NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

Canada

An Analysis/Simulation Program For Power Electronic Circuits

Donato Vincenti

A Thesis

in

The Department

of

Electrical Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Electrical Engineering
Concordia University
Montréal, Québec, Canada

October 1987

ISBN 0-315-41646-7

ABSTRACT

An Analysis/Simulation Program For
Power Electronic Circuits

Donato Vincenti

The central role of power electronics in the development of fully automated production machinery has prompted an intensive research effort directed towards the improvement of the associated converter circuits. Part of this activity has been focused on the development of efficient programs for computer-aided circuit analysis and design. A number of such programs exist for both mainframes and personal computers (PCs) and all perform well with varying degrees of versatility and flexibility. Despite their adaptability, however, these general circuit simulators are not designed to work fast with power electronic circuits - especially when these packages are run on PC's. Moredver, there is a need for circuit simulation and analysis programs that can run faster and be more effective because they are geared towards a specific family of circuits and components. This strategy has been employed in writing the analysis and simulation program for power electronic circuits (ASPPEC) presented in this thesis. It is focused on applications in power electronics equipment circuits which, due to their nonlinearities and switching patterns, tend to be difficult to analyze with existing

circuit simulation programs. The algorithm developed in this thesis, is used to design a program targeted for use with desk-top computers. It features extensive graphics, low memory requirements, fast execution times and above all - user friendliness. This thesis will show how ASPPEC performs schematic capture to analyze power circuits on PCs. Finally, the program is used to produce simulated and experimental results of actual converter circuits in order to establish its effectiveness.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

1.1 General Introduction to Static Power Amplifiers

Full automation in manufacturing processes has become the leading edge of industrial renewal. Full automation requires the successful functional integration of three basic system components: the controller, the power amplifier with its respective driver, and the production machinery. Of these three components, the controller and the production machinery have received an enormous amount of attention. There are 2 good examples of controller applications: the research into the development of special purpose digital computers and associated software which function as industrial controllers, and the research into the development of intelligent features such as vision and touch for specialized production machinery (robots). Despite spectacular successes in the aforementioned two areas, the fact remains that, unless, equally sophisticated and reliable power amplifiers are developed, then the overall production system cannot reach its full potential. A major step in this direction has been the development of effective analysis and simulation programs for power electronic circuits such as PSPICE [1] or SPICE [2], IS-SPICE [3], MICRO-CAP [4], ECA-2 [5], VANGUARD [6], ACNAP3 and DCNAP2 [7], ATOSEC5 [8], COSMIR [9], and SWEAP [10].

1.2 Related Simulation Packages

As with many other research areas, digital simulation of power amplifier circuits can significantly facilitate the process of product

design and development. The various circuit simulation packages available today can be classified into 2 categories of electronic simulators:

1) Simulators which use complex modeling.

2) Simulators which use simple modeling or ideal models.

The circuit simulators of the first category, such as PSPICE [1], SPICE [2], IS-SPICE [3], MICRO-CAP [4], ECA-2 [5], VANGUARD [6], ACNAP3, and DCNAP2 [7] use complex models for their semiconductors (transistors). Therefore analysis of circuits with several transistors (power electronic circuits) use large amounts of memory, require excessive run times, and need process information that can be supplied only by the switch manufacturers. All of these PC software packages either use SPICE for their circuit simulation or they are imitations of SPICE. For instance, VANGUARD [6], the computer-aided engineering system from CASE technology is a $82,105.00 package that has a schematic graphic circuit editor producing a PCB layout, timing verification, logic simulation, and has a SPICE circuit simulator integrated in the package. Another circuit simulator, PSPICE, which was one of the first circuit simulators on the market (January 1984), was developed by MicroSim [1]. PSPICE is a member of the SPICE family of circuit simulators. This program comes from the SPICE [2] family of circuit simulators developed at the University of California at Berkeley during the early 1970's. PSPICE uses the same algorithm as SPICE and is considered more powerful than IS-SPICE, MICRO-CAP, ACNAP3, DCNAP2, and ECA-2 [5]. For example,

MICRO-CAP has a self-contained schematic editor whereas PSPICE must interface with PC-based schematic editors (FutureNet, Omation, OrCAD, P-CAD, or Viewlogic) which are somewhat cumbersome since these editors are slow and use unnecessary components which PSPICE does not recognize. Reference [5] states that ECA-2 performs essentially the same functions as SPICE (PSPICE) or less. Reference [3] states that two or more independent sinusoidal sources in IS-SPICE cannot be phase shifted with respect to one another except for small signal AC sources. References [3] and [5] are examples of competitive circuit simulators that use PSPICE (SPICE). By comparison, references [3] [6] [7] and [11] have a similar PSPICE input file syntax. Furthermore, special software (PRE-SPICE) is used in IS-SPICE [3] to prepare circuit analysis files for either IS-SPICE or mainframe SPICE. Therefore, from references [2] [3] [4] [5] [6] [7] and [11] one can conclude that PSPICE is the PC industry standard of the first category of circuit simulator. For many years SPICE has been successfully employed to predict the transient and steady state behaviour of various power amplifier circuits on a mainframe or super minicomputer, such as the VAX11/780. However, one can agree with reference [11] and argue that jobs on a PC AT, for example, typically run about 1/3 as fast as those run on a VAX11/780 minicomputer. Although this run-time is insignificant compared with the time required to define a problem, it is an advantage to designers when results can be viewed immediately on a PC display rather than after the lengthy delays incurred by remote facilities. System parameters usually presented by approximate values and component tolerance limits, for example, call for numerous runs. Therefore, the time spent waiting for a

batch run and the output from a multi-user mainframe often offset the advantage of higher speed. With the arrival of the 80386 MPU systems and the numeric coprocessor IC for floating-point calculations - mainframes use floating-point software - it is possible to run iterative software faster on the PC than on the mainframe. This has many advantages from the designers point of view. Since run times on PCs are somewhat longer than corresponding simulation algorithms applied on mainframes, many algorithms using the second category of simulators such as ATOSEC5 [8], COSMIR [9], and SWEAP [10] are more suitable for PCs. These programs fall under the second category in which the nonlinearities of switches are modelled by an ideal open-circuit in the non-conduction state and an ideal short-circuit in the conduction state (COSMIR and SWEAP), and a large inductance and a small inductance (1μH for a bipolar transistor) respectively in ATOSEC5. The problem with ATOSEC5's algorithm is that it requires that each semiconductor switch present in the system be paralleled with an RC circuit (snubber). This will increase the order of the system by 2 for every semiconductor switch, implying that a solution using ATOSEC5 will be slow when incorporated on a PC. Moreover, ATOSEC5 requires a lengthy input file which does not make this type of program user friendly. The other two programs, COSMIR and SWEAP, have also been developed specifically for power electronic circuits and applications. In the COSMIR program, semiconductors are represented by piecewise-linear switches where each mode of the converter is represented by a linear circuit and is described by a corresponding set of linear state-space equations [9]. A similar algorithm is used by SWEAP [10] which puts constraints on the number of states, inputs, and

topological modes the program is capable of accommodating.

1.3 Proposed Simulation Package

The circuit simulation program ASPPEC (analysis/simulation program for power electronics equipment) presented in this thesis has also been written for power electronic circuits. Its main features are; that it is very user friendly (e.g. one can input a circuit by drawing on the CRT), it includes extensive graphics features, and it is compatible with the memory capabilities of desk-top computers. However, at the present state of development, it cannot accommodate feedback loops. Nevertheless, it is envisaged that the subject simulation program can assist in the development of power amplifiers and of power electronic circuits in general, in four ways:

i) It can be used to provide performance insights before the actual circuit is built. Because of its short execution time and extensive graphics, the program can be used repeatedly to predict transient and steady state current and voltage waveforms through and across various circuit components. These features are especially useful during the "getting acquainted with the circuit" phase of research and development.

ii) It can be used to provide relevant circuit design data such as rms, peak, and average voltage and current ratings for various active and passive circuit components under the worst operating conditions. In this case, the effects, on the

overall system performance of changing certain component values can be evaluated quickly and inexpensively.

iii) It can be conveniently employed to perform nondestructive fault circuit testing by simulating various system faults such as component failures and load short circuits.

iv). Finally, it can be also used as a self-teaching, self learning tool by nonexpert members of the circuit development team.

2.0 PROGRAM STRUCTURE

A description of the hardware support and of the internal structure of the program is provided in this chapter. The program itself is designed to introduce the reader in a systematic way to the hardware and software required and to allow him to produce a similar program on a different computer such as an IBM-PC. Appendices containing the program can be referenced to the algorithm presented in this thesis to aid in the understanding of the methods used here and their application to any other high-level programming language. Moreover, the program is not as important as is the usage of the algorithm. Although this thesis was implemented on a Hewlett Packard 9836 series desk top computer, it is important to stress that the program presented here is not restricted to this machine. Thus, specific hardware requirements such as the knob-control on the HP 9836 can be replaced by a comparable hardware item such as a PC-mouse, which functions in a similar fashion.

2.1 General Description of the Program Structure

The program actually consists of two subprograms. The first subprogram is a user-friendly interactive circuit-drawing package which makes use of the computer's extensive graphic capabilities. A listing of the circuit drawing program is given in Appendix 1. The second program is the tree-link program which performs circuit analysis based on linear network theory [12]. After having drawn the circuit in the computer, the analysis section recalls the coordinates of circuit elements from a file

created by the circuit drawing program. Subsequently, it performs the necessary conversions to be used in the tree algorithm and finally produces output voltage/current waveforms following the formulated results of chapters 3 and 4. A listing of this tree-link program is given in Appendix 2. Throughout this thesis both of these subprograms are referred to as "the program" or ASPPEE unless indicated otherwise.

The program is written in 3.1 Hewlett Packard BASIC for the HP 9836 series desk-top computer. In HP BASIC there exists a family of special instructions that facilitates arithmetic operations on numeric arrays. This family of special instructions is used extensively in the tree-link program, hence, familiarity with this special instruction group will aid in introducing the reader to the program logic. In most high level languages this special instruction group does not exist, and therefore would be replaced by subroutines to perform the task. The special instruction is the MAT command. It can quickly initialize numeric arrays to constant values or copy numeric arrays. It also performs arithmetic operations on numeric arrays. Finally, through the use of secondary keywords, it can be used to manipulate numeric data stored in arrays. For example, matrix multiplication is implemented by the following instruction;

```
MAT C = A * B
```

The asterisk is the matrix multiplication operator. In this case it is multiplying matrix B by matrix A and storing the solution in C. The asterisk can also be used in scalar multiplication as follows;

MAT C = k * B

where matrix B is multiplied by a scalar number k and the solution is stored in C. The calculation is done in REAL math, unless both operands are integers, in which case the computation is also INTEGER. With the use of secondary keywords in conjunction with the MAT instruction, the following functions can be executed. There are 3 secondary keywords:

i) IDN is a secondary keyword which turns a square matrix into an identity matrix.

ii) INV is a secondary keyword which finds the inverse of a square matrix. The inverse is found using the pivot-point method. If the determinant of the inverse matrix is very small compared with the elements in the argument matrix then the inverse may be invalid. A zero determinant would indicate an invalid inverse matrix.

iii) TRN is a secondary keyword which produces the transpose of a matrix. The transpose is produced by exchanging rows for columns and columns for rows. Wrong shaped result matrix will be redimensioned by the computer.

Examples of the use of these functions are found within the tree-link program (Appendix 2).

2.2 Subsections of the Program

In this section a description of hardware support and program structure is provided with the help of the flowchart shown in Figure 2.1. The START block begins the flowchart. Here the general hardware support description is introduced. The next section of the flowchart is the INPUT CIRCUIT block. In this stage, located in the circuit drawing program (Appendix 1), interactive graphics are used to input circuit elements. The program passes directly to the STORE ON FLOPPY DISK section. This block is useful when repeated circuit analyses are desired. Here, the circuit description can be recovered or stored on the mass storage medium - the floppy disk. Both the INPUT CIRCUIT and STORE ON FLOPPY DISK steps are found in the circuit drawing program (Appendix 1), incorporating the many "user friendly" features mentioned in the next subsections. In the ENTER VALUES block, circuit element parameters are entered followed by the search for previously stored data files in the PWM TREE section. If the search is positive the program goes to the CAPACITOR LOOP INDUCTOR CUTSET stage, otherwise the program switches to the PWM SCHEME block where either a standard or a custom input scheme can be selected. The LINEAR GRAPH SETUP section produces the tree file using linear graph theory [12]. Both the tree file and the PWM scheme file are stored when the program is in the STORE PWM SCHEME and the CIRCUIT TREE ON FLOPPY blocks. The search for capacitor loops or inductor cut-sets is performed in the CAPACITOR LOOP INDUCTOR CUTSET stage. If this block finds none of the degeneracies shown in chapter 3, the program allows the user to select the output waveforms in the OUTPUT SELECTION step. The next three sections are discussed in chapter 3 and

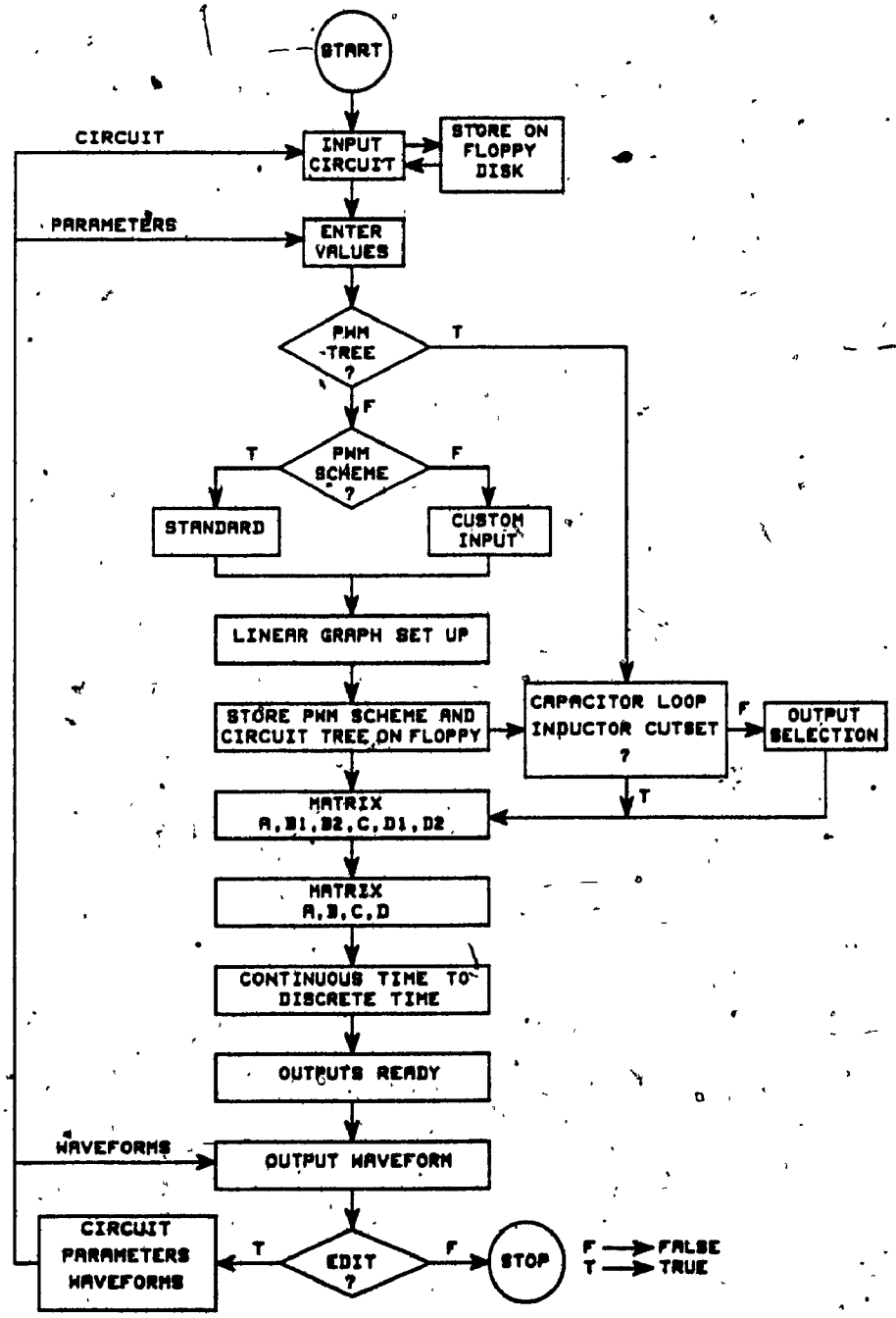


Figure 2.1. Flowchart of the program.

chapter 4 and use the State Space approach - references [12][13][14]. After the output waveforms are solved and drawn by the OUTPUTS READY block the program goes to the OUTPUT WAVEFORM step where the user chooses the destination of the waveforms. Finally, the EDIT block simplifies the task of editing either the circuit, the circuit parameters or the presentation of the waveforms.

2.2.1 Hardware Support

The program is implemented on the Hewlett Packard 9836 series desk-top computer with a Motorola 68000 based micro-processer. Some of its hardware features include a single 1.0 Mbyte memory board, an HP-IB bus interfaced with a HP plotter, rotary knob control with extended graphics support, two disk drives, a RS-232C standard interface, and a high resolution graphic screen. The computer's graphic hardware features initiated the task of developing the program's graphic orientation. The visual interaction of the user with the system makes the inputting and outputting of data illustrative.

2.2.2 Inputting the Circuit

The circuit under simulation is entered and displayed on the CRT (as shown in Figure 2.2) by using the rotary knob and the pre-defined function keys (i.e. soft-keys). These pre-defined function keys include a wide selection of circuit elements such as dc/ac voltage and current sources, resistors, capacitors, inductors, switches, single and three phase converter structures, and connecting lines. Additional graphical features designed to enhance circuit illustration include arrows, loops,

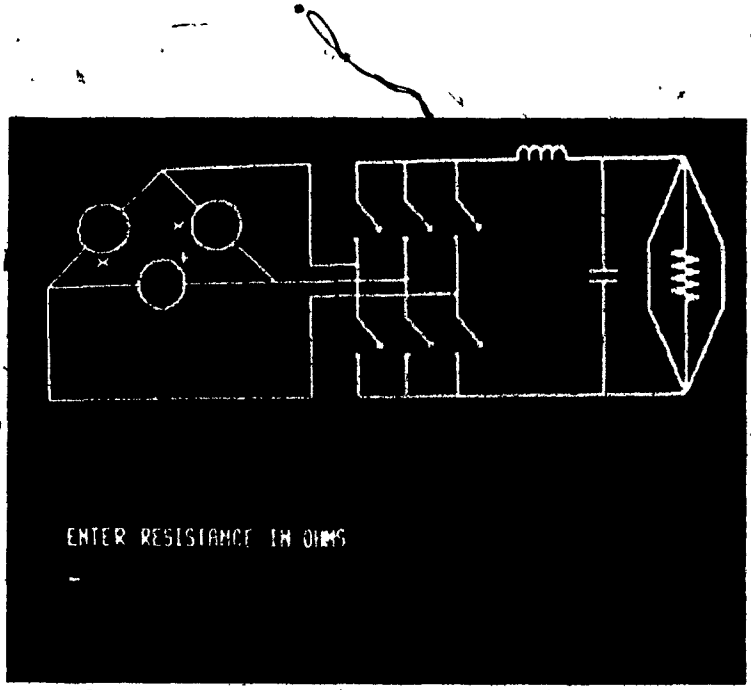


Figure 2.2. Picture of the CRT display of a 3-phase rectifier circuit during the entering value period.

circles, and squares. Circuit manipulation in graphic mode also allows extensive labelling and editing such as the deletion or addition of circuit elements, character deletion, element sizing, and repositioning of the entire circuit. Hard copies can be obtained with the use of an HP plotter. Moreover, all the circuits illustrated in this thesis have been obtained with the use of this program.

2.2.3 Storage on the Floppy Disk

The visual representation of the circuit on the CRT can be stored or retrieved from a data file on the floppy disk. The file name can consist of up to 9 alphanumeric characters. An underscore character is appended to the data file name allowing the program to immediately identify the file as a circuit diagram data file. Circuit element positions are stored using integer coordinates and element codes. This information is packed in integer form to economize disk space and is repacked upon deletions of circuit elements. To avoid duplication of file names, a directory is displayed when prompted to input the file name.

2.2.4 Entering the Parameters

The circuit element values are entered interactively. For resistors, capacitors, and inductors, values are in Ohms, Henrys, and Farads respectively. The values of the independent voltage sources and current sources for any schematic representation are entered in volts or amperes respectively. These source inputs can also be a data file containing the desired waveforms. This procedure involves the following: using a data acquisition device such as a digital scope interfaced to the computer, actual source waveforms can be retrieved from the digital scope and stored on a floppy disk. At the source entry prompt, the file name of this waveform is retained in memory as being one source waveform. Preprogrammed source waveforms are similarly entered. Entry of the base frequency, the waveform resolution (number of points), and the number of cycles are then prompted. Upon completing the above element

entries, if a switch configuration exists, the entry for the switching scheme (PWM scheme) will be prompted.

2.2.5 Previously Stored Data Files

To speedup repeated runs, the program searches the data floppy disk for previously defined inputs such as circuit PWM and tree files. If and when such data is found, the program proceeds with the investigation of capacitor loops and/or inductor cut-sets. This situation also occurs when only component values are changed while the respective circuit topology and control scheme remain the same. When the resolution (number of points) selected by the user is high or the circuit is large, the usefulness of this storage feature becomes apparent. In conjunction with ram storage, run-times can be sharply reduced with this storage feature.

2.2.6 The PWM Scheme

At this point the program prompts the user to define the required PWM (pulse width modulation) voltage scheme from an existing menu. If, however, the user wants to employ a customized PWM scheme this section can be bypassed. Some standard PWM schemes, such as the square wave scheme, original sine PWM scheme, modified sine PWM scheme and the third harmonic injection PWM scheme [15], are incorporated in this section. With 4 popular carrier frequencies the 9, 15, 21, and 27 triangles are selected based on the criteria for unwanted harmonics. Also, a .1 to 1 modulation index can be selected.

2.2.7 Linear Graph Setup

From the circuit diagram drawn on the CRT, the program constructs the respective circuit tree by using established principles of linear graph and circuit analysis theory [12]. Accordingly, all voltage sources and as many capacitors as possible become twigs, while current sources and as many inductors as possible become links. Resistors can become either links or twigs. This method is elaborated in chapter 3.

2.2.8 Storage of the PWM Scheme and Circuit Tree on Floppy Disk

The PWM control scheme and respective circuit tree obtained earlier are stored in binary files on a floppy disk. The PWM control scheme data file consists of real number data that describe the switch configuration scheme. A single cycle is developed by ASPPEE for the switch configuration scheme. To reduce run-time, the cycle is repeated up to the number of cycles selected by the user. In a 3-phase system, other phases are derived from this single cycle. The respective circuit tree data file consists of integer number data describing the circuit tree [12].

2.2.9 Search for Capacitor Loops or Inductor Cut-sets

A unique feature of the proposed simulation program is that it can handle capacitor loops and/or inductor cut-sets. This feature frees the user from employing additional components (such as series and shunt resistors) to eliminate unavoidable loops and cut-sets. The restriction resulting from the presence of such circuit conditions is that only the readily available state variables (i.e. capacitor twig voltages,

inductor link currents) are used as outputs. If, however, the circuit does not include any capacitor loops or inductor cut-sets, then the user is free to ask for any valid circuit output. To compute these outputs the program introduces a zero-value voltage source in series , or a zero-value current source in parallel with the respective circuit element. As expected, the voltage and current sources are used to compute current and voltage quantities respectively.

2.2.10 The Output Waveforms

When a requested output has been computed, the program notifies the user by appropriately labeling the respective circuit component(s). Once a requested output becomes available, it can be displayed on the CRT, or sent to a plotter, or stored on the data floppy disk for further analysis. Such analysis might involve taking the FFT (fast fourier transform) of a waveform to obtain its frequency spectrum.

2.2.11 Editing The Circuit

A circuit editing capability has been included when repeated program runs are required for different circuit component values . It enables the user to bypass time consuming parts of the program (such as tree formation, etc.) by using relevant data from previous runs. EDIT also allows the user to customize his output plots by adjusting vertical and/or horizontal axes and to perform operations on outputs such as the product of two waveforms. Some of these features are shown in chapter 4.

3.0 FORMULATION OF THE CIRCUIT EQUATIONS.

In order to establish the methods by which ASPPEC formulates the circuit equations used to solve the output waveforms, identification of the circuit components and sub-circuits must be introduced. It should be noted that the trivial and impractical case of a purely resistive network is not considered by ASPPEC since Kirchhoff's voltage law (KVL) and Kirchhoff's current law (KCL) equations would be static - no variations in the voltage or current of the circuit would occur in time. Therefore, a very impractical circuit to incorporate into the program since it is both time consuming (computer running time) and memory consuming. This type of circuit and also resistive circuits containing only sources are not allowed by the program since the routines to analyse such circuits were not incorporated.

In general, circuits are classified by the type of electrical components making up the circuit and by the voltage and current characteristics of these electrical components. Therefore, this section will first present the type of circuit components and circuit networks (sub-circuits) that can be analysed by ASPPEC. Secondly, a systematic formulation of the circuit equations will be developed.

3.1 Circuit Elements

Six basic circuit elements are available to the user for creating a network of elements:

1) Independent voltage sources.

2) Independent current sources.

3) Resistors.

4) Inductors.

5) Capacitors.

6)-Switch configurations.

The schematic representation of these circuit elements are shown in Figure 3.1.

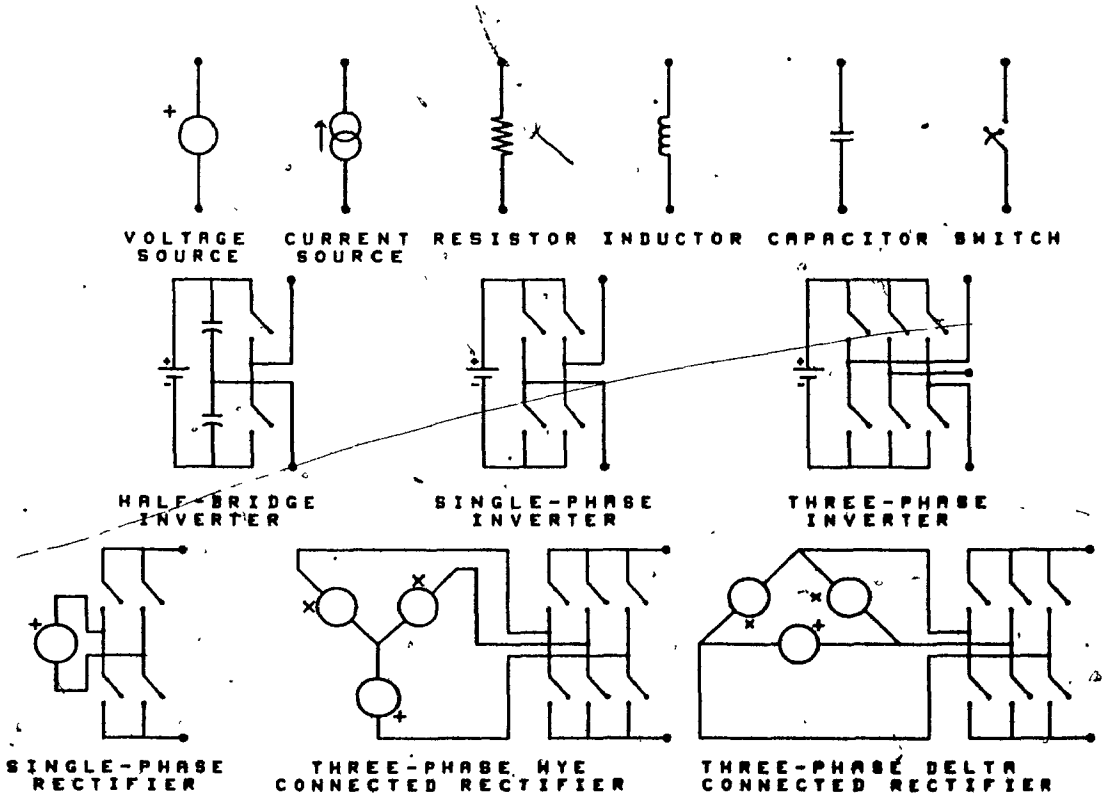


Figure 3.1. Schematic representation of circuit elements.

Independent voltage and current sources can be any time varying waveform created by the user (customized waveform) or anyone of the common sinusoidal or dc waveforms is produced by ASPPEC. The customized sources are introduced to the program by entering the data file name during the "entering the parameters" period of the program. The customized source data files can be produced by the user and stored in the format mentioned in chapter 2. They are mathematically represented as v(t) and i(t), where v(t) represents the voltage at any given time across the voltage source terminals shown in Figure 3.1. The plus sign seen next to the source (Figure 3.1) indicates the positive terminal. The current i(t) represents the current in the direction of the arrow at any instant of time through the two terminal current source device also shown in Figure 3.1. Besides these two schematic representations of the voltage source and current sources the program has also incorporated three other schematic sources:

1) Constant voltage source.
2) Battery source.
3) Sinusoidal voltage.

All the voltage sources represent the voltage across a two-terminal device where the voltage can be described by the user using a data file and the default time varying voltage, is a sinusoidal voltage. For example, if the user enters only a numeric value instead of an ASCII name during the "entering the parameters" period (described in chapter 2) and the source is schematically represented by a sinusoidal voltage

source, ASPPEC will default to the sinusoidal source with a defined base frequency and root-mean-square (rms) value of entered amplitude. All types of sources are assumed to be ideal. This means that no matter what network of elements is connected to the terminals of a voltage or current source, the voltage across the voltage source and the current through the current source will maintain its magnitude and waveform. This is true no matter what amount of current is demanded by the network connected across the terminals of a voltage source and no matter what amount of voltage is demanded by the network connected to a current source.

The next set of electrical elements which are also schematically shown in Figure 3.1 are resistors, inductors, and capacitors, all of which are classified by ASPPEC under the heading "linear elements". The voltage and current characteristic of each of these three passive elements can be written as

$$v(t) = Ri(t)$$

or

$$v = Ri \tag{3.1}$$

$$v = \frac{Ldi}{dt} \tag{3.2}$$

and

$$v = \frac{1}{C}\int i\, dt \tag{3.3}$$

for the resistor, inductor, and capacitor respectively. Assuming the three passive elements RLC are lumped and their values remain constant in time (time-invariant). The resistor, inductor and capacitor behave as linear time-invariant elements since their values are assumed to remain constant. Therefore, a network composed of these elements is classified as a time-invariant linear network.

The final category of elements which are incorporated into the program are the switch configurations which are shown schematically in Figure 3.1. Six switch configurations are used in power electronic sub-circuits in power electronic equipment. They consist of the four basic elements mentioned earlier and the constant voltage sources, sinusoidal voltage sources, capacitors, and a new element-the switch. Before describing the switch and then the switch configurations as a whole, something should be said about the capacitors in the half-bridge inverter shown in Figure 3.1 (the only switch configuration with capacitors). It is assumed-and is most often the case in real applications-that the values of these capacitors are large enough to allow the replacement of the constant voltage source and the two capacitors by two ideal constant voltage sources each having half the amplitude of the original source (Figure 3.2). For this reason this type of sub-circuit are called a half-bridge inverter. Using this assumption all the switch configurations in Figure 3.1 are composed of at least one type of voltage source and a number of switches. Therefore, the switch must be defined to better understand the switch configurations.

The switch is shown in Figure 3.3 and can be visualized as a

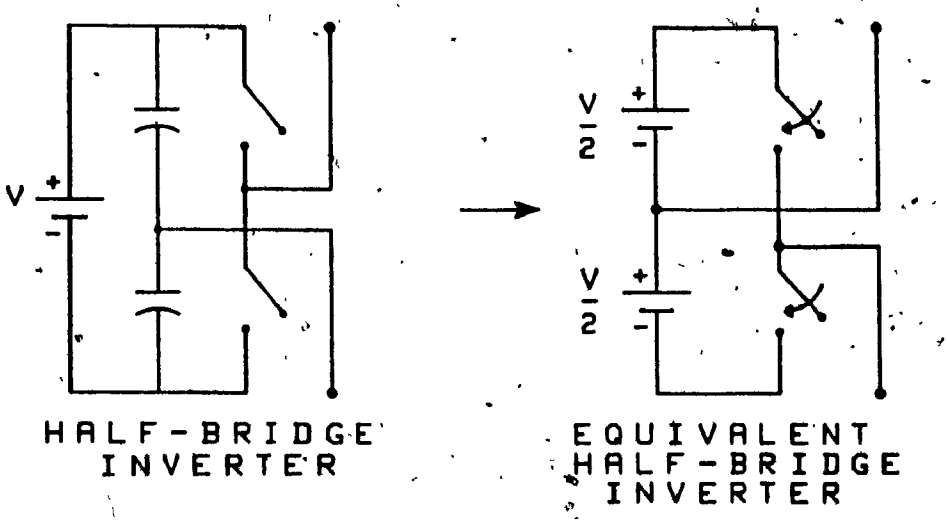


Figure 3.2. Half-bridge inverter equivalent circuit.

section of wire having two terminals that can be in one of two modes at at any given instant of time. The first mode is the open mode or off state and the second mode is the closed mode or on state. These states represent the cutoff region and the saturation region of a power

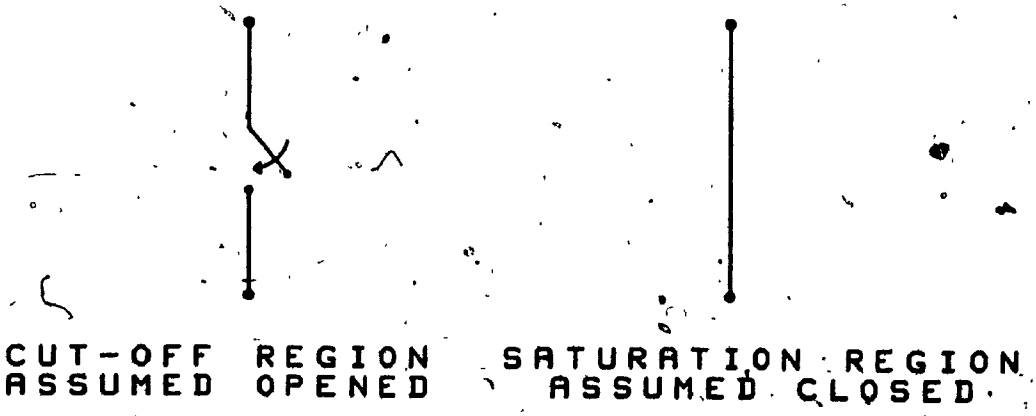


Figure 3.3. The ideal switch.

electronic transistor respectively. Also, the switch is idealized it has zero current flowing through the two terminals in the off state; it has zero voltage across the two terminals in the on state; and it is bidirectional. These 3 conditions are not true for the diode, thyristor, triac, GTO, bipolar transistor and MOSFET transistor. Moreover the switch has no losses or does not store any charge, the voltage can be any polarity between the two terminals, and the current can flow in either direction. Like the non-ideal switch, however, they are nonlinear. Therefore, the switch and the switch configuration can be found under the program sub-menu (circuit drawing program in Appendix 1) titled "nonlinear elements". These elements are classified as nonlinear because they fail the linearity test. One can quickly see that as the switch is opened the current is zero. When the switch is closed, the current can be any value up to infinity (which is determined by the rest of the circuit) and the voltage is zero. Therefore, the switch does not conform to the proportionality criteria for linearity.

For those who would like to imagine a more realistic representation of the switch, one can think of the switch as consisting of four diodes and a power electronic NPN bipolar transistor with ideal properties as shown in Figure 3.4. The realization of an ideal bidirectional switch using ideal components indicates that a third terminal (the gate) is present. More information is needed to allow the switch to alternate from one mode to the other. This extra information is provided by the gating signal. This gating signal is supplied to the gate of the transistor in Figure 3.4. The amplitude of this gating signal (either

current or voltage depending on the selected transistor) usually determines the state of the transistor. Since a power transistor can be in only one of two states, a simpler representation is required in order to indicate the state at any given instant in time. If this is not done the amplitude of the gating signal of a non-power transistor would determine if it is in the cutoff region, saturation region or active region.

Three functions describe the switching patterns the existence functions [16], gain functions [17], and switching functions [18]. Even

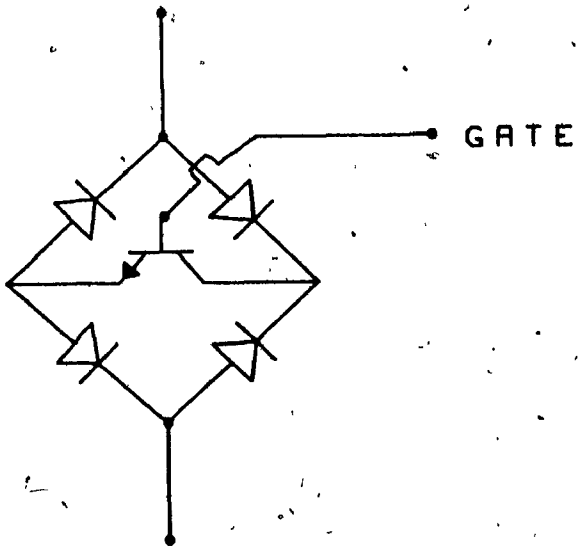


Figure 3.4. Realization of an ideal bidirectional switch using ideal components.

though the names vary from reference to reference, these functions mathematically describe switch states in much the same way. By looking at an example one can better understand a switching function. In Figure

3.5a the input voltage is shown to be $v_i(t)$, the output voltage across the resistor is $v_o(t)$, the current through the switch is $i_s(t)$, and the current in the output resistance is $i_o(t)$. Also, the switch is following an on-state and repetitive off-state switching pattern as shown in Figure 3.5b called the switching function of the switch. Switching

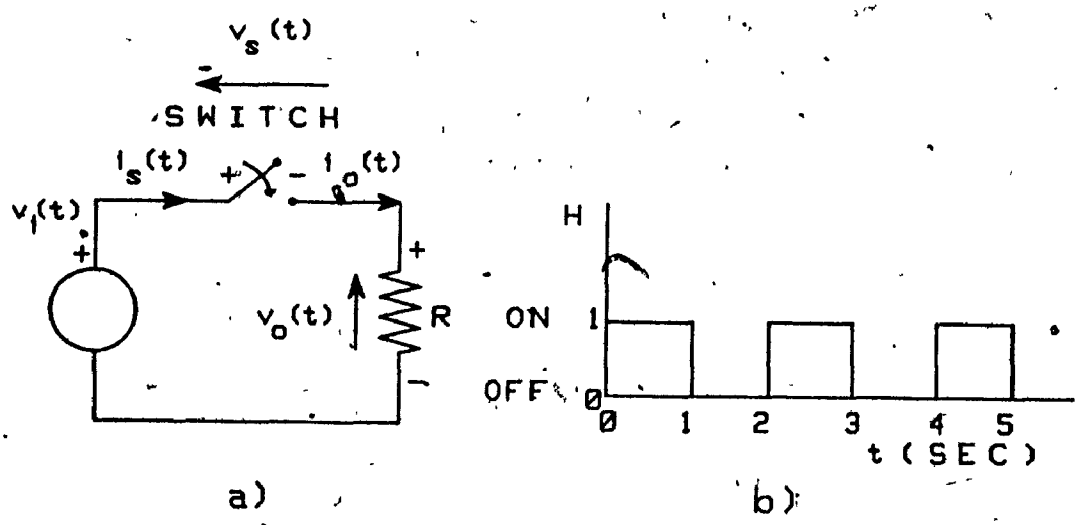


Figure 3.5 Test circuit a) using a switch and a resistor b) and the switching function of the switch.

functions are mathematically expressed by using the Fourier expansion

$$H(\omega t) = \sum_{n=0}^{\infty} \left[A_n \cos(n\omega t) + B_n \sin(n\omega t) \right]$$

where

$$A_0 = \frac{1}{T} \int_{-T/2}^{T/2} H \, dt$$

$$A_n = \frac{2}{T}\int_{-T/2}^{T/2} H\cos(n\omega t)\, dt \qquad n \neq 0$$

$$B_n = \frac{2}{T}\int_{-T/2}^{T/2} H\sin(n\omega t)\, dt$$

and

$n = 0, 1, 2, \ldots$

f = repetitive frequency of the pulses

$T = 1/f$

$\omega = 2\pi f$ the angular frequency

The variable H can have the values 1 or 0 which can be found from the switching function.

