


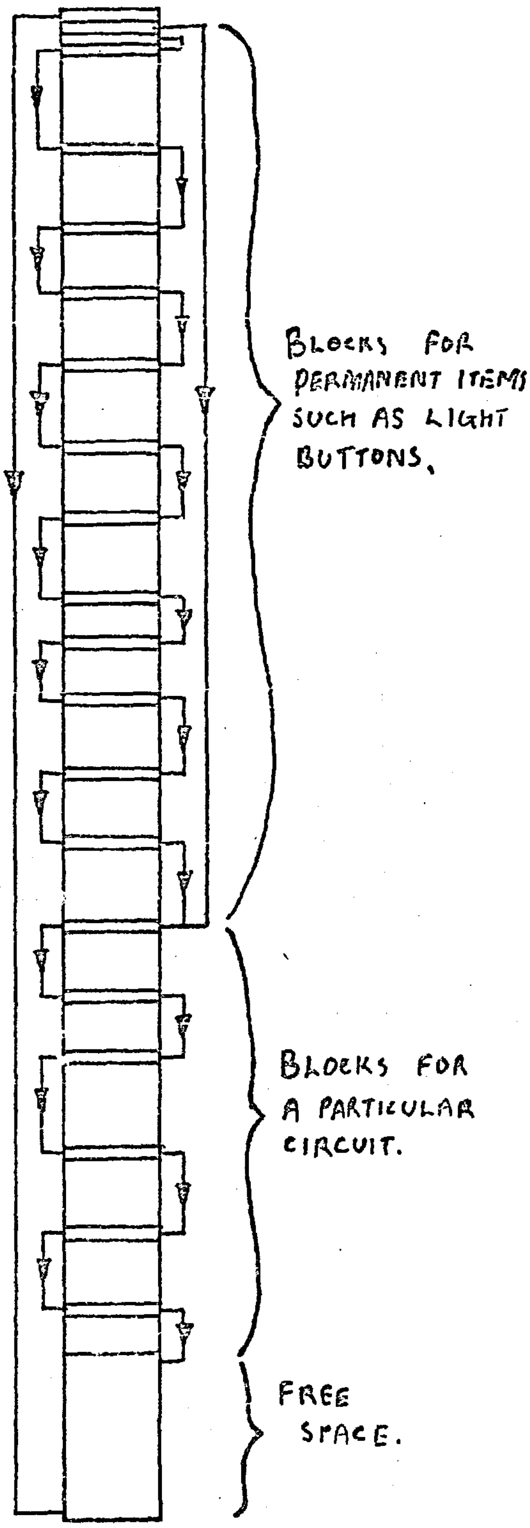
FIGURE 11B GRAPHIC INPUT – C2

the display of that item; the machine then waits in subroutine WATE for an interrupt. This interrupt will usually indicate that the display has finished and if so the next item is taken from the list and the new item is displayed. This sequence continues until the whole list has been displayed when it is restarted. Thus a complete picture is produced with a cycle time of about ten frames a second.

Interrupts from the machine clock occur every 20mS and give entry to subroutine DRAW. DRAW causes either the search pattern or the tracking cross to appear and then processes the resulting pen co-ordinates. These are used to build additions to the display list so that wire segments may be drawn. The routines for displaying the search pattern and tracking cross are held in PFINIT.