Having mathematically represented the switching function as $H(\omega t)$, the switching function can be multiplied by the input voltage $v_i(t)$ of Figure 3.5a to give a new voltage $v_i'(t)$, where

$$H(\omega t) \times v_i(t) = v_i'(t) \qquad (3.4)$$

This concept is fundamental to the development of the ASPPEE program. For example, in the circuit of Figure 3.5a let

$$v_i(t) = 10 \text{ V}$$

$$R = 5\ \Omega$$

and the switching function H(ωt) is the one shown in Figure 3.5b. Also by using KCL and Ohm's law

$$v(t) = i(t) \times R$$

The output voltage $v_o(t)$, output current $i_o(t)$, switch current $i_s(t)$, and switch voltage $v_s(t)$ are shown in Figure 3.6. This is a very simple

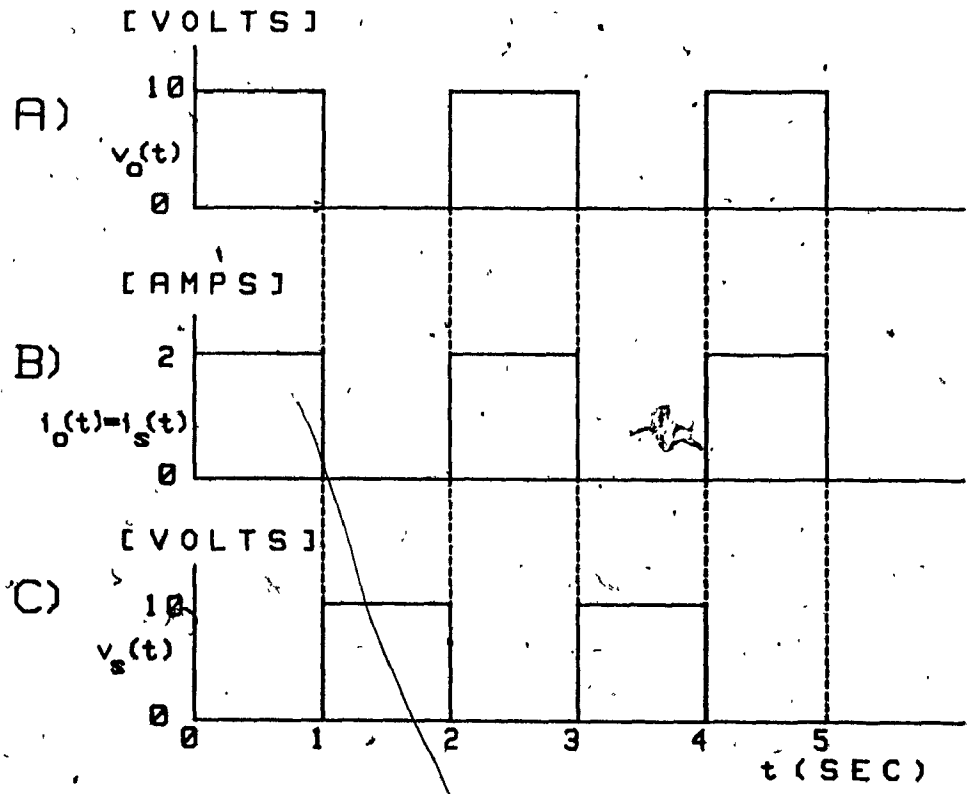


Figure 3.6. Ideal power circuit output waveforms.

but good illustrative example showing the voltage and current waveforms of a simple power electronic circuit. It should be emphasized that it is assumed that the switch is an ideal switch. No modeling is performed to represent the switch and certain electrical characteristics of the switch are neglected, such as the on-voltage drop or the effects of parasitic capacitance between terminals. One is tempted to say , therefore, that the dynamical solutions presented by ASPPEC will not represent the actual solution waveforms since certain electrical characteristics of the standard transistor are ignored. Moreover, since the validity of the final result is questionable the solution is an approximation of the real solution. This is not true, however, because the power electronic transistor and its technology actually approaches the performance of the ideal switch. To illustrate the validity of this statement actual examples of power electronic circuit waveforms and their simulated results by ASPPEC are presented in chapters 5 and 6.

Getting back to the example of Figure 3.5a, Figure 3.6 illustrates all the voltage and current waveforms of the circuit. Another way of presenting this example is by replacing the dc voltage, $v_1(t)$ (10 volts), and the switch in the example by a new voltage $v_1'(t)$ as shown in Figure 3.7. The new voltage source $v_1'(t)$ is just the switching function multiplied by the original voltage source $v_1(t)$ (equation 3.4). This results in another circuit which has the same output voltage and current waveforms as shown in Figure 3.6a and b. A problem in doing this new modified circuit, which is not apparent in this simple example, is that all the voltage and current waveforms on the right of the switch or

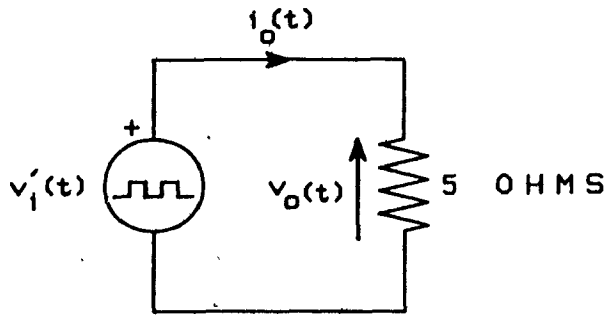


Figure 3.7. Modification of the circuit after removing the switch.

configuration can be derived using techniques of linear network theory [12]. However, the voltage/current waveforms of the circuit on the left of the switch configuration seems to be lost (Figure 3.7), since the supply and switch configuration is replaced by a varying power supply. Thus vital information about the transistor electrical rating requirement and original power supply ratings is lost. But actually, using the techniques described in reference [16], one can reconstruct the voltage and current waveforms of the individual switches and power supply. For example, by referring to the example of Figure 3.5 the output current $i_o(t)$ can be derived by solving for the current of the modified circuit shown in Figure 3.7, and multiplying the result by the switching function described by the Fourier expansion of the switching function shown in Figure 3.5b. The result will be the current $i_s(t)$ through the power electronic switch and in this case the input current

delivered from the supply. This result may be obvious in this example but the process and result holds true for many power electronic equipment configurations.

A power electronic circuit with linear and nonlinear elements can be replaced by an equivalent linear circuit. Replacing the supply and switch configuration in this fashion, transforms a nonlinear circuit into a purely linear circuit. This means that the output voltage and current waveforms can be derived by using the State Space method for linear time-invariant systems where there is a wealth of information [9] [12] [13] [14] [19]. In the next section the State Space method is used to formulate the circuit equations in a systematic way.

3.2 Systematic Formulation of Circuit Equations

At this point in the program, the subroutine called "Time-Domain", beginning at line 16035 and ending at line 22020 of the tree-link program (Appendix 2), constructs the normal tree, the formation of circuit equations, and the discretized solution of these equations. Only the circuit tree and formation of circuit equations are discussed in this section. There are 9 main blocks:

1) Construction of a tree, lines 16240 to 16960 (Appendix 2)
2) Partitioning of the corresponding voltage and current vectors from the element table (inputs).
3) Using KCL equations on the tree to produce the fundamental cut-set link matrix Q_l, lines 17990 to 18710 (Appendix 2).

4) From the fundamental cut-set link matrix the fundamental loop twig matrix (B_t) can be derived.

5) Partitioning of the fundamental cut-set link matrix into submatrices.

6) Expansion of the equation can be derived when B_t and Q_l matrices are inserted back into Kirchhoff's equations.

7) By using the voltage/current relationship of the elements, unwanted variables can be eliminated so that only the state variables are left.

8) The general state equations, a set of first-order differential equations, are constructed.

9) The state equations for a network having no derivatives of the sources are constructed.

In the general state equations (block 8), a set of first-order differential equations

$$\frac{dx}{dt} = \mathcal{A}x + \mathcal{B}_1 e + \mathcal{B}_2 \frac{de}{dt} \qquad (3.5)$$

and the output equation

$$w = \mathcal{C}x + \mathcal{D}_1 e + \mathcal{D}_2 \frac{de}{dt} \qquad (3.6)$$

are derived. Both equations are referred to as state equations. Where $\mathcal{A}$, $\mathcal{B}_1$, $\mathcal{C}$, $\mathcal{D}_1$, $\mathcal{D}_2$ are matrices, **x** is the state vector, the elements of **x** are state variables, and **e** is the input source vector. Using the

transformation

$$x' = x - \mathcal{B}_2 e \tag{3.7}$$

one finds that

$$\frac{dx'}{dt} = \frac{d\left[x - \mathcal{B}_2 e\right]}{dt}$$

and

$$\frac{dx'}{dt} = \frac{dx}{dt} - \mathcal{B}_2 \frac{de}{dt}$$

therefore

$$\frac{dx}{dt} = \frac{dx'}{dt} + \mathcal{B}_2 \frac{de}{dt}$$

Equation 3.5 becomes

$$\frac{dx'}{dt} = \mathcal{A}x' + \mathcal{B}e \tag{3.8}$$

Using the linear combination of the original state vector x

$$x = x' + \mathcal{B}_2 e$$

The output equation is

$$w = \mathcal{C}x + \mathcal{D}e \tag{3.9}$$

where $\mathcal{B} = \mathcal{B}_1 + \mathcal{A}\mathcal{B}_2$, $\mathcal{D} = \mathcal{D}_1 + \mathcal{C}\mathcal{B}_2$. Since the outputs are restricted to capacitor twigs and inductor links, $\mathcal{D}_2$ will always be zero. Finally, for a causal system (block 9), a term later explained in this section, the state equations for selected outputs become

$$\frac{d\mathbf{x}}{dt} = \mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{e} \tag{3.10}$$

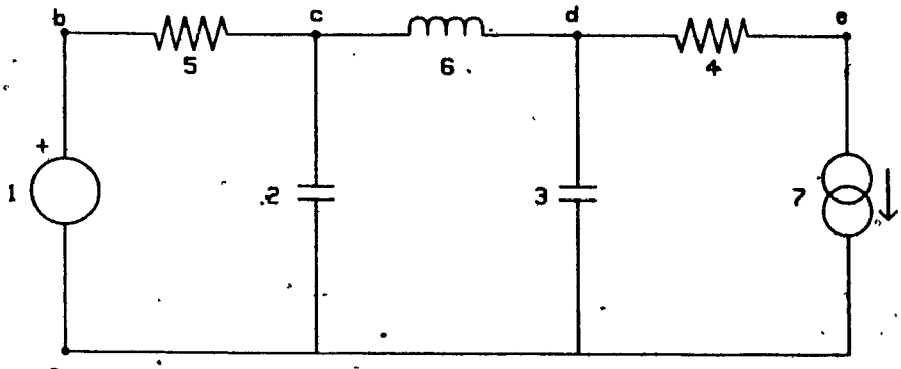
and the output equation

$$\mathbf{w} = \mathcal{C}\mathbf{x} + \mathcal{D}\mathbf{e} \tag{3.11}$$

where again $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ are matrices, $\mathbf{x}$ is the state vector whose elements are state variables, and $\mathbf{e}$ is the input source vector.

In the previous section (section 3.1) it was shown that the final dynamical time-invariant system consists of voltage sources, current sources, and RLC linear elements with voltage/current relationships described by expressions 3.1, 3.2 and 3.3. Before calling the "Time Domain" subroutine the program has developed a table consisting of element codes (codes for each type of element), element values, and the actual element terminal coordinate values. Using this table the program solves for a normal tree [12].

A tree is a different representation of a circuit. It involves the following conditions in producing a tree consisting of twigs and links. For example take the circuit shown in Figure 3.8, where the solid line and dotted line in the tree represents the twig and link respectively. The tree is constructed following the rules outlined in reference [12]



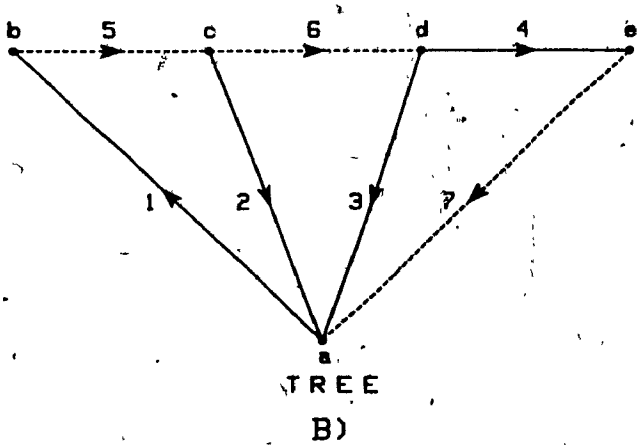


Figure 3.8 Example circuit and its tree.

which states that all nodes must be connected by at least one twig and there can be no twig loops (loops produced entirely by twigs). There are 4 rules for assigning twigs and links:

a) Voltage sources must be twigs and current sources must be links.

b) As much as possible capacitors should be twigs and as much as possible inductor should be links.

c) Resistor twigs and resistor links which are assigned so that the two previous rules are not violated.

d) As few as possible inductor twigs and capacitor links.

At this point it is important to mention that the program analysis section distinguishes between two types of circuits which are

1) Capacitor-loop-inductor-cut-set circuit (CLICS).

2) Non-capacitor-loop-inductor-cut-set circuit (NCLICS).

A CLICS circuit (also called a non-causal system) is a circuit that has at least one inductor twig or capacitor link in its tree representation. The test location for such a circuit is shown in the flowchart (Figure 2.1) of the program and is located under the block name called "capacitor loop inductor cut-set". Schematically, the presence of any capacitor-loops; a loop containing only capacitors or only capacitors and independent voltage sources, and the presence of any inductor cut-set; a cut-set containing only inductors or only inductors and independent current sources where a cut-set of branches of a connected graph (more specifically a tree) where removal will cause the graph to

become unconnected into exactly two subgraphs (subtrees), is classified as a CLICS circuit. When the test for such a circuit indicates a positive result, the program prevents the user from selecting output waveforms. This is so because we no longer have only solutions where output waveforms are a linear combinations of the state variables; instead we have degeneracies where the capacitor voltage and inductor currents are restricted by some constraint. This introduces the derivative of the input source vector (e) term in the state equation 3.5. Thus, the order of complexity of the circuit equals the number of independent energy-storing branches (capacitors plus inductors) less the number of capacitor links and inductor twigs in the tree (capacitor loops and inductor cut-sets) in the circuit. Therefore, the solution is restricted to solving the output voltage of capacitor twigs and current waveforms through inductor links. In this case the transformation mentioned earlier to solve the general expressions 3.8 and 3.9 is used. It should be emphasized that this transformation is a very important feature since it allows the user to construct circuits containing either capacitor loops or inductor cut-sets. These circuits are very important ones and often arise in actual applications especially in power electronic three-phase circuits where filter components inherently produce capacitor loops and/or inductor cut-sets. A simple resistor placed in the proper location can alleviate this problem as is recommended by the authors of ATOSEC5 [8], but this resistor puts a constraint on the user and reduces the program's circuit generalization ability. Adding this resistor will also increase the size of the fundamental cut-set link matrix, and the program in turn will need more

memory and the matrix operation speed is reduced. Not giving the user the option for selection is not really a problem since the solutions of the state variables can be used to evaluate any output waveforms desired by the user.

In the NCLICS circuit, (causal system) [13], ASPPEC will give the user the option to select the desired output waveforms. This is possible since the state equations 3.9 and 3.10 do not contain any terms involving future predictions of the input sources ($\frac{de}{dt}$). Therefore, output waveforms can be selected freely by the user. Transparent to the user, the program inserts zero value voltage sources in series with the branch element where output current waveforms were selected, and zero value current sources in between branch terminals where output voltage waveforms are selected. The voltage polarities and current direction remain consistent with the respective branch voltage polarities and current direction followed in the circuit tree representation. These zero valued voltage sources (represented as Eo), and zero valued current sources (represented as Jo) are implemented in the analysis until the final solution where the output waveforms are systematically derived.

The actual routine implemented in the program for checking capacitor loops or inductor cut-sets is simply the tree routine, which functions as if it was solving a tree until one of the degeneracies is found. When this happens the program then ignores the output selection routine.

The arrowheads seen on the branches in the tree of Figure 3.8, represent the direction of the current. They are placed in the direction of the entered element align in the direction of the second element

terminal entry, selected by the user. The voltage source polarity indicated by the plus sign and current source arrow direction are consistent with the assigned current direction of the branches. Moreover, the output current waveform selection direction-indicator during the output selection part of the program also is consistent with the branch current directions. Therefore, the user knows the directions and polarity of the respective output current and voltage waveforms. Moving along to the second part of the outline mentioned earlier (block 2), the partitioning of the voltage and current vectors due to the tree arrangement and the ease in recovering the zero voltage and current source submatrices (using the convenience of the REDIM statement of the HP program language), the partitioning of the source vectors are

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_t \\ \mathbf{v}_l \end{bmatrix} \Rightarrow \mathbf{v}_t = \begin{bmatrix} v_{Eo} \\ v_{E1} \\ v_{Ct} \\ v_{Rt} \\ v_{Lt} \end{bmatrix} \quad \mathbf{v}_l = \begin{bmatrix} v_{Cl} \\ v_{Rl} \\ v_{Ll} \\ v_{Jo} \\ v_{J1} \end{bmatrix} \qquad (3.12a)$$

$$\mathbf{i} = \begin{bmatrix} \mathbf{i}_t \\ \mathbf{i}_l \end{bmatrix} \Rightarrow \mathbf{i}_t = \begin{bmatrix} i_{Eo} \\ i_{E1} \\ i_{Ct} \\ i_{Rt} \\ i_{Lt} \end{bmatrix} \quad \mathbf{i}_l = \begin{bmatrix} i_{Cl} \\ i_{Rl} \\ i_{Ll} \\ i_{Jo} \\ i_{J1} \end{bmatrix} \qquad (3.12b)$$

Where, for example, v_{Eo} is the vector of zero voltage source twig voltage, and the subscripts t and l represent twig and link

respectively. These subscripts are understood for the input voltage quantities (E_1 and E_o) and input current quantities (J_1 and J_o) therefore, not used. Also, the voltage quantities E_o and E_1 are partitions of the larger quantity E, similarly the current quantities J_o and J_1 are partitions of the larger quantity J.

Using the tree of Figure 3.8, the twig currents in terms of link currents (in scalar form) using KCL equations are

$$i_1 = i_5 \tag{3.13a}$$

$$i_2 = i_5 - i_6 \tag{3.13b}$$

$$i_3 = i_6 - i_7 \tag{3.13c}$$

$$i_4 = i_7 \tag{3.13d}$$

In matrix form we know that

$$Q_f = \left[Q_t \; Q_l\right] = \left[I \; Q_l\right]$$

Where the subscript f stands for fundamental and therefore Q_f is the fundamental cut-set matrix. This expression can also be written as

$$Q_f = \left[I \; Q_l\right]\begin{bmatrix} i_t \\ i_l \end{bmatrix} = i_t + Q_l i_l = 0$$

or

$$i_t = -Q_l i_l \tag{3.14}$$

Similarly using KVL equations it can be shown that

$$v_l = -B_t v_t$$

or

$$v_l = Q_l^T v_t \tag{3.15}$$

Where ^{T} signifies the matrix transpose operation, and B_t is the fundamental loop twig matrix. From the twig current equations 3.13, the Q_l matrix can be written as

$$Q_l = \begin{array}{c} \text{branch} \\ \\ \text{twig} \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 5 & 6 & 7 \\ \text{link} \rightarrow \text{R} & \text{L} & \text{J} \end{array} \\ \begin{array}{c} \text{E} \\ \\ \text{C} \\ \text{R} \end{array} \begin{bmatrix} -1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix} \end{array} \tag{3.16}$$

which is less than full but gives the general idea behind the partitioning process. The Q_l matrix in a general form can be written as

$$
Q_l = \begin{array}{c} \text{link} \rightarrow \\ \text{twig} \downarrow \\ \\ E \\ \\ \\ \\ \\ \end{array}
\begin{array}{c}
\begin{array}{ccccc} C & R & L & J_o & J_1 \end{array} \\
\begin{bmatrix}
Q_{EoC} & Q_{EoR} & Q_{EoL} & Q_{EoJo} & Q_{EoJ1} \\
Q_{E1C} & Q_{E1R} & Q_{E1L} & Q_{E1Jo} & Q_{E1J1} \\
Q_{CC} & Q_{CR} & Q_{CL} & Q_{CJo} & Q_{CJ1} \\
0 & Q_{RR} & Q_{RL} & Q_{RJo} & Q_{RJ1} \\
0 & 0 & Q_{LL} & Q_{LJo} & Q_{LJ1}
\end{bmatrix}
\end{array}
\begin{array}{l} E_o \\ E_1 \\ C \\ R \\ L \end{array}
\qquad (3.17)
$$

Where, for example, Q_{CC} is a submatrix from the partitioned Q_l matrix. Take for example the Q_l matrix developed from the tree shown in Figure 3.8. The submatrices are

$$
Q_{ER} = \begin{bmatrix} -1 \end{bmatrix} \quad Q_{EL} = \begin{bmatrix} 0 \end{bmatrix} \quad Q_{EJ} = \begin{bmatrix} 0 \end{bmatrix}
$$

$$
Q_{CR} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad Q_{CL} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad Q_{CJ} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad (3.18)
$$

$$
Q_{RR} = \begin{bmatrix} 0 \end{bmatrix} \quad Q_{RL} = \begin{bmatrix} 0 \end{bmatrix} \quad Q_{RJ} = \begin{bmatrix} -1 \end{bmatrix}
$$

The three zero entries come directly from the CLICS circuit definition. If a capacitor link exists, by definition there can only be voltage sources and/or other capacitors in the loop. Therefore, under the capacitor link column, the R and L twig entries must be zero. Similarly, the C and R entries in the inductor twig must also be zero. Once the Q_l matrix is constructed, the program can quickly find the KVL equations in

matrix form from equation 3.15. Where

$$B_t = -Q_l^T$$

Now that the Q_l matrix is formed, using the voltage vectors 3.12a, current vectors 3.12b, KCL equations in the form of equation 3.14, and KVL equations in the form of equation 3.15, the result will be an expanded form showing all the twig currents and link voltages. To simplify the derivation, first the CLICS circuit with no zero sources (Eo and Jo terms in the Q_l matrix do not need to be separately considered) case will be derived. Then, the derivation for the NCLICS circuit will be derived.

3.2.1 CLICS Circuit

When the Eo and Jo terms are not considered, the expanded form of equations 3.14 and 3.15 is

$$i_E = -Q_{EC}i_{Cl} - Q_{ER}i_{Rl} - Q_{EL}i_{Ll} - Q_{EJ}i_J \qquad (3.19a)$$

$$i_{Ct} = -Q_{CC}i_{Cl} - Q_{CR}i_{Rl} - Q_{CL}i_{Ll} - Q_{CJ}i_J \qquad (3.19b)$$

$$i_{Rt} = 0 - Q_{RR}i_{Rl} - Q_{RL}i_{Ll} - Q_{RJ}i_J \qquad (3.19c)$$

$$i_{Lt} = 0 \quad 0 - Q_{LL}i_{Ll} - Q_{LJ}i_J \qquad (3.19d)$$

and

$$v_{Cl} = Q_{EC}^T v_E + Q_{CC}^T v_{Ct} \quad 0 \quad 0 \tag{3.20a}$$

$$v_{Rl} = Q_{ER}^T v_E + Q_{CR}^T v_{Ct} + Q_{RR}^T v_{Rt} \quad 0 \tag{3.20b}$$

$$v_{Ll} = Q_{EL}^T v_E + Q_{CL}^T v_{Ct} + Q_{RL}^T v_{Rt} + Q_{LL}^T v_{Lt} \tag{3.20c}$$

$$v_J = Q_{EJ}^T v_E + Q_{CJ}^T v_{Ct} + Q_{RJ}^T v_{Rt} + Q_{LJ}^T v_{Lt} \tag{3.20d}$$

By using the voltage/current relationships of the network elements (first seen in scalar form as equations 3.1, 3.2, and 3.3) the resistor, inductor, and capacitor elements can be rewritten in a matrix form which describes the link and twig voltage/current relationships as

$$\begin{bmatrix} i_{Rl} \\ v_{Rt} \end{bmatrix} = \begin{bmatrix} G_l & 0 \\ 0 & R_t \end{bmatrix} \begin{bmatrix} v_{Rl} \\ i_{Rt} \end{bmatrix} \tag{3.21}$$

$$\begin{bmatrix} v_{Ll} \\ v_{Lt} \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} L_l & 0 \\ 0 & L_t \end{bmatrix} \begin{bmatrix} i_{Ll} \\ i_{Lt} \end{bmatrix} \tag{3.22}$$

$$\begin{bmatrix} i_{Ct} \\ i_{Cl} \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} C_t & 0 \\ 0 & C_l \end{bmatrix} \begin{bmatrix} v_{Ct} \\ v_{Cl} \end{bmatrix} \tag{3.23}$$

The final variables of interest are v_{Ct} and i_{Ll} - the state variables. The derivation process consists of deriving expressions for v_{Ct} and i_{Ll}.

using equations 3.19b and 3.20c respectively, while all other variables using equations 3.19, 3.20, 3.21, 3.22, and 3.23 are eliminated.

Equation 3.19b can be rearranged as

$$i_{Ct} + Q_{CC} i_{Cl} = -Q_{CR} i_{Rl} - Q_{CL} i_{Ll} Q_{CJ} i_{J} \tag{3.24}$$

The left-hand side can be rewritten as

$$i_{Ct} + Q_{CC} i_{Cl} = \begin{bmatrix} I & Q_{CC} \end{bmatrix} \begin{bmatrix} i_{Ct} \\ i_{Cl} \end{bmatrix} \tag{3.25}$$

Then, by substituting equation 3.23 into equation 3.25, variables i_{Ct} and i_{Cl} can be eliminated from the right-hand side. That is

$$i_{Ct} + Q_{CC} i_{Cl} = \begin{bmatrix} I & Q_{CC} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} C_t & 0 \\ 0 & C_l \end{bmatrix} \begin{bmatrix} v_{Ct} \\ v_{Cl} \end{bmatrix} \tag{3.26}$$

Substituting equation 3.20a into equation 3.26 the variable v_{Cl} can also be eliminated giving a final equation

$$i_{Ct} + Q_{CC} i_{Cl} = \begin{bmatrix} I & Q_{CC} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} C_t & 0 \\ 0 & C_l \end{bmatrix} \begin{bmatrix} v_{Ct} \\ Q_{EC}^T v_E + Q_{CC}^T v_{Ct} \end{bmatrix}$$

With variables i_{Ct}, i_{Cl}, and v_{Cl} eliminated on the right-hand side

$$i_{Ct} + Q_{CC} i_{Cl} = \frac{d}{dt} \begin{bmatrix} I & Q_{CC} \end{bmatrix} \begin{bmatrix} C_t & 0 \\ 0 & C_l \end{bmatrix} \begin{bmatrix} I \\ Q_{CC}^T \end{bmatrix} v_{Ct}$$

$$+ \frac{d}{dt}\begin{bmatrix} I & Q_{CC} \end{bmatrix}\begin{bmatrix} C_t & 0 \\ 0 & C_l \end{bmatrix}\begin{bmatrix} 0 \\ Q_{EC}^T \end{bmatrix} v_E$$

or

$$i_{Ct} + Q_{CC} i_{Cl} = \frac{d}{dt}\mathcal{E} v_{Ct} - \frac{d}{dt}\mathcal{E}' v_E \qquad (3.27)$$

where

$$\mathcal{E} = \begin{bmatrix} I & Q_{CC} \end{bmatrix}\begin{bmatrix} C_t & 0 \\ 0 & C_l \end{bmatrix}\begin{bmatrix} I \\ Q_{CC}^T \end{bmatrix} = C_t + Q_{CC} C_l Q_{CC}^T$$

and

$$\mathcal{E}' = -\begin{bmatrix} I & Q_{CC} \end{bmatrix}\begin{bmatrix} C_t & 0 \\ 0 & C_l \end{bmatrix}\begin{bmatrix} 0 \\ Q_{EC}^T \end{bmatrix} = -Q_{CC} C_l Q_{EC}^T$$

Substituting equation 3.27 back into equation 3.24

$$\frac{d}{dt}\mathcal{E} v_{Ct} = -Q_{CR} i_{Rl} - Q_{CL} i_{Ll} - Q_{CJ} i_J + \frac{d}{dt}\mathcal{E}' v_E \qquad (3.28)$$

The task now is to eliminate i_{Rl}, which can be done by substituting equation 3.20b into

$$i_{Rl} = G_l v_{Rl}$$

an expression which comes from equation 3.21 giving

$$i_{Rl} = G_l Q_{ER}^T v_E + G_l Q_{CR}^T v_{Ct} + G_l Q_{RR}^T v_{Rt} \qquad (3.29)$$

Substituting equation 3.19c into

$$v_{Rt} = R_t i_{Rt}$$

(from equation 3.21) will eliminate i_{Rt} from the equation giving

$$v_{Rt} = -R_t Q_{RR} i_{Rl} - R_t Q_{Rl} i_{Ll} - R_t Q_{RJ} i_J \tag{3.30}$$

Substitution of equation 3.30 into equation 3.29 will give

$$i_{Rl} = G_l Q_{ER}^T v_E + G_l Q_{CR}^T v_{Ct} - G_l Q_{RR}^T R_t Q_{RR} i_{Rl} - G_l Q_{RR}^T R_t Q_{RL} i_{Ll} \tag{3.31}$$

$$- G_l Q_{RR}^T R_t Q_{RJ} i_J$$

Equation 3.31 can be rewritten as

$$i_{Rl} + G_l Q_{RR}^T R_t Q_{RR} i_{Rl} = G_l Q_{ER} v_E^T + G_l Q_{CR} v_{Ct}^T - G_l Q_{RR} R_t^T Q_{RL} i_{Ll}$$

$$- G_l Q_{RR}^T R_t Q_{RJ} i_J$$

or

$$\left[I + G_l Q_{RR}^T R_t Q_{RR}\right] i_{Rl} = G_l Q_{ER}^T v_E + G_l Q_{CR}^T v_{Ct} - G_l Q_{RR}^T R_t Q_{RL} i_{Ll}$$

$$- \quad G_l Q_{RR}^T R_t Q_{RJ} i_J$$

Three variables $\mathcal{K}_1$, R, and G_l are defined as

$$\mathcal{K}_1 = \left[I_m + G_l Q_{RR}^T R_t Q_{RR} \right]$$

$$R = R_l + Q_{RR}^T R_t Q_{RR}$$

and

$$G_l = R^{-1}$$

Therefore

$$\mathcal{K}_1 = G_l R$$

where R is the loop-parameter matrix and is non-singular [1]. As a consequence $\mathcal{K}_1$ is non-singular and $\mathcal{K}_1^{-1}$ always exists. Moreover

$$i_{Rl} = R^{-1}G_l^{-1}G_l Q_{ER}^T v_E + R^{-1}G_l^{-1}G_l Q_{CR}^T v_{Ct} - R^{-1}G_l^{-1}G_l Q_{RR}^T R_t Q_{RL} i_{Ll}$$

$$- R^{-1}G_l^{-1}G_l Q_{RR}^T R_t Q_{RJ} i_J$$

$$i_{Rl} = R^{-1}Q_{ER}^T v_E + R^{-1}Q_{CR}^T v_{Ct} - R^{-1}Q_{RR}^T R_t Q_{RL} i_{Ll} - R^{-1}Q_{RR}^T R_t Q_{RJ} i_J \qquad (3.32)$$

By substituting equation 3.32 back into equation 3.28, an equation free from non-state variables is found

$$\frac{d}{dt}\mathcal{C}v_{Ct} = -Q_{CR}R^{-1}G_l Q_{ER}^T v_E - Q_{CR}R^{-1}Q_{Ct}^{-1}v_{Ct} + Q_{CR}R^{-1}Q_{RR}^T R_t Q_{RL} i_{Ll}$$

$$+ Q_{CR}R^{-1}Q_{RR}^T R_t Q_{RJ} i_J - Q_{CL} i_{Ll} - Q_{CJ} i_J + \frac{d}{dt}\mathcal{C}' v_E$$

$$\frac{d}{dt}\mathcal{C}v_{Ct} = -\left[Q_{CR}R^{-1}Q_{CR}^T\right]v_{Ct} + \left[-Q_{CL} + Q_{CR}R^{-1}Q_{RR}R_t Q_{RL}\right] - \left[Q_{CR}R^{-1}G_l Q_{ER}^T\right]v_E$$

$$+ \left[-Q_{CJ} + Q_{CR}R^{-1}Q_{RR}^T R_t Q_{RJ}\right] i_J + \frac{d}{dt}\mathcal{C}' v_E$$

or more compactly

$$\frac{d}{dt}\mathcal{C}v_{Ct} = -\mathcal{Y} i_{Ct} + \mathcal{H} i_{Ll} - \mathcal{Y}' v_E + \mathcal{H}' i_J + \frac{d}{dt}\mathcal{C}' v_E \qquad (3.33)$$

where

$$\mathcal{Y} = Q_{CR}R^{-1}Q_{Ct}^T$$

$$\mathcal{H} = -Q_{CL} + Q_{CR}R^{-1}Q_{RR}^T R_t Q_{RL}$$

$$\mathcal{Y}' = Q_{CR}R^{-1}Q_{ER}^T$$

$$\mathcal{H}' = -Q_{CJ} + Q_{CR}R^{-1}Q_{RR}^T R_t Q_{RJ}$$

Similarly, using equation 3.20c and rearranging as

$$v_{Ll} - Q_{LL}^T v_{Lt} = Q_{EL}^T v_E + Q_{CL}^T v_{Ct} + Q_{RL}^T v_{Rt} \tag{3.34}$$

the left-hand side can be rearranged as

$$v_{Ll} - Q_{LL}^T v_{Lt} = \begin{bmatrix} I & -Q_{LL}^T \end{bmatrix} \begin{bmatrix} v_{Ll} \\ v_{Lt} \end{bmatrix} \tag{3.35}$$

Substituting equation 3.22 (the voltage/current relationship for an inductor) into equation 3.35, variables v_{Ll} and v_{Lt} can be eliminated from the right-hand side

$$v_{Ll} - Q_{LL}^T v_{Lt} = \begin{bmatrix} I & -Q_{LL}^T \end{bmatrix} \frac{d}{dt} \begin{bmatrix} L_l & 0 \\ 0 & L_t \end{bmatrix} \begin{bmatrix} i_{Ll} \\ i_{Lt} \end{bmatrix} \tag{3.36}$$

The variable i_{Lt} can be eliminated from equation 3.36 by replacing the variable with equation 3.19d. Therefore, equation 3.36 becomes

$$v_{Ll} - Q_{LL}^T v_{Lt} = \begin{bmatrix} I & -Q_{LL}^T \end{bmatrix} \frac{d}{dt} \begin{bmatrix} L_l & 0 \\ 0 & L_t \end{bmatrix} \begin{bmatrix} i_{Ll} \\ -Q_{LL} i - Q_{LJ} i_J \end{bmatrix}.$$

$$v_{Ll} - Q_{LL}^T v_{Lt} = \frac{d}{dt} \begin{bmatrix} I & -Q_{LL}^T \end{bmatrix} \begin{bmatrix} L_l & 0 \\ 0 & L_t \end{bmatrix} \begin{bmatrix} I \\ -Q_{LL} \end{bmatrix} i_{Ll}$$

$$+ \frac{d}{dt} \begin{bmatrix} I & -Q_{LL}^T \end{bmatrix} \begin{bmatrix} L_l & 0 \\ 0 & L_t \end{bmatrix} \begin{bmatrix} I \\ -Q_{LJ} \end{bmatrix} i_J$$

Or

$$v_{Ll} - Q_{LL}^T v_{Lt} = \frac{d}{dt}\mathcal{L}_{Ll} - \frac{d}{dt}\mathcal{L}' i_J \qquad (3.37)$$

where

$$\mathcal{L} = \begin{bmatrix} I & -Q_{LL}^T \end{bmatrix} \begin{bmatrix} L_l & 0 \\ 0 & L_t \end{bmatrix} \begin{bmatrix} I \\ -Q_{LL} \end{bmatrix} = L_l + Q_{LL}^T L_t Q_{LL}$$

and

$$\mathcal{L}' = -\begin{bmatrix} I & -Q_{LL}^T \end{bmatrix} \begin{bmatrix} L_l & 0 \\ 0 & L_t \end{bmatrix} \begin{bmatrix} 0 \\ -Q_{LJ} \end{bmatrix} = -Q_{LL}^T L_t Q_{LJ}$$

By substituting equation 3.37 back into equation 3.34 and rearranging

$$\frac{d}{dt}\mathcal{L} i_{Ll} = Q_{CL}^T v_{Ct} + Q_{RL}^T v_{Rt} + Q_{EL}^T v_E + \frac{d}{dt}\mathcal{L}' i_J \qquad (3.38)$$

The final task is to eliminate the variable v_{Rt} from equation 3.38. This is done by substituting

$$v_{Rt} = R_t i_{Rt} \qquad \text{(from equation 3.21)}$$

into equation 3.38 however i_{Rt} must first be eliminated in this expression. Substitution of equation 3.19c will give

$$v_{Rt} = -R_t Q_{RR} i_{Rl} - R_t Q_{RL} i_{Ll} - R_t Q_{RJ} i_J \qquad (3.39)$$

The variable i_{Rl} in equation 3.39 can be eliminated by substituting equation 3.29, the result is

$$v_{Rt} = -R_t Q_{RR} G_l Q_{ER}^T v_E - R_t Q_{RR} G_l Q_{CR}^T v_{Ct} - R_t Q_{RR} G_l Q_{RR}^T v_{Rt}$$

$$- R_t Q_{RL} i_{Ll} - R_t Q_{RJ} i_J$$

Also

$$v_{Rt} + R_t Q_{RR} G_l Q_{RR}^T v_{Rt} = -R_t Q_{RR} G_l Q_{ER}^T v_E - R_t Q_{RR} G_l Q_{CR}^T v_{Ct} - R_t Q_{RL} i_{Ll} - R_t Q_{RJ} i_J$$

or

$$\left[I + R_t Q_{RR} G_l Q_{RR}^T\right] v_{Rt} = -R_t Q_{RR} G_l Q_{ER}^T v_E - R_t Q_{RR} G_l Q_{CR}^T v_{Ct} \qquad (3.40)$$

$$- R_t Q_{RL} i_{Ll} - R_t Q_{RJ} i_J$$

Again three variables K_2, R, and G_l are defined as

$$K_2 = I + R_t Q_{RR} G_l Q_{RR}^T$$

$$G = G_t + Q_{RR} G_l Q_{RR}^T$$

and

$$R_t = G_t^{-1}$$

Therefore

$$K_2 = R_t G$$

where G is the node-parameter matrix [12], and like the loop-parameter matrix R, the G matrix is non-singular. Therefore, K_2^{-1} always exists. Equation 3.40 can now be written as

$$\left[R_t G\right] v_{Rt} = -R_t Q_{RR} G_l Q_{ER}^T v_E - R_t Q_{RR} G_l Q_{CR}^T v_{Ct} - R_t Q_{RL} i_{Ll} - R_t Q_{RJ} i_J$$

$$v_{Rt} = -G^{-1} R_t^{-1} R_t Q_{RR} G_l Q_{ER}^T v_E - G^{-1} R_t^{-1} R_t Q_{RR} G_l Q_{CR}^T v_{Ct}$$

$$- G^{-1} R_t^{-1} R_t Q_{RL} i_{Ll} - G^{-1} R_t^{-1} R_t Q_{RJ} i_J$$

Therefore

$$v_{Rt} = -G^{-1} Q_{RR} G_l Q_{ER}^T v_E - G^{-1} Q_{RR} G_l Q_{CR}^T v_{Ct} - G^{-1} Q_{RL} i_{Ll} - G^{-1} Q_{RJ} i_J \qquad (3.41)$$

Substituting equation 3.41 back into equation 3.38

$$\frac{d}{dt} \varphi_{Ll} = Q_{CL}^T v_{Ct} - Q_{RL}^T G^{-1} Q_{RR} G_l Q_{ER}^T v_E - Q_{RL}^T G^{-1} Q_{RR} G_l Q_{CR}^T v_{Ct}$$

$$- Q_{RL}^T G^{-1} Q_{RL} i_{Ll} - Q_{RL}^T G^{-1} Q_{RJ} i_J + Q_{EL}^T v_E + \frac{d}{dt} \mathcal{L}' i_J$$

$$\frac{d}{dt} \mathcal{L} i_{Ll} = \left[Q_{CL}^T - Q_{RL}^T G^{-1} Q_{RR} G_l Q_{CR}^T \right] v_{Ct} - \left[Q_{RL}^T G^{-1} Q_{RL} \right] i_{Ll}$$

$$+ \left[Q_{EL}^T - Q_{RL}^T G^{-1} Q_{RR} G_l Q_{ER}^T \right] v_E - \left[Q_{RL}^T G^{-1} Q_{RJ} \right] i_J + \frac{d}{dt} \mathcal{L}' i_J$$

In a more compact form

$$\frac{d}{dt} \mathcal{L} i_{Ll} = \mathcal{G} v_{Ct} - \mathcal{F} i_{Ll} + \mathcal{G}' v_E - \mathcal{F}' i_J + \frac{d}{dt} \mathcal{L}' i_J \qquad (3.42)$$

where

$$\mathcal{G} = Q_{CL}^T - Q_{RL}^T G^{-1} Q_{RR} G_l Q_{CR}^T$$

$$\mathcal{F} = Q_{RL}^T G^{-1} Q_{RL}$$

$$\mathcal{G}' = Q_{EL}^T - Q_{RL}^T G^{-1} Q_{RR} G_l Q_{ER}^T$$

$$\mathcal{F}' = Q_{RL}^T G^{-1} Q_{RJ}$$

Rearranging and rewriting equation 3.33 and equation 3.42 in matrix form, they both can be presented as

$$\begin{bmatrix} \mathcal{C} & 0 \\ 0 & \mathcal{L} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix} = \begin{bmatrix} -\mathcal{Y} & \mathcal{H} \\ \mathcal{G} & -\mathcal{F} \end{bmatrix} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix} + \begin{bmatrix} -\mathcal{Y}' & \mathcal{H}' \\ \mathcal{G}' & -\mathcal{F}' \end{bmatrix} \begin{bmatrix} v_E \\ i_J \end{bmatrix} + \begin{bmatrix} \mathcal{C}' & 0 \\ 0 & \mathcal{L}' \end{bmatrix} \frac{d}{dt} \begin{bmatrix} v_E \\ i_J \end{bmatrix} \qquad (3.43)$$

Since matrix $\mathcal{E}$ and matrix $\mathcal{L}$ are time-invariant diagonnal matrices, $\mathcal{E}^{-1}$ and $\mathcal{L}^{-1}$ exist. Equation 3.43 can thus be written as

$$\frac{dx}{dt} = \mathcal{A}x + \mathcal{B}_1 e + \mathcal{B}_2 \frac{de}{dt} \tag{3.44}$$

where

$$x = \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix} \quad , \quad e = \begin{bmatrix} v_E \\ i_J \end{bmatrix}$$

$$\mathcal{A} = \begin{bmatrix} \mathcal{E}^{-1} & 0 \\ 0 & \mathcal{L}^{-1} \end{bmatrix} \begin{bmatrix} -\mathcal{Y} & \mathcal{H} \\ \mathcal{G} & -\mathcal{F} \end{bmatrix}$$

$$\mathcal{B}_1 = \begin{bmatrix} \mathcal{E}^{-1} & 0 \\ 0 & \mathcal{L}^{-1} \end{bmatrix} \begin{bmatrix} -\mathcal{Y}' & \mathcal{H}' \\ \mathcal{G}' & -\mathcal{F}' \end{bmatrix}$$

and

$$\mathcal{B}_2 = \begin{bmatrix} \mathcal{E}^{-1} & 0 \\ 0 & \mathcal{L}^{-1} \end{bmatrix} \begin{bmatrix} \mathcal{E}' & 0 \\ 0 & \mathcal{L}' \end{bmatrix}$$

and the submatrices are

$$\mathcal{E} = C_t + Q_{CC} C_l Q_{CC}^T \quad , \quad \mathcal{E}' = -Q_{CC} C_l Q_{EC}^T$$

$$\mathcal{L} = L_l + Q_{LL}^T L_t Q_{LL} \quad , \quad \mathcal{L}' = Q_{LL}^T L_t Q_{LJ}$$

$$R = R_l + Q_{RR}^T R_t Q_{RR} \quad , \quad G = G_t + Q_{RR} G_l Q_{RR}^T$$

$$\mathcal{Y} = Q_{CR}R^{-1}Q_{CR}^{T} \qquad , \qquad \mathcal{Y}' = Q_{CR}R^{-1}Q_{ER}$$

$$\mathcal{H} = -Q_{CC} + Q_{CR}R^{-1}Q_{RR}^{T}R_{t}Q_{RC} \qquad , \qquad \mathcal{H}' = -Q_{CJ} + Q_{CR}R^{-1}Q_{RR}^{T}R_{t}Q_{RJ}$$

$$\mathcal{G} = Q_{CL}^{T} - Q_{RL}^{T}G^{-1}Q_{RR}G_{l}Q_{CR}^{T} \qquad , \qquad \mathcal{G}' = Q_{EL}^{T} - Q_{RL}^{T}G^{-1}Q_{RR}G_{l}Q_{ER}^{T}$$

$$\mathcal{F} = Q_{RL}^{T}G^{-1}Q_{RL} \qquad , \qquad \mathcal{F}' = Q_{RL}^{T}G^{-1}Q_{RJ}$$

The final equation 3.44 can be put in the form of equation 3.8, and by using the linear combination

$$x = x' + \mathcal{B}_{2}e \tag{3.45}$$

the output equation can be shown to be

$$w = \mathcal{C}x \tag{3.46}$$

where $\mathcal{C}$ is the identity matrix I, and x is the state variable. The $\mathcal{D}$ matrix seen in the general form (equation 3.9) by definition is zero. The next type of circuit considered by the program is the NCLIC circuit.

3.2.2 NCLIC Circuit

The second type of circuit which can be analyzed by the program is the NCLIC circuit. In the NCLIC circuit the Eo and Jo terms must be considered separately from the E and J partitioned matrix, therefore the expansion of equation 3.14 and equation 3.15 is

$$i_{Eo} = -Q_{EoC}i_{Cl} - Q_{EoR}i_{Rl} - Q_{EoL}i_{Ll} - Q_{EoJo}i_{Jo} - Q_{EoJ1}i_{J1} \quad (3.47a)$$

$$i_{E1} = -Q_{E1C}i_{Cl} - Q_{E1R}i_{Rl} - Q_{E1L}i_{Ll} - Q_{E1Jo}i_{Jo} - Q_{E1J1}i_{J1} \quad (3.47b)$$

$$i_{Ct} = -Q_{CC}i_{Cl} - Q_{CR}i_{Rl} - Q_{CL}i_{Ll} - Q_{CJo}i_{Jo} - Q_{CJ1}i_{J1} \quad (3.47c)$$

$$i_{Rt} = 0 - Q_{RR}i_{Rl} - Q_{RL}i_{Ll} - Q_{RJo}i_{Jo} - Q_{RJ1}i_{J1} \quad (3.47d)$$

$$i_{Lt} = 0 \quad 0 - Q_{LL}i_{Lt} - Q_{LJo}i_{Jo} - Q_{LJ1}i_{J1} \quad (3.47e)$$

and

$$v_{Cl} = Q^T_{EoC}v_{Eo} + Q^T_{E1C}v_{E1} + Q^T_{CC}v_{Ct} + 0 + 0 \quad (3.48a)$$

$$v_{Rl} = Q^T_{EoR}v_{Eo} + Q^T_{E1R}v_{E1} + Q^T_{CR}v_{Ct} + 0 + 0 \quad (3.48b)$$

$$v_{Ll} = Q^T_{EoL}v_{Eo} + Q^T_{E1L}v_{E1} + Q^T_{CL}v_{Ct} + Q^T_{RL}v_{Rt} + 0 \quad (3.48c)$$

$$v_{Jo} = Q^T_{EoJo}v_{Eo} + Q^T_{E1Jo}v_{E1} + Q^T_{CJo}v_{Ct} + Q^T_{RJo}v_{Rt} + Q^T_{LJo}v_{Lt} \quad (3.48d)$$

$$v_{J1} = Q^T_{EoJ1}v_{Eo} + Q^T_{E1J1}v_{E1} + Q^T_{CJ1}v_{Ct} + Q^T_{RJ1}v_{Rt} + Q^T_{LJ1}v_{Lt} \quad (3.48e)$$

The output solution of the desired waveforms i_{Eo} and v_{Jo}, can be solved using equations 3.47a and 3.48d. First, equation 3.47a can be rewritten as

$$i_{Eo} = -Q_{EoC} i_{Cl} - Q_{EoR} i_{Rl} - Q_{EoL} i_{Ll} - Q_{EoJo} i_{Jo} - Q_{EoJ1} i_{J1}$$

$$i_{Eo} = -Q_{EoC} i_{Cl} - Q_{EoR} i_{Rl} - Q_{EoL} i_{Ll} - Q_{EoJ} i_{J} \qquad (3.49)$$

The task now is to eliminate variables i_{Cl} and i_{Rl} from equation 3.49. This process is considered below. From equation 3.23

$$i_{Cl} = C_l \frac{d}{dt} v_{Cl}$$

Substituting v_{Cl} by equation 3.48a it can be shown that

$$i_{Cl} = C_l Q_{EoC} \frac{d}{dt} v_{Eo} + C_l Q_{EC}^T \frac{d}{dt} v_{E1} + C_l Q_{CC}^T \frac{d}{dt} v_{Ct}$$

This can be simplified since a NCLIC circuit by definition has no derivatives of the input source. Therefore

$$\frac{d}{dt} v_{Eo} = \frac{d}{dt} v_{E1} = 0$$

and

$$i_{Ct} = C_l Q_{CC}^T \frac{d}{dt} v_{Ct} \tag{3.50}$$

Substituting equation 3.50 and equation 3.32 into equation 3.49, the result can be shown to be

$$i_{Eo} = -Q_{EoC} C_l Q_{CC}^T \frac{d}{dt} v_{Ct} - Q_{EoR} R^{-1} Q_{ER}^T v_E - Q_{EoR} R^{-1} Q_{CR}^T v_{Ct}$$

$$+ Q_{EoR} R^{-1} Q_{RR}^T R_t Q_{RL} i_{Ll} + Q_{EoR} R^{-1} Q_{RR}^T R_t Q_{RJ} i_J - Q_{EoL} i_{Ll} - Q_{EoJ} i_J$$

or

$$i_{Eo} = -Q_{EoC} C_l Q_{CC}^T \frac{d}{dt} v_{Ct}$$

$$+ \begin{bmatrix} -Q_{EoR} R^{-1} Q_{CR}^T & Q_{EoR} R^{-1} Q_{RR}^T R_t Q_{RL} - Q_{EoL} \end{bmatrix} \begin{bmatrix} v_{Ct} \\ i_J \end{bmatrix}$$

$$+ \begin{bmatrix} -Q_{EoR} R^{-1} Q_{ER}^T & Q_{EoR} R^{-1} Q_{RR}^T R_t Q_{RJ} - Q_{EoJ} \end{bmatrix} \begin{bmatrix} v_E \\ i_J \end{bmatrix} \tag{3.51}$$

Similarly, equation 3.48d is rewritten as

$$v_{Jo} = Q_{EoJo}^T v_{Eo} + Q_{E1Jo}^T v_{E1} + Q_{CJo}^T v_{Ct} + Q_{RJo}^T v_{Rt} + Q_{LJo}^T v_{Lt}$$

or

$$v_{Jo} = Q_{EJo}^T v_E + Q_{CJo}^T v_{Ct} + Q_{RJo}^T v_{Rt} + Q_{LJo}^T v_{Lt} \tag{3.52}$$

The task now is to eliminate the variables v_{Lt} and v_{Rt}. From equation 3.22

$$v_{Lt} = L_t \frac{d}{dt} i_{Lt}$$

By replacing variable i_{Lt} with equation 3.47e, it can be shown that

$$v_{Lt} = -L_t Q_{LL} \frac{d}{dt} i_{Ll} - L_t Q_{LJo} \frac{d}{dt} i_{Jo} - L_t Q_{LJ1} \frac{d}{dt} i_{J1}$$

This can be simplified, since a NCLIC circuit, by definition can have no derivatives of the input sources, that is

$$\frac{d}{dt} i_{Jo} = \frac{d}{dt} i_{J1} = 0$$

Therefore

$$v_{Lt} = -L_t Q_{LL} \frac{d}{dt} i_{Ll} \qquad (3.53)$$

Substituting equation 3.53 and equations 3.41 into equation 3.52, the final result is

$$v_{Jo} = -Q_{LJo}^T L_t Q_{LL} \frac{d}{dt} i_{Ll} + Q_{EJo}^T v_E + Q_{CJo}^T v_{Ct} - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{ER}^T v_E$$

$$- Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{CR}^T v_{Ct} - Q_{RJo}^T G^{-1} Q_{RL} i_{Ll} - Q_{RJo}^T G^{-1} Q_{RJ} i_J$$

or

$$v_{Jo} = -Q_{LJo}^T L_t Q_{LL} \frac{d}{dt} i_{Ll}$$

$$+ \begin{bmatrix} Q_{CJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{CR}^T & -Q_{RJo}^T G^{-1} Q_{RL} \end{bmatrix} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix}$$

$$+ \begin{bmatrix} Q_{EJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{ER}^T & -Q_{RJo}^T G^{-1} Q_{RJ} \end{bmatrix} \begin{bmatrix} v_E \\ i_J \end{bmatrix} \qquad (3.54)$$

Finally, equation 3.51 and equation 3.54 can be combined as

$$\begin{bmatrix} i_{Eo} \\ v_{Jo} \end{bmatrix} = \begin{bmatrix} -Q_{EoC} C_l Q_{CC}^T & 0 \\ 0 & -Q_{LJo}^T L_t Q_{LL} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix}$$

$$+ \begin{bmatrix} -Q_{EoR} R^{-1} Q_{CR}^T & Q_{EoR} R^{-1} Q_{RR}^T R_t Q_{RL} - Q_{EoL} \\ Q_{CJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{CR}^T & -Q_{RJo}^T G^{-1} Q_{RL} \end{bmatrix} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix}$$

$$+ \begin{bmatrix} -Q_{EoR} R^{-1} Q_{ER}^T & Q_{EoR} R^{-1} Q_{RR}^T R_t Q_{RJ} - Q_{EoJ} \\ Q_{EJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{ER}^T & -Q_{RJo}^T G^{-1} Q_{RJ} \end{bmatrix} \begin{bmatrix} v_E \\ i_J \end{bmatrix} \qquad (3.55)$$

From the NCLIC circuit definition, equation 3.44 becomes

$$\frac{d}{dt}\begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix} = \mathcal{A}x + \mathcal{B}e$$

$$\frac{d}{dt}\begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix} = \mathcal{A}\begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix} + \mathcal{B}\begin{bmatrix} v_{E} \\ i_{J} \end{bmatrix} \qquad (3.56)$$

Substituting equation 3.56 into equation 3.55 will give an expression without derivatives, that is

$$\begin{bmatrix} i_{Eo} \\ v_{Jo} \end{bmatrix} = \begin{bmatrix} -Q_{EoC}C_{l}Q_{CC}^{T} & 0 \\ 0 & -Q_{LJo}^{T}L_{t}Q_{LL} \end{bmatrix} \mathcal{A} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix}$$

$$+ \begin{bmatrix} -Q_{EoC}C_{l}Q_{CC}^{T} & 0 \\ 0 & -Q_{LJo}^{T}L_{t}Q_{LL} \end{bmatrix} \mathcal{B} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix}$$

$$+ \begin{bmatrix} -Q_{EoR}R^{-1}Q_{CR}^{T} & Q_{EoR}R^{-1}Q_{RR}^{T}R_{t}Q_{RL} - Q_{EoL} \\ Q_{CJo}^{T} - Q_{RJo}^{T}G^{-1}Q_{RR}G_{l}Q_{CR}^{T} & -Q_{RJo}^{T}G^{-1}Q_{RL} \end{bmatrix} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix}$$

$$+ \begin{bmatrix} -Q_{EoR}R^{-1}Q_{ER}^{T} & Q_{EoR}R^{-1}Q_{RR}^{T}R_{t}Q_{RJ} - Q_{EoJ} \\ Q_{EJo}^{T} - Q_{RJo}^{T}G^{-1}Q_{RR}G_{l}Q_{ER}^{T} & -Q_{RJo}^{T}G^{-1}Q_{RJ} \end{bmatrix} \begin{bmatrix} v_{E} \\ i_{J} \end{bmatrix}$$

or

$$\begin{bmatrix} i_{Eo} \\ v_{Jo} \end{bmatrix} = \left\{ \begin{bmatrix} -Q_{EoC}C_{l}Q_{CC}^{T} & 0 \\ 0 & -Q_{LJo}^{T}L_{t}Q_{LL} \end{bmatrix} \mathcal{A} \right.$$

$$+\begin{bmatrix} -Q_{EoR}R^{-1}Q_{CR}^T & Q_{EoR}R^{-1}Q_{RR}^T R_t Q_{RL} - Q_{EoL} \\ Q_{CJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{CR}^T & -Q_{RJo}^T G^{-1} Q_{RL} \end{bmatrix}\Bigg\} \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix}$$

$$+\left\{\begin{bmatrix} -Q_{EoC}C_l Q_{CC}^T & 0 \\ 0 & -Q_{LJo}^T L_t Q_{LL} \end{bmatrix}\mathcal{B}\right.$$

$$+\begin{bmatrix} -Q_{EoR}R^{-1}Q_{ER}^T & Q_{EoR}R^{-1}Q_{RR}^T R_t Q_{RJ} - Q_{EoJ} \\ Q_{EJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{ER}^T & -Q_{RJo}^T G^{-1} Q_{RJ} \end{bmatrix}\Bigg\} \begin{bmatrix} v_E \\ i_J \end{bmatrix}$$

This can be written more compactly as

$$\begin{bmatrix} i_{Eo} \\ v_{Jo} \end{bmatrix} = \left\{\begin{bmatrix} \mathcal{M} & 0 \\ 0 & \mathcal{N} \end{bmatrix}\mathcal{A} + \begin{bmatrix} \mathcal{O} & \mathcal{P} \\ \mathcal{S} & \mathcal{U} \end{bmatrix}\right\}\begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix} + \left\{\begin{bmatrix} \mathcal{M} & 0 \\ 0 & \mathcal{N} \end{bmatrix}\mathcal{B} + \begin{bmatrix} \mathcal{V} & \mathcal{W} \\ \mathcal{X} & \mathcal{Z} \end{bmatrix}\right\}\begin{bmatrix} v_E \\ i_J \end{bmatrix} \qquad (3.57)$$

Where

$$\mathcal{M} = -Q_{EoC}C_t Q_{CC}^T$$

$$\mathcal{N} = -Q_{LJo}^T L_t Q_{LL}$$

$$\mathcal{O} = -Q_{EoR}R^{-1}Q_{CR}^T$$

$$\mathcal{P} = Q_{EoR}R^{-1}Q_{RR}^T R_t Q_{RL} - Q_{EoL}$$

$$\mathcal{S} = Q_{CJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{CR}^T$$

$$\mathcal{U} = -Q_{RJo}^T G^{-1} Q_{RL}$$

$$\mathcal{V} = -Q_{EoR}R^{-1}Q_{ER}^T$$

$$\mathcal{W} = Q_{EoR}R^{-1}Q_{RR}^T R_t Q_{RJ} - Q_{EoJ}$$

$$\mathcal{X} = Q_{EJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{ER}^T$$

$$\mathcal{Z} = -Q_{RJo}^T G^{-1} Q_{RJ}$$

In the general form equation 3.57 can be written as;

$$w' = \mathcal{C}x + \mathcal{D}e$$

where

$$w = \begin{bmatrix} i_{Eo} \\ v_{Jo} \end{bmatrix} \quad , \quad x = \begin{bmatrix} v_{Ct} \\ i_{Ll} \end{bmatrix} \quad , \quad e = \begin{bmatrix} v_E \\ i_J \end{bmatrix}$$

$$\mathcal{C} = \left\{ \begin{bmatrix} \mathcal{M} & 0 \\ 0 & \mathcal{N} \end{bmatrix} \mathcal{A} + \begin{bmatrix} 0 & \mathcal{P} \\ \mathcal{Y} & \mathcal{U} \end{bmatrix} \right\} \tag{3.58}$$

and

$$\mathcal{D} = \left\{ \begin{bmatrix} \mathcal{M} & 0 \\ 0 & \mathcal{N} \end{bmatrix} \mathcal{B}_1 + \begin{bmatrix} \mathcal{V} & \mathcal{W} \\ \mathcal{X} & \mathcal{Z} \end{bmatrix} \right\} \tag{3.59}$$

In the next section the discretized solution of these state equations are considered also, a NCLICS circuit example is worked out to summarize the procedure outlined in this section.

4.0 SOLUTION OF THE CIRCUIT EQUATIONS

The state equation is solved through the discretization of linear, time-invariant, continuous-time system technique described in reference [13], which is summarized here.

Assuming that over a sampling time ($\mathcal{T}$), all components of the input vector e are constant, the discretized expression for equation 3.8 and equation 3.10 is

$$\mathbf{x}\Big[(k+1)\mathcal{T}\Big] = e^{\mathcal{A}\mathcal{T}}\mathbf{x}(k\mathcal{T}) + \int_0^{\mathcal{T}} e^{\mathcal{A}\tau}\mathcal{B}\, d\tau\ \mathbf{e}(k\mathcal{T})$$

or

$$\mathbf{x}\Big[(k+1)\mathcal{T}\Big] = \mathcal{U}(\mathcal{T})\mathbf{x}(k\mathcal{T}) + \mathcal{B}(\mathcal{T})\mathbf{e}(k\mathcal{T}) \qquad (4.1)$$

where

$$k = 0, 1, 2, 3, \ldots$$

$$\mathcal{T} = \text{sampling time (seconds)}$$

$$\mathcal{U}(\mathcal{T}) = e^{\mathcal{A}\mathcal{T}} \qquad (4.2)$$

and

$$\mathcal{B}(\mathcal{T}) = \int_0^{\mathcal{T}} e^{\mathcal{A}\tau}\mathcal{B}\, d\tau \qquad (4.3)$$

Therefore, the $\mathcal{A}$ and $\mathcal{B}$ matrices of the continuous-time systems expressed by equation 3.8 and equation 3.10 can be discretized for a certain sampling time $\mathcal{T}$, giving the discretized matrices $\mathcal{U}(\mathcal{T})$ and $\mathcal{B}(\mathcal{T})$ respectively. The output equation for the corresponding discrete-time system for both the CLICS and NCLICS systems is

$$w(k\mathcal{T}) = \mathcal{C}x(k\mathcal{T}) + \mathcal{D}e(k\mathcal{T}) \tag{4.4}$$

The solution of equation 4.4, which represents the discrete time solution of the CLICS and NCLICS system, can be achieved if equation 4.1 is solved using equation 4.2 and equation 4.3. The method of solution selected for this program does not depend on the order of the matrices ($\mathcal{A}$ and $\mathcal{B}$) or on their eigenvalues. The fundamental power series is

$$e^{\mathcal{A}\mathcal{T}} = \sum_{k=0}^{\infty} \frac{(\mathcal{A}\mathcal{T})^k}{k!} \tag{4.5}$$

and

$$\int_0^{\mathcal{T}} e^{\mathcal{A}\tau}\, d\tau = \int_0^{\mathcal{T}} \sum_{k=0}^{\infty} \frac{(\mathcal{A}\tau)^k}{k!} = \sum_{k=0}^{\infty} \frac{\mathcal{A}^k \mathcal{T}^{k+1}}{(k+1)k!}$$

Therefore

$$\int_0^{\mathcal{T}} e^{\mathcal{A}\tau}\, d\tau = \sum_{k=0}^{\infty} \frac{(\mathcal{A}^k \mathcal{T}^k)\mathcal{T}}{(k+1)!} \qquad (4.6)$$

Reference [14] emphasized that the difficulty in the calculated error is the so called "catastrophic cancellation" in the floating point arithmetic and the truncation of arithmetic roundoff of the computer and not the truncation of the series. However, the HP9836 computer has a floating point card which uses a 64 bit floating number to represent a real number. One bit is used for the sign, 11 bits for the exponent, and 52 bits for the mantissa giving approximately 15 significant digits of accuracy. For most circuits the roundoff errors are not a problem. Furthermore, the number of terms used in the series (k) is 6. This number of terms gave good results in most examples tested and some of these are presented in chapter 6. Therefore, the $\mathcal{U}(\mathcal{T})$ matrix and the $\mathcal{B}(\mathcal{T})$ matrix can be evaluated approximately using equations 4.5 and 4.6 as

$$\mathcal{U}(\mathcal{T}) = I + \mathcal{T}\mathcal{A} + \frac{(\mathcal{T}\mathcal{A})^2}{2} + \frac{(\mathcal{T}\mathcal{A})^3}{6} + \frac{(\mathcal{T}\mathcal{A})^4}{24} + \frac{(\mathcal{T}\mathcal{A})^5}{120} + \frac{(\mathcal{T}\mathcal{A})^6}{720} \qquad (4.7)$$

and

$$\mathcal{B}(\mathcal{T}) = \left[I + \frac{\mathcal{T}\mathcal{A}}{2} + \frac{(\mathcal{T}\mathcal{A})^2}{6} + \frac{(\mathcal{T}\mathcal{A})^3}{24} + \frac{(\mathcal{T}\mathcal{A})^4}{120} + \frac{(\mathcal{T}\mathcal{A})^5}{720} + \frac{(\mathcal{T}\mathcal{A})^6}{5040} \right] \mathcal{T}\mathcal{B} \qquad (4.8)$$

This type of algorithm, which avoids use of the eigenvalues, has a problem if matrix $\mathcal{A}$ is large (usually due to small reactive element

values). Taking $\mathcal{A}$ to a power will tend to increase its terms causing numbers to be greater than $\pm 10^{308}$ or less than $\pm 10^{-308}$ which results in a real number overflow or a real number underflow error during execution time. This problem can be eliminated by selecting more points or less points either automatically through ASPPEC or manually by the user. As the number of points increases, $\mathcal{T}$ decreases. Therefore, the multiplication of $\mathcal{A}$ by $\mathcal{T}$ will decrease the element values thus eliminating the possibility of a real overflow error. However, this will tend to require considerably more computer time.

4.1 Summarized Procedure in Solving Circuit Equations

In order to understand the process ASPPEC uses to analyse a power electronic circuit, an NCLIC circuit example is presented in this section. The example used is a single phase isolated inverter shown in Figure 4.1. It consists of a dc-to-ac converter, an isolation

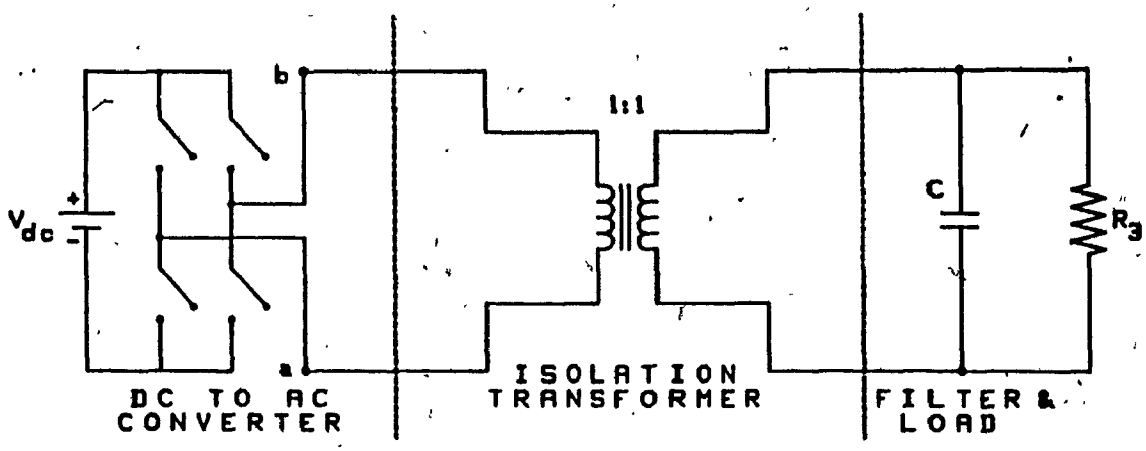


Figure 4.1. Single phase isolated inverter.

transformer, and the output filter and load section. This circuit is used often in power electronics equipment such as part of an auxilary power supply for a critical load - a load which cannot tolerate a power supply interruption. A good example of such a load is a computer system. Any interruption of the Hydro power supply, even for a few milliseconds, may cost time and money due to loss of information. The squarewave scheme is used in this example to illustrate the process, but more advanced schemes such as the harmonic injection PWM technique [15], could have been used to obtain better output spectra results. The objective of this example is to observe the peak current and voltage waveforms caused by the magnetizing inductance and the linkage inductance of the transformer. The waveforms are observed at the primary side of the transformer, at the secondary side of the transformer, and at the filter and load. The user can start the procedure by graphically

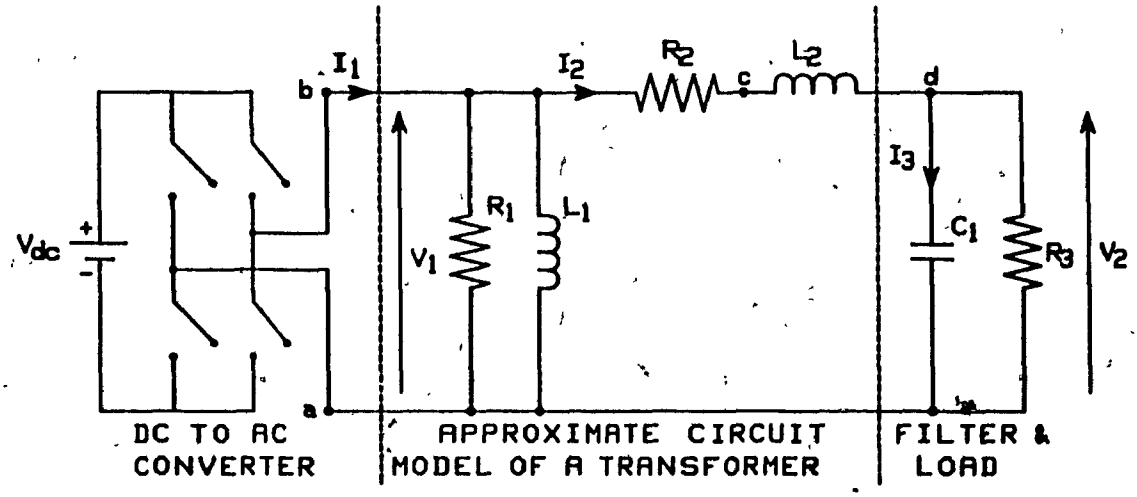


Figure 4.2. An inverter with an approximate circuit model of a transformer.

entering the schematic shown in Figure 4.2. The transformer, which is replaced by its equivalent circuit model, and the selected output voltage/current waveforms are shown in Figure 4.2. From the procedure outlined in chapter 3, the first step performed by ASPPEC is to replace the dc voltage source and the switching configuration by an equivalent time varying voltage source placed between nodes a and b as shown in Figure 4.3. Next, the program constructs an element table identifying

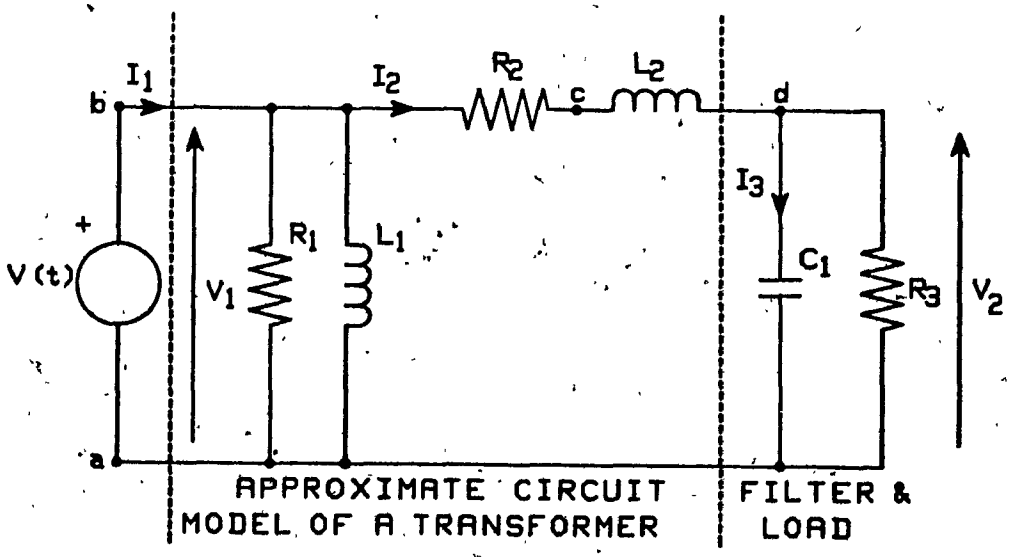


Figure 4.3. Transformed inverter circuit.

all elements in the network, the element values, and their coordinates. Then the circuit tree, which also includes the zero valued voltage sources and zero valued current sources, is inserted by the program for the output. The tree and current directions of each branch created by ASPPEC are illustrated in Figure 4.4. The primed nodes are created by

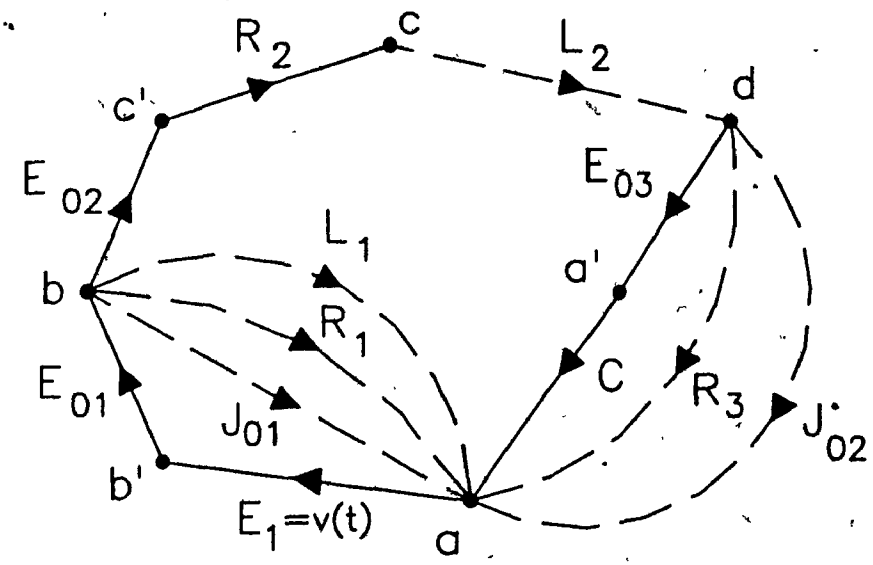


Figure 4.4. Example inverter circuit tree.

the program for the insertion of the zero valued voltage sources, where the E_0 terms represent the zero valued voltage sources, and the J_0 terms represent the zero valued current sources. From the element value table produced by the program for this example, the element values can be

taken as

$$R_1 = 2000\ \Omega$$

$$R_2 = 1\ \Omega$$

$$R_3 = 5\ \Omega$$

$$C = 5\ \mu F$$

$$L_1 = 1\ H$$

and

$$L_2 = 20\ \mu H$$

By using this information and the circuit tree information illustrated in Figure 4.4, the element matrices and the Q_t matrix can be constructed as

$$G_t = \begin{bmatrix} .0005 & 0 \\ 0 & .2 \end{bmatrix}$$

$$R_t = \begin{bmatrix} 2000 & 0 \\ 0 & 5 \end{bmatrix}$$

$$L_t = \begin{bmatrix} 1 & 0 \\ 0 & 20\times10^{-6} \end{bmatrix}$$

$$G_t = [1]$$

$$R_t = [1]$$

$$C_t = \left[5x10^{-6}\right]$$

and

$$Q_l = \begin{array}{c} \text{LINK} \rightarrow \\ \text{TWIG} \downarrow \end{array} \begin{array}{c} \\ E \\ \end{array} \left[\begin{array}{c} \\ E_o \\ \\ \\ E_1 \\ C \\ R_2 \end{array} \left[\begin{array}{c} E_{o1} \\ E_{o2} \\ E_{o3} \\ \\ \\ \\ \end{array}\right.\right.$$

	R1	R3	L1	L2	Jo & J: Jo1	Jo & J: Jo2
E_{o1}	-1	0	-1	-1	-1	0
E_{o2}	0	0	0	-1	0	0
E_{o3}	0	1	0	-1	0	1
E_1	-1	0	-1	-1	-1	0
C	0	1	0	-1	0	1
R_2	0	0	0	-1	0	0

Therefore the Q submatrices are

$$Q_{EoR} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q_{EoL} = \begin{bmatrix} -1 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, \quad Q_{EoJo} = Q_{EoR} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q_{E1R} = \begin{bmatrix} -1 & 0 \end{bmatrix}, \quad Q_{E1R} = \begin{bmatrix} -1 & -1 \end{bmatrix}, \quad Q_{E1Jo} = Q_{E1J} = \begin{bmatrix} -1 & 0 \end{bmatrix}$$

$$Q_{ER} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad Q_{EL} = \begin{bmatrix} -1 & -1 \\ 0 & -1 \\ 0 & -1 \\ -1 & -1 \end{bmatrix}, \quad Q_{EJo} = Q_{EJ} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$Q_{CR} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad Q_{CL} = \begin{bmatrix} 0 & -1 \end{bmatrix}, \quad Q_{CJo} = Q_{CJ} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

and

$$Q_{RR} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \quad Q_{RL} = \begin{bmatrix} 0 & -1 \end{bmatrix}, \quad Q_{RJo} = Q_{RJ} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Furthermore

$$\mathcal{E} = C_t + Q_{CC} C_l Q_{CC}^T = \begin{bmatrix} 5\times10^{-6} \end{bmatrix}$$

$$\mathcal{E}' = -Q_{CC} C_l Q_{EC}^T = \begin{bmatrix} 0 \end{bmatrix}$$

$$R = R_l + Q_{RR}^T R_t Q_{RR} = \begin{bmatrix} 2000 & 0 \\ 0 & 5 \end{bmatrix}$$

$$\mathcal{L} = L_t + Q_{LL}^T L_t Q_{LL} = \begin{bmatrix} 1 & 0 \\ 0 & 20\times10^{-6} \end{bmatrix}$$

$$\mathcal{L}' = -Q_{LL}^T L_t Q_{LJ} = \begin{bmatrix} 0 \end{bmatrix}$$

$$G = G_t + R_t Q_{RR} G_l Q_{RR}^T = \begin{bmatrix} 1 \end{bmatrix}$$

$$\mathcal{Y} = Q_{CR} R^{-1} Q_{CR}^T = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .0005 & 0 \\ 0 & .2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} .2 \end{bmatrix}$$

$$\mathcal{Y}' = Q_{CR} R^{-1} Q_{ER}^T = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .0005 & 0 \\ 0 & .2 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & .2 & 0 \end{bmatrix}$$

$$\mathcal{H} = -Q_{CL} + Q_{CR} R^{-1} Q_{RR}^T R_t Q_{RL} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\mathcal{H}' = -Q_{CJ} + Q_{CR} R^{-1} Q_{RR}^T R_t Q_{RJ} = \begin{bmatrix} 0 & -1 \end{bmatrix}$$

$$\mathcal{G} = Q_{CL}^T - Q_{RL}^T G^{-1} Q_{RR} G_l Q_{CR}^T = -\mathcal{H}^T = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$\mathcal{G}' = Q_{EL}^T - Q_{RL}^T G^{-1} Q_{RR} G_l Q_{ER}^T = \begin{bmatrix} -1 & 0 & 0 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix}$$

$$\mathcal{F} = Q_{RL}^T G^{-1} Q_{RL} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$\mathcal{F}' = Q_{RL}^T G^{-1} Q_{RJ} = \begin{bmatrix} 0 \end{bmatrix}$$