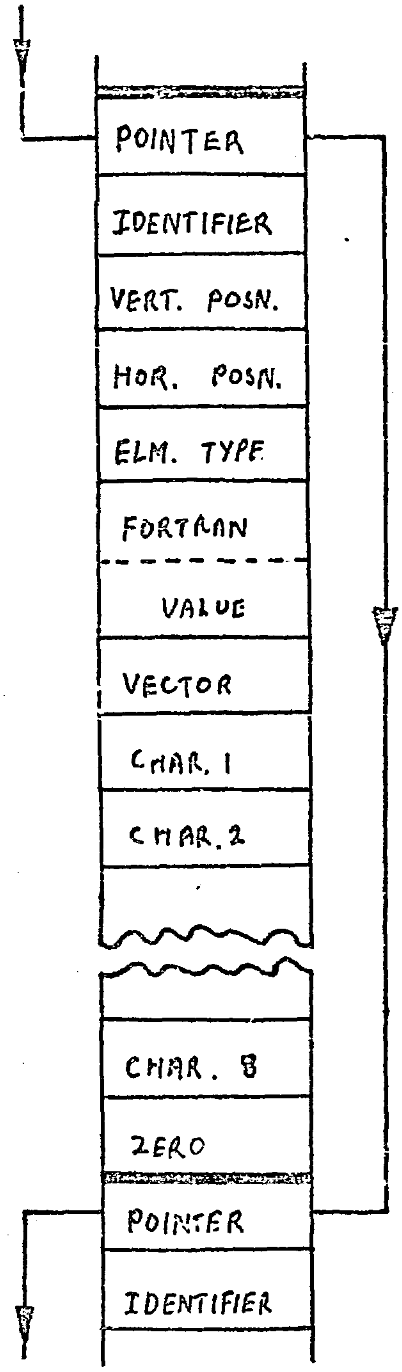
Light pen interrupts are handled by subroutine LITBUT which has program for the addition of elements and the deletion of elements and wires. Program for the association of controlled sources with their voltage detectors and for the reception of characters for element values is also present. However the processing of the characters to form a floating point number is done by WATE.

The list of items to be displayed forms a simple data structure in free storage which is arranged as shown in figure 11C. Each item has a block of storage the first word of which holds the address of the first word of the next block. The first word of the last block holds the address of the first unused word. Each block is responsible for an item on the screen, such as a straight line segment, an element, a light button, an earth symbol, etc., and is taken in turn and used to build a display



FREE STORAGE AND DATA STRUCTURE

FIGURE 11C



ELEMENT BLOCK

FIGURE 11D

for that item. The first 60 or so blocks are for the permanent items such as light buttons and frame lines while the remainder are defined in turn by the user as he draws a circuit. The data in a typical block is shown in Figure 11D. After the pointer to the next block comes the identifier which comprises a code for the item type and a serial number. What is next depends on the item but it is usually a pair of display words to define the screen position of the item, followed by display data. The example shown is an element block and the position words are followed by a code for the element type. The floating point value of the element occupies the next two words and then comes a vector which expresses the difference between the element position and the position of its value caption. Lastly there come up to eight codes for the characters of this caption.

The data structure has no pointers except those described and so knows nothing of nodes, elements, networks, etc., There is one exception to this rule and that is in the case of the controlled sources. Each source utilises three blocks, one each for the voltage detector and current source items and an association block. This last holds the display data for the association line and also pointers to the other two items. The voltage detector also has pointers to the current source and association blocks but the current source only has a pointer to the association block.

Exit from the graphics is initiated by a light button hit which stops the display and sets a flag to indicate the option required. If this was the operating instructions then C1 is entered, while if paper tape input or output or the desk calculator routine were required C6 is selected. For any other option the electrical circuit is first extracted from the data structure by subroutine MOUSE. This is accomplished by using the exact coincidence of the ends of items to indicate that they are connected. Four Fortran arrays are filled, the first one with a descriptor for the item, for example 1 for a resistor, 2 for an inductor etc. The next two contain "node" numbers derived from the actual screen co-ordinates of the item's ends and the last one its value, if it has one. All items are treated in this manner, including elements, voltage detectors, current sources and wires. Subroutine MOUSE is followed by subroutine CAT which reduces the circuit description to a more conventional arrangement involving only elements. During this process the circuit is checked for number of nodes, elements and controlled sources, for the presence of input and output symbols and for element values and frequencies having been given. If any errors are found an entry is made to C6 which gives a diagnostic on the typewriter. If there are no errors then C3 is entered for a local analysis (PDP 9) and C7 if the analysis is to be done on the remote machine (UNIVAC 1108). If a display of the stored output is required then C4 is entered.

The analysis is by pivotal condensation of the indefinite admittance matrix. Originally a straightforward method[35] was used and was found to give occasional numerical errors, especially when the elements had a large range in their values. A DIC student[36] in the Department reprogrammed the pivotal condensation according to a method of his own and this version, which is now being used, very rarely runs into any difficulties.

After the analysis the program in C4 scales the results and builds the display file for the output graphs. This file is displayed under control of subroutines in C5 which also contains routines for the single frequency interrogation of the circuit and for the tabular hard-copy output. Because C5 necessarily contains the analysis subroutine it uses very nearly all the available core.

Ultimately C2 is re-entered from C1, C5 or C6. Each change from section to section takes from 1 to 2 seconds while the data is read from the disc into the core.

The choice of the program version is made by setting the accumulator switches before loading. One switch controls which of two files is put into blank common by C1 and so gives the alternative display layout for the input graphics. Another switch sets a flag which ensures that when an analysis is requested C7 is entered rather than C3. A third switch sets another flag and as a result each significant use of the light pen is recorded, with its time of occurance, on magnetic tape. The default option for the switches is the normal display with the PDP 9 analysis.

12. FUTURE PROGRAMS

12.1. Developement of SLAP

This section gives a brief description of some of the techniques that I would employ should I set up another interactive graphical system for the computer aided design of electric networks. This applies mainly to the graphical drawing technique and should not be considered as complete.

Assuming that the input is a light pen the system must be such that the response to a typical command such as input or deletion of an element can be carried to completion within 100 mS. If this is not possible then a graphical interactive input is not likely to be satisfactory. If the response time is from 100mS up to a few seconds then a typewriter input should be considered still retaining the graphics for output.

The distance between the user's console and the machine which carries out the analysis should be short, i.e. traversable by the programmer in a few minutes, at least while the system is being developed[37, 38]. The delay, due for example to transmission and queuing time, extra to the analysis time should not exceed about one minute at the worst.

If these conditions for an interactive graphical input are met then I would adopt the following techniques in its implementation. The light pen would have a switch attached to its shutter so that a machine interrupt occurs very soon after the shutter is opened. The search pattern would not occupy the screen almost continuously as at present but would display once only each time the pen shutter was opened. If the pen were found then the tracking cross would be displayed as before, if not then neither the search pattern nor the tracking cross would be displayed.

The elements would again be chosen from a menu of light buttons at the bottom of the screen, the four most used buttons would be duplicated around the tracking cross, what they were would depend on the type of circuit most studied. More buttons would be required to cater for other types of analysis. In addition to resistors, inductors and capacitors, both linear and non-linear, voltage and current source symbols would be needed for controlled and uncontrolled sources of AC, DC and pulse signals. Voltage detector and current detector symbols and the diode and transistor symbols would also be required. The controlled sources would be used to simulate operational amplifiers and user models of semi-conductors and other special devices, while the diodes and transistors would be used to specify devices by makers' code number. Either a voltage or current detector would be assigned to control a voltage or current source thus giving any one of the four basic amplifier types, in addition an inductor could be assigned to another inductor to give an ideal transformer. The transfer parameters would be a transfer immittance, a voltage or current ratio or a turns (voltage) ratio.

Wires would be straight but not restricted to horizontal or vertical, they would be "rubber band" wires. This would make the joining procedure much more certain, otherwise the procedure would be very similar to the present program. There would be a "MOVE" facility so that when an element or the junction point of two or more straight sections of wire was pointed at by the pen it could be dragged around the screen. All the immediately associated wires would alter their lengths and directions to accommodate the movement while maintaining the circuit topology. This feature would help enormously in the drawing of a complete circuit and would also enable the user to check whether items were joined or not.

The input of numerical values and semi-conductor codes would be done by using the light pen and light buttons to input an unchecked character string to an auxiliary and visible buffer. This buffer would then be assigned to an element, or several elements and a check made to see if it were suitable, that is, made up a sensible number or was a code held in the library.