Therefore one can construct

$$\mathcal{A} = \begin{bmatrix} \mathcal{E}^{-1} & 0 \\ 0 & \mathcal{L}^{-1} \end{bmatrix} \begin{bmatrix} -\mathcal{Y} & \mathcal{H} \\ \mathcal{G} & -\mathcal{F} \end{bmatrix} = \begin{bmatrix} 2x10^5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 50000 \end{bmatrix} \begin{bmatrix} -.2 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

$$\mathcal{A} = \begin{bmatrix} -40000 & 0 & 2x10^5 \\ 0 & 0 & 0 \\ -50000 & 0 & -50000 \end{bmatrix} \qquad (4.9)$$

$$\mathcal{B} = \begin{bmatrix} \mathcal{E}^{-1} & 0 \\ 0 & \mathcal{L}^{-1} \end{bmatrix} \begin{bmatrix} -\mathcal{Y}' & \mathcal{H}' \\ \mathcal{G}' & -\mathcal{F}' \end{bmatrix} = \begin{bmatrix} 2x10^5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 50000 \end{bmatrix} \begin{bmatrix} 0 & 0 & -.2 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & -1 & -1 & -1 & 0 & 0 \end{bmatrix}$$

and

$$\mathcal{B} = \begin{bmatrix} 0 & 0 & -4x10^4 & 0 & 0 & 0 & -2x10^5 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -5x10^4 & 0 & -5x10^4 & -5x10^4 & -5x10^4 & 0 & 0 \end{bmatrix} \qquad (4.10)$$

The submatrices of the output equation 3.57 are necessary to construct and finally evaluate the output, therefore

$$M = -Q_{EoC}^{T} C Q_{CC} = \begin{bmatrix} 0 \end{bmatrix}$$

$$N = -Q_{LJo}^{T} L_t Q_{LL} = \begin{bmatrix} 0 \end{bmatrix}$$

$$O = -Q_{EoR}R^{-1}Q_{CR}^{T} = \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}.0005 & 0\\ 0 & .2\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = \begin{bmatrix}0\\ 0\\ -.2\end{bmatrix}$$

$$\mathcal{P} = Q_{EoR}R^{-1}Q_{RR}^{T}R_{t}Q_{RL} - Q_{EoL} = \begin{bmatrix}1 & 1\\ 0 & 1\\ 0 & 1\end{bmatrix}$$

$$\mathcal{Y} = Q_{CJo}^{T} - Q_{RJo}^{T}G^{-1}Q_{RR}G_{t}Q_{CR}^{T} = \begin{bmatrix}0\\ 1\end{bmatrix}$$

$$\mathcal{U} = -Q_{RJo}^{T}G^{-1}Q_{RL} = \begin{bmatrix}0\end{bmatrix}$$

$$V = -Q_{EoR}R^{-1}Q_{ER}^{T} = \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}.0005 & 0\\ 0 & .2\end{bmatrix}\begin{bmatrix}-1 & 0 & 0 & -1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

or

$$V = \begin{bmatrix}-.0005 & 0 & 0 & -.0005\\ 0 & 0 & 0 & 0\\ 0 & 0 & -.2 & 0\end{bmatrix}$$

$$W = Q_{EoR}R^{-1}Q_{RR}^{T}R_{t}Q_{RJ} - Q_{EoJ} = \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & -1\end{bmatrix}$$

$$\mathcal{X} = Q_{EJo}^T - Q_{RJo}^T G^{-1} Q_{RR} G_l Q_{ER}^T = \begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$\mathcal{Z} = -Q_{RJo} G^{-1} Q_{RJ} = \begin{bmatrix} 0 \end{bmatrix}$$

The task now is to discretize the State Space equations. Using the procedure outlined previously in this chapter

$$\mathcal{T} = 4.16667 \times 10^{-6}$$

$$\mathcal{U}(\mathcal{T}) = I + \mathcal{TA} + \frac{(\mathcal{TA})^2}{2} + \frac{(\mathcal{TA})^3}{6} + \frac{(\mathcal{TA})^4}{24} + \frac{(\mathcal{TA})^5}{120} + \frac{(\mathcal{TA})^6}{720}$$

$$\mathcal{U}(\mathcal{T}) = \begin{bmatrix} .77051780 & 0 & .671087841 \\ 0 & 1 & 0 \\ -.167771960 & 0 & .741497388 \end{bmatrix}$$

$$\mathcal{B}(\mathcal{T}) = \left[I + \frac{\mathcal{TA}}{2} + \frac{(\mathcal{TA})^2}{6} + \frac{(\mathcal{TA})^3}{24} + \frac{(\mathcal{TA})^4}{120} + \frac{(\mathcal{TA})^5}{720} + \frac{(\mathcal{TA})^6}{5040} \right] \mathcal{TB}$$

and

$$\mathcal{B}(\mathcal{T}) = \begin{bmatrix} -.75084525 & -.07560845 & -.224948034 & -.075608452 & 0 & -.746697908 \\ -4.16667 \times 10^{-6} & 0 & 0 & 0 & 0 & 0. \\ -.18289405 & -.18289405 & -.167772363 & -.182894054 & 0 & -.075608452 \end{bmatrix}$$

From equation 3.58 and equation 3.59, we have

$$\mathcal{C} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ -.2 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.11}$$

$$\mathcal{D} = \begin{bmatrix} -.0005 & 0 & 0 & -.0005 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -.2 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.12}$$

Finally, after constructing the output state equation 3.11, where the $\mathcal{C}$ and $\mathcal{D}$ matrices are given by equations 4.11 and 4.12, the example was simulated by ASPPEC and the output results agreed with the results presented in this section. The output results were plotted and are shown in Figure 4.5. Using the features available in the plotting routine, the current I_1 was observed more closely. The x-axis is expanded and the x-offset function is used to observe the transient current I_1 (Figure 4.6a), the peak positive current of I_1 on the second positive pulse (Figure 4.6b), and the peak negative current of I_1 on the second negative pulse (Figure 4.6c). Neither the accuracy of the algorithm in chapter 3 nor the error associated with the solution in this

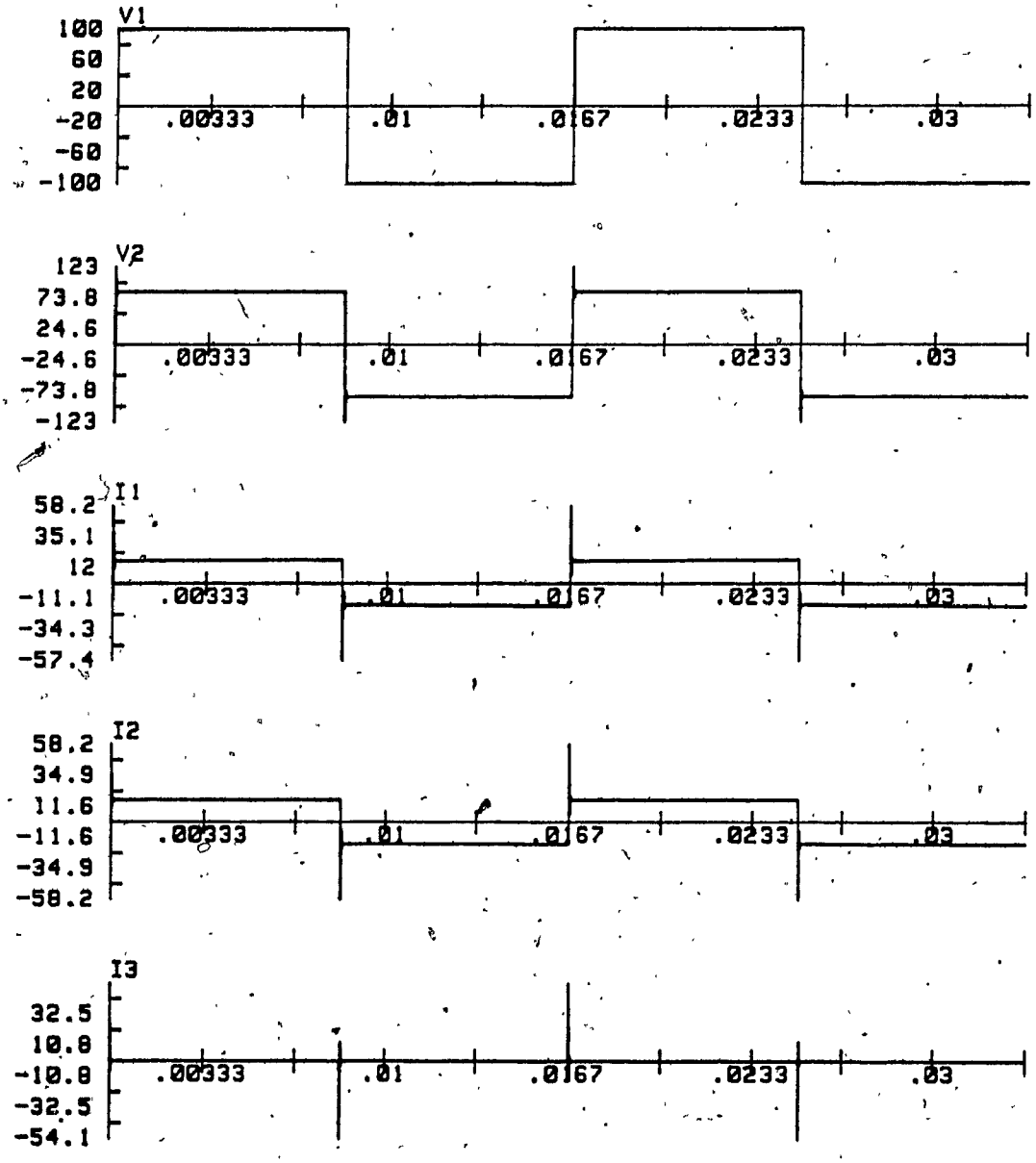


Figure 4.5. Inverter waveforms.

chapter has been investigated. Therefore, in order to verify the validity of the methods considered previously, examples are considered

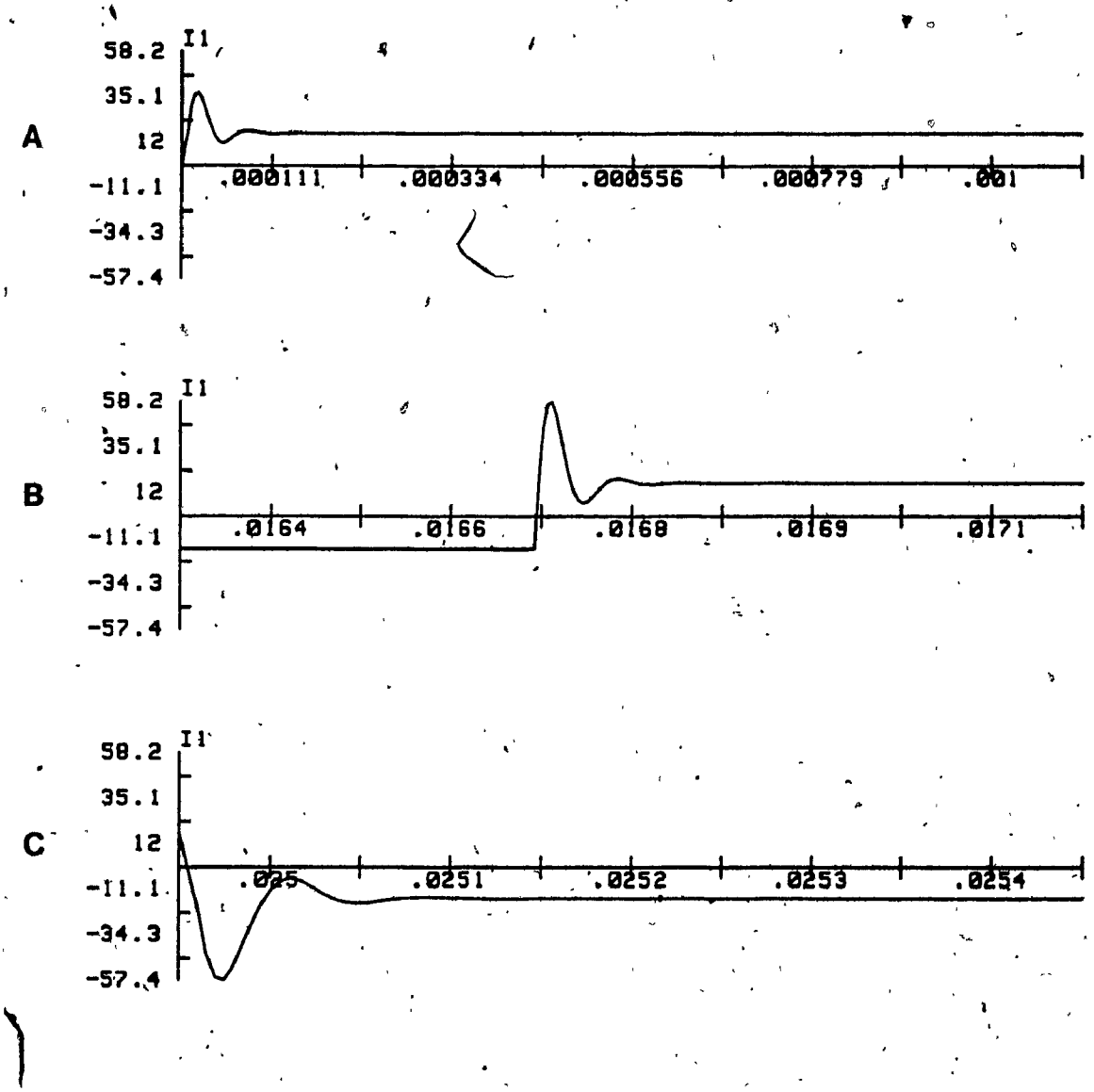


Figure 4.6. Current I1 spikes are expanded.

in chapter 5 and verified in chapter 6. Chapter 5 develops a solution based on the methods of chapters 3 and 4, while chapter 6 provides a solution based on results obtained from an actual circuit. Comparison of the solutions derived in these two chapters demonstrates the effectiveness of the program.

5.0 SIMULATED RESULTS AND EVALUATION

In order to further illustrate the usefulness and capabilities of the simulation program, six representative examples are discussed in this section. These examples involve two of the most frequently used types of static converters rectifiers and inverters. Three examples using two 3-phase rectifier circuits controlled by two different switch configuration schemes are considered as well as three 3-phase voltage source inverters. To begin, two 3-phase rectifier circuits Figure 5.1 and Figure 5.3 are simulated by ASPPEC.

5.1 3-Phase Rectifier

Figure 2.1 illustrates the actual format used to enter the circuit of a 3-phase line commutated rectifier into the ASPPEC program. Component values and load conditions have been entered interactively. The resulting output voltage and current waveforms obtained from the simulation of the circuit shown in Figure 5.1 are shown in Figure 5.2a, b, c, and d. The output voltage V_1, dc line current I_1, load voltage V_2, and capacitor current I_2 are shown in Figure 5.2a, b, c, and d respectively for rated load conditions at control angle $\alpha = 0$. These types of rectifiers are used extensively in industry for applications such as dc motor drives for the pulp and paper industry. An attempt was made to compare the output results of Figure 5.1 using PSpice but one quickly realizes that PSpice is unable to analyze a voltage loop (a capacitor loop made up entirely of voltage sources). Therefore, modifications of the circuit were first needed to observe the results

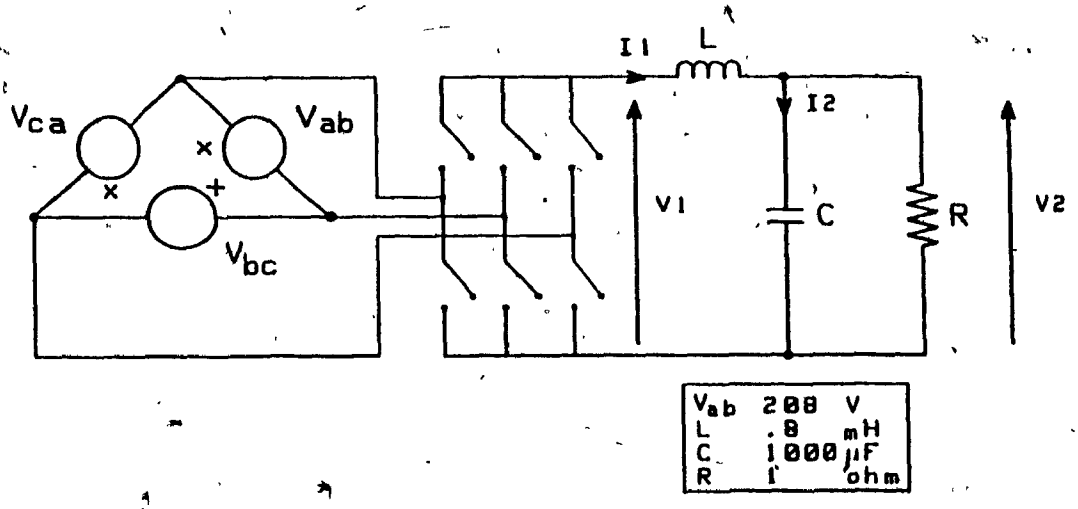
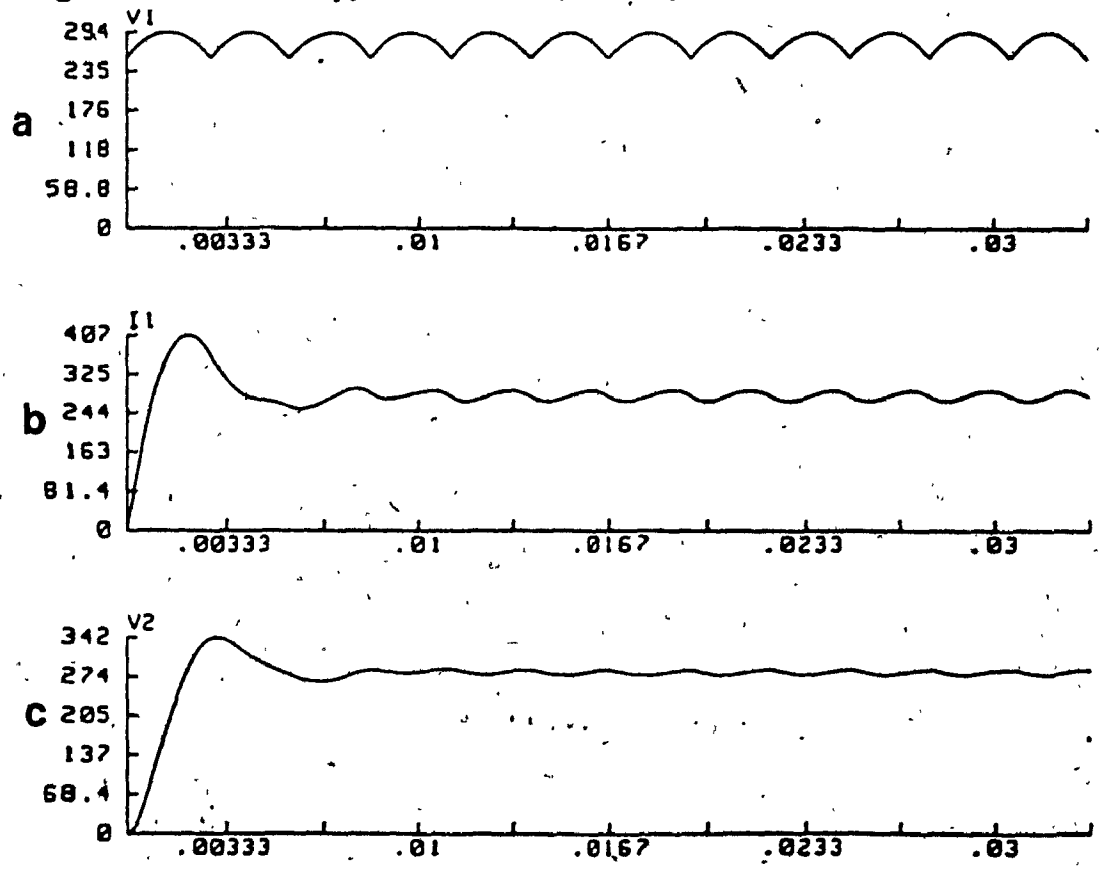


Figure 5.1. Hardcopy of the analysed 3-phase rectifier circuit



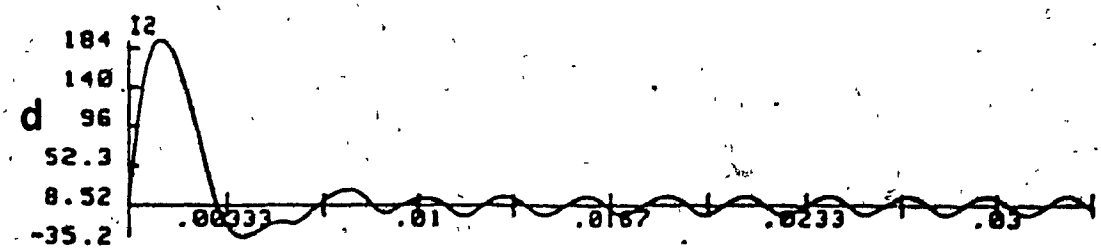


Figure 5.2. Simulated 3-phase line commutated rectifier (Figure 5.1) waveforms.

presented by PSpice. The input data file and output waveforms of the equivalent 3-phase rectifier shown in Figure 5.1 as analyzed by PSpice are presented in Appendix 3. The output results shown in Appendix 3 are similar to the results given by ASPPEC (shown in Figure 5.2) except that the resolution of the PSpice output waveforms are poorer. This is so even though the transient analysis internal step size was selected to be .01μsec. The time step used in the input data file may not correspond to the printer or CRT time steps. These values (printer and CRT display time steps) are obtained by a 2^{nd} order polynomial interpolation [1]. Although the waveform resolutions from PSpice for both the printer and CRT output is poor, run-time compared to ASPPEC is approximately the same. This PSpice example is shown to emphasize the problem PSpice has in simulating power electronic circuits which very often contain degeneracies. The appropriate circuit modifications needed for PSpice analysis implies that a certain background knowledge is needed thereby reducing the desirability of using PSpice as a self-teaching, self-learning tool. To further illustrate the usefulness of the ASPPEC program, Figure 5.3 illustrates the circuit of Figure 5.1 with twice the number of filter elements. The output voltage V_1, dc line current I_1, load voltage V_2, and the second capacitor current I_2 for this rectifier are shown in Figures 5.4a, b, c, and d respectively. Such circuit configurations are frequently used in applications with very strict voltage ripple specifications. A different approach, which allows the reduction of the rectifier output ripple voltage (V_2) without compromising rectifier circuit size, is illustrated in Figure 5.5. In this case the line commutated switches (i.e. thyristors used in Figures 5.1 and 5.3) are replaced by the more advanced gate-commutated type of

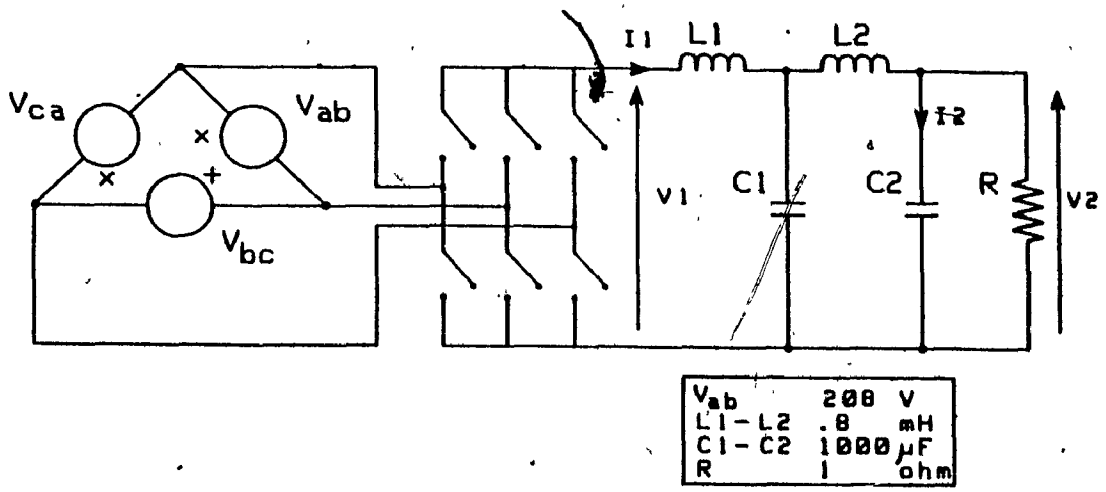


Figure 5.3. Simulated 3-phase rectifier with a fourth order filter

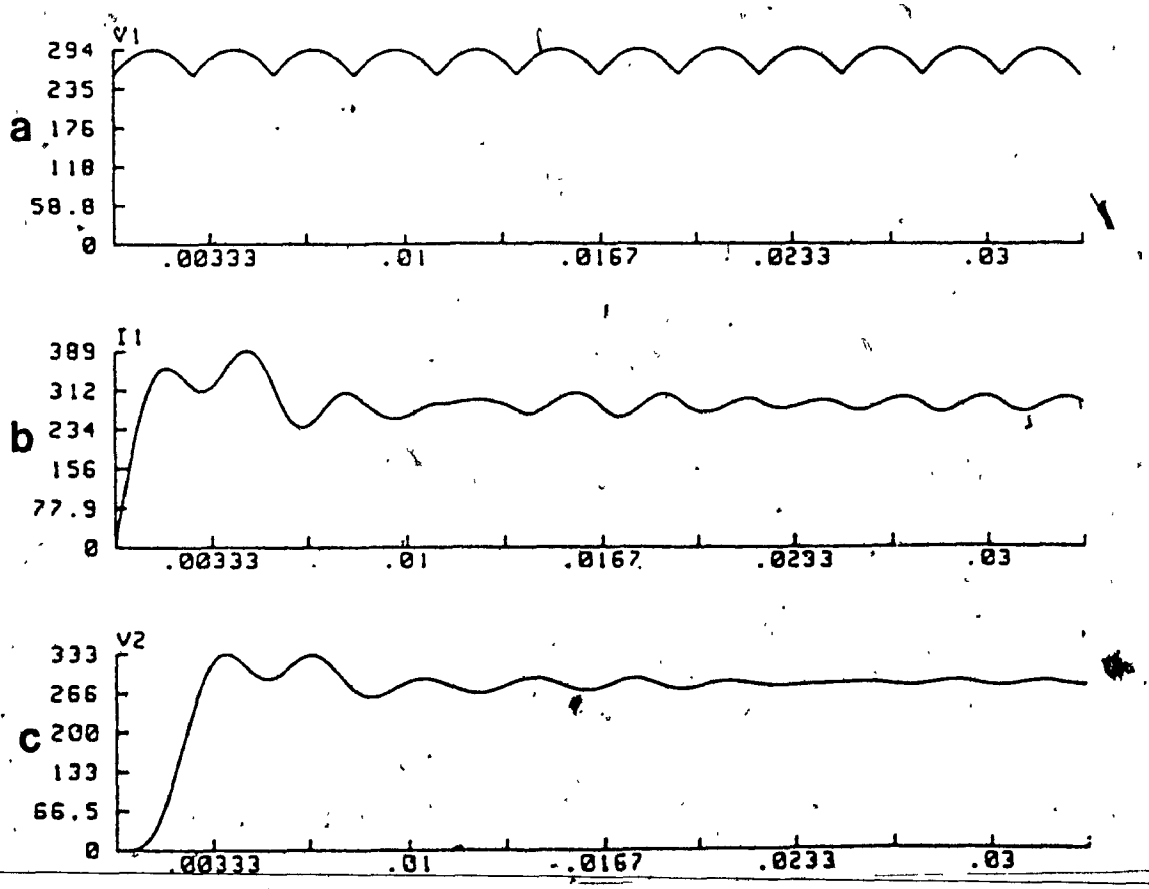


Figure 5.4. Simulated 3-phase line commutated rectifier (Figure 5.3) waveforms.

switches (such as power transistors). Also, by adopting previously optimized pulse width modulation harmonic reduction techniques [15] [18] [20].

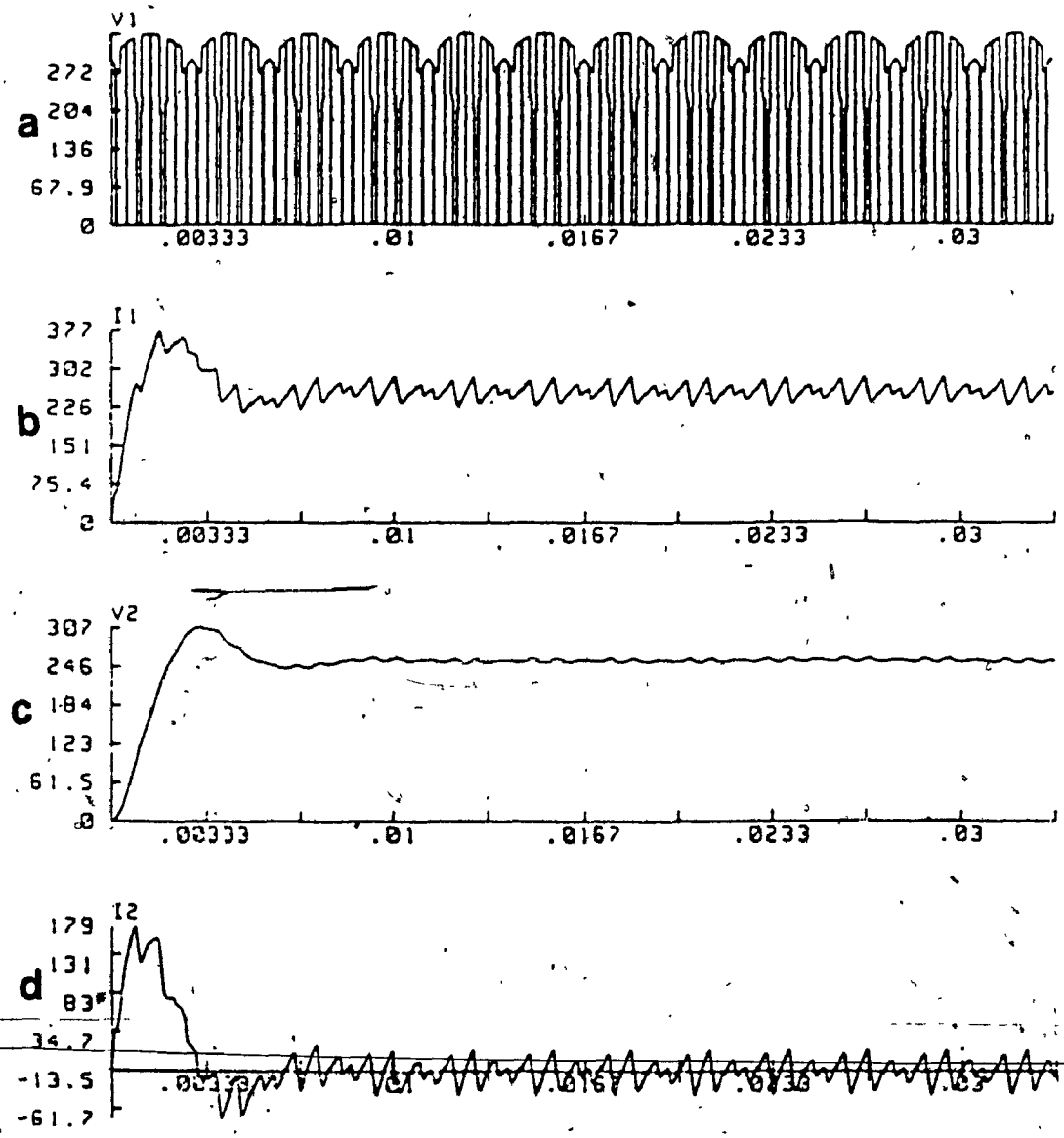


Figure 5.5. Simulated 3-phase PWM rectifier (circuit of Figure 5.1) waveforms.

the (simulated) output current and voltage waveforms shown in Figures 5.5a,b,c and d are obtained. Moreover, Figure 5.5b shows that the resulting transient peak output current with the PWM harmonic control (with modulation 1) is smaller than the respective current obtained with the phase control technique (Figure 5.4b). In the next section another type of power amplifier, the inverter, that can be easily treated with the proposed simulation package is considered.

5.2 3-Phase Voltage Source Inverter

The inverter amplifier, shown in Figure 5.6, is used extensively with ac motors in ac drive applications. These applications can be as simple as speed controlled fans and pumps or as sophisticated as high precision multi-axes ac servodrives. It is also used in applications requiring uninterruptible high quality power supplies (UPS), such as digital computers, life supporting electronic devices, and other types of critical equipment. Simulation results depicting current and voltage waveforms generated by this amplifier (inverter) under balanced and unbalanced load conditions are shown in Figures 5.7a and b, and Figures 5.9a, b, c, and d respectively. Additional information can be obtained from Figure 5.8 depicting the same circuit under fault conditions (i.e. output line-to-line short circuit). Another inverter amplifier using a new type of PWM scheme [21] has its circuit, and output voltage/current waveforms shown in Figure 5.10 and 5.11 respectively. The high frequency sharp current ripple of the simulated current waveform of Figure 5.11 is a difficult waveform to obtain quickly using PSpice since, PSpice uses a variable time step size. In the next section experimental results of the inverter output voltage and current waveforms from the circuit of

Figures 5.6 and 5.10 [21] are shown.

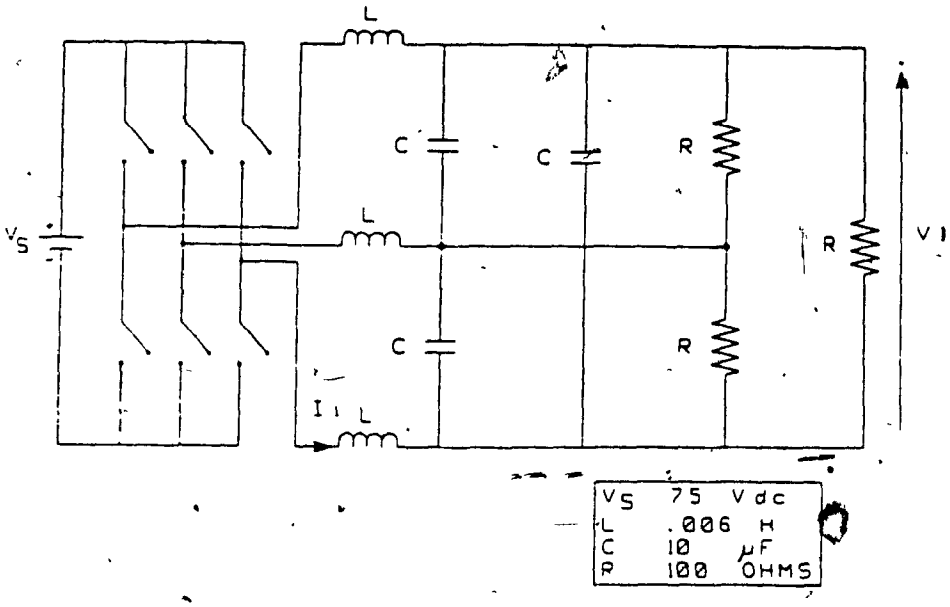


Figure 5.6. Hard-copy of a 3-phase PWM voltage source inverter circuit.

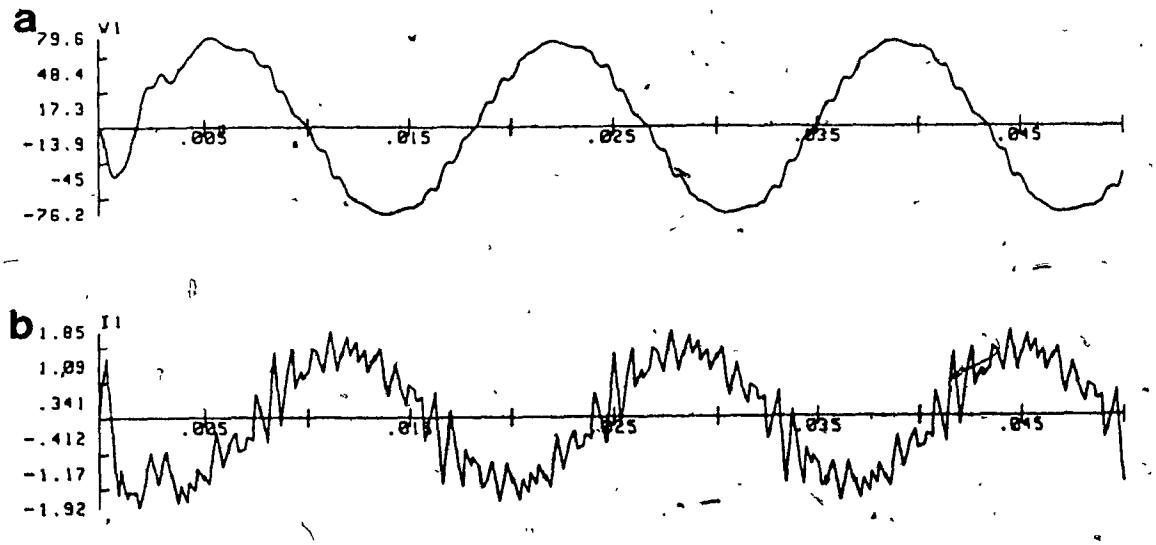


Figure 5.7. Inverter output voltage and current waveforms for the circuit of Figure 5.6.

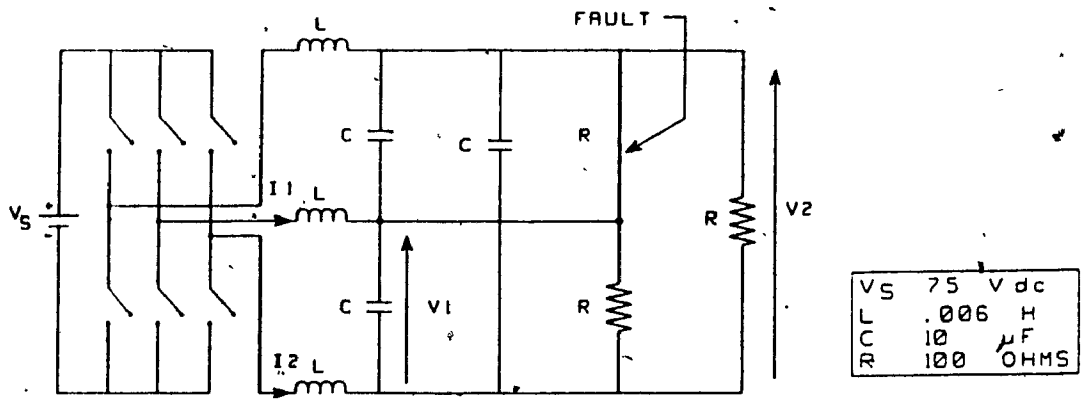


Figure 5.8. Hard-copy of the 3-phase PWM voltage source inverter with a short-circuit fault.