If the display area were no larger than for SLAP ($9\frac{1}{2}$" x $9\frac{1}{2}$") then I would use a windowing technique. This would differ from the usual method because it would only apply to the diagram with element captions. The elements and other circuit symbols would be reduced in size so that they were only just clearly legible and when the circuit was being drawn no windowing would be used. Due to the small symbols the screen would hold perhaps 20 times as much as for SLAP*. When some operation involved the captions or values the diagram would be magnified and viewed by windowing so that they were both legible and in proportion to the element symbols.

When an analysis was required it would first be necessary to indicate what type, DC, AC or TR, then the user would be asked for the extra data necessary, e.g. the values of independent sources, frequency ranges, time steps and range. The circuit would then be checked for errors and if any were found a graphical caption would be given.

* Elements are 96 units long for SLAP, this would be reduced to 20.

When the analysis was completed a circuit description and a set of results would be output on a line printer and/or magnetic tape unless the user specifically said not. He would have the option of terminating the analysis before completion, any partial results being displayed. A complete set of node voltages and element currents would be available after the analysis and the user would be able to change his output display. All these options would be chosen by means of light buttons as these compare favourably with the typewriter for speed of use and are likely to be more reliable than that mechanical device.

12.2. Developement of Graphical Interactive CAD

The facilities provided by SLAP represent only a beginning in the application of interactive graphics to circuit design. A comprehensive program of the future may be expected to provide aid in all the steps of a design, from the initial conception to the final manufacture and sale of the network. It will also provide a number of ancillary facilities, not used in the direct design sequence but necessary nevertheless.

It will consist of a number of sub-programs, independent of each other except that they will share a common bank of data. This data, which will describe the circuit in all particulars, will be provided initially by the interactive graphical input program or by a data file or tape. It will be used as input by the other programs such as analyses, optimisation, layout etc. The output from these programs, in addition to being displayed, will, when relevant, be added to the common bank of data, thus allowing the results of one evaluation to affect the input data for the others.

For such a facility some sort of monitor or control program will be required, the purpose of which will be to discover what it is that the user wishes to do and then to call the appropriate program having first made certain that all the required input data has been provided. When the user has finished with that particular program the control will take over again and shepherd him into the next stage of his design. The control program will also select which of several possibilities is used, for example if the user wishes to make a linear AC analysis then it might be carried out by either a symbolic type of program or by numerical inversion of the indefinite addmittance matrix or by a method only suited to passive ladder networks. Similar possibilities exist for optimisation and transient analysis proceedures. If an analysis lands in difficulties, such as lack of convergance of an iterative process, then the control program would try another method or ask the user for advice.

Among the ancillary features provided might be the interactive design of inductors and transformers and facilities similar to those at present provided by the interactive on-line conversational programing languages. The latter however would use the graphic screen for the output of graphs and text and would also enable the user to edit his program by means of a graphical editor.

Of course it will be seen that all these ideas are only developments of the existing techniques of circuit design but the interactive use of a large computer on-line might permit an altogether now solution to the design problem. If for example a really good optimisation program became available the difficulties with the input of a circuit might be replaced by difficulties with the graphical output of a recognisable circuit.

In any case what ever the system is the program will need to be slick and "easy to get on with".

13. CONCLUSIONS

There are three conclusions that I draw from this work on graphics. They relate first to the use of graphics in interactive design using computers, secondly to the more general use of computer graphics, and lastly to the conditions for the more extensive use of graphics. Briefly they may be summarised as the need forinteractive input graphics to be "slick", the fact that all human input to computers should be by means of graphics, and the need for a high level graphics language.

13.1. Graphics for Interactive Design.

This covers the input of pictures as part of the design procedure for such purposes as the design of electric networks, ships, buildings, structures, vehicles and so on. The important point here is that the designer's mind must be concentrated on the thing that he is designing if he is to do his job satisfactorily. He can only afford to give the smallest part of his attention to the operation of the graphics. The program is a tool, not an end on itself, it must "respond in a friendly, courteous, forgiving manner"[40] so that the user does not need to worry all the time about the possibility of making a catastrophic mistake. He must be able to see at once the effect of his actions and be able easily to reverse them. Only in this way will he be able to give his (almost) undivided attention to his design problem.

13.2. General Computer Input by Humans.

Whenever the input data has to be prepared by a human then interactive graphics should be used[39]. At the moment the data is usually transferred from a paper page to either punched cards or paper tape by an operator sitting at a punching machine. The procedure is basically typing and no help is given in checking, sometimes indeed it is not even possible to discover what the last character typed was, without typing more. An interactive graphical program would allow the input to be displayed and checked (as far as possible) and, if a mistake were detected by the machine or the operator, it could be edited out without the necessity of retyping at least some of the correct input, as is at present the case. Of course an experienced operator would not make many mistakes and when one was made just the indication of a mistake would be sufficient. We have here the familiar dilemma that a beginner needs a very full error diagnostic while the experienced user needs none. Many users will be experts in some respects but beginners in others. The difficulty can easily be resolved by the graphic display which is capable of presenting a comparatively large body of tutorial or diagnositc text, quickly and quietly. I visualise an interactive display with light pen and light buttons for all characters and commands. These characters and commands would be duplicated on the typewriter so that the experienced operator could type in the data rapidly without much reference to the display, while the inexperienced would probably use the light pen and display without much reference to the typewriter.

13.3. Graphic Language.

It seems very clear that before the use of interactive graphics can become widespread there must be a standard (or a set of standard) high level languages. At the moment a large proportion of the effort involved in writing a graphic program is lost when a transfer to another newer or larger machine is required, because the language used is often special to the machine in which the graphics was developed. Sometimes the solution has been attempted by producing a set of subroutines callable from an existing high level language like Fortran. This suffers from the difficulty that such languages, though readily available, are not really suitable for the control of graphics. They represent a single sequence of operations and only one thing is being done at a time. With graphics, however, apparently nothing is being done for long periods, and then some sequence will occur, initiated from the outside world and quite impossible for the computer program to predict. A high level graphic language is required which is suited to these features and which is compatible, for example, with Fortran. It needs to be able to define objects both at load time and run time, and build items from them which can be put in groups as part of a data structure. Then operations can be carried out on these groups by sections of program initiated by interrupts from outside. For example, delete a group of items, or flash it, or display a group of light buttons, and if one of them is selected go to a particular portion of the program. An important feature of such a language would be the WAIT instruction, meaning continue to display what has been set up for display and wait for something to happen.

14. ACKNOWLEDGEMENTS

I am pleased to acknowledge the financial support received from Ferranti Limited, The Plessey Company, the U.K. Science Research Council and especially the Department of Electrical Engineering, Imperial College. Also the supply of the computer to Professor W.S. Elliott by the Ministry of Technology and the U.K. Science Research Council.

I should like to thank Professor Elliott for his aid with the computer and E.N. Jacob and Miss J. Iremonger for help with some of the programming.

I particularly wish to thank Dr. Robert Spence, who supervised the project, for his continuous encouragement and assistance.

15. REFERENCES

1. TEMES, G.C.: "Filter Design in Transformed Frequency Variable", (See 3).

2. TEMES, G.C.: "Optimization Methods in Circuit Design", (See 3).

3. KUO, F.F. & MAGNUSON, W.G.: Editors of "Computer Orientated Circuit Design", Englewood Cliffs, N.J., Prentice-Hall, Inc., 1969.

4. "1620 Electronic Circuit Analysis Program (ECAP)(1620-EE-02X) User's Manual", I.B.M. Corporation, Data Processing Div, White Plains, N.Y., I.B.M. Application Program File, H20-0170-1, 1965.