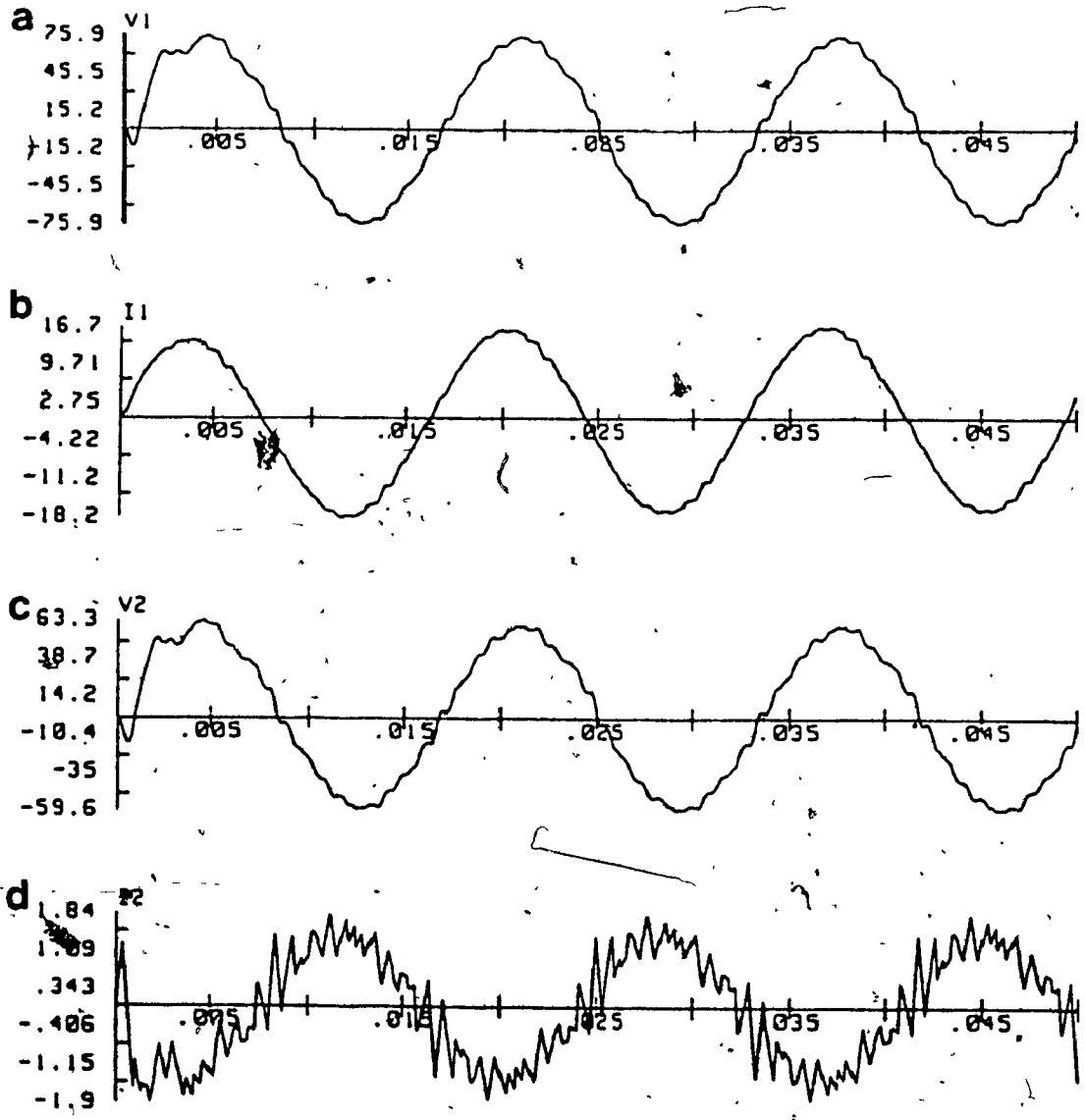


Figure 5.9. Inverter output waveforms for the circuit of Figure 5.8 under fault condition.

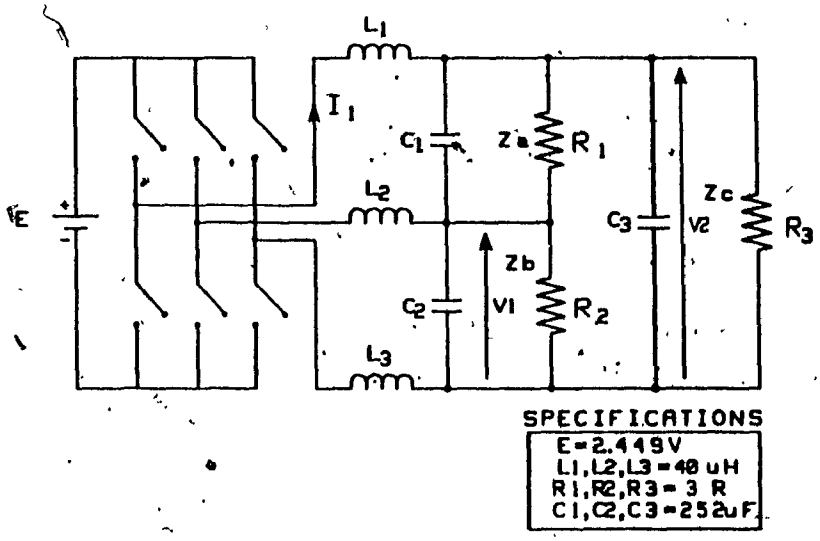


Figure 5.10. Simulated test circuit of reference [21].

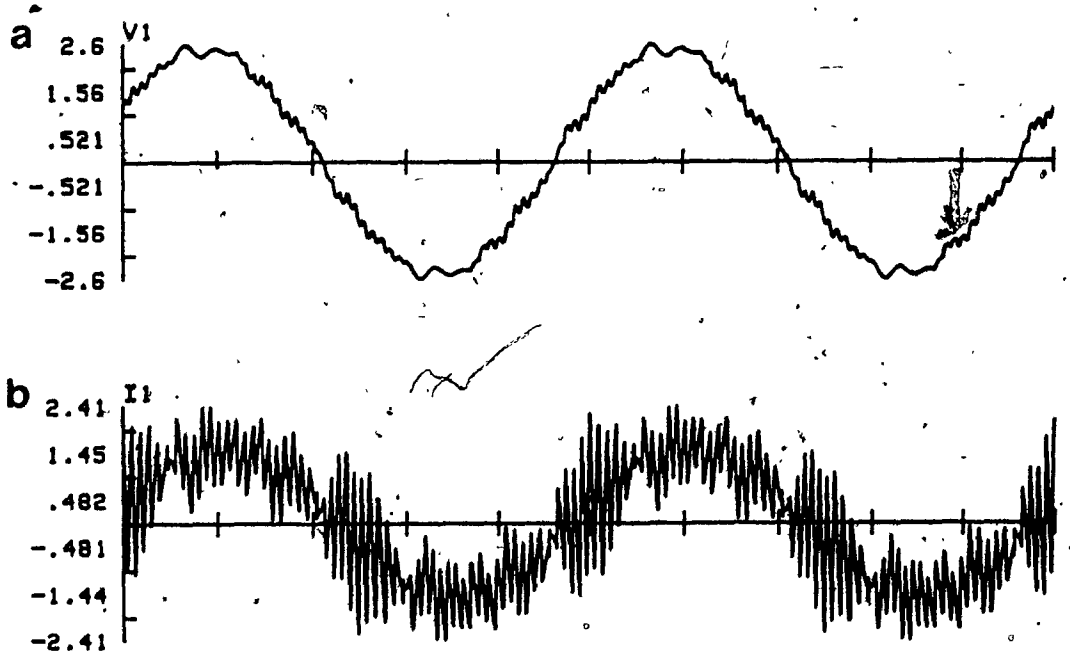


Figure 5.11. Reference [21] output voltage and current waveforms.

6.0 EXPERIMENTAL RESULTS

In order to establish the effectiveness of the proposed simulation program in predicting actual converter voltage/current waveforms, a sample of predicted results were verified experimentally on a labbratory prototype inverter. The first circuit implemented is the 3-phase voltage source inverter shown in Figure 5.6, using the well know sinusoidal PWM technique [22]. The second example utilizes the harmonic injection PWM scheme [21] which is a newly optimized fixed pattern switching scheme.

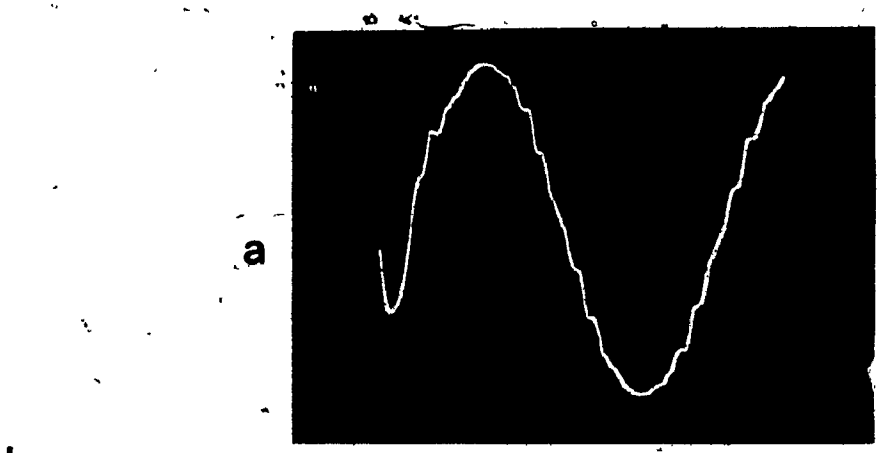
6.1 3-Phase Voltage Source Inverter Using Sinusoidal PWM

The experimental results of the 3-phase voltage source inverter circuit illustrated in Figure 5.6 are shown in Figures 6.1 and 6.2. Specifically, Figures 6.1a and c show the actual waveforms of the inverter output load voltage V_1 and line current I_1 under transient (start-up) conditions. The same waveforms under steady state conditions are shown in Figures 6.2a and b. Comparison between these experimental waveforms and their predicted counterparts, shown in Figures 5.7a and b which are reshown in Figure 6.1b and d, show that they are in close agreement. Also, to further illustrate the flexibility of the proposed program these experimental waveforms are read into the computer (through the use of an appropriate data acquisition device) and their frequency spectra were obtained through further waveform analysis. These spectra are shown at the bottom of Figures 6.2a and b. This input/output waveform feature is important since customized PWM waveforms can be

quickly produced by external waveform generators such as the Hewlett-Packard 8340A signal generator. These hardware implemented voltage/current waveforms which can be down-loaded into a file and then called by ASPPEC. This facilitates the process of manufacturing complex source waveforms or switching configuration waveforms. To illustrate the power of this feature the customized PWM scheme of reference [21] is used in the next example.

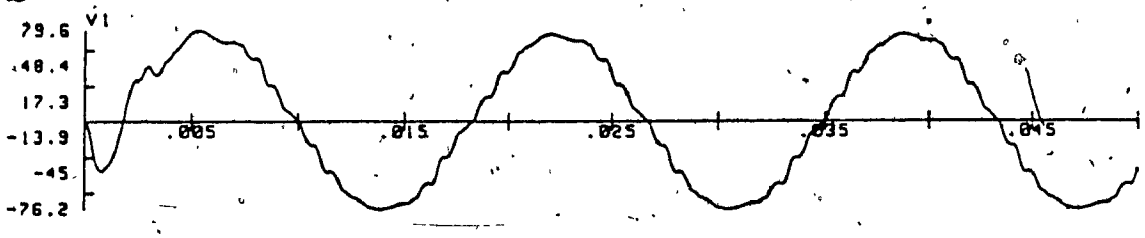
6.2 3-Phase Voltage Source Inverter Using Harmonic Injection PWM

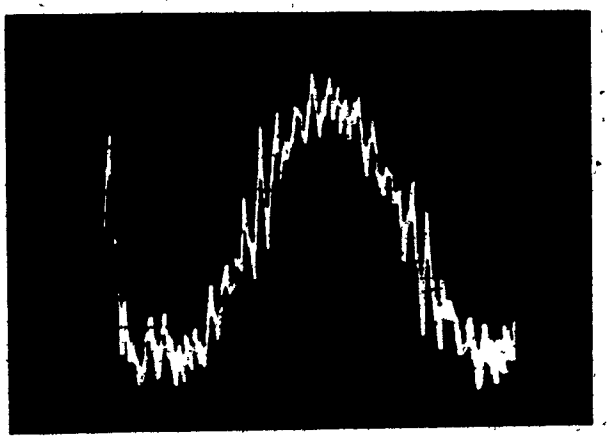
Utilizing the customized PWM produced by the authors of reference [21], the experimental steady state line-to-line voltage and line current of the circuit illustrated in Figure 5.10 are shown in Figures 6.3a and b respectively. These waveforms agree with their respective simulated waveforms reshown in Figures 6.3c and d. Furthermore, the observations deducted from the simulated results such as the voltage total harmonic distortion and peak line current agree with the experimental results. This example was selected and presented for its low line inductance. Because of this low inductance, the output line current has many sharp current spikes. Such an output waveform would take hours to solve using PSPICE while ASPPEC took just a few minutes.



a

b





c

d

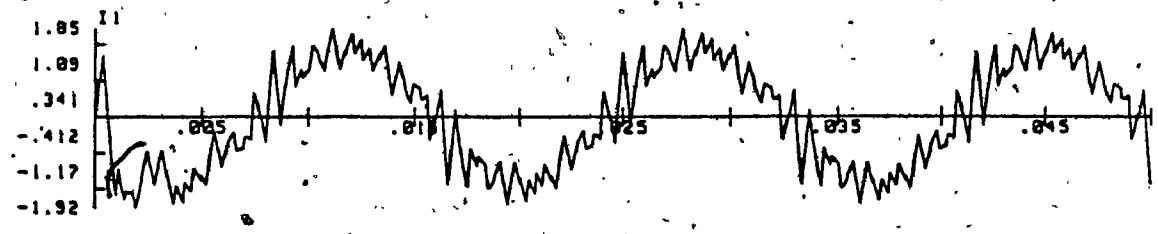


Figure 6.1. Experimental transient inverter waveforms and their respective simulated results.

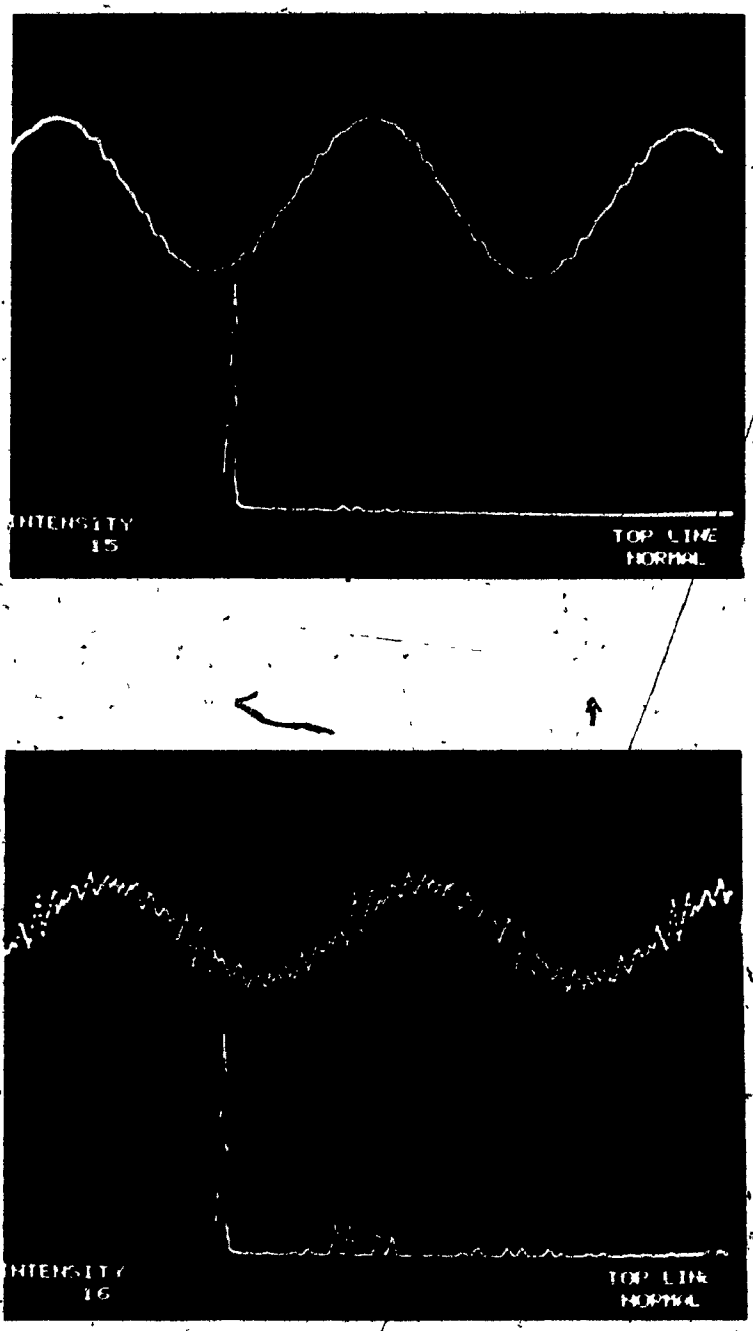


Figure 6.2. Experimental steady state inverter waveforms and their respective frequency spectra.

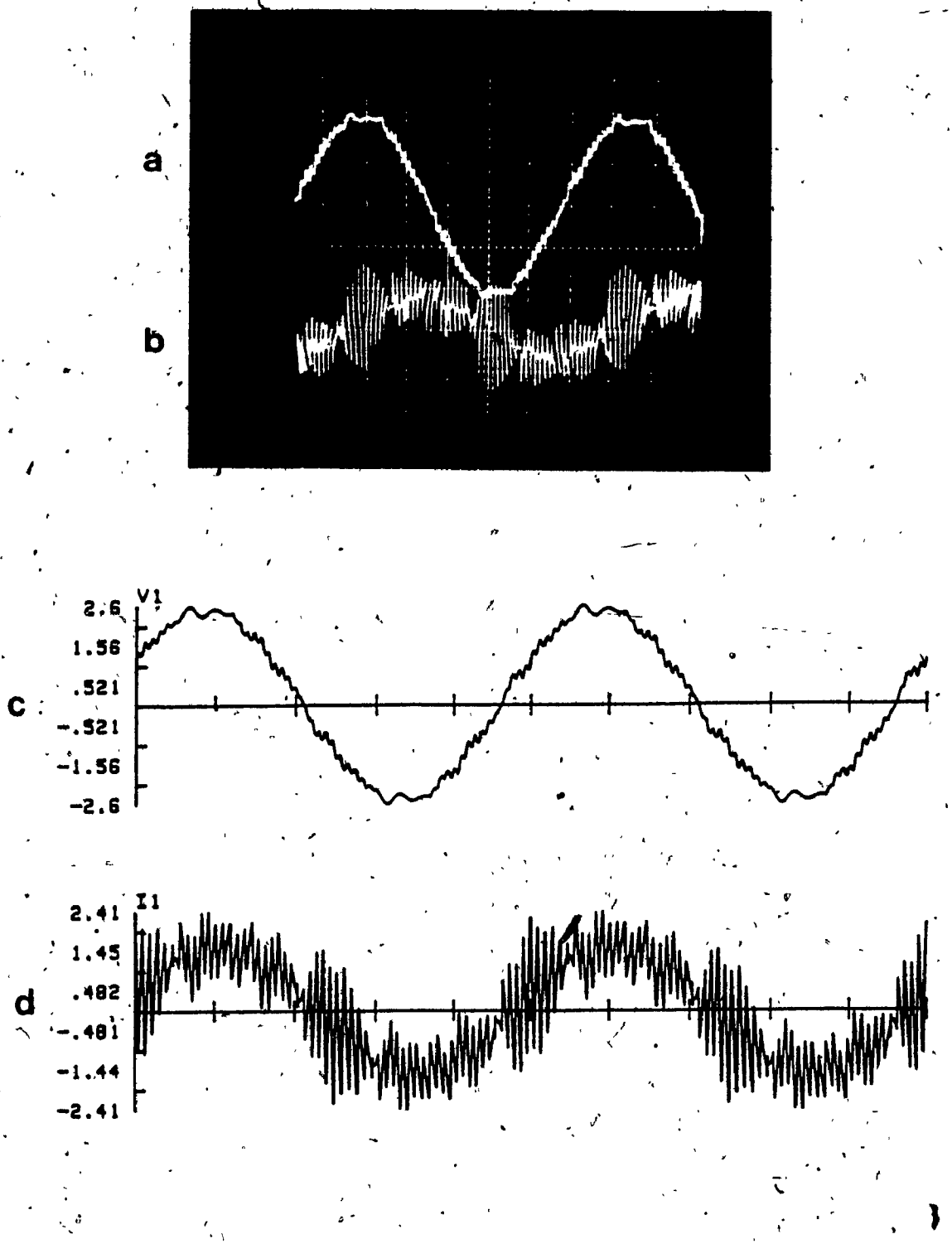


Figure 6.3. Experimental results and simulated results of the test circuit of reference [21].

7.0 CONCLUSIONS

Existing simulation/analysis programs have shown to be lacking in simplicity and user friendliness when dealing with electronic power converters - especially when PWM switch control is used. In contrast to these simulation/analysis packages, however, this thesis has produced a simplified algorithm that can develop a versatile simulation package suitable for analysis, design, and teaching purposes. The strength of State Space analysis combined with switching functions has enabled the presentation of nonlinear power bridges in a linear form. The discretization of the state equations allows direct use of powerful matrix manipulations thereby allowing execution on PCs. Implementation of the algorithm on a desk-top computer and experimental examination of its results have also been presented.

Arguments for the development of a software package dedicated to the simulation and analysis of power electronic circuits were presented in chapter 1. These arguments are based on an analysis of the unique requirements of the power electronics circuit designer and the shortcomings of the existing circuit simulators. Moreover, the required program was to emphasize three main objectives: user friendliness, low memory requirements, and fast execution times. In chapter 2 the program structure was presented and the features incorporated in the program to facilitate the interface between the user and computer for both inputting and outputting of data were described. These features contribute a great deal to the user friendliness of the package. Many of

the procedures described in this chapter and in Appendices 1 and 2 may aid programmers in achieving similar objectives. In chapter 3 combinations of the switching concepts presented in reference [16], and the State Space equations of reference [12] were extended and combined. These concepts contribute to the reduction of the amount of time and RAM memory necessary for execution. Circuit representation and systematic formulation of circuit equations for both the CLICS and NCLICS circuits are reduced to compact output system equations containing only linear elements. These equations were derived step-by-step and presented as equations 3.45 and 3.57 for the CLICS and NCLICS circuits respectively. Furthermore, the transformation used for the output CLICS equations, equation 3.7, minimizes the number of extra elements required for the CLICS solutions thereby increasing the execution speed and reducing the learning time required by inexperienced circuit designers. The discretized solution of these equations according to reference [13] was selected and described in chapter 4. This type of solution - a matter of calculating the exponential of a matrix using the power series - avoids lengthy evaluations thus increasing the overall speed of the solutions. To summarize the algorithm presented in chapters 3 and 4, an example of a single phase inverter was also fully implemented. Though the isolated inverter circuit used has 4 transistors and 6 linear components the state equation matrix A (equations 4.9) has dimensions of 3 X 3 . When PSPICE is used for the same task the dimensions of it's matrix A will be more than 3 times greater since one transistor (BIPOLAR) is modelled by 8 reactive elements and 7 passive elements. Moreover, the system state matrices, are smaller and thus easier and more quickly manipulated.

Validity of the program and the algorithm presented in this thesis were verified through the simulation of the 2 power electronic circuits presented in chapter 5 and through their respective experimental results presented in chapter 6. Finally, a fully functional program using the algorithm described in this thesis was presented in Appendices 1 and 2. The final program can assist in the development of power electronic equipment circuits, in 5 ways:

1) It can be used to provide performance insights before any actual hardware implementation is performed.
2) It can be used interactively with data acquisition devices to enter a sophisticated hardware implemented switching scheme.
3) It can be used to provide relevant circuit design data such as rms, peak, and average voltage and current ratings for various active and passive circuit components.
4) It could be employed to perform nondestructive circuit testing.
5) A self-teaching, self-learning tool by nonexpert members of the circuit development team.

Even though this program achieves the objectives stated in the introduction, further work could be continued in 4 areas:

1) Methods of optimum sampling time can be investigated to minimize the PWM waveforms and output waveform construction time.
2) Further research is needed into speeds of different types of solutions and their accuracy based on the PCs emerging on the market.
3) Since, very often the program interacts with mass storage peripherals in storing relatively long waveform files, ultra fast RAM storage (such as the virtual disk) and compact disk recorders can be investigated to lower computer run time during simulation.
4) Program rum times can be shortened through the development of hardware which can perform mathematical operations such as integration and matrix manipulation.

8.0 REFERENCES

[1] PSpice. Laguna Hills California: MircoSim Corp., 1987.

[2] Vladimirescu A.; Zhang Kaihe; Newton A.R.; Pederson D.O.; and Sangiovanni Vincentelli A. SPICE Version 2G User's Guide. Berkeley California: University of California Department of Electrical Engineering and Computer Sciences, 1981.

[3] IS-SPICE. San Pedro: Intusoft, 1986.

[4] Thompson Andrew V. Mirco-Cap Mircocomputer Circuit Analysis Program. California: Spectrum Software, 1987.

[5] Software Catalog: Michigan: Tatum labs Inc., May 1987.

[6] The CASE Vanguard CAE Design System Price/Configuration Guide. Case Technology Inc., 1987.

[7] Hardware/Software Catalog 6 California: Professional Software, 1987.

[8] Rajagopalan V., and Rao Sankara K. ATOSEC5 Users Manual. Quebec: Department D'Ingenierie Universite du Quebec a Trois Rivieries, August 1985.

[9] Hsiao C.J.; Ridley R.B.; Naitoh H.; and Lee F.C. Circuit-Oriented Discrete-Time Modeling and Simulation for Switching Converters. PESC Record, June 21-26, 1987, pp. 167-176.

[10] Dirkman R.J. The Simulation of Gereral Circuits Containing Ideal Switches. PESC Record, June 21-26, 1987, pp. 185-194.

[11] Carlisle B.H. Solving Big Problems With Little Computers. Machine Design Magazine, May 1986, pp. 75-78.

[12] Balabanian N., and Bickart T.A. Electrical Network Theory. New York: John Wiley & Sons Inc., 1969.

[13] Patel R.V. Time Domain Analysis and Design. Concordia University, 1978.

[14] Moler C., and Van Loan C. Nineteen Dubious Ways to Compute the Exponential of a Matrix. Society for Industrial and Applied Mathematics, Siam Review, Vol. 20, No. 4, October 1978.

[15] Boost M., and Ziogas P.D. State of the Art PWM Techniques: A Critical Evaluation. PESC Conference Record, June 1986.

[16] Wood P. Switching Power Converters. New York: Van Nostrand Reinhold Company, 1981.

[17] Ziogas P.D. Synthesis of Optimum Gain Functions for Static Power Converters. IEEE Transactions on Industry Applications, Vol. IA-19, No. 3, May/June 1983.

[18] Ziogas P.D.; Wiechmann E.; and Stefanovic V. A Computer Aided Analysis and Design Approach for Static Voltage Source Inverters. IEEE Transactions on Industry Application, Sept./Oct. 1985.

[19] Nelmo R.M., and Grigsby L.L. Simulation of AC Spacecraft Power Systems Using a Modular State Variable Approach. PESC Record, June 21-26, 1987, pp. 177-184.

[20] Bowes S.R., and Mount M.J. Microprocessor Control of PWM Inverters. IEE Proceedings, Vol. 128, Pt. B, No. 6, November 1981.

[21] Boost, Mike and Ziogas, Phoivos D. Towards a Zero Output Impedance UPS System. Quebec: Department of Electrical Engineering of Concordia University, 1987.

[22] Bose Bimal K. Adjustable Speed AC Drive Systems. New York: IEEE Industry Applications Society, John WITey & Sons, 1980.

[23] Vincenti D.; Boost M.; Ziogas P.D.; and Patel R.V. A Novel/Simulation Program for Power Electronics Equipment. Canadian Conference on Industrial Computer Systems, May 1986.

[24] Hsiao C.J.; Ridley R.B.; Naitoh H.; and Lee F.C. Circuit-Oriented Discrete-Time Modeling and Simulation for Switching Converters. PESC Record, June 21-26, 1987, pp. 167-176.

[25] Bowes S.R., and Clements R.R. Digital Computer Simulation of Variable-Speed PWM Inverter-Machine Drives. IEE Proceedings, Vol. 130, Pt. B, No. 3, May 1983.

[26] Bowes S.R., and Clare J. Steady-State Performance of PWM Inverter Drives. IEE Proceedings, Vol. 130, Pt. B, No. 4, July 1983.

[27] Ma X. Modeling Analysis and Digital Simulation for Power Electronic Converters. IEEE IAS, Conference Record, 1985, pp. 1195-1202.

[28] Harrington R.J., and Gawish S.A. A Digital Computer Transient Model of a Three-Phase Naturally-Commutated Inverter Including Line and Inverter Faults. IEEE IAS, Conference Record, 1985, pp. 1203-1212.

APPENDICES

APPENDIX 1: CIRCUIT DRAWING PROGRAM

APPENDIX 2: TREE-LINK PROGRAM

APPENDIX 3: PSPICE OUTPUT

APPENDIX 1

CIRCUIT DRAWING PROGRAM

```
5 !   ************************************************************
10!   *                                                          *
15!   *              CIRCUIT  DRAWING  PROGRAM                   *
20!   * USING  BASIC 2.0  OR 3.0  WITH  EXTENDED  GRAPHICS      *
25!   *     IF USING BASIC 2.0 DELETE  LINE WITH THE "*"         *
30!   *                                                          *
35!   ************************************************************
40!   PROGRAM CHECKS WHICH HP BASIC VERSION IS BEING USED
45     IF VAL(SYSTEM$("VERSION:BASIC"))=3 THEN Bas3=1
50     OUTPUT 2;CHR$(255)&"K";
55     DIM Zx$(8)[10],Cat1$(50)[80],Ab$[80],Lab$(2000)[1]
60     DIM Hihi$[19]
65     INTEGER E(70,200),Screen1(12480),Ss,Li,Hi
70     INTEGER Lab(2000,4),C11
75     S11$=CHR$(255)&CHR$(80)&CHR$(255)&CHR$(88)
80     S22$=CHR$(255)&CHR$(88)&CHR$(255)&CHR$(67)
85     OUTPUT 2;S11$&"SCRATCH KEY"&S22$; ! SCRATCH KEYS
90     GINIT
95     GRAPHICS OFF
100    ALPHA OFF
105    ON ERROR GOTO Extendg
110    RECTANGLE 0,0
115    GCLEAR
120    GOTO Startpro
125!   IF EXTENDED GRAPHICS IS NOT LOADED THE
130!   PROGRAM IS ABORTED
135 Extendg:GRAPHICS ON
140    CSIZE 6,.3
145    MOVE 20,70
150    LABEL "SORRY, YOU NEED EXTENDED GRAPHICS"
155    MOVE 20,60
160    LABEL "FOR THIS PROGRAM"
165    MOVE 40,10
170    LABEL "* * * * P R O G R A M  A B O R T E D * * * *"
175    WAIT 5
180    GOTO End
185 Startpro:  !  START PROGRAM
190    MOVE 20,70
195    GRAPHICS ON
200    CSIZE 6,.3
205    MOVE 20,60
210    LABEL "CIRCUIT DRAWING AND"
215    LABEL "ANALYZING PROGRAM"
220    MOVE 20,40
225    LABEL "USING BASIC ";VAL(SYSTEM$("VERSION: BASIC"))
230    CSIZE 8,.2
235    MOVE 40,10
240    LABEL "* * P R O G R A M  N O W  S T A R T I N G * *"
245    WAIT 3
250    GCLEAR
```

```
255     VIEWPORT 0,131,0,100
260     WINDOW 0,800,0,611
265     ! ON ERROR GOSUB Err1
270     DEG
275     CSIZE 3
280     LORG 4
285     FRAME
290!  SIZE OF USABLE GRID WINDOW FOR DRAWING IS 8 TO 792
295!  (HORIZONTAL) BY 8 TO 600 (VERTICAL)
300     CLIP 8,792,8,600
305!  CIRCUIT ELEMENT STARTING SIZE
310     S=.8
315!  A 4 GRID BY 4 GRID RECTANGLE IS USED AS THE GRAPHIC
320!  CURSOR
325!  THE GRAPHIC CURSOR STARTING POSITION IS AT 8,8
330     Px=8
335     Py=8
340     MOVE Px,Py
345     RECTANGLE 4,4
350!  ******************************************
355!  *A USEFUL COMPUTER AID IS THE ROTATING KNOB*
360!  *   WHICH IS LOCATED ON THE KEY BOARD      *
365!  *   IT IS USED HERE TO SELECT THE POINTS   *
370!  ******************************************
375 Beg:     ! BEGINNING OF THE DRAWING ROUTINE
380     Back_to_analyze=0
385     LORG 4
390     OFF ERROR
395     Dflag=1
400     MOVE Px,Py
405     RECTANGLE 4,4
410     OFF KEY
415     Ab$=""
420     Aa=1
425!  IF KNOB IS SENSED PROGRAM JUMPS TO XY ROUTINE
430     ON KNOB .030 GOTO Xy
435!  IF THE KNOB IS NOT TOUCHED THE USER HAS CONTROL
440!  OF DRAWING FUNCTIONS THROUGH ACTIVE KEYS ON THE KEY BOARD
445!  THE "S" KEY SCANS FOR THE CLOSEST POINT AROUND THE CURSOR
450!  AND MOVES THE CURSOR THERE
455!  THE "R" KEY REDRAWS THE CIRCUIT
460!  THE "M" KEY CAUSES THE PROGRAM TO JUMP TO THE MAIN MENU
465!  PRESSING THE "H" KEY WILL CAUSE THE PROGRAM TO JUMP TO
470!  HELP ROUTINE
475!  THE SPECIAL UP, DOWN, LEFT, AND RIGHT ARROW KEYS
480!  AVAILABLE ON THIS KEY BOARD ARE USED TO JUMP TO ALREAY
485!  SELECTED POINTS (NODES) FOR VERY FAST SYMMETRICAL
490!  DRAWING
495 Xy:!
500     OFF ERROR
505     ON KBD GOTO 510
510     Ab$=KBD$
515     IF Ab$="S" THEN Scan
```

```
520    IF Ab$="R" THEN GOSUB Redraw
525    IF Ab$="M" THEN Main_menu
530    IF Ab$="E" THEN Beg
535    IF Ab$="H" THEN GOSUB Help
540    IF Ab$=CHR$(255)&"^" THEN GOTO Up_dir
545    IF Ab$=CHR$(255)&"V" THEN GOTO Down_dir
550    IF Ab$=CHR$(255)&"<" THEN GOTO Left_dir
555    IF Ab$=CHR$(255)&">" THEN GOTO Right_dir
560    IF Ab$=CHR$(255)&"K" THEN
565     MAT E= (0)
570     MAT Lab= (0)
575     MAT Lab$= ("")
580     GINIT
585     VIEWPORT 0,131,0,100
590     WINDOW 0,800,0,611
595     FRAME
600     CLIP 8,792,8,600
605     GOTO Beg
610    END IF
615    IF Ab$=CHR$(255)&"E" THEN
620!  A SOUND PULSE (3000 HZ FREQUENCY) IS USED TO
625!  INDICATE SELECTION OF THE FIRST POINT
630     BEEP 3000,.02
635     IF Flag=0 THEN
640      BEEP 81,.02
645      Px1=Px
650      Py1=Py
655      Flag=1
660      GOTO Beg
665     END IF
670     IF Flag=1 THEN
675!  A SOUND (800 HZ FREQUENCY) IS USED TO
680!  INDICATE SELECTION OF THE SECOND POINT
685      BEEP 800,.02
690      Px2=Px
695      Py2=Py
700      Flag=2
705      Analyze_draw=0
710      OFF KNOB
715!  AFTER SELECTION OF THE SECOND POINT THE
720!  PROGRAM AUTOMATICALY JUMPS TO THE MAIN MENU
725      GOTO Main_menu
730     END IF
735    END IF
740!  STATUS OF THE "SHIFT" KEY IS SENSED
745!  IF "SHIFT" KEY IS NOT USED WITH THE KNOB THE HORIZONTAL
750!  DIRECTION IS SELECTED
755!  IF THE "SHIFT" KEY IS USED WITH THE KNOB THE VERTICAL
760!  DIRECTION IS SELECTED
765    STATUS 2,10;Xy
770    IF Xy=0 THEN Px
775    IF Xy=1 THEN Py
780    GOTO Beg
```

```
785 Px:! VERTICAL GRID MOVE
790     X=KNOBX*4
795     IF X>=0 THEN
800      PEN -1
805      MOVE Px,Py
810      RECTANGLE 4,4
815      Px=Px+X
820      MOVE Px,Py
825!  RIGHT GRID LIMIT
830      IF Px>788 THEN Px=788
835      PEN 1
840      RECTANGLE 4,4
845     ELSE
850      PEN -1
855      MOVE Px,Py
860      RECTANGLE 4,4
865      PEN 1
870      Px=Px+X
875      MOVE Px,Py
880!  LEFT GRID LIMIT
885      IF Px<8 THEN Px=8
890      PEN 1
895      RECTANGLE 4,4
900     END IF
905     GOTO Beg
910 Py:! HORIZONTAL MOVE
915     IF Bas3=1 THEN
920!     ******IF YOU ARE USING BASIC 2 DELETE NEXT LINE******
925      Y=KNOBY*(-4)     !  THE LINE WITH THE  "*"
930     ELSE
935      Y=KNOBX*(-4)
940     END IF
945     IF Y>=0 THEN
950      PEN -1
955      MOVE Px,Py
960      RECTANGLE 4,4
965      Py=Py+Y
970      MOVE Px,Py
975! TOP GRID LIMIT
980      IF Py>596 THEN Py=596
985      PEN 1
990      RECTANGLE 4,4
995      GOTO Beg
1000    ELSE
1005     PEN -1
1010     MOVE Px,Py
1015     RECTANGLE 4,4
1020     PEN 1
1025     Py=Py+Y
1030     MOVE Px,Py
1035!  BOTTOM GRID LIMIT
1040     IF Py<8 THEN Py=8
1045     PEN 1
```

```
1050    RECTANGLE 4,4
1055   END IF
1060   GOTO Beg
1065 Label2:    ! ROUTINE FOR LABELING THE CIRCUIT
1070   OFF KNOB
1075   PEN -1
1080   MOVE Px,Py
1085   RECTANGLE 4,4
1090   PEN 0
1095   LORG 5
1100   Cw1=0
1105   Ch1=0
1110   Flagdir=0
1115 Label3:    ! CHARACTER SIZE
1120   OFF KNOB
1125   OFF KBD
1130!  ORIENTATION OF THE CHARACTERS ARE SELECTED HERE
1135!  EITHER HORIZONTAL OR VERTICAL SELECTION
1140!  CHARACTER SPACING OR GRID POINT SPACING
1145   ON KEY 0 LABEL "NORMAL/WIDEWIDTH" GOTO Nw
1150   ON KEY 5 LABEL "HOR/VER" GOTO Hflag
1155   ON KEY 1 LABEL "SIZE 1/2" GOTO C1
1160   ON KEY 2 LABEL "SIZE 3/4" GOTO C3
1165   ON KEY 3 LABEL "SIZE 5/6" GOTO C5
1170   ON KEY 4 LABEL "SIZE 7/8" GOTO C7
1175   ON KEY 8 LABEL "SIZE 9/10" GOTO C9
1180   ON KEY 6 LABEL "CHAR.JUMP/NOT" GOTO Cj
1185   ON KEY 16 LABEL "NO CHAR.JUMP" GOTO Ncj
1190   ON KEY 11 GOTO C2
1195   ON KEY 12 GOTO C4
1200   ON KEY 13 GOTO C6
1205   ON KEY 14 GOTO C8
1210   ON KEY 18 GOTO C10
1215   ON KEY 10 GOTO Ww
1220   ON KEY 15 GOTO Vflag
1225   ON KEY 9 LABEL "EXIT" GOTO Endlabel
1230   GOTO Label3
1235 C1:  !
1240   Ch=8
1245   Ff=1
1250   GOTO Start1
1255 C2:  !
1260   Ch=12
1265   Ff1=1
1270   GOTO Start1
1275 C3:  !
1280   Ch=16
1285   GOTO Start1
1290 C4:  !
1295   Ch=20
1300   GOTO Start1
1305 C5:  !
1310   Ch=24
```

```
1315   GOTO Start1
1320 C6:   !
1325   Ch=28
1330   GOTO Start1
1335 C7:   !
1340   Ch=32
1345   GOTO Start1
1350 C8:   !
1355   Ch=36
1360   GOTO Start1
1365 C9:   !
1370   Ch=44
1375   GOTO Start1
1380 C10:  !
1385   Ch=60
1390   GOTO Start1
1395 Nw:   ! NORMAL
1400   Lflag=0
1405   GOTO Start1
1410 Ww:   ! WIDE
1415   Lflag=1
1420   GOTO Start1
1425 Hflag:    !
1430   Flagdir=0
1435   GOTO Start1
1440 Vflag:            !
1445   Flagdir=1
1450   GOTO Start1
1455 Cj:Ff1=1
1460   GOTO Start1
1465 Ncj:Ff1=0
1470 Start1:    !
1475   ALPHA OFF
1480   Cw=Ch
1485   Cc=-.743+.243*Ch
1490   PEN 0
1495   CSIZE Cc
1500   Flag=3
1505   MOVE Px,Py
1510! A RECTANGLE (THE SIZE WAS SELECTED PREVIOUSLY BY THE
1515! USER) IS  USED TO INDICATE THE SIZE AND LOCATION OF
1520! THE CHARACTER
1525   RECTANGLE Cw1,Ch1
1530   MOVE Px,Py
1535   RECTANGLE Cw,Ch
1540 Label1:  !
1545   ALPHA OFF
1550   Cw1=Cw
1555   Ch1=Ch
1560   Ab$=""
1565   Az=1
1570   ON KBD GOTO 1575
1575   Ab$=KBD$
```

```
1580   Chh=4
1585   Cww=4
1590! UP ARROW KEY WILL CAUSE THE CURSOR (NOW CW X CH SIZE) TO
1595! JUMP ONE SPACE (EITHER A CHARACTER SPACE OR 4 GRID POINT
1600! SPACE) UP
1605   IF Ab$=CHR$(255)&"^" THEN
1610    OFF KBD
1615    MOVE Px,Py
1620    RECTANGLE Cw,Ch
1625    IF Ff1=1 THEN Chh=Ch
1630    IF Py+Chh<=580 THEN Py=Py+Chh
1635    MOVE Px,Py
1640    RECTANGLE Cw,Ch
1645    GOTO Label1
1650    ON KBD GOTO 1575
1655   END IF
1660! THE DOWN ARROW KEY WILL CAUSE THE CURSOR TO JUMP ONE
1665! SPACE DOWN
1670   IF Ab$=CHR$(255)&"V" THEN
1675    OFF KBD
1680    MOVE Px,Py
1685    RECTANGLE Cw,Ch
1690    IF Ff1=1 THEN Chh=Ch
1695    IF Py-Chh>=8 THEN Py=Py-Chh
1700    MOVE Px,Py
1705    RECTANGLE Cw,Ch
1710    GOTO Label1
1715    ON KBD GOTO 1575
1720   END IF
1725! THE LEFT ARROW KEY WILL CAUSE THE CURSOR TO JUMP ONE
1730! SPACE TO THE LEFT
1735   IF Ab$=CHR$(255)&"<" THEN
1740    OFF KBD
1745    MOVE Px,Py
1750    RECTANGLE Cw,Ch
1755    IF Ff1=1 THEN Cww=Cw
1760    IF Px-Cww>=8 THEN Px=Px-Cww
1765    MOVE Px,Py
1770    RECTANGLE Cw,Ch
1775    GOTO Label1
1780    ON KBD GOTO 1575
1785   END IF
1790! THE RIGHT ARROW KEY WILL CAUSE THE CURSOR TO
1795! JUMP ONE SPACE TO THE RIGHT
1800   IF Ab$=CHR$(255)&">" THEN
1805    OFF KBD
1810    MOVE Px,Py
1815    RECTANGLE Cw,Ch
1820    IF Ff1=1 THEN Cww=Cw
1825    IF Px+Cww<=780 THEN Px=Px+Cww
1830    MOVE Px,Py
1835    RECTANGLE Cw,Ch
1840    GOTO Label1
```

```
1845    ON KBD GOTO 1575
1850   END IF
1855! THE ENTER KEY WILL MOVE THE CURSOR TO THE NEXT LINE
1860! ALIGNED WITH THE FIRST CHARACTER OF THE PREVIOUS LINE
1865   IF Ab$=CHR$(255)&"E" THEN
1870    OFF KBD
1875    IF Flagdir=0 THEN
1880     Minx=Px
1885     FOR I=1 TO Lab(0,0)
1890      IF Lab(I,3)<Minx AND Lab(I,4)=Py THEN Minx=Lab(I,3)
1895     NEXT I
1900     MOVE Px,Py
1905     RECTANGLE Cw,Ch
1910     Chh=Ch
1915     IF Py-Chh>=8 THEN Py=Py-Chh
1920     MOVE Minx,Py
1925     RECTANGLE Cw,Ch
1930     Px=Minx
1935    ELSE
1940     Minx=Py
1945     FOR I=1 TO Lab(0,0)
1950      IF Lab(I,4)<Minx AND Lab(I,3)=Px THEN Minx=Lab(I,4)
1955     NEXT I
1960     MOVE Px,Py
1965     RECTANGLE Cw,Ch
1970     Cww=Cw
1975     IF Px+Cww<=780 THEN Px=Px+Cww
1980     MOVE Px,Minx
1985     RECTANGLE Cw,Ch
1990     Py=Minx
1995    END IF
2000    ON KBD GOTO Label1
2005    GOTO Label1
2010   END IF
2015! CHARACTER IS PRINTED ON THE SCREEN
2020! CHARACTER IS TEMPORARILY STORED IN AB$
2025   IF Ab$<>"" AND Ab$[1,1]<>CHR$(255) THEN
2030    OFF KBD
2035    IF Ab$=" " THEN
2040     AREA PEN -1
2045     FOR I=1 TO Lab(0,0)
2050      IF Lab(I,3)<Px+Cw AND Lab(I,3)>=Px AND Lab(I,4)>=Py AND Lab(I,4)<Py+Ch
 THEN Lab(I,2)=0
2055     NEXT I
2060     BEEP 1500,.01
2065     MOVE Px+2,Py+2
2070     RECTANGLE Cw-4,Ch-4,FILL
2075     ON KBD GOTO Start1
2080     GOTO Start1
2085    END IF
2090! CHARACTER IS STORED IN THE ARRAY IAB$
2095    Lab(0,0)=Lab(0,0)+1
2100    Jj=Lab(0,0)
```

```
2105    Lab$(Jj)=Ab$[1,1]
2110! SIZE, ORIENTATION AND LOCATION OF THE CHARACTER
2115! IS STORED IN THE ARRAY LAB
2120    Lab(Jj,0)=Lflag
2125    Lab(Jj,1)=Flagdir
2130    Lab(Jj,2)=Cc
2135    Lab(Jj,3)=Px
2140    Lab(Jj,4)=Py
2145    PEN 1
2150! POSITION ON THE SCREEN
2155    IF Flagdir=0 THEN
2160     MOVE Px+(Cc+.743)/(2*(.243)),Py+(Cc+.743)/(2*(+.243))
2165     LDIR 0
2170    ELSE
2175     LORG 5
2180     LDIR 90
2185     MOVE Px+(Cc+.743)/(2*(.243)),Py+(Cc+.743)/(2*(.243))
2190    END IF
2195! PRINT THE CHARACTER ON THE SCREEN
2200    LABEL Ab$
2205    PEN 0
2210    MOVE Px,Py
2215    RECTANGLE Cw,Ch
2220    IF Ff1=0 THEN
2225     IF Flagdir=0 THEN
2230      IF Px+4<=780 THEN Px=Px+4
2235     ELSE
2240      IF Py+4<=580 THEN Py=Py+4
2245     END IF
2250    ELSE
2255     IF Flagdir=0 THEN
2260      IF Px+Cw<=780 THEN Px=Px+Cw
2265     ELSE
2270      IF Py+Ch<=580 THEN Py=Py+Ch
2275     END IF
2280    END IF
2285    MOVE Px,Py
2290    RECTANGLE Cw,Ch
2295! SOUND (4500 HZ FREQUENCY) INDICATING THAT THE CHARACTER
2300! WAS SUCCESFULLY RECORDED IN THE ARRAY BEFORE AN
2305! INTERRUPTION OCCURED
2310    BEEP 4500,.007
2315    ON KBD GOTO Label1
2320    GOTO Label1
2325   END IF
2330   GOTO Label1
2335 Endlabel:  ! END OF LABEL ROUTINE
2340   LDIR 0
2345   PEN 0
2350   MOVE Px,Py
2355   RECTANGLE Cw,Ch
2360   Flag=0
2365! GO BACK TO BEG
```

```
2370   GOTO Beg
2375!
2380 Turn_c:     ! TURN CIRCUIT AROUND
2385   RETURN
2390!
2395 Move_c:     ! MOVE CIRCUIT ON PAGE ROUTINE
2400   OFF KNOB
2405   Smallx=800
2410   Smally=800
2415   Largex=0
2420   Largey=0
2425! FIND SIZE OF CIRCUIT
2430   FOR I=1 TO 70
2435    IF E(I,0)=0 THEN 2490
2440    FOR J=4 TO E(I,0)*4 STEP 4
2445     IF E(I,J-3)<Smallx THEN Smallx=E(I,J-3)
2450     IF E(I,J-1)<Smallx THEN Smallx=E(I,J-1)
2455     IF E(I,J-3)>Largex THEN Largex=E(I,J-3)
2460     IF E(I,J-1)>Largex THEN Largex=E(I,J-1)
2465     IF E(I,J-2)<Smally THEN Smally=E(I,J-2)
2470     IF E(I,J)<Smally THEN Smally=E(I,J)
2475     IF E(I,J-2)>Largey THEN Largey=E(I,J-2)
2480     IF E(I,J)>Largey THEN Largey=E(I,J)
2485    NEXT J
2490   NEXT I
2495! ADD 20 OR SUBTRACT 20 GRID POINTS FROM THE
2500! DIMENSION OF THE CIRCUIT PERIMETER
2505   Largex=Largex+20
2510   Largey=Largey+20
2515   Smallx=Smallx-20
2520   Smally=Smally-20
2525   IF Largex>792 THEN Largex=792
2530   IF Largey>600 THEN Largey=600
2535   IF Smallx<8 THEN Smallx=8
2540   IF Smally<8 THEN Smally=8
2545   Sx=Smallx
2550   Sy=Smally
2555   GINIT
2560   VIEWPORT 0,131,0,100
2565   WINDOW 0,800,0,611
2570! CIRCUIT IS CLEARED FROM SCREEN TEMPORARILY
2575   GCLEAR
2580   GRAPHICS ON
2585   ALPHA OFF
2590   FRAME
2595   CLIP 8,792,8,600
2600   Smx=0
2605   Smy=0
2610 Move_sq:          ! MOVE RECTANGLE
2615   OFF KEY
2620   PEN 1
2625! A RECTANGLE IS DRAWN AROUND THE CIRCUIT (THE
2630! CIRCUIT WHICH WAS CLEARED) TO SPEED UP POSITIONING
```

```
2635! OF THE CIRCUIT
2640   PLOT Smallx,Largey,1
2645   PLOT Largex,Largey
2650   PLOT Largex,Smally
2655   PLOT Smallx,Smally
2660   PLOT Smallx,Largey,2
2665   Flag=0
2670   Ab$=""
2675   ON KBD GOTO 2690
2680! USING THE KNOB OR ARROW KEYS THE RECTANGLE CAN BE
2685! POSITIONED ANYWHERE ON THE SCREEN QUICKLY
2690   Ab$=KBD$
2695   IF Ab$=CHR$(255)&"?" THEN GOTO Label2
2700   IF Ab$=CHR$(255)&"^" THEN
2705    Smy=Smy+2
2710    GOTO Drsq
2715   END IF
2720   IF Ab$=CHR$(255)&"V" THEN
2725    Smy=Smy-2
2730    GOTO Drsq
2735   END IF
2740   IF Ab$=CHR$(255)&"<" THEN
2745    Smx=Smx-2
2750    GOTO Drsq
2755   END IF
2760   IF Ab$=CHR$(255)&">" THEN
2765    Smx=Smx+2
2770    GOTO Drsq
2775   END IF
2780! THE POSITION IS SELECTED BY PRESSING THE ENTER KEY
2785   IF Ab$=CHR$(255)&"E" THEN Flag=1
2790   IF Ab$=CHR$(255)&"C" THEN
2795    OFF KBD
2800! REDRAW CIRCUIT BACK ON THE SCREEN IN ITS
2805! NEW POSITION
2810    GOSUB Redraw
2815    GOTO Beg
2820   END IF
2825 Drsq:                    !
2830   PEN -1
2835   PLOT Smallx,Largey,1
2840   PLOT Largex,Largey
2845   PLOT Largex,Smally
2850   PLOT Smallx,Smally
2855   PLOT Smallx,Largey,2
2860   IF Largey+Smy<=600 AND Smally+Smy>=8 THEN
2865    Smally=Smally+Smy
2870    Largey=Largey+Smy
2875   ELSE
2880    IF Smy>0 THEN Smy=600-Largey
2885    IF Smy<0 THEN Smy=8-Smally
2890    Smally=Smally+Smy
2895    Largey=Largey+Smy
```

```
2900    END IF
2905    IF Largex+Smx<=792 AND Smallx+Smx>=8 THEN
2910     Smallx=Smallx+Smx
2915     Largex=Largex+Smx
2920    ELSE
2925     IF Smx>0 THEN Smx=792-Largex
2930     IF Smx<0 THEN Smx=8-Smallx
2935     Smallx=Smallx+Smx
2940     Largex=Largex+Smx
2945    END IF
2950    IF Flag=1 THEN 2965
2955    GOTO Move_sq
2960    OFF KBD
2965    FOR I=1 TO 70
2970     IF E(I,0)=0 THEN 3005
2975     FOR J=4 TO E(I,0)*4 STEP 4
2980      E(I,J-3)=E(I,J-3)+(Smallx-Sx)
2985      E(I,J-2)=E(I,J-2)+(Smally-Sy)
2990      E(I,J-1)=E(I,J-1)+(Smallx-Sx)
2995      E(I,J)=E(I,J)+(Smally-Sy)
3000     NEXT J
3005    NEXT I
3010    GOSUB Redraw
3015    GOTO Beg
3020 Scan:     !  SCAN ROUTINE
3025! THE SCAN ROUTINE IS USED TO LOCATE AND MOVE THE CURSOR
3030! TO THE CLOSEST ENTERED POINT AROUND THE PRESENT CURSOR
3035! POSITION FOR QUICK CONNECTION OF ELEMENTS
3040    PEN -1
3045    MOVE Px,Py
3050    RECTANGLE 4,4
3055    PEN 1
3060    Aa=Px
3065    Bb=Py
3070    D=1.E+6
3075! CLOSEST POINT IS FOUND BY CREATING CIRCLES AND FINDING
3080! THE POINT WITH THE SMALLEST RADIUS SQUARED
3085    FOR I=1 TO 70
3090     IF E(I,0)=0 THEN Loop1
3095     FOR J=4 TO E(I,0)*4 STEP 4
3100      D1=(Px-E(I,J-3))^2+(Py-E(I,J-2))^2
3105      IF D1<D THEN
3110       D=D1
3115       Aa=E(I,J-3)
3120       Bb=E(I,J-2)
3125      END IF
3130      D1=(Px-E(I,J-1))^2+(Py-E(I,J))^2
3135      IF D1<D THEN
3140       Aa=E(I,J-1)
3145       Bb=E(I,J)
3150      END IF
3155     NEXT J
3160 Loop1:          !
```

```
3165   NEXT I
3170   Px=Aa
3175   Py=Bb
3180! END OF THE SCAN ROUTINE
3185! GO BACK TO BEG
3190   GOTO Beg
3195!
3200 Up_dir:   ! UP ARROW KEY SELECTION ROUTINE
3205   Tes=800
3210   PEN -1
3215   MOVE Px,Py
3220   RECTANGLE 4,4
3225   PEN 1
3230   Ax=Px
3235   Ay=Py
3240   FOR I=1 TO 70
3245    IF E(I,0)=0 THEN 3300
3250    FOR J=4 TO E(I,0)*4 STEP 4
3255     IF E(I,J-2)>Ay AND E(I,J-2)<Tes THEN
3260      Py=E(I,J-2)
3265      Tes=Py
3270     END IF
3275     IF E(I,J)>Ay AND E(I,J)<Tes THEN
3280      Py=E(I,J)
3285      Tes=Py
3290     END IF
3295    NEXT J
3300   NEXT I
3305   GOTO Beg
3310!
3315 Down_dir: ! DOWN ARROW KEY SELECTION ROUTINE
3320   Tes=0
3325   PEN -1
3330   MOVE Px,Py
3335   RECTANGLE 4,4
3340   PEN 1
3345   Ax=Px
3350   Ay=Py
3355   FOR I=1 TO 70
3360    IF E(I,0)=0 THEN 3415
3365    FOR J=4 TO E(I,0)*4 STEP 4
3370     IF E(I,J-2)<Ay AND E(I,J-2)>Tes THEN
3375      Py=E(I,J-2)
3380      Tes=Py
3385     END IF
3390     IF E(I,J)<Ay AND E(I,J)>Tes THEN
3395      Py=E(I,J)
3400      Tes=Py
3405     END IF
3410    NEXT J
3415   NEXT I
3420   GOTO Beg
3425!
```

```
3430 Left_dir: ! LEFT ARROW KEY SELECTION ROUTINE
3435   Tes=0
3440   PEN -1
3445   MOVE Px,Py
3450   RECTANGLE 4,4
3455   PEN 1
3460   Ax=Px
3465   Ay=Py
3470   FOR I=1 TO 70
3475    IF E(I,0)=0 THEN 3530
3480    FOR J=4 TO E(I,0)*4 STEP 4
3485     IF E(I,J-3)<Ax AND E(I,J-3)>Tes THEN
3490      Px=E(I,J-3)
3495      Tes=Px
3500     END IF
3505     IF E(I,J-1)<Ax AND E(I,J-1)>Tes THEN
3510      Px=E(I,J-1)
3515      Tes=Px
3520     END IF
3525    NEXT J
3530   NEXT I
3535   GOTO Beg
3540!
3545 Right_dir: ! RIGHT ARROW KEY SELECTION ROUTINE
3550   Tes=800
3555   PEN -1
3560   MOVE Px,Py
3565   RECTANGLE 4,4
3570   PEN 1
3575   Ax=Px
3580   Ay=Py
3585   FOR I=1 TO 70
3590    IF E(I,0)=0 THEN 3645
3595    FOR J=4 TO E(I,0)*4 STEP 4
3600     IF E(I,J-3)>Ax AND E(I,J-3)<Tes THEN
3605      Px=E(I,J-3)
3610      Tes=Px
3615     END IF
3620     IF E(I,J-1)>Ax AND E(I,J-1)<Tes THEN
3625      Px=E(I,J-1)
3630      Tes=Px
3635     END IF
3640    NEXT J
3645   NEXT I
3650   GOTO Beg
3655! ************************************************
3660! *AFTER THE SELECTION OF THE SECOND POINT       *
3665! *FUNCTION KEYS  ARE DISPLAYED AT THE BOTTOM    *
3670! *OF THE SCREEN                                 *
3675! *                                             *
3680! *SELECTION OF ONE OF THESE KEYS                *
3685! *WILL PERFORM A FUCTION                        *
3690! *                                             *
```

```
3695! *   MENUS WILL LOOP THEN JUMP BACK TO BEG      *
3700! ************************************************
3705 Main_menu:    !
3710   Flag=0
3715   OFF KEY
3720   FOR I=1 TO 500
3725    ON KEY 0 LABEL "SOURCES" GOTO Sources
3730    ON KEY 1 LABEL "LINEAR ELEMENT" GOTO Linear_element
3735    ON KEY 2 LABEL "NON-LINEAR" GOTO Non_linear
3740    ON KEY 3 LABEL "LOGIC ELEMENT" GOTO Logic_element
3745    ON KEY 4 LABEL "DELETE" GOTO Undraw
3750    ON KEY 5 LABEL "LINE/DOT" GOTO Li
3755    ON KEY 6 LABEL "CONFIGURATIONS" GOTO Configurations
3760    ON KEY 7 LABEL "LABEL" GOTO Label2
3765    ON KEY 8 LABEL "MANIPULATION" GOTO Manipulation
3770    ON KEY 9 LABEL "DOTTED LINE" GOTO Dot_l
3775    ON KEY 15 GOTO Do
3780    ON KBD,15 RECOVER 3790
3785    GOTO 3825
3790    A$=KBD$
3795! AT ANY TIME (WHILE IN THIS MENU) PRESSING THE E KEY,
3800! R KEY OR THE CLEAR SCREEN KEY WILL INTERRUPT AND EXIT
3805! REDRAW OR CLEAR THE SCREEN RESPECTIVELY
3810    IF A$="E" THEN Beg
3815    IF A$="R" THEN GOSUB Redraw
3820    IF A$=CHR$(255)&"#" THEN En
3825   NEXT I
3830   OFF KEY
3835   GOTO Beg
3840    !
3845 Undraw: ! UNDRAW ROUTINE
3850   OFF ERROR
3855! AFTER THE SELECTION OF TWO POINTS AN ELEMENT CAN BE
3860! DELETED USING THE DELETE FUNCTION KEY IN THE MAIN MENU
3865   Aa=-1
3870   FOR I=1 TO 70
3875    IF E(I,0)=0 THEN 3935
3880    FOR J=4 TO E(I,0)*4 STEP 4
3885     IF E(I,J-3)=0 THEN 3930
3890     Jj=J
3895     Ii=I
3900     IF E(I,J-3)=Px1 AND E(I,J-2)=Py1 THEN
3905      IF E(I,J-1)=Px2 AND E(I,J)=Py2 THEN 3950
3910     END IF
3915     IF E(I,J-3)=Px2 AND E(I,J-2)=Py2 THEN
3920      IF E(I,J-1)=Px1 AND E(I,J)=Py1 THEN 3950
3925     END IF
3930    NEXT J
3935   NEXT I
3940                    !
3945   GOTO 4095
3950   Ee=I
3955   Px1=E(I,J-3)
```

```
3960   Py1=E(I,J-2)
3965   Px2=E(I,J-1)
3970   Py2=E(I,J)
3975   Flag=0
3980   SELECT Be
3985   CASE =1
3990    GOSUB Pdo
3995   CASE =2
4000    GOSUB Pli
4005   CASE ELSE
4010    Flag=1
4015    GOSUB Pdraw1
4020   END SELECT
4025! THE DELETED ELEMENT IS REMOVED
4030! IMMEDIATELY FROM THE 2 DIMENSIONAL
4035! ARRAY E AND THEN THE ARRAY IS REPACKED
4040! SO THAT LOOPING THROUGH THE ELEMENTS
4045! IS NOT SLOWED DOWN
4050          ! REPACK ARRAY
4055   FOR K=J TO E(I,0)*4 STEP 4
4060    E(I,K-3)=E(I,(K+4)-3)
4065    E(I,K-2)=E(I,(K+4)-2)
4070    E(I,K-1)=E(I,(K+4)-1)
4075    E(I,K)=E(I,(K+4))
4080   NEXT K
4085   E(I,0)=E(I,0)-1
4090                !
4095   Flag=0
4100   Aa=1
4105   GOTO Beg
4110          !
4115 Manipulation: ! MANIPULATION FUNCTION KEY MENU
4120   OFF KEY
4125   Dflag=1
4130   FOR I=1 TO 500
4135    ALPHA OFF
4140    ON KEY 0 LABEL "STORE" GOTO Store1
4145    ON KEY 5 LABEL "PLOTTER" GOTO Plot1
4150    ON KEY 4 LABEL "MAIN MENU" GOTO Main_menu
4155    ON KEY 1 LABEL "ELEMENT BIGGER" GOSUB S_bigger
4160    ON KEY 6 LABEL "ELEMENT SMALLER" GOSUB S_smaller
4165    ON KEY 9 LABEL "END" GOTO En
4170    ON KEY 2 LABEL "MOVE CIRCUIT" GOTO Move_c
4175! THE ANALYZE FUNCTION KEY IS LOCATED IN THE
4180! MANIPULATION FUNCTION KEY MENU
4185    ON KEY 8 LABEL "ANALYZE" GOTO Analyze
4190    ON KEY 7 LABEL "TURN CIRCUIT" GOTO Turn_c
4195    ON KEY 3 LABEL "RECALL" GOTO Recall
4200    OFF KBD
4205    ON KBD,15 RECOVER 4235
4210    GOTO 4255
4215! AT ANY GIVEN TIME (IN THE MENU MODE) EXIT TO BEG,
4220! REDRAW OR GO TO MAIN MENU INTERRUPTIONS ARE POSSIBLE BY
```

```
4225! PRESSING THE E KEY, R KEY, OR THE M KEY RESPECITVELY
4230! THESE ARE ACTIVE IN ALL MENUS
4235   A$[1]=KBD$
4240   IF A$="E" THEN Beg
4245   IF A$="R" THEN GOSUB Redraw
4250   IF A$="M" THEN GOTO Main_menu
4255  NEXT I
4260  OFF KEY
4265  GOTO Beg
4270    !
4275 Sources: ! LINEAR ELEMENT FUNCTION KEY MENU
4280  OFF KEY
4285  FOR I=1 TO 500
4290   ON KEY 0 LABEL "BATTERY" GOTO Ba
4295   ON KEY 1 LABEL "CURRENT SOURCE" GOTO Cs
4300   ON KEY 2 LABEL "AC VOLTAGE-+" GOTO Ac
4305   ON KEY 3 LABEL "ARROW" GOTO Ar
4310   ON KEY 4 LABEL "DIRECTION" GOTO Di
4315   ON KEY 5 LABEL "DC SOURCE" GOTO Dc
4320   ON KEY 6 LABEL "CURRENT" GOTO Cu
4325   ON KEY 7 LABEL "AC VOLTAGE" GOTO A_
4330   ON KEY 8 LABEL "LOOP" GOTO Lo
4335   ON KEY 9 LABEL "GROUND" GOTO Gr
4340   ON KBD,15 RECOVER 4350
4345   GOTO 4370
4350   A$=KBD$
4355   IF A$="E" THEN Beg
4360   IF A$="R" THEN GOSUB Redraw
4365   IF A$="M" THEN GOTO Main_menu
4370  NEXT I
4375  OFF KEY
4380  Flag=0
4385  GOTO Beg
4390    !
4395 Linear_element:    !
4400  OFF KEY
4405  FOR I=1 TO 500
4410   ON KEY 0 LABEL "RESISTER" GOTO Re1
4415   ON KEY 1 LABEL "CAPACITOR" GOTO Ca
4420   ON KEY 2 LABEL "INDUCTOR" GOTO In1
4425   ON KEY 3 LABEL "IMPEDANCE" GOTO Im
4430   ON KEY 6 LABEL "ELECIROLYTIC" GOTO El
4435   ON KEY 8 LABEL "ADMITANCE" GOTO Ad
4440   ON KEY 7 LABEL "CHOKE" GOTO Ch
4445   ON KBD,15 RECOVER 4455
4450   GOTO 4475
4455   A$=KBD$
4460   IF A$="E" THEN Beg
4465   IF A$="R" THEN GOSUB Redraw
4470   IF A$="M" THEN GOTO Main_menu
4475  NEXT I
4480  OFF KEY
4485  GOTO Beg
```

```
4490     !
4495 Non_linear: ! NON-LINEAR ELEMENT FUNCTION KEY MENU
4500   OFF KEY
4505   FOR I=1 TO 500
4510    ON KEY 0 LABEL "DIODE" GOTO Di1
4515    ON KEY 1 LABEL "THYRISTOR" GOTO Th
4520    ON KEY 2 LABEL "BIPOLAR" GOTO Bi
4525    ON KEY 3 LABEL "MOSFET" GOTO Mo
4530    ON KEY 4 LABEL "SWITCH" GOTO Sw
4535    ON KEY 5 LABEL "ZENER" GOTO Ze
4540    ON KEY 6 LABEL "THYRISTOR D" GOTO Td
4545    ON KEY 7 LABEL "BIPOLAR DIODE" GOTO Bd
4550    ON KEY 8 LABEL "MOSFET DIODE" GOTO Md
4555    ON KEY 9 LABEL "NEXT" GOTO Non_linear2
4560    ON KBD,15 RECOVER 4570
4565    GOTO 4590
4570    A$=KBD$
4575    IF A$="E" THEN Beg
4580    IF A$="R" THEN GOSUB Redraw
4585    IF A$="M" THEN GOTO Main_menu
4590   NEXT I
4595   GOTO Beg
4600     !
4605 Non_linear2: ! SECOND NON-LINEAR ELEMENT FUNCTION KEY MENU
4610   OFF KEY
4615   FOR I=1 TO 500
4620    ON KEY 0 LABEL "TRIAC" GOTO Tr
4625    ON KEY 1 LABEL "T SUPRESSOR" GOTO Si
4630    ON KEY 2 LABEL "HALF-BRID.INV." GOTO Inver1
4635    ON KEY 3 LABEL "FULL-BRID.INV." GOTO Inver2
4640    ON KEY 4 LABEL "FULL-B.3-0-INV." GOTO Inver3
4645    ON KEY 7 LABEL "1-0 RECTIFIER" GOTO Rectifier1
4650    ON KEY 8 LABEL "3-0 DELTA-RECT" GOTO Rectifier2
4655    ON KEY 9 LABEL "3-0 Y-RECT" GOTO Rectifier3
4660    ON KBD,15 RECOVER 4670
4665    GOTO 4690
4670    A$=KBD$
4675    IF A$="E" THEN Beg
4680    IF A$="R" THEN GOSUB Redraw
4685    IF A$="M" THEN GOTO Main_menu
4690   NEXT I
4695   OFF KEY
4700   GOTO Beg
4705     !
4710 Configurations:!DIFFERENT CONFIGRATION FUNCTION KEY MENU
4715   OFF KEY
4720   FOR I=1 TO 500
4725    ON KEY 0 LABEL "SINGLE 0 T." GOTO St
4730    ON KEY 1 LABEL "3 0 T. D-Y" GOTO Tt
4735    ON KEY 2 LABEL "DELTA" GOTO De
4740    ON KEY 3 LABEL "WYE" GOTO Wy
4745    ON KEY 5 LABEL "CENTER TAP T." GOTO Ct
4750    ON KEY 6 LABEL "4 DIODE BRIDGE" GOTO Db
```

```
4755    ON KEY 9 LABEL "SHAPES" GOTO Shapes
4760    ON KEY 7 LABEL "ARROW" GOTO Ar
4765    ON KEY 8 LABEL "DIRECTION" GOTO Di
4770    ON KEY 4 LABEL "LOOP" GOTO Lo
4775    ON KBD,15 RECOVER 4785
4780    GOTO 4805
4785    A$=KBD$
4790    IF A$="E" THEN Beg
4795    IF A$="R" THEN GOSUB Redraw
4800    IF A$="M" THEN GOTO Main_menu
4805  NEXT I
4810  OFF KEY
4815  GOTO Beg
4820    !
4825 Logic_element: ! LOGIC ELEMENT FUNCTION KEY MENU
4830  OFF KEY
4835  FOR I=1 TO 500
4840    ON KEY 0 LABEL "AND GATE" GOTO Ag
4845    ON KEY 1 LABEL "OR GATE" GOTO Og
4850    ON KEY 2 LABEL "INVERTER" GOTO In2
4855    ON KEY 3 LABEL "EXCLUSIVE OR" GOTO Er
4860    ON KEY 5 LABEL "NAND GATE" GOTO Ng
4865    ON KEY 6 LABEL "NOR GATE" GOTO Nr
4870    ON KEY 7 LABEL "D-FLIP FLOP" GOTO Df
4875    ON KEY 8 LABEL "JK-FLIP FLOP" GOTO Jk
4880    ON KBD,15 RECOVER 4890
4885    GOTO 4910
4890    A$=KBD$
4895    IF A$="E" THEN Beg
4900    IF A$="R" THEN GOSUB Redraw
4905    IF A$="M" THEN GOTO Main_menu
4910  NEXT I
4915  OFF KEY
4920  GOTO Beg
4925    !
4930 Shapes: ! SHAPE FUNCTION KEY MENU
4935  OFF KEY
4940  FOR I=1 TO 500
4945    ON KEY 0 LABEL "CIRCLE LINES" GOTO Ci
4950    ON KEY 2 LABEL "SQUARE LINES" GOTO Lq
4955    ON KEY 7 LABEL "SQUARE" GOTO Sq
4960    ON KEY 5 LABEL "CIRCLE" GOTO Cl
4965    ON KEY 1 LABEL "SUMMER" GOTO Summ
4970    ON KEY 8 LABEL "MOTOR" GOTO Mot
4975    ON KEY 4 LABEL "BIRD" GOTO Br
4980    ON KEY 3 LABEL "RECTANGLE" GOTO Re2
4985    ON KEY 9 LABEL "CHIPS" GOTO Chips
4990    ON KBD,15 RECOVER 5000
4995    GOTO 5020
5000    A$=KBD$
5005    IF A$="E" THEN Beg
5010    IF A$="R" THEN GOSUB Redraw
5015    IF A$="M" THEN GOTO Main_menu
```

```
5020   NEXT I
5025   OFF KEY
5030   GOTO Beg
5035     !
5040 Chips: ! CHIP PACKAGE FUNCTION KEY MENU
5045   OFF KEY
5050   FOR I=1 TO 500
5055    ON KEY 0 LABEL "14-PIN" GOTO Chip14
5060    ON KEY 1 LABEL "16-PIN" GOTO Chip16
5065    ON KEY 2 LABEL "24-PIN" GOTO Chip24
5070    ON KBD,15 RECOVER 5080
5075    GOTO 5100
5080    A$=KBD$
5085    IF A$="E" THEN Beg
5090    IF A$="R" THEN GOSUB Redraw
5095    IF A$="M" THEN GOTO Main_menu
5100   NEXT I
5105   OFF KEY
5110   Flag=0
5115   GOTO Beg
5120    !
5125! ********************************************
5130! *   FAST ACCESS TO THE TREE-LINK PROGRAM   *
5135! *   DIRECTLY FROM THE MANIPULATION MENU    *
5140! ********************************************
5145 Analyze:  !
5150   Back_to_analyze=1
5155! CHECKING TO SEE IF THE CIRCUIT WAS STORED BEFORE
5160! GOING TO THE ANALYZE PROGRAM
5165   IF Analyze_draw=0 THEN
5170    GCLEAR
5175    CSIZE 4
5180    MOVE 300,400
5185    LABEL "PLEASE STORE YOUR CIRCUIT FIRST"
5190    LABEL "OR IT WILL BE PURGED"
5195   ELSE
5200    GCLEAR
5205    CSIZE 4
5210    MOVE 300,400
5215    LABEL "PLEASE SELECT"
5220   END IF
5225! ANOTHER CHANCE TO CONFIRM THE DECISION
5230! TO ANALYZE THE CIRCUIT BY CALLING
5235! THE TREE-LINK PROGRAM
5240   OFF KEY
5245   FOR I=1 TO 700
5250    ON KEY 0 LABEL "STORE" GOTO Store1
5255    ON KEY 5 LABEL "RETURN" GOTO 5315
5260    ON KEY 2 LABEL "ANALYZE" GOTO Call_tree
5265    IF Analyze_draw=0 THEN ON KEY 2 LABEL "" GOTO Call_tree
5270    ON KBD,15 RECOVER 5285
5275    GOTO 5300
5280! R AND M KEY ACTIVE
```

```
5285    A$=KEY$
5290    IF A$="R" THEN GOSUB Redraw
5295    IF A$="M" THEN GOTO Main_menu
5300   NEXT I
5305   OFF KEY
5310   Flag=0
5315   Back_to_analyze=0
5320   GOSUB Redraw
5325   GOTO Beg
5330   !
5335 Call_tree: !
5340! LOADING THE TREE-LINK PROGRAM
5345   LOAD "TREE"
5350! END OF THE CIRCUIT DRAWING PROGRAM
5355    !
5360! ****************************************
5365! * CIRCUIT ELEMENTS ARE GIVEN A CODE    *
5370! * REPRESENTED BY VARIABLE EE           *
5375! ****************************************
5380 Ac:Ee=3
5385   GOTO Draw1
5390 A_:Ee=4
5395   GOTO Draw1
5400 Cs:Ee=5
5405   GOTO Draw1
5410 Cu:Ee=6
5415   GOTO Draw1
5420 Ba:Ee=7
5425   GOTO Draw1
5430 Dc:Ee=8
5435   GOTO Draw1
5440 Ar:Ee=9
5445   GOTO Draw1
5450 Lo:Ee=10
5455   GOTO Draw1
5460 Di:Ee=11
5465   GOTO Draw1
5470 Gr:Ee=12
5475   GOTO Draw1
5480 Rel:Ee=13
5485   GOTO Draw1
5490 Ca:Ee=14
5495   GOTO Draw1
5500 El:Ee=15
5505   GOTO Draw1
5510 In1:Ee=16
5515   GOTO Draw1
5520 Ch:Ee=17
5525   GOTO Draw1
5530 Im:Ee=18
5535   GOTO Draw1
5540 Ad:Ee=19
5545   GOTO Draw1
```

```
5550 Di1:Ee=20
5555   GOTO Draw1
5560 Th:Ee=21
5565   GOTO Draw1
5570 Ze:Ee=22
5575   GOTO Draw1
5580 Td:Ee=23
5585   GOTO Draw1
5590 Bi:Ee=24
5595   GOTO Draw1
5600 Bd:Ee=25
5605   GOTO Draw1
5610 Mo:Ee=26
5615   GOTO Draw1
5620 Md:Ee=27
5625   GOTO Draw1
5630 Sw:Ee=28
5635   GOTO Draw1
5640 Tr:Ee=29
5645   GOTO Draw1
5650 Si:Ee=30
5655   GOTO Draw1
5660 St:Ee=31
5665   GOTO Draw1
5670 Tt:Ee=32
5675   GOTO Draw1
5680 De:Ee=33
5685   GOTO Draw1
5690 Wy:Ee=34
5695   GOTO Draw1
5700 Ct:Ee=35
5705   GOTO Draw1
5710 Db:Ee=36
5715   GOTO Draw1
5720 Ag:Ee=37
5725   GOTO Draw1
5730 Og:Ee=38
5735   GOTO Draw1
5740 In2:Ee=39
5745   GOTO Draw1
5750 Ng:Ee=40
5755   GOTO Draw1
5760 Nr:Ee=41
5765   GOTO Draw1
5770 Er:Ee=42
5775   GOTO Draw1
5780 Df:Ee=43
5785   GOTO Draw1
5790 Jk:Ee=44
5795   GOTO Draw1
5800 Re2:Ee=45
5805   GOTO Draw1
5810 Ci:Ee=46
```

```
5815   GOTO Draw1
5820 C1:Ee=47
5825   GOTO Draw1
5830 Lq:Ee=48
5835   GOTO Draw1
5840 Sq:Ee=49
5845   GOTO Draw1
5850 Br:Ee=50
5855   GOTO Draw1
5860 Mot:Ee=52
5865   GOTO Draw1
5870 Summ:Ee=53
5875   GOTO Draw1
5880 Chip14:Ee=54
5885   GOTO Draw1
5890 Chip16:Ee=55
5895   GOTO Draw1
5900 Chip24:Ee=56
5905   GOTO Draw1
5910 Inver1:Ee=60
5915   GOTO Draw1
5920 Inver2:Ee=61
5925   GOTO Draw1
5930 Inver3:Ee=62
5935   GOTO Draw1
5940 Rectifier1:Ee=63
5945   GOTO Draw1
5950 Rectifier2:Ee=64
5955   GOTO Draw1
5960 Rectifier3:Ee=65
5965   GOTO Draw1
5970     !
5975 Do:      ! DOT ROUTINE
5980   E(1,0)=E(1,0)+1
5985   E(1,E(1,0)*4-3)=Px1
5990   E(1,E(1,0)*4-2)=Py1
5995   E(1,E(1,0)*4-1)=Px2
6000   E(1,E(1,0)*4)=Py2
6005   PEN Aa
6010   MOVE Px1+2,Py1+2
6015   POLYGON 3.5,30
6020   POLYGON 3,30
6025   POLYGON 2,30
6030   POLYGON 1,30
6035   POLYGON .5,30
6040   OFF KEY
6045   GOTO Beg
6050     !
6055 Dot_1:    ! DOTTED LINE ROUTINE
6060   E(51,0)=E(51,0)+1
6065   E(51,E(51,0)*4-3)=Px1
6070   E(51,E(51,0)*4-2)=Py1
6075   E(51,E(51,0)*4-1)=Px2
```

```
6080   E(51,E(51,0)*4)=Py2
6085   PEN Aa
6090   LINE TYPE 4
6095   MOVE Px1+2,Py1+2
6100   DRAW Px2+2,Py2+2
6105   LINE TYPE 1
6110   OFF KEY
6115   GOTO Beg
6120 Li:     ! STRAIGHT LINE ROUTINE
6125   E(2,0)=E(2,0)+1
6130   E(2,E(2,0)*4-3)=Px1
6135   E(2,E(2,0)*4-2)=Py1
6140   E(2,E(2,0)*4-1)=Px2
6145   E(2,E(2,0)*4)=Py2
6150   PEN Aa
6155   MOVE Px1+2,Py1+2
6160   DRAW Px2+2,Py2+2
6165   OFF KEY
6170   GOTO Beg
6175 Draw1:     ! CIRCUIT ELEMENT DRAWING ROUTINE
6180   GOSUB Error_control
6185   OFF KEY
6190! ONLY FOR EE(INVER 3). IF PHASE B DOES NOT FALL
6195! ON GRID, CHANGE PHASE A SO THAT PHASE B DOES FALL
6200! ON A GRID POINT
6205! THIS IS IMPORTANT SINCE THE CIRCUIT WILL HAVE
6210! CONTINUITY FOR ANY ELEMENTS CONNECTED ON PHASE B OF
6215! THE THREE PHASE INVERTER (A THREE TERMINAL DEVICE)
6220   IF Ee=62 THEN
6225    Ass=(Py2-Py1)/2/4
6230    Ass1=INT((Py2-Py1)/2/4)+.1
6235    IF Ass>Ass1 THEN Py2=Py2+4
6240   END IF
6245   E(Ee,0)=E(Ee,0)+1
6250   E(Ee,E(Ee,0)*4-3)=Px1
6255   E(Ee,E(Ee,0)*4-2)=Py1
6260   E(Ee,E(Ee,0)*4-1)=Px2
6265   E(Ee,E(Ee,0)*4)=Py2
6270   PEN Aa
6275   Angle=0
6280   ON ERROR GOTO Angle1
6285! FINDING THE ANGLE OF THE ELEMENT
6290   Angle=ATN((Py2-Py1)/(Px2-Px1))
6295   GOTO Start1
6300 Angle1:Angle=90
6305 Start1:                !
6310! FINDING THE MIDDLE POINT
6315   M1=(Px1+Px2)/2
6320   M2=(Py1+Py2)/2
6325   IF Py2-Py1=0 AND Px2<Px1 THEN Angle=180
6330   IF Py2<Py1 AND Px2-Px1=0 THEN Angle=270
6335   IF Py2-Py1<>0 AND Px1-Px2<>0 THEN
6340    IF Px1>Px2 AND Py1>Py2 THEN Angle=Angle+180
```

```
6345   IF Px1>Px2 AND Py1<Py2 THEN Angle=Angle+180
6350   END IF
6355! THE END POINT OF THE ELEMENT IS CALCULATED
6360! AND STORED IN A VARIABLE PZ
6365   Pz=SQR((Px2-Px1)^2+(Py2-Py1)^2)/2
6370   SELECT Ee
6375   CASE =6
6380    Ff=20
6385   CASE =10
6390    Ff=16/(S+1.E-39)
6395   CASE ELSE
6400    Ff=16
6405   END SELECT
6410   Aaa=Ff*(Px2-Px1)/((Px2-Px1)^2+(Py2-Py1)^2)^.5
6415   Bbb=Ff*(Py2-Py1)/((Px2-Px1)^2+(Py2-Py1)^2)^.5
6420   Bbb=Bbb*S
6425   Aaa=Aaa*S
6430   Pz=SQR((Px2-Px1)^2+(Py2-Py1)^2)/2
6435   MOVE M1+2,M2+2
6440   PIVOT Angle-90
6445! THE TYPE OF ELEMENT IS CALLED UP FROM THE ELEMENT
6450! SUBROUTINES LOCATED AT THE END OF THIS PROGRAM
6455! THEY ARE LABELED OR CALLED BY NAME
6460   SELECT Ee
6465   CASE =3
6470    GOSUB Ac_volt
6475   CASE =4
6480    GOSUB A_volt
6485   CASE =5
6490    GOSUB Current_sour
6495   CASE =6
6500    GOSUB Current
6505   CASE =7
6510    GOSUB Battery
6515   CASE =8
6520    GOSUB Dc_source
6525   CASE =9
6530    GOSUB Arrow
6535   CASE =10
6540    GOSUB Looped
6545   CASE =11
6550    GOSUB Direc
6555   CASE =12
6560    GOSUB Ground1
6565   CASE =13
6570    GOSUB Resistor
6575   CASE =14
6580    GOSUB Capacitor
6585   CASE =15
6590    GOSUB Electrolytic
6595   CASE =16
6600    GOSUB Inductor
6605   CASE =17
```

```
6610    GOSUB Choke
6615   CASE =18
6620    GOSUB Impedance
6625   CASE =19
6630    GOSUB Admitance
6635   CASE =20
6640    GOSUB Diode
6645   CASE =21
6650    GOSUB Thyristor
6655   CASE =22
6660    GOSUB Zener
6665   CASE =23
6670    GOSUB Thyristor_d
6675   CASE =24
6680    GOSUB Bipolar
6685   CASE =25
6690    GOSUB Bipolar_d
6695   CASE =26
6700    GOSUB Mosfet
6705   CASE =27
6710    GOSUB Mosfet_d
6715   CASE =28
6720    GOSUB Switch
6725   CASE =29
6730    GOSUB Triac
6735   CASE =30
6740    GOSUB T_supressor
6745   CASE =31
6750    GOSUB Single_t
6755   CASE =32
6760    GOSUB Three_t
6765   CASE =33
6770    GOSUB Delta
6775   CASE =34
6780    GOSUB Wye
6785   CASE =35
6790    GOSUB Center_t
6795   CASE =36
6800    GOSUB Diode_brid
6805   CASE =37
6810    GOSUB And_g
6815   CASE =38
6820    GOSUB Or_g
6825   CASE =39
6830    GOSUB Inver
6835   CASE =40
6840    GOSUB Nand_g
6845   CASE =41
6850    GOSUB Nor_g
6855   CASE =42
6860    GOSUB Ex_or
6865   CASE =43
6870    GOSUB D_flip
```

```
6875   CASE =44
6880    GOSUB Jk_flip
6885   CASE =45
6890    GOSUB Rect1
6895   CASE =46
6900    GOSUB Circle_1
6905   CASE =47
6910    GOSUB Circle1
6915   CASE =48
6920    GOSUB Square_1
6925   CASE =49
6930    GOSUB Square
6935   CASE =50
6940    GOSUB Bird
6945   CASE =52
6950    GOSUB Motor
6955   CASE =53
6960    GOSUB Summer
6965   CASE =54
6970    GOSUB C14_pin
6975   CASE =55
6980    GOSUB C16_pin
6985   CASE =56
6990    GOSUB C24_pin
6995   CASE =60
7000    GOSUB Half_bridge_in
7005   CASE =61
7010    GOSUB Full_bridge_1in
7015   CASE =62
7020    GOSUB Full_bridge_3in
7025   CASE =63
7030    GOSUB Phase1_rect
7035   CASE =64
7040    GOSUB Phase3_rect_d
7045   CASE =65
7050    GOSUB Phase3_rect_y
7055   END SELECT
7060   PIVOT 0
7065   OFF ERROR
7070   ON ERROR GOSUB Err1
7075   Flag=0
7080   GOTO Beg
7085          !
7090 Store1: ! FLOPPY DISK STORAGE ROUTINE
7095! FIND WHICH DRIVE HAS THE DATA DISK
7100    Hihi$=SYSTEM$("MSI")&"    "
7105    IF Hihi$[15,15]="1" THEN
7110     MASS STORAGE IS ":INTERNAL,4,0"
7115    ELSE
7120     MASS STORAGE IS ":INTERNAL,4,1"
7125    END IF
7130    OFF KBD
7135    OFF KNOB
```

```
7140   PEN 1
7145   Nn1=0
7150   OFF ERROR
7155   OFF KEY
7160   LORG 4
7165   GCLEAR
7170   CSIZE 6,.4
7175   MOVE 300,500
7180   LABEL "CHECKING FOR CIRCUIT FILES"
7185   FOR I=0 TO 8
7190    Zx$(I)=""
7195   NEXT I
7200   CLIP OFF
7205   CSIZE 3,.5
7210! LOAD ALREADY EXISTING FILE NAMES IN ARRAY CALLED CAT1$
7215   CAT TO Cat1$(*);SELECT "_",COUNT Nn,NO HEADER
7220   GCLEAR
7225   MOVE 160,550
7230   LABEL "* ALREADY EXISTING CIRCUIT FILES *"
7235! PRINT ALL FILE NAMES ON SCREEN
7240   FOR I=0 TO Nn
7245    IF Cat1$(I)="" THEN 7270
7250    Nn1=Nn1+1
7255    LABEL Cat1$(I)[2,9]
7260    IF Nn1=17 THEN MOVE 300;530
7265    IF Nn1=34 THEN MOVE 420,530
7270   NEXT I
7275   MOVE 240,210
7280! LIST OF EXISTING FILES ARE SHOWN SO THAT THE
7285! USER DOES NOT COPY AN-EXISTING FILE NAME
7290   LABEL "NOTE: USAGE OF THESE FILE NAMES WILL PURGE OLD FILE"
7295   CSIZE 5,.3
7300! A SMALL FRAMED RECTANGLE IS DRAWN ON THE SCREEN SO
7305! THAT THE USER CAN ENTER A FILE NAME WITH THE
7310! RIGHT NUMBER OF CHARACTERS QUICKLY
7315   MOVE 150,180
7320   LABEL "PLEASE ENTER FILE NAME"
7325   AREA INTENSITY 1,1,1
7330   MOVE 176,80
7335   RECTANGLE 330,72,FILL
7340   CSIZE 5
7345   AREA INTENSITY 0,0,0
7350   MOVE 196,100
7355   RECTANGLE 286,32,FILL
7360   CLIP 196,484,0,200
7365   Ppx=196
7370   Ppy=100
7375   MOVE Ppx,Ppy
7380   RECTANGLE 32,32
7385 Beg1:    !
7390   PEN 1
7395   ALPHA OFF
7400   Dflag=1
```

```
7405   MOVE Ppx,Ppy
7410   RECTANGLE 32,32
7415   Aa=1
7420! THE KNOB CAN BE USED TO MOVE THE CURSOR FOR
7425! EDITING A CHARACTER
7430!
7435!,CHARACTERS ARE REPLACED BY PLACING THE CURSOR
7440! OVER THE CHARACTER AND PRESSING A NEW CHARACTER
7445   ON KNOB .030'GOTO Xxy
7450 Xxy:!
7455   ON KBD GOTO 7460
7460   Ab$=""
7465   Ab$=KBD$
7470   Hh=2
7475! UNACCEPTABLE CHARACTERS FOR FILE NAMES ARE NOT ALLOWED
7480! TO BE USED
7485   IF Ab$=" " OR Ab$="!" OR Ab$="@" OR Ab$="#" OR Ab$="$" THEN Beg1
7490   IF Ab$="%" OR Ab$="^" OR Ab$="&" OR Ab$="*" OR Ab$="(" THEN Beg1
7495   IF Ab$=")" OR Ab$="-" OR Ab$="+" OR Ab$="=" OR Ab$="(" THEN Beg1
7500   IF Ab$=")" OR Ab$="[" OR Ab$="]" OR Ab$="<" OR Ab$=">" THEN Beg1
7505   IF Ab$="?" OR Ab$="/" OR Ab$="." OR Ab$="," OR Ab$=";" THEN Beg1
7510   IF Ab$=":" OR Ab$="'" OR Ab$="""" THEN Beg1
7515   IF Ab$="_" THEN Hh=-5
7520! PRESSING THE CONTINUE KEY WILL FORCE THE PROGRAM TO
7525! IGNORE THIS ROUTINE, REDRAW THE CIRCUIT, AND
7530! GO BACK TO BEG
7535   IF Ab$=CHR$(255)&"C" THEN
7540    OFF KNOB
7545    OFF KBD
7550    GOTO 8400
7555   END IF
7560! THE LEFT AND RIGHT ARROW KEYS CAN ALSO BE USED
7565! USED FOR EDITING
7570   IF Ab$=CHR$(255)&"<" THEN GOTO L_dir
7575   IF Ab$=CHR$(255)&">" THEN GOTO R_dir
7580! PRESSING THE CLEAR LINE KEY WILL DELETE
7585! THE WHOLE NAME
7590   IF Ab$=CHR$(255)&"#" THEN GOTO Clr_ln
7595! PRESSING THE DELETE CHARACTER KEY WILL DELETE
7600! THE CHARACTER
7605   IF Ab$=CHR$(255)&"-" THEN GOSUB Del_chr
7610! THE ENTER KEY WILL ENTER THE NAME
7615   IF Ab$=CHR$(255)&"E" THEN
7620    OFF KNOB
7625    GOTO Fin
7630   END IF
7635   IF Ab$[1,1]=CHR$(255) THEN GOTO Beg1
7640   Ab$=Ab$[1,1]
7645   IF Ab$<>"" THEN
7650    IF Zx$((Ppx-196)/32)<>"" THEN GOSUB Del_chr
7655    Zx$((Ppx-196)/32)=Ab$
7660    PEN 1
7665    MOVE Ppx+17,Ppy-Hh
```

```
7670    LABEL Ab$
7675    WAIT .2
7680    X=32
7685    GOTO Cont1
7690   END IF
7695 Ppx:!
7700    X=KNOBX*32
7705 Cont1: !
7710    IF X>=0 THEN
7715     PEN -1
7720     MOVE Ppx,Ppy
7725     RECTANGLE 32,32
7730     Ppx=Ppx+X
7735     MOVE Ppx,Ppy
7740     IF Ppx>452 THEN Ppx=452
7745     PEN 1
7750     RECTANGLE 32,32
7755    ELSE
7760     PEN -1
7765     MOVE Ppx,Ppy
7770     RECTANGLE 32,32
7775     PEN 1
7780     Ppx=Ppx+X
7785     MOVE Ppx,Ppy
7790     IF Ppx<196 THEN Ppx=196
7795     PEN 1
7800     RECTANGLE 32,32
7805    END IF
7810    GOTO Beg1
7815 L_dir:  ! MOVE GRAPHIC CURSOR ONE SPACE TO THE LEFT
7820    X=-32
7825    GOTO Cont1
7830 R_dir: ! MOVE GRAPHIC CURSOR ONE SPACE TO THE RIGHT
7835    X=32
7840    GOTO Cont1
7845 Clr_ln: ! CLEAR WHOLE LINE ROUTINE
7850    MOVE 196,100
7855    RECTANGLE 288,32,FILL
7860    Ppx=196
7865    GOTO Beg1
7870 Del_chr:       ! DELETE CHARACTER ROUTINE
7875    Zx$((Ppx-196)/32)=""
7880    MOVE Ppx,Ppy
7885    RECTANGLE 32,32,FILL
7890    PEN 1
7895    RECTANGLE 32,32
7900! END OF ENTERING THE FILE NAME FOR STORING THE CIRCUIT
7905    RETURN
7910 Fin: ! FILE NAME
7915    Hi=0
7920    OFF KBD
7925    PEN 1
7930    CSIZE 5,.3
```

```
7935   MOVE 430,50
7940! PROGRAM BEING POLITE TO THE USER
7945   LABEL "THANK-YOU"
7950! CIRCUIT DRAWING IDENTIFICATION CHARACTER IS APPENED TO
7955! IS THE FIRST CHARACTER OF THE FILE NAME
7960! USED FOR FAST RETRIEVING
7965   Z$=TRIM$("_"&Zx$(0)&Zx$(1)&Zx$(2)&Zx$(3)&Zx$(4)&Zx$(5)&Zx$(6)&Zx$(7)&Zx$(
8))
7970! TOTAL NUMBER OF ELEMENTS IN FILE (HI)
7975   FOR I=1 TO 70
7980    IF E(I,0)=0 THEN 7990
7985    Hi=Hi+E(I,0)
7990   NEXT I
7995   Cll=0
8000! TOTAL NUMBER OF LABEL CHARACTERS IN FILE (CLL)
8005   FOR I=1 TO Lab(0,0)
8010    IF Lab(I,2)=0 THEN 8020
8015    Cll=Cll+1
8020   NEXT I
8025   IF Hi=0 THEN
8030    ALLOCATE INTEGER El(0)
8035    GOTO 8065
8040   END IF
8045! ALLOCATE A NEW ARRAY CALLED EL FOR THE ELEMENT ARRAY
8050! ELAB ARRAY FOR THE LABEL CHARACTER LOCATION ARRAY
8055! AND ELAB$ ARRAY FOR THE LABEL CHARACTER ARRAY
8060   ALLOCATE INTEGER El(Hi*5+1)
8065   ALLOCATE INTEGER Elab(Cll*5)
8070   ALLOCATE Elab$(Cll)[1]
8075   Ii=1
8080! THE NEXT TWO LOOPS MINIMIZE THE FLOPPY
8085! DISK SPACE THE FILE WILL OCCUPY
8090!
8095! THIS IS DONE BY REPACKING THE INFORMATION
8100! FROM THE TWO DIMENSIONAL ARRAYS E AND LAB
8105! INTO THE ONE DIMENSIONAL  ARRAYS EL AND ELAB
8110! RESPECTIVELY
8115   IF Hi=0 THEN 8180
8120   FOR I=1 TO 70
8125    IF E(I,0)=0 THEN 8170
8130    FOR J=4 TO E(I,0)*4 STEP 4
8135     Ii=Ii+5
8140     El(Ii-4)=I
8145     El(Ii-3)=E(I,J-3)
8150     El(Ii-2)=E(I,J-2)
8155     El(Ii-1)=E(I,J-1)
8160     El(Ii)=E(I,J)
8165    NEXT J
8170   NEXT I
8175   IF Cll=0 THEN 8265
8180   J=0
8185   Elab(0)=Cll
8190   FOR I=1 TO Lab(0,0)
```

```
8195    IF Lab(I,2)=0 THEN 8235
8200    J=J+1
8205    Elab(J*5-4)=Lab(I,0)
8210    Elab(J*5-3)=Lab(I,1)
8215    Elab(J*5-2)=Lab(I,2)*1000
8220    Elab(J*5-1)=Lab(I,3)
8225    Elab(J*5)=Lab(I,4)
8230    Elab$(J)=Lab$(I)
8235   NEXT I
8240! REPACKING INFORMATION INTO SINGLE DIMENSIONAL ARRAY
8245! ROUTINE IS FINISHED
8250!
8255! NUMBER OF CIRCUIT ELEMENTS IS STORED INTO THE FIRST
8260! ELEMENT OF THE ARRAY EL
8265   El(0)=Ii
8270! THE SIZE OF THE CIRCUIT ELEMENTS (S) IS STORED IN THE
8275! SECOND ELEMENT OF THE ARRAY EL ALSO TO MAKE THE SIZE
8280! AN INTEGER NUMBER IT IS MULTIPLIED BY 100
8285   IF Hi<>0 THEN El(1)=S*100
8290   IF Hi=0 THEN El(0)=0
8295   ON ERROR GOTO 8305
8300   PURGE Z$
8305   OFF ERROR
8310! FILES ARE CREATED WITH RELATIVE SIZES
8315   CREATE BDAT Z$,1,(Ii*5)*2+(Cll*5)*2+(Cll)*8
8320   ASSIGN @Path TO Z$
8325   OUTPUT @Path;El(*);
8330   OUTPUT @Path;Elab(*);
8335   OUTPUT @Path;Elab$(*)
8340   ASSIGN @Path TO *
8345   DEALLOCATE El(*)
8350   DEALLOCATE Elab(*),Elab$(*)
8355! MASS STORAGE DEVICE IS RETURNED TO THE PROGRAM DISK
8360   Hihi$=SYSTEM$("MSI")&"   "
8365   IF Hihi$[15,15]="1" THEN
8370    MASS STORAGE IS ":INTERNAL,4,0"
8375   ELSE
8380    MASS STORAGE IS ":INTERNAL,4,1"
8385   END IF
8390   Analyze_draw=1
8395   IF Back_to_analyze=1 THEN Analyze
8400   GCLEAR
8405   GOSUB Redraw
8410   GOTO Beg
8415          !
8420          !
8425 Recall: ! RECALL ROUTINE TO LOAD A CIRCUIT INTO MEMORY
8430   Hihi$=SYSTEM$("MSI")&"   "
8435! DATA DISK DRIVE IS FOUND
8440   IF Hihi$[15,15]="1" THEN
8445    MASS STORAGE IS ":INTERNAL,4,0"
8450   ELSE
8455    MASS STORAGE IS ":INTERNAL,4,1"
```

```
8460   END IF
8465   LORG 4
8470   OFF KEY
8475   OFF ERROR
8480   GCLEAR
8485   PEN 1
8490   F=0
8495   CSIZE 5,.4
8500   MOVE 400,400
8505! ONLY CIRCUIT FILES WITH THE SPECIAL
8510! IDENTIFICATION CHARACTER ARE CHECKED
8515   LABEL "CHECKING FOR CIRCUIT FILES"
8520   CAT TO Cat1$(*);SELECT "_",COUNT Nn,NO HEADER
8525   IF Nn=0 THEN
8530    LABEL "SORRY, NO CIRCUIT FILES FOUND"
8535    WAIT 3
8540    GOTO Out1
8545   END IF
8550   Analyze_draw=1
8555   GCLEAR
8560   CSIZE 4,.4
8565   MOVE 500,100
8570   LABEL "PLEASE ENTER FILE NAME"
8575   CSIZE 2.6,.8
8580   MOVE 145,565
8585   LABEL "NO. FILE NAME"
8590   MOVE 154,545
8595! LIST OF CIRCUIT FILES STORED ON THE FLOPPY DISK ARE
8600! DISPLAYED ON THE SCREEN
8605   FOR I=0 TO Nn
8610    IF Cat1$(I)="" THEN 8625
8615    Nn1=Nn1+1
8620    LABEL I+1;"...";Cat1$(I)[2,10]
8625   NEXT I
8630   ALLOCATE Arr(7,2)
8635   READ Arr(*)
8640   DATA -32,8,1,  0,8,1,   0,16,1,  32,0,1,  0,-16,1,
8645   DATA 0,-8,1,   -32,-8,1,    -32,8,2
8650   RESTORE
8655   OFF KNOB
8660   OFF ERROR
8665   GRAPHICS ON
8670   CLIP 8,792,16,586
8675   Ppy=554
8680   Ppx=40
8685 Beg2:    !
8690   CSIZE 3,.5
8695   AREA PEN 1
8700   MOVE Ppx,Ppy
8705! A LARGE ARROW IS DRAWN BESIDE THE COLUMN
8710! OF THE FILE NAMES
8715   SYMBOL Arr(*),FILL
8720! THE KNOB IS USED TO MOVE THE ARROW VERTICALLY
```

```
8725! TO POINT AT ANY FILE NAME
8730   ON KNOB .04 GOTO Xy2
8735 Xy2:!
8740   ALPHA OFF
8745   Ppy2=Ppy
8750   Ppx2=Ppx
8755   OFF ERROR
8760   ON KBD GOTO 8765
8765   Ab$=KBD$
8770! THE NUMERICAL KEYS ARE USED TO MOVE THE ARROW
8775! IN JUMPS FOR QUICK SELECTION OF FILES (1 TO 10)
8780   IF Ab$="1" OR Ab$="2" OR Ab$="3" OR Ab$="4" OR Ab$="5" OR Ab$="6" OR Ab$=
"7" OR Ab$="8" OR Ab$="9" OR Ab$="0" THEN
8785     Ppy=554-((VAL(Ab$)-1)*16)
8790     IF Ab$="0" THEN Ppy=410
8795     OFF KBD
8800     GOTO 9195
8805   END IF
8810! THE NUMERICAL KEYS USING THE SHIFT KEY ARE
8815! FOR FILES 11 TO 20
8820   IF Ab$="!" OR Ab$="@" OR Ab$="#" OR Ab$="$" OR Ab$="%" OR Ab$="^" OR Ab$=
"&" OR Ab$="*" OR Ab$="(" OR Ab$=")" THEN
8825     IF Ab$="!" THEN Ppy=394
8830     IF Ab$="@" THEN Ppy=378
8835     IF Ab$="#" THEN Ppy=362
8840     IF Ab$="$" THEN Ppy=346
8845     IF Ab$="%" THEN Ppy=330
8850     IF Ab$="^" THEN Ppy=314
8855     IF Ab$="&" THEN Ppy=298
8860     IF Ab$="*" THEN Ppy=282
8865     IF Ab$="(" THEN Ppy=266
8870     IF Ab$=")" THEN Ppy=250
8875     OFF KBD
8880     GOTO 9195
8885   END IF
8890! PRESSING THE CONTINUE KEY WILL FORCE THE PROGRAM
8895! TO IGNORE THIS ROUTINE, REDRAW THE PREVIOUS CIRCUIT
8900! AND GO BACK TO BEG
8905   IF Ab$=CHR$(255)&"C" THEN Ou
8910! PRESSING THE SHIFT AND UP ARROW KEYS WILL MOVE THE
8915! ARROW TO THE TOP OF THE SCREEN
8920   IF Ab$=CHR$(255)&"T" THEN
8925     Ppy=38
8930     OFF KBD
8935     GOTO 9195
8940   END IF
8945! PRESSING THE SHIFT AND DOWN ARROW KEYS WILL MOVE THE
8950! ARROW TO THE BOTTOM OF THE SCREEN
8955   IF Ab$=CHR$(255)&"W" THEN
8960     Ppy=554
8965     OFF KBD
8970     GOTO 9195
8975   END IF
```

```
8980! THE UP ARROW KEY IS ALSO USED TO MOVE THE ARROW
8985! UP ONE FILE NAME AT A TIME
8990   IF Ab$=CHR$(255)&"^" THEN
8995    OFF KBD
9000    Y=Y+16
9005    IF Ppy2+Y<38 OR Ppy2+Y>554 THEN Beg2
9010    OFF KBD
9015    GOTO 9195
9020   END IF
9025! THE DOWN ARROW KEY IS ALSO USED TO MOVE THE ARROW
9030! DOWN ONE FILE NAME AT A TIME
9035   IF Ab$=CHR$(255)&"V" THEN
9040    Y=Y-16
9045    IF Ppy2+Y<38 OR Ppy2+Y>554 THEN Beg2
9050    OFF KBD
9055    GOTO 9195
9060   END IF
9065! PRESSING THE ENTER KEY WILL LOAD THE CIRCUIT FILE
9070! INDICATED GRAPHICALLY BY THE LARGE
9075! ARROW DISPLAYED ON THE SCREEN
9080   IF Ab$=CHR$(255)&"E" THEN
9085    CSIZE 4,.4
9090    IF (570-Ppy)/16>Nn THEN
9095     IF F=1 THEN Ou
9100     PEN -1
9105     MOVE 500,100
9110     LABEL "PLEASE ENTER FILE NAME"
9115     PEN 1
9120     MOVE 500,100
9125     LABEL "SELECT A FILE OR PRESS CONTINUE"
9130     F=1
9135     GOTO Py2
9140    END IF
9145    Z$=Cat1$((554-Ppy)/16)[1,10]
9150    MOVE 510,80
9155    LABEL "THANK-YOU"
9160    WAIT .5
9165    GOTO Rec
9170   END IF
9175 Py2: !
9180   Y=KNOBX*(-16)
9185   IF Y=0 THEN Beg2
9190   IF Ppy2+Y<38 OR Ppy2+Y>554 THEN Beg2
9195   IF Y>=0 THEN
9200    AREA PEN -1
9205    PEN -1
9210    MOVE Ppx2,Ppy2
9215    SYMBOL Arr(*),FILL
9220    Ppy=Ppy+Y
9225    MOVE Ppx,Ppy
9230    AREA PEN 1
9235    PEN 1
9240    SYMBOL Arr(*),FILL
```

```
9245    GOTO Beg2
9250   ELSE
9255    AREA PEN -1
9260    PEN -1
9265    MOVE Ppx2,Ppy2
9270    SYMBOL Arr(*),FILL
9275    AREA PEN 1
9280    PEN 1
9285    Ppy=Ppy+Y
9290    MOVE Ppx,Ppy
9295    PEN 1
9300    SYMBOL Arr(*),FILL
9305   END IF
9310   GOTO Beg2
9315 Rec:    !
9320   MAT E= (0)
9325   OFF KNOB
9330   GCLEAR
9335   MOVE 400,400
9340   LABEL "LOADING CIRCUIT FILE ";Z$[2,9]
9345   ASSIGN @Path TO Z$
9350   ENTER @Path;Li
9355   IF Li<>0 THEN
9360    ALLOCATE INTEGER El(Li-1)
9365   ELSE
9370    ALLOCATE INTEGER El(Li)
9375   END IF
9380   IF Li<>0 THEN ENTER @Path;El(*)
9385   ENTER @Path;Cll
9390   IF Cll<>0 THEN
9395    ALLOCATE INTEGER Elab(Cll*5-1)
9400   ELSE
9405    ALLOCATE INTEGER Elab(Cll*5)
9410   END IF
9415! FILE IS LOADED AND THE DATA IS TRANSFERED INTO THE
9420! TWO DIMENSIONAL ARRAYS E AND LAB
9425   ALLOCATE Elab$(Cll)[1]
9430   IF Cll<>0 THEN ENTER @Path;Elab(*);Elab$(*)
9435   ASSIGN @Path TO *
9440   S=El(0)/100
9445   K=0
9450   FOR I=1 TO Li-1 STEP 5
9455    IF K=El(I) THEN
9460     J=J+4
9465    ELSE
9470     J=4
9475    END IF
9480    K=El(I)
9485    E(El(I),0)=E(El(I),0)+1
9490    E(El(I),J-3)=El(I+1)
9495    E(El(I),J-2)=El(I+2)
9500    E(El(I),J-1)=El(I+3)
9505    E(El(I),J)=El(I+4)
```

```
9510   NEXT I
9515     !
9520   MAT Lab= (0)
9525   MAT Lab$= ("")
9530   Lab(0,0)=Cll
9535   FOR I=1 TO Lab(0,0)
9540    Lab(I,0)=Elab(I*5-5)
9545    Lab(I,1)=Elab(I*5-4)
9550    Lab(I,2)=Elab(I*5-3)/1000
9555    Lab(I,3)=Elab(I*5-2)
9560    Lab(I,4)=Elab(I*5-1)
9565    Lab$(I)=Elab$(I)
9570   NEXT I
9575   DEALLOCATE El(*)
9580   DEALLOCATE Elab(*),Elab$(*)
9585 Ou:DEALLOCATE Arr(*)
9590 Out1:GOSUB Redraw
9595   Hihi$=""
9600   Hihi$=SYSTEM$("MEI")&"    "
9605   IF Hihi$[15,15]="1" THEN
9610    MASS STORAGE IS ":INTERNAL,4,0"
9615   ELSE
9620    MASS STORAGE IS ":INTERNAL,4,1"
9625   END IF
9630! END OF THE RECALL ROUTINE
9635   GOTO Beg
9640   GOTO En
9645          !
9650 Plot1:  ! ROUTINE TO SEND CIRCUIT TO A HP PLOTTER
9655   Dflag=0
9660 !ON TIMEOUT 7,.1 GOTO 8420
9665   PLOTTER IS 705,"HPGL"
9670   GOTO 9690
9675   DISP "SORRY, PLOTTER IS NOT CONNECTED PLEASE TRY AGAIN"
9680   WAIT 3
9685   GOTO Manipulation
9690   GOSUB Redraw
9695   GOTO Beg
9700 Redraw: !
9705   OFF KNOB
9710   OFF KBD
9715   IF Dflag=1 THEN
9720    GINIT
9725    VIEWPORT 0,131,0,100
9730    WINDOW 0,800,0,611
9735    GCLEAR
9740    GRAPHICS ON
9745    ALPHA OFF
9750    FRAME
9755    CLIP 8,792,8,600
9760   END IF
9765   FOR I=1 TO 70
9770    IF E(I,0)=0 THEN 9885
```

```
9775    FOR J=4 TO E(I,0)*4 STEP 4
9780     IF E(I,J-3)=0 THEN 9880
9785     Ee=I
9790     Px1=E(I,J-3)
9795     Py1=E(I,J-2)
9800     Px2=E(I,J-1)
9805     Py2=E(I,J)
9810                    !
9815     Flag=0
9820     SELECT Ee
9825     CASE =1
9830      GOSUB Pdo
9835     CASE =2
9840      GOSUB Pli
9845     CASE =51
9850      GOSUB Pdot_1
9855     CASE ELSE
9860      GOSUB Pdraw1
9865     END SELECT
9870                    !
9875    !
9880    NEXT J
9885   NEXT I
9890   LORG 5
9895   FOR I=1 TO Lab(0,0)
9900    IF Lab(I,3)=0 THEN 9945
9905    Cc=Lab(I,2)
9910    Px6=Lab(I,3)
9915    Py6=Lab(I,4)
9920    LDIR 0
9925    IF Lab(I,1)=1 THEN LDIR 90
9930    MOVE Px6+(Cc+.743)/(2*(.243)),Py6+(Cc+.743)/(2*(.243))
9935    CSIZE Cc
9940    LABEL Lab$(I)
9945   NEXT I
9950 Kkkkk:    !
9955   PEN Up
9960   LDIR 0
9965   Flag=0
9970   PLOTTER IS 3,"INTERNAL"
9975   RETURN
9980              !
9985              !
9990              !
9995 Pdot_1:              !
10000  PEN Aa
10005  LINE TYPE 4,.5
10010  MOVE Px1+2,Py1+2
10015  DRAW Px2+2,Py2+2
10020  LINE TYPE 1
10025  RETURN
10030                   !
10035 Pdo:   ! DRAW DOTS
```

```
10040  PEN Aa
10045  MOVE Px1+2,Py1+2
10050  POLYGON 2,30
10055  POLYGON 1,30
10060  POLYGON .5,30
10065  OFF KEY
10070  RETURN
10075    !
10080 Pli:    ! DRAW LINE
10085  PEN Aa
10090  MOVE Px1+2,Py1+2
10095  DRAW Px2+2,Py2+2
10100  OFF KEY
10105  RETURN
10110 Pdraw1:    !
10115  GOSUB Error_control
10120  OFF KEY
10125  PEN Aa
10130  Angle=0
10135  ON ERROR GOTO Pangle1
10140  Angle=ATN((Py2-Py1)/(Px2-Px1))
10145  GOTO Pstart1
10150 Pangle1:Angle=90
10155 Pstart1:                !
10160  M1=(Px1+Px2)/2
10165  M2=(Py1+Py2)/2
10170  IF Py2-Py1=0 AND Px2<Px1 THEN Angle=180
10175  IF Py2<Py1 AND Px2-Px1=0 THEN Angle=270
10180  IF Py2-Py1<>0 AND Px1-Px2<>0 THEN
10185   IF Px1>Px2 AND Py1>Py2 THEN Angle=Angle+180
10190   IF Px1>Px2 AND Py1<Py2 THEN Angle=Angle+180
10195  END IF
10200  SELECT Ee
10205  CASE =6
10210   Ff=20
10215  CASE =10
10220   Ff=16/(S+1.E-39)
10225  CASE ELSE
10230   Ff=16
10235  END SELECT
10240  Aaa=Ff*(Px2-Px1)/((Px2-Px1)^2+(Py2-Py1)^2)^.5
10245  Bbb=Ff*(Py2-Py1)/((Px2-Px1)^2+(Py2-Py1)^2)^.5
10250  Bbb=Bbb*S
10255  Aaa=Aaa*S
10260  Pz=SQR((Px2-Px1)^2+(Py2-Py1)^2)/2
10265  MOVE M1+2,M2+2
10270  PIVOT Angle-90
10275  SELECT Ee
10280  CASE =3
10285   GOSUB Ac_volt
10290  CASE =4
10295   GOSUB A_volt
10300  CASE =5
```

```
10305   GOSUB Current_sour
10310  CASE =6
10315   GOSUB Current
10320  CASE =7
10325   GOSUB Battery
10330  CASE =8
10335   GOSUB Dc_source
10340  CASE =9
10345   GOSUB Arrow
10350  CASE =10
10355   GOSUB Looped
10360  CASE =11
10365   GOSUB Direc
10370  CASE =12
10375   GOSUB Ground1
10380  CASE =13
10385   GOSUB Resistor
10390  CASE =14
10395   GOSUB Capacitor
10400  CASE =15
10405   GOSUB Electrolytic
10410  CASE =16
10415   GOSUB Inductor
10420  CASE =17
10425   GOSUB Choke
10430  CASE =18
10435   GOSUB Impedance
10440  CASE =19
10445   GOSUB Admitance
10450  CASE =20
10455   GOSUB Diode
10460  CASE =21
10465   GOSUB Thyristor
10470  CASE =22
10475   GOSUB Zener
10480  CASE =23
10485   GOSUB Thyristor_d
10490  CASE =24
10495   GOSUB Bipolar
10500  CASE =25
10505   GOSUB Bipolar_d
10510  CASE =26
10515   GOSUB Mosfet
10520  CASE =27
10525   GOSUB Mosfet_d
10530  CASE =28
10535   GOSUB Switch
10540  CASE =29
10545   GOSUB Triac
10550  CASE =30
10555   GOSUB T_supressor
10560  CASE =31
10565   GOSUB Single_t
```

```
10570  CASE =32
10575   GOSUB Three_t
10580  CASE =33
10585   GOSUB Delta
10590  CASE =34
10595   GOSUB Wye
10600  CASE =35
10605   GOSUB Center_t
10610  CASE =36
10615   GOSUB Diode_brid
10620  CASE =37
10625   GOSUB And_g
10630  CASE =38
10635   GOSUB Or_g
10640  CASE =39
10645   GOSUB Inver
10650  CASE =40
10655   GOSUB Nand_g
10660  CASE =41
10665   GOSUB Nor_g
10670  CASE =42
10675   GOSUB Ex_or
10680  CASE =43
10685   GOSUB D_flip
10690  CASE =44
10695   GOSUB Jk_flip
10700  CASE =45
10705   GOSUB Rect1
10710  CASE =46
10715   GOSUB Circle_1
10720  CASE =47
10725   GOSUB Circle1
10730  CASE =48
10735   GOSUB Square_1
10740  CASE =49
10745   GOSUB Square
10750  CASE =50
10755   GOSUB Bird
10760  CASE =52
10765   GOSUB Motor
10770  CASE =53
10775   GOSUB Summer
10780  CASE =54
10785   GOSUB C14_pin
10790  CASE =55
10795   GOSUB C16_pin
10800  CASE =56
10805   GOSUB C24_pin
10810  CASE =60
10815   GOSUB Half_bridge_in
10820  CASE =61
10825   GOSUB Full_bridge_1in
10830  CASE =62
```

```
10835   GOSUB Full_bridge_3in
10840  CASE =63
10845   GOSUB Phase1_rect
10850  CASE =64
10855   GOSUB Phase3_rect_d
10860  CASE =65
10865   GOSUB Phase3_rect_y
10870  END SELECT
10875  PIVOT 0
10880  RETURN
10885 Error_control:          !
10890  RETURN
10895! END OF PLOTTER ROUTINE
10900           !
10905 S_bigger:! INCREASE SIZE OF ELEMENT ROUTINE
10910  Dflag=1
10915  S=S+.1
10920  GOSUB Redraw
10925  RETURN
10930 S_smaller:! DECREASE SIZE OF ELEMENT ROUTINE
10935  Dflag=1
10940  S=S-.1
10945  GOSUB Redraw
10950  RETURN
10955 Help:   ! ON LINE HELP ROUTINE
10960  ALPHA OFF
10965  OFF KNOB
10970  OFF KBD
10975  GSTORE Screen1(*)
10980  GCLEAR
10985    !
10990    !          ***********  HELP SCREEN  **********
10995  ALLOCATE INTEGER Eee(60,200),Z2(13),Z1(11)
11000  FOR I=1 TO 70
11005   IF E(I,0)=0 THEN 11045
11010   FOR J=4 TO E(I,0)*4 STEP 4
11015    Eee(I,0)=E(I,0)
11020    Eee(I,J-3)=E(I,J-3)
11025    Eee(I,J-2)=E(I,J-2)
11030    Eee(I,J-1)=E(I,J-1)
11035    Eee(I,J)=E(I,J)
11040   NEXT J
11045  NEXT I
11050              !
11055              !
11060              !
11065  FOR I=1 TO 24
11070   READ Aaa
11075  NEXT I
11080  READ Z2(*)
11085  READ Z1(*)
11090  E(1,0)=4
11095  FOR J=1 TO 12
```

```
11100   E(1,J)=Z1(J-1)
11105  NEXT J
11110  DATA 28,86,142,200,258,316,374,432,490,548,606,664
11115  DATA 722,780,28,485,28,485,28,523,28,523,28,561,28,561
11120  RESTORE
11125  FOR I=2 TO 14
11130   E(I,0)=1
11135   E(I,1)=Z2(I-1)
11140   E(I,2)=485
11145   E(I,3)=Z2(I-1)
11150   E(I,4)=561
11155  NEXT I
11160  FOR I=15 TO 28
11165   IF I=25 OR I=27 THEN 11195
11170   E(I,0)=1
11175   E(I,1)=Z2(I-15)
11180   E(I,2)=371
11185   E(I,3)=Z2(I-15)
11190   E(I,4)=447
11195  NEXT I
11200  FOR I=29 TO 42
11205   IF I=29 OR I=32 OR I=33 OR I=34 OR I=35 THEN 11235
11210   E(I,0)=1
11215   E(I,1)=Z2(I-29)
11220   E(I,2)=257
11225   E(I,3)=Z2(I-29)
11230   E(I,4)=333
11235  NEXT I
11240  FOR I=43 TO 56
11245   IF I=43 OR I=44 OR I>52 THEN 11275
11250   E(I,0)=1
11255   E(I,1)=Z2(I-43)
11260   E(I,2)=143
11265   E(I,3)=Z2(I-43)
11270   E(I,4)=219
11275  NEXT I
11280  FOR I=56 TO 60
11285   IF I=56 OR I=57 OR I=58 OR I=59 OR I=60 THEN 11315
11290   E(I,0)=1
11295   E(I,1)=Z2(I-56)
11300   E(I,2)=29
11305   E(I,3)=Z2(I-56)
11310   E(I,4)=105
11315  NEXT I
11320  GOSUB Redraw
11325  MAT E= (0)
11330    !
11335    !
11340    !
11345    !
11350  FOR I=1 TO 60
11355   IF Eee(I,0)=0 THEN 11395
11360   FOR J=4 TO Eee(I,0)*4 STEP 4
```

```
11365    E(I,0)=Eee(I,0)
11370    E(I,J-3)=Eee(I,J-3)
11375    E(I,J-2)=Eee(I,J-2)
11380    E(I,J-1)=Eee(I,J-1)
11385    E(I,J)=Eee(I,J)
11390   NEXT J
11395  NEXT I
11400  DEALLOCATE Eee(*),Z2(*),Z1(*)
11405  GCLEAR
11410  GLOAD Screen1(*)
11415  GRAPHICS ON
11420  RETURN
11425 ! END OF HELP ROUTINE
11430 !
11435 ! ALL THE CIRCUIT ELEMENTS AND GRAPHICAL ELEMENTS
11440 ! USED IN THIS CIRCUIT DRAWING PROGRAM ARE PLACED
11445 ! HERE ALL THE GRAPHICAL ELEMENTS ARE INTENDED TO
11450 ! HELP DESCRIBE THE CIRCUIT AND ARE IGNORED IN THE
11455 ! TREE-LINK PROGRAM ALL THESE CIRCUIT ELEMENT ROUTINES
11460 ! CONTAIN ONLY FUNDAMENTAL STATEMENTS
11465 ! ALL ELEMENTS ARE PRODUCED WITH SIMPLE GEOMETRIC
11470 ! EQUATIONS NO SPECIAL GRAPHICS FUNCTIONS ARE USED
11475  RPLOT 0,-Pz,1
11480  RPLOT 0,Pz
11485  FOR K9=0 TO 3 STEP .5
11490   RPLOT 0,(Pz-K9)
11495   RPLOT (-4+K9),(Pz-12+K9)
11500   RPLOT 0,(Pz-8+K9)
11505   RPLOT (4-K9),(Pz-12+K9)
11510  NEXT K9
11515  RPLOT 0,(Pz-K9),2
11520  RETURN
11525 Summer:   !
11530  RPLOT -Pz,0,1
11535  FOR Xx=-Pz TO Pz STEP .5
11540   Yy=SQR((Pz)^2-Xx^2)
11545   RPLOT Xx,Yy
11550  NEXT Xx
11555  FOR Xx=Pz TO -Pz STEP -.5
11560   Yy=-SQR(Pz^2-Xx^2)
11565   RPLOT Xx,Yy
11570  NEXT Xx
11575  RPLOT -Pz,0,2
11580  IF Pz<20 THEN Pz=20
11585  RPLOT -(Pz/1.5-9),Pz/1.5-11,1
11590  RPLOT -(Pz/1.5-8),Pz/1.5-12
11595  RPLOT -(Pz/1.5-8),-(Pz/1.5-8)
11600  RPLOT 0,0
11605  RPLOT (Pz/1.5-8),-(Pz/1.5-8)
11610  RPLOT (Pz/1.5-8),Pz/1.5-12
11615  RPLOT (Pz/1.5-9),Pz/1.5-11,2
11620  RETURN
11625 Motor:   !
```

```
11630  RPLOT 0,Pz,1
11635  RPLOT 0,20*S,2
11640  RPLOT -4*S,SQR(256-4^2)*S,1
11645  RPLOT -4*S,20*S
11650  RPLOT 4*S,20*S
11655  RPLOT 4*S,SQR(256-4^2)*S,2
11660  RPLOT -16*S,0,1
11665  FOR Xx=-16 TO 16 STEP .5
11670    Yy=SQR(256-Xx^2)
11675    RPLOT Xx*S,Yy*S
11680  NEXT Xx
11685  FOR Xx=16 TO -16 STEP -.5
11690    Yy=-SQR(256-Xx^2)
11695    RPLOT Xx*S,Yy*S
11700  NEXT Xx
11705  RPLOT -16*S,0,2
11710  RPLOT -4*S,-SQR(256-4^2)*S,1
11715  RPLOT -4*S,-20*S
11720  RPLOT 4*S,-20*S
11725  RPLOT 4*S,-SQR(256-4^2)*S,2
11730  RPLOT 0,-20*S,1
11735  RPLOT 0,-Pz,2
11740  RETURN
11745 Current:   !
11750  RPLOT 0,Pz,1
11755  RPLOT 0,20*S,2
11760  RPLOT -12*S,8*S,1
11765  FOR Xx=-12 TO 12 STEP .5
11770    Yy=SQR(144-Xx^2)+8
11775    RPLOT Xx*S,Yy*S
11780  NEXT Xx
11785  FOR Xx=12 TO -12 STEP -.5
11790    Yy=-SQR(144-Xx^2)+8
11795    RPLOT Xx*S,Yy*S
11800  NEXT Xx
11805  RPLOT -12*S,8*S,2
11810  RPLOT -12*S,-8*S,1
11815  FOR Xx=-12 TO 12 STEP .5
11820    Yy=SQR(144-Xx^2)-8
11825    RPLOT Xx*S,Yy*S
11830  NEXT Xx
11835  FOR Xx=12 TO -12 STEP -.5
11840    Yy=-SQR(144-Xx^2)-8
11845    RPLOT Xx*S,Yy*S
11850  NEXT Xx
11855  RPLOT -12*S,-8*S,2
11860  RPLOT -20*S,12*S,1
11865  RPLOT -24*S,8*S,2
11870  RPLOT -20*S,12*S,1
11875  RPLOT -16*S,8*S,2
11880  RPLOT -20*S,12*S,1
11885  RPLOT -20*S,-12*S,2
11890  RPLOT 0,-20*S,1
```

```
11895  RPLOT 0,-Pz,2
11900  RETURN
11905 Current_sour:   !
11910  RPLOT 0,Pz,1
11915  RPLOT 0,16*S,2
11920  RPLOT -16*S,0,1
11925  FOR Xx=-16 TO 16 STEP .5
11930   Yy=SQR(256-Xx^2)
11935   RPLOT Xx*S,Yy*S
11940  NEXT Xx
11945  FOR Xx=16 TO -16 STEP -.5
11950   Yy=-SQR(256-Xx^2)
11955   RPLOT Xx*S,Yy*S
11960  NEXT Xx
11965  RPLOT -16*S,Yy*S,2
11970  RPLOT 0,-16*S,1
11975  RPLOT 0,-Pz,2
11980  RPLOT -16*S,0,2
11985  RPLOT 0,12*S,1
11990  RPLOT 0,-12*S,2
11995  RPLOT 0,12*S,1
12000  RPLOT -4*S,8*S,2
12005  RPLOT 0,12*S,1
12010  RPLOT 4*S,8*S,2
12015                 !
12020                 !
12025  RETURN
12030 Dc_source:    !
12035  RPLOT 0,Pz,1
12040  RPLOT 0,3.5*S,2
12045  RPLOT -12*S,3.5*S,1
12050  RPLOT 12*S,3.5*S,2
12055  RPLOT -6*S,-3.5*S,1
12060  RPLOT 6*S,-3.5*S,2
12065  RPLOT 0,-3.5*S,1
12070  RPLOT 0,-Pz,2
12075  RPLOT -8*S,10*S,1
12080  RPLOT -4*S,10*S,2
12085  RPLOT -6*S,12*S,1
12090  RPLOT -6*S,8*S,2
12095  RPLOT -8*S,-10*S,1
12100  RPLOT -4*S,-10*S,2
12105  RETURN
12110 Battery:      !
12115  RPLOT 0,Pz,1
12120  RPLOT 0,12*S,2
12125  RPLOT -12*S,12*S
12130  RPLOT 12*S,12*S,2
12135  RPLOT -6*S,4*S,1
12140  RPLOT 6*S,4*S,2
12145  RPLOT -12*S,-4*S,1
12150  RPLOT 12*S,-4*S,2
12155  RPLOT -6*S,-12*S,1
```

```
12160  RPLOT 6*S,-12*S,2
12165  RPLOT 0,-12*S,1
12170  RPLOT 0,-Pz,2
12175  RETURN
12180 A_volt:   !
12185  RPLOT 0,Pz,1
12190  RPLOT 0,16*S,2
12195  RPLOT -16*S,0,1
12200  FOR Xx=-16 TO 16 STEP .5
12205   Yy=SQR(256-Xx^2)
12210   RPLOT Xx*S,Yy*S
12215  NEXT Xx
12220  FOR Xx=16 TO -16 STEP -.5
12225   Yy=-SQR(256-Xx^2)
12230   RPLOT Xx*S,Yy*S
12235  NEXT Xx
12240  RPLOT -16*S,0,2
12245  MOVE M1+2,M2+2
12250  FOR Xx=9 TO -9 STEP -.5
12255   Yy=-SIN(Xx*20)
12260   RPLOT Xx*S,(Yy*5)*S
12265  NEXT Xx
12270  RPLOT -9*S,Yy*S,2
12275  RPLOT 0,-16*S,1
12280  RPLOT 0,-Pz,2
12285  RETURN
12290    !
12295    !
12300 Ac_volt:   !
12305  RPLOT 0,Pz,1
12310  RPLOT 0,16*S,2
12315  RPLOT -16*S,0,1
12320  FOR Xx=-16 TO 16 STEP .5
12325   Yy=SQR(256-Xx^2)
12330   RPLOT Xx*S,Yy*S
12335  NEXT Xx
12340  FOR Xx=16 TO -16 STEP -.5
12345   Yy=-SQR(256-Xx^2)
12350   RPLOT Xx*S,Yy*S
12355  NEXT Xx
12360  RPLOT -16*S,0,2
12365  MOVE M1+2,M2+2
12370  FOR Xx=9 TO -9 STEP -.5
12375   Yy=-SIN(Xx*20)
12380   RPLOT Xx*S,(Yy*5)*S
12385  NEXT Xx
12390  RPLOT -9*S,Yy*S,2
12395  RPLOT 0,-16*S,1
12400  RPLOT 0,-Pz,2
12405  RPLOT -16*S,20*S,1
12410  RPLOT -16*S,12*S,2
12415  RPLOT -20*S,16*S,1
12420  RPLOT -12*S,16*S,2
```

```
12425  RETURN
12430    !
12435    !
12440    !
12445 Resistor:    !
12450  RPLOT 0,Pz,1
12455  RPLOT 0,16*S
12460  RPLOT -8*S,14*S
12465  RPLOT 8*S,10*S
12470  RPLOT -8*S,6*S
12475  RPLOT 8*S,2*S
12480  RPLOT -8*S,-2*S
12485  RPLOT 8*S,-6*S
12490  RPLOT -8*S,-10*S
12495  RPLOT 8*S,-14*S
12500  RPLOT 0,-16*S
12505  RPLOT 0,-Pz,2
12510  RETURN
12515 Diode:    !
12520  RPLOT 0,Pz,1
12525  RPLOT 0,6*S,2
12530  RPLOT -10*S,6*S,1
12535  RPLOT 10*S,6*S,2
12540  RPLOT 0,6*S,1
12545  RPLOT -10*S,-6*S
12550  RPLOT 10*S,-6*S
12555  RPLOT 0,6*S,2
12560  RPLOT 0,-6*S,1
12565  RPLOT 0,-Pz,2
12570  RETURN
12575          !
12580 Zener:       !
12585  RPLOT 0,Pz,1
12590  RPLOT 0,6*S,2
12595  RPLOT -10*S,6*S,1
12600  RPLOT 10*S,6*S,2
12605  RPLOT 0,6*S,1
12610  RPLOT -10*S,-6*S
12615  RPLOT 10*S,-6*S
12620  RPLOT 0,6*S,2
12625  RPLOT 0,-6*S,1
12630  RPLOT 0,-Pz,2
12635  RPLOT -10*S,6*S,1
12640  RPLOT -14*S,2*S,2
12645  RPLOT 10*S,6*S,1
12650  RPLOT 14*S,10*S,2
12655  RETURN
12660          !
12665 Thyristor:    !
12670  RPLOT 0,Pz,1
12675  RPLOT 0,6*S,2
12680  RPLOT 0,6*S,1
12685  RPLOT 7*S,13*S,2
```

```
12690  RPLOT -10*S,6*S,1
12695  RPLOT 10*S,6*S,2
12700  RPLOT 0,6*S,1
12705  RPLOT -10*S,-6*S
12710  RPLOT 10*S,-6*S
12715  RPLOT 0,6*S,2
12720  RPLOT 0,-6*S,1
12725  RPLOT 0,-Pz,2
12730  RETURN
12735       !
12740 Inductor:      !
12745  RPLOT 0,Pz,1
12750  RPLOT 0,16*S
12755  RPLOT -4*S,16*S
12760  FOR Yy=16 TO 8 STEP -.5
12765   Xx=-SQR(16-(Yy-12)^2)-4
12770   RPLOT Xx*S,Yy*S
12775  NEXT Yy
12780  RPLOT 0,8*S,2
12785  RPLOT -4*S,8*S,1
12790  FOR Yy=8 TO 0 STEP -.5
12795   Xx=-SQR(16-(Yy-4)^2)-4
12800   RPLOT Xx*S,Yy*S
12805  NEXT Yy
12810  RPLOT 0,0,2
12815  RPLOT -4*S,0,1
12820  FOR Yy=0 TO -8 STEP -.5
12825   Xx=-SQR(16-(Yy+4)^2)-4
12830   RPLOT Xx*S,Yy*S
12835  NEXT Yy
12840  RPLOT 0,-8*S,2
12845  RPLOT -4*S,-8*S,1
12850  FOR Yy=-8 TO -16 STEP -.5
12855   Xx=-SQR(16-(Yy+12)^2)-4
12860   RPLOT Xx*S,Yy*S
12865  NEXT Yy
12870  RPLOT 0,-16*S
12875  RPLOT 0,-Pz,2
12880  RETURN
12885     !
12890 Choke:    !
12895  RPLOT 0,Pz,1
12900  RPLOT 0,16*S
12905  RPLOT -4*S,16*S
12910  FOR Yy=16 TO 8 STEP -.5
12915   Xx=-SQR(16-(Yy-12)^2)-4
12920   RPLOT Xx*S,Yy*S
12925  NEXT Yy
12930  RPLOT 0,8*S,2
12935  RPLOT -4*S,8*S,1
12940  FOR Yy=8 TO 0 STEP -.5
12945   Xx=-SQR(16-(Yy-4)^2)-4
12950   RPLOT Xx*S,Yy*S
```

```
12955  NEXT Yy
12960  RPLOT 0,0,2
12965  RPLOT -4*S,0,1
12970  FOR Yy=0 TO -8 STEP -.5
12975   Xx=-SQR(16-(Yy+4)^2)-4
12980   RPLOT Xx*S,Yy*S
12985  NEXT Yy
12990  RPLOT 0,-8*S,2
12995  RPLOT -4*S,-8*S,1
13000  FOR Yy=-8 TO -16 STEP -.5
13005   Xx=-SQR(16-(Yy+12)^2)-4
13010   RPLOT Xx*S,Yy*S
13015  NEXT Yy
13020  RPLOT 0,-16*S
13025  RPLOT 0,-Pz,2
13030  RPLOT -10*S,16*S,1
13035  RPLOT -10*S,-16*S,2
13040  RPLOT -14*S,16*S,1
13045  RPLOT -14*S,-16*S,2
13050  RETURN
13055          !
13060 Impedance:          !
13065  RPLOT 0,Pz,1
13070  RPLOT 0,16*S,2
13075  RPLOT -12*S,16*S,1
13080  RPLOT 12*S,16*S
13085  RPLOT 12*S,-16*S
13090  RPLOT -12*S,-16*S
13095  RPLOT -12*S,16*S,2
13100  RPLOT 12*S,16*S,1
13105  RPLOT -12*S,-16*S,2
13110  RPLOT 0,-16*S,1
13115  RPLOT 0,-Pz,2
13120  RETURN
13125 Admitance:      !
13130  RPLOT 0,Pz,1
13135  RPLOT 0,16*S,2
13140  RPLOT -12*S,16*S,1
13145  RPLOT 12*S,16*S
13150  RPLOT 12*S,-16*S
13155  RPLOT -12*S,-16*S
13160  RPLOT -12*S,16*S,2
13165  RPLOT 0,16*S,1
13170  RPLOT 0,0
13175  RPLOT -12*S,-16*S,2
13180  RPLOT 0,0,1
13185  RPLOT 12*S,-16*S,2
13190  RPLOT 0,16*S,1
13195  RPLOT 0,-Pz,2
13200  RETURN
13205 Capacitor:      !
13210  RPLOT 0,Pz,1
13215  RPLOT 0,3.2*S,2
```

```
13220  RPLOT -8*S,3.2*S,1
13225  RPLOT 8*S,3.2*S,2
13230  RPLOT -8*S,-3.2*S,1
13235  RPLOT 8*S,-3.2*S,2
13240  RPLOT 0,-3.2*S,1
13245  RPLOT 0,-Pz,2
13250  RETURN
13255 Electrolytic:!
13260  RPLOT 0,Pz,1
13265  RPLOT 0,3.2*S,2
13270  RPLOT -8*S,3.2*S,1
13275  RPLOT 8*S,3.2*S,2
13280  Xx=-8
13285  Yy=SQR(256-Xx^2)-19.2
13290  RPLOT Xx*S,Yy*S
13295  FOR Xx=-7.6 TO 7.6 STEP .4
13300   Yy=SQR(256-Xx^2)-19.2
13305   RPLOT Xx*S,Yy*S
13310  NEXT Xx
13315  Xx=8
13320  Yy=SQR(256-Xx^2)-19.2
13325  RPLOT Xx*S,Yy*S,2
13330  RPLOT 0,-3.2*S,1
13335  RPLOT 0,-Pz,2
13340  RETURN
13345 Looped:   !
13350  RPLOT 0,Pz,1
13355  RPLOT 0,7
13360  RPLOT -4,7
13365  FOR Xx=-4 TO -8 STEP -.5
13370   Yy=SQR(49-(Xx+4)^2)
13375   RPLOT Xx,Yy
13380  NEXT Xx
13385  FOR Xx=-8 TO -4 STEP .5
13390   Yy=-SQR(49-(Xx+4)^2)
13395   RPLOT Xx,Yy
13400  NEXT Xx
13405  RPLOT -4,-7,1
13410  RPLOT 0,-7
13415  RPLOT 0,-Pz,2
13420  RETURN
13425 Direc:    !
13430  FOR Cx=0 TO 4 STEP .5
13435   RPLOT 0,6-Cx,1
13440   RPLOT -4+Cx,-6+Cx
13445   RPLOT 0,-2+Cx
13450   RPLOT 4-Cx,-6+Cx
13455   RPLOT 0,6-Cx,2
13460  NEXT Cx
13465  RETURN
13470 Ground1:    !
13475  RPLOT 0,-Pz,1
13480  RPLOT 0,Pz-9.6*S,2
```

```
13485  RPLOT -10*S,Pz-9.6*S,1
13490  RPLOT 10*S,Pz-9.6*S,2
13495  RPLOT -7.5*S,Pz-7.2*S,1
13500  RPLOT 7.5*S,Pz-7.2*S,2
13505  RPLOT -4*S,Pz-4.8*S,1
13510  RPLOT 4*S,Pz-4.8*S,2
13515  RPLOT -2.4*S,Pz-2.4*S,1
13520  RPLOT 2.4*S,Pz-2.4*S,2
13525  RPLOT -1*S,Pz,1
13530  RPLOT 1*S,Pz,2
13535  RETURN
13540 Thyristor_d:    !
13545  RPLOT 0,Pz,1
13550  RPLOT 0,16*S,2
13555  RPLOT -16*S,16*S,1
13560  RPLOT 16*S,16*S,2
13565  RPLOT -16*S,16*S,1
13570  RPLOT -16*S,6*S,2
13575  RPLOT -26*S,6*S,1
13580  RPLOT -6*S,6*S
13585  RPLOT -16*S,-6*S
13590  RPLOT -26*S,6*S,2
13595  RPLOT -26*S,-6*S,1
13600  RPLOT -6*S,-6*S,2
13605  RPLOT -16*S,-6*S,1
13610  RPLOT -16*S,-16*S
13615  RPLOT 16*S,-16*S
13620  RPLOT 16*S,-6*S,2
13625  RPLOT 6*S,-6*S,1
13630  RPLOT 26*S,-6*S
13635  RPLOT 16*S,6*S
13640  RPLOT 6*S,-6*S,2
13645  RPLOT 6*S,6*S,1
13650  RPLOT 26*S,6*S,2
13655  RPLOT 23*S,13*S,1
13660  RPLOT 16*S,6*S
13665  RPLOT 16*S,16*S,2
13670  RPLOT 0,-16*S,1
13675  RPLOT 0,-Pz,2
13680  RETURN
13685 Switch:    !
13690  RPLOT 0,Pz,1
13695  RPLOT 0,13*S,2
13700  RPLOT -1*S,12*S,1
13705  FOR Xx=-1 TO 1 STEP .2
13710    Yy=SQR(1-Xx^2)+12
13715    RPLOT Xx*S,Yy*S
13720  NEXT Xx
13725  FOR Xx=1 TO -1 STEP -.2
13730    Yy=-SQR(1-Xx^2)+12
13735    RPLOT Xx*S,Yy*S
13740  NEXT Xx
13745  RPLOT -1*S,12*S,2
```

```
13750  RPLOT 0,-12*S,1
13755  RPLOT -14.3*S,5.3*S,2
13760  RPLOT 0,-12*S,1
13765  RPLOT 0,-Pz,2
13770  RPLOT -16*S,6*S,1
13775  FOR Xx=-16 TO -14 STEP .2
13780   Yy=SQR(1-(Xx+15)^2)+6
13785   RPLOT Xx*S,Yy*S
13790  NEXT Xx
13795  FOR Xx=-14 TO -16 STEP -.2
13800   Yy=-SQR(1-(Xx+15)^2)+6
13805   RPLOT Xx*S,Yy*S
13810  NEXT Xx
13815  RPLOT -16*S,6*S,2
13820  RPLOT -15*S,-6.6*S,1
13825  FOR Xx=-15 TO -4 STEP .5
13830   Yy=SQR(256-Xx^2)-12
13835   RPLOT Xx*S,Yy*S
13840  NEXT Xx
13845  FOR K=0 TO 1 STEP .25
13850   RPLOT (-4.2+K)*S,(Yy+2-K)*S
13855   RPLOT (-2-K)*S,S*(SQR(256-(-2-K)^2)-12)
13860   RPLOT (-3.8+K-.2)*S,(Yy-2+K+.37)*S
13865   RPLOT (-4+K)*S,S*(SQR(256-(-4+K)^2)-12)
13870  NEXT K
13875  RPLOT -3*S,S*(SQR(256-(-3)^2)-12),2
13880  RETURN
13885 Bipolar:  !
13890  Don=S
13895 !S=.8
13900  RPLOT 0,Pz,1
13905  RPLOT 0,16*S
13910  RPLOT -12*S,6*S,2
13915  RPLOT -12*S,12*S,1
13920  RPLOT -12*S,-12*S,2
13925  RPLOT -24*S,0,1
13930  RPLOT -12*S,0,2
13935  RPLOT -12*S,-6*S,1
13940  RPLOT 0,-16*S
13945  RPLOT 0,-Pz,2
13950    !   OPTION
13955  FOR K=0 TO 2 STEP .5
13960   RPLOT (-8+K*1.1)*S,(-14+K*1.1)*S
13965   RPLOT (-2.26-K*1.1)*S,(-14.2+K*.9)*S
13970   RPLOT (-3.6-K*.9)*S,(-8.8-K*1.1)*S
13975   RPLOT (-8+K*1.1)*S,(-14+K*1.1)*S
13980  NEXT K
13985  RETURN
13990    !
13995 T_supressor:                    !
14000  RPLOT 0,Pz,1
14005  RPLOT 0,20*S,2
14010  RPLOT -10*S,20*S,1
```

```
14015  RPLOT 10*S,20*S
14020  RPLOT 10*S,-20*S
14025  RPLOT -10*S,-20*S
14030  RPLOT -10*S,20*S,2
14035  RPLOT 26*S,20*S,1
14040  RPLOT 18*S,20*S
14045  RPLOT -18*S,-20*S
14050  RPLOT -26*S,-20*S,2
14055  RPLOT 0,-20*S,1
14060  RPLOT 0,-Pz,2
14065  RETURN
14070 Mosfet:   !
14075  RPLOT 0,Pz,1
14080  RPLOT 0,6*S
14085  RPLOT -12*S,6*S,2
14090  RPLOT -12*S,12*S,1
14095  RPLOT -12*S,-12*S,2
14100  RPLOT -24*S,0,1
14105  RPLOT -12*S,0,2
14110  RPLOT -12*S,-6*S,1
14115  RPLOT 0,-6*S
14120  RPLOT 0,-Pz,2
14125  RPLOT -5*S,-10*S,1
14130    ! OPTION
14135  FOR K=0 TO 1.4 STEP .2
14140   RPLOT (-5+K)*S,(-10+K*1.4)*S
14145   RPLOT (-2-K)*S,-6*S
14150   RPLOT (-5+K)*S,(-2-K*1.4)*S
14155   RPLOT (-5+K)*S,(-10+K*1.4)*S
14160  NEXT K
14165  RPLOT (-5*S)+1,(-10*S)+1,2
14170  RETURN
14175 Single_t:   !
14180  T=S
14185  IF T>3 THEN T=3
14190  RPLOT -56,-Pz,1
14195  RPLOT -56,-14*T
14200  RPLOT -16*T,-14*T
14205  RPLOT -16*T,-10*T
14210  FOR Xx=-16 TO -8 STEP .5
14215   Yy=-SQR(16-(Xx+12)^2)-10
14220   RPLOT Xx*T,Yy*T
14225  NEXT Xx
14230  RPLOT -8*T,-14*T,2
14235  RPLOT -8*T,-10*T,1
14240  FOR Xx=-8 TO 0 STEP .5
14245   Yy=SQR(16-(Xx+4)^2)-10
14250   RPLOT Xx*T,Yy*T
14255  NEXT Xx
14260  RPLOT 0,-14*T,2
14265  RPLOT 0,-10*T,1
14270  FOR Xx=0 TO 8 STEP .5
14275   Yy=SQR(16-(Xx-4)^2)-10
```

```
14280   RPLOT Xx*T,Yy*T
14285  NEXT Xx
14290  RPLOT 8*T,-14*T,2
14295  RPLOT 8*T,-10*T,1
14300  FOR Xx=8 TO 16 STEP .5
14305   Yy=SQR(16-(Xx-12)^2)-10
14310   RPLOT Xx*T,Yy*T
14315  NEXT Xx
14320  RPLOT 16*T,-14*T
14325  RPLOT 56,-14*T
14330  RPLOT 56,-Pz,2
14335    !
14340  RPLOT -16*T,-2*T,1
14345  RPLOT 16*T,-2*T,2
14350  RPLOT -16*T,2*T,1
14355  RPLOT 16*T,2*T,2
14360  RPLOT -56,Pz,1
14365  RPLOT -56,14*T
14370  RPLOT -16*T,14*T
14375  RPLOT -16*T,10*T
14380  FOR Xx=-16 TO -8 STEP .5
14385   Yy=-SQR(16-(Xx+12)^2)+10
14390   RPLOT Xx*T,Yy*T
14395  NEXT Xx
14400  RPLOT -8*T,10*T,2
14405  RPLOT -8*T,14*T,1
14410  FOR Xx=-8 TO 0 STEP .5
14415   Yy=-SQR(16-(Xx+4)^2)+10
14420   RPLOT Xx*T,Yy*T
14425  NEXT Xx
14430  RPLOT 0,10*T,2
14435  RPLOT 0,14*T,1
14440  FOR Xx=0 TO 8 STEP .5
14445   Yy=-SQR(16-(Xx-4)^2)+10
14450   RPLOT Xx*T,Yy*T
14455  NEXT Xx
14460  RPLOT 8*T,10*T,2
14465  RPLOT 8*T,14*T,1
14470  FOR Xx=8 TO 16 STEP .5
14475   Yy=-SQR(16-(Xx-12)^2)+10
14480   RPLOT Xx*T,Yy*T
14485  NEXT Xx
14490  RPLOT 16*T,14*T
14495  RPLOT 56,14*T
14500  RPLOT 56,Pz,2
14505    !
14510  RETURN
14515 Diode_brid:    !
14520  F=0
14525  Pz=32
14530  Ss=2
14535  RPLOT 0,Pz,1
14540  RPLOT (-6-F)*Ss,(10+F)*Ss,2
```

```
14545  RPLOT (-8-F)*Ss,(12+F)*Ss,1
14550  RPLOT (-4-F)*Ss,(8+F)*Ss
14555  RPLOT (-10-F)*Ss,(6+F)*Ss
14560  RPLOT (-8-F)*Ss,(12+F)*Ss,2
14565  RPLOT (-12-F)*Ss,(8+F)*Ss,1
14570  RPLOT (-8-F)*Ss,(4+F)*Ss,2
14575  RPLOT (-10-F)*Ss,(6+F)*Ss,1
14580  RPLOT -Pz,0
14585  RPLOT (-10-F)*Ss,(-6-F)*Ss,2
14590  RPLOT (-12-F)*Ss,(-8-F)*Ss,1
14595  RPLOT (-8-F)*Ss,(-4-F)*Ss,2
14600  RPLOT (-10-F)*Ss,(-6-F)*Ss,1
14605  RPLOT (-8-F)*Ss,(-12-F)*Ss
14610  RPLOT (-4-F)*Ss,(-8-F)*Ss
14615  RPLOT (-10-F)*Ss,(-6-F)*Ss,2
14620  RPLOT (-6-F)*Ss,(-10-F)*Ss,1
14625  RPLOT 0,-Pz
14630  RPLOT (6+F)*Ss,(-10-F)*Ss,2
14635  RPLOT (4+F)*Ss,(-8-F)*Ss,1
14640  RPLOT (8+F)*Ss,(-12-F)*Ss,2
14645  RPLOT (6+F)*Ss,(-10-F)*Ss,1
14650  RPLOT (8+F)*Ss,(-4-F)*Ss
14655  RPLOT (12+F)*Ss,(-8-F)*Ss
14660  RPLOT (6+F)*Ss,(-10-F)*Ss,2
14665  RPLOT (10+F)*Ss,(-6-F)*Ss,1
14670  RPLOT Pz,0
14675  RPLOT (10+F)*Ss,(6+F)*Ss,2
14680  RPLOT (8+F)*Ss,(4+F)*Ss,1
14685  RPLOT (6+F)*Ss,(10+F)*Ss
14690  RPLOT (12+F)*Ss,(8+F)*Ss
14695  RPLOT (8+F)*Ss,(4+F)*Ss,2
14700  RPLOT (4+F)*Ss,(8+F)*Ss,1
14705  RPLOT (8+F)*Ss,(12+F)*Ss,2
14710  RPLOT (6+F)*Ss,(10+F)*Ss,1
14715  RPLOT 0,Pz,2
14720  RETURN
14725 Circle1:    !
14730  RPLOT -Pz,0,1
14735  FOR Xx=-Pz TO Pz STEP .5
14740    Yy=SQR((Pz)^2-Xx^2)
14745    RPLOT Xx,Yy
14750  NEXT Xx
14755  FOR Xx=Pz TO -Pz STEP -.5
14760    Yy=-SQR(Pz^2-Xx^2)
14765    RPLOT Xx,Yy
14770  NEXT Xx
14775  RPLOT -Pz,0,2
14780  RETURN
14785 Square_1:          !
14790  RPLOT 0,Pz,1
14795  RPLOT 0,16*S,2
14800  RPLOT -16*S,16*S
14805  RPLOT 16*S,16*S
```

```
14810  RPLOT 16*S,-16*S
14815  RPLOT -16*S,-16*S
14820  RPLOT -16*S,16*S,2
14825  RPLOT 0,-16*S,1
14830  RPLOT 0,-Pz,2
14835  RETURN
14840 Square:                    !
14845  RPLOT -Pz,-Pz,1
14850  RPLOT -Pz,Pz
14855  RPLOT Pz,Pz
14860  RPLOT Pz,-Pz
14865  RPLOT -Pz,-Pz,2
14870  RETURN
14875 Or_g:                      !
14880  RPLOT 0,Pz,1
14885  RPLOT 0,18*S,2
14890  RPLOT -14*S,-18.5064*S,1
14895  RPLOT -14*S,-6.2487*S
14900  FOR Xx=-14 TO 0 STEP .5
14905   Yy=SQR(-1*(Xx-14)^2+784)-6.2487
14910   RPLOT Xx*S,Yy*S
14915  NEXT Xx
14920  FOR Xx=0 TO 14 STEP .5
14925   Yy=SQR(-1*(Xx+14)^2+784)-6.2487
14930   RPLOT Xx*S,Yy*S
14935  NEXT Xx
14940  RPLOT 14*S,-18.5064*S
14945  FOR Xx=14 TO -14 STEP -.5
14950   Yy=SQR(576-Xx^2)-38
14955   RPLOT Xx*S,Yy*S
14960  NEXT Xx
14965  REM    THE   INPUTS
14970  RPLOT -14*S,-18.5064*S,2
14975  RPLOT -8,(SQR(576-(8/S)^2)-38)*S,1
14980  RPLOT -8,-Pz,2
14985  RPLOT 8,(SQR(576-(8/S)^2)-38)*S,1
14990  RPLOT 8,-Pz,2
14995  RETURN
15000 And_g:                     !
15005  RPLOT 0,Pz,1
15010  RPLOT 0,18*S,2
15015  RPLOT -14*S,4*S,1
15020  FOR Xx=-14 TO 14 STEP .5
15025   Yy=SQR(196-Xx^2)+4
15030   RPLOT Xx*S,Yy*S
15035  NEXT Xx
15040  RPLOT 14*S,-18*S
15045  RPLOT -14*S,-18*S
15050  RPLOT -14*S,4*S,2
15055  RPLOT -8,-18*S,1
15060  RPLOT -8,-Pz,2
15065  RPLOT 8,-18*S,1
15070  RPLOT 8,-Pz,2
```

```
15075 RETURN
15080 Inver:                !
15085 RPLOT 0,Pz,1
15090 RPLOT 0,16*S,2
15095 RPLOT -4*S,12*S,1
15100 FOR Xx=-4 TO 4 STEP .5
15105  Yy=SQR(16-Xx^2)+12
15110  RPLOT Xx*S,Yy*S
15115 NEXT Xx
15120 FOR Xx=4 TO -4 STEP -.5
15125  Yy=-SQR(16-Xx^2)+12
15130  RPLOT Xx*S,Yy*S
15135 NEXT Xx
15140 RPLOT -4*S,Yy*S,2
15145 RPLOT 0,8*S,1
15150 RPLOT -14*S,-10*S
15155 RPLOT 14*S,-10*S
15160 RPLOT 0,8*S,2
15165 RPLOT 0,-10*S,1
15170 RPLOT 0,-Pz,2
15175 RETURN
15180 Nand_g:               !
15185 RPLOT 0,Pz,1
15190 RPLOT 0,18*S,2
15195 RPLOT -14*S,4*S,1
15200 FOR Xx=-14 TO 14 STEP .5
15205  Yy=SQR(196-Xx^2)+4
15210  RPLOT Xx*S,Yy*S
15215 NEXT Xx
15220 RPLOT 14*S,-18*S
15225 RPLOT -14*S,-18*S
15230 RPLOT -14*S,4*S,2
15235 RPLOT -8,-18*S,1
15240 RPLOT -8,-Pz,2
15245 RPLOT 8,-18*S,1
15250 RPLOT 8,-Pz,2
15255 RETURN
15260 Nor_g:        !
15265 RPLOT 0,Pz,1
15270 RPLOT 0,18*S,2
15275 RPLOT -14*S,-18.5064*S,1
15280 RPLOT -14*S,-6.2487*S
15285 FOR Xx=-14 TO 0 STEP .5
15290  Yy=SQR(-1*(Xx-14)^2+784)-6.2487
15295  RPLOT Xx*S,Yy*S
15300 NEXT Xx
15305 FOR Xx=0 TO 14 STEP .5
15310  Yy=SQR(-1*(Xx+14)^2+784)-6.2487
15315  RPLOT Xx*S,Yy*S
15320 NEXT Xx
15325 RPLOT 14*S,-18.5064*S
15330 FOR Xx=14 TO -14 STEP -.5
15335  Yy=SQR(576-Xx^2)-38
```

```
15340   RPLOT Xx*S,Yy*S
15345  NEXT Xx
15350  REM    THE   INPUTS
15355  RPLOT -14*S,-18.5064*S,2
15360  RPLOT -8,(SQR(576-(8/S)^2)-38)*S,1
15365  RPLOT -8,-Pz,2
15370  RPLOT 8,(SQR(576-(8/S)^2)-38)*S,1
15375  RPLOT 8,-Pz,2
15380  RETURN
15385 Rect1:  !
15390  RPLOT 0,Pz,1
15395  RPLOT 0,20*S,2
15400  RPLOT -12*S,20*S,1
15405  RPLOT 12*S,20*S
15410  RPLOT 12*S,-20*S
15415  RPLOT -12*S,-20*S
15420  RPLOT -12*S,20*S,2
15425  RPLOT 0,-20*S,1
15430  RPLOT 0,-Pz,2
15435  RETURN
15440              !
15445 Circle_1:   !
15450  RPLOT 0,Pz,1
15455  RPLOT 0,16*S,2
15460  RPLOT -16*S,0,1
15465  FOR Xx=-16 TO 16 STEP .5
15470   Yy=SQR(256-Xx^2)
15475   RPLOT Xx*S,Yy*S
15480  NEXT Xx
15485  FOR Xx=16 TO -16 STEP -.5
15490   Yy=-SQR(256-Xx^2)
15495   RPLOT Xx*S,Yy*S
15500  NEXT Xx
15505  RPLOT -16*S,0,2
15510  RPLOT 0,-16*S,1
15515  RPLOT 0,-Pz,2
15520  RETURN
15525 Bird:   !
15530  OFF ERROR
15535  D=Pz/40
15540  RPLOT -8*D,18*D,1
15545  FOR Xx=-8 TO 0 STEP .5
15550   Yy=SQR(16-(Xx+4)^2)+18
15555   RPLOT Xx*D,Yy*D
15560  NEXT Xx
15565  FOR Xx=0 TO -8 STEP -.5
15570   Yy=-SQR(16-(Xx