5. MALMBERG, A.F., CORNWELL, F.L. & HOFER, F.N.: "NET-1 Network Analysis Program", Los Alamos Scientific Laboratory, Los Alamos, N. Mexico, Report LA-3119, 7090/94 version, August, 1964.

6. FASS, V.H.: "Conversational Computing", Proceedings of the Conference on CACD, March, 1968, Sheffield, Design Electronics, Sheffield University and Kingston College of Technology.

7. DERTOUZOS, M.L.: "CIRCAL : On-Line Circuit Design", Proc. I.E.E.E., Vol. 55, No.5., May, 1967.

8. LUNDAY, P.A.: "The Lessons of the '60's'", (see 30).

9. DAY, C.W., HOGSETT, G.R., NISEWANGER, D.A. & MALONE, W.H.: "S360 General Program Library, 2250 ECAP, April, 1967, 16.4.002", I.B.M. Corporation, 3777 Long Beach, California.

10. DAVIES, M.R. & ELLIS, T.O.: "The Rand Tablet; A Man-Machine Graphical Communication Device", A.F.I.P. Proc. 1963, Fall Joint Computer Conf.

11. TEIXEIRA, J.F. & SALLEN, R.P.: "The Sylvania Data Tablet; A New Approach to Graphic Data Input", A.F.I.P. Proc. 1968, Spring Joint Computer Conference.

12. MACHOVER, C.: "Computer Graphics in the United States", (See 28).

13. MICHAEL, G.A. & CRALLE, R.K.: "A Survey of Graphic Data Processing Equipment for Computers", (See 3).

14. WISEMAN, N.E.: "Man-Machine Communication", Proceedings of the Conference on CACD, March 1968, Sheffield, Design Electronics, Sheffield University and Kingston College of Technology.

15. IRANI, K.B., WALLACE, V.L. & JACKSON, J.H.: "Conversational Design of Stochastic Service Systems From a Graphical Terminal", (See 30).

16. NEWMAN, W.M.: "A Graphical Technique for Numerical Input", Computer Technology Group Report, Centre for Computing and Automation, Imperial College, 1967.

17. Seen at CISE, Segrate, Milan, Italy.

18. SPENCE, R.: Private Communication.

19. BENNETT, E., HAINES, E.C. & SUMMERS, J.K.: "A.E.S.O.P. : A Prototype For On-Line User Control of Organizational Data Storage, Retrieval and Processing", A.F.I.P. Proceedings, 1965, Fall Joint Computer Conference.

20. NICKERSON, R.S.: "Man-Computer Interaction : A Challenge for Human Factors Research", Trans. I.E.E.E. on Man-Machine Systems, MMS-10, December, 1969.

21. CHUA, L.O.: "Computer Aided Analysis of Non-Linear Networks", (See 3).

22. CARPENTER, R.M. & HAPP, W.W.: "Symbolic Approach to Computer-Aided Circuit Design", Electronics, Vol. 39, No.5., Dec. 1966.

23. GODDARD, P.J. & SPENCE, R.: "Efficient Method For the Calculation of First and Second-Order Network Sensitivities", Electronics Letters, Vol. 5., No.16, 7th. Aug., 1969.

24. MAGNUSON, W.G., KUO, F.F. & WALSH, W.J.: "Graphic Input for Network Analysis", (See 3).

25. Described in Chapter 9.

26. MURRAY-YASSO, M.A.: "On-Line Circuit Design on Commercial Time-

Shared Systems", (See 27).

27. HERSKOWITZ , G.J.: Editor of "Computer-Aided Integrated Circuit Design", McGraw-Hill Book Company, New York, 1968.

28. PARSLOW, R.D., PROWSE, R.W. & ELLIOT GREEN, R.: Editors of "Computer Graphics, Techniques and Applications", Plenum Press, London and New York, 1969.

29. MACLEAN, M.A. & BOWN, H.G.: "Interactive Graphical Input for Circuit Analysis", (See 30).

30. "International Symposium on Computer Graphics '70", Brunel University, Department of Computer Science, Uxbridge, Middlesex, England, April, 1970.

31. ELLIOTT, W.S., JENKINS, A.P. & JONES, C.B.: "An Interactive Graphical System Using Computers Linked by Voice Grade Line", (See 30).

32. Private Communication from S. Kassum.

33. LAFUENTE, J.M.: "Interactive Graphics in Data Processing-Cam Design on a Graphics Consol", I.B.M. System Journal Volume 7, No.3 & 4, 1968.

34. LICKLIDER, J.C.R.: "A Picture is Worth a Thousand Words and Costs....", A.F.I.P. Proceedings, Spring Joint Computer Conference, 1969.

35. ZADEH, L.A.: "Multipole Analysis of Active Networks", I.R.E. Transactions of the Professional Group on Circuit Theory, Volume CT-4, No.3., Sept. 1957.

36. JACOB, E.N.; Imperial College, 1971.

37. GEAR, C.W.: Answering questions on "Graphical Computer Aided Programming Systems", (See 28).

38. HERZOG, B.: Commenting on 37.

39. HARRIS, D.G.: "Tesco and the Visual Display Unit", (See 30).

40. JACKS, E.L.: "Observations on a Graphic Console System", from "Emerging Concepts in Computer Graphics", Editors, D. Secrest and J. Nievergelt, W.A. Benjamin, Inc., New York, 1968.

APPENDIX

COMMENTS ON THE SLAP DATA STRUCTURE

There are several programming features which distinguish the graphic input of SLAP from those of similar programs. The most important of these arise from the extreme simplicity of the data structure. Most interactive graphic programs have a data structure which has many pointers from one item to another. These pointers tie the items into lists or rings so that, for example, all the elements, or all the nodes, or all the items forming or terminating on a node, are on the same ring*. Thus when, for example, the user points to a circuit wire it is possible to determine from the data structure that he is referring to either that wire, or to a node formed from that wire and other items, or to one of several circuits, etc.

These more elaborate structures store the diagram as a circuit and are modified as the drawing proceeds; for example, node numbers may be allocated when the node is drawn. This can of course lengthen the response time. If the data structure is large it will be necessary to divide it into sections which are kept on disc or drum, there only being room in the core memory for that section relevant to the operation being performed. The display of the permanent screen items, such as light buttons and comments, will be quite separate and independent of the data structure.

The SLAP data structure knows nothing of nodes or the inter-connections of the circuit, it merely describes the diagram as an assembly of apparently independent items. When, for example, the user points at a wire only that wire is indicated; if it were desired to indicate the node then a search of the data structure would have to be made to identify the components (i.e.

* Reference: WISEMAN, N.E. & HILES, J.O.: "A Ring Structure Processor for a Small Computer", Computer Journal, Vol.10, No.4., Feb, 1968.

attached wires etc.) of the node. This again would take time and lead to a slower response.

It seems generally accepted that a fairly complex data structure is necessary and there are some procedures which use the data structure in the analysis. However, it does not follow that the data structure should be built while the circuit is being drawn. The user will alternate between regarding the diagram as a circuit and as a collection of items, depending on what he finds more convenient. It seems illogical to try and make electrical sense (i.e. build a data structure) out of a partly drawn circuit, this can lead to difficulties, especially when items are deleted. It is more reasonable to have a minimal data structure while the drawing is in progress and only try to make electrical sense of the diagram when it is essential, as when an analysis is requested. When this occurs there is usually more time available.

The method of connecting items by arranging that their ends are coincident, as used in SLAP, may very well have advantages since it agrees so well with the instinctive and intuitive approach of most users. The node numbers are derived from the screen co-ordinates of the ends of the items, the most significant half of the number being formed by the X co-ordinate and the Y co-ordinate forming the least significant half. Items which have coincident ends will share a node number. The numbers are of course rather odd and are also assigned to segments of wire as if they were elements. It is the purpose of subroutine CAT to remove these unwanted items and to re-number the nodes, while retaining the circuit topology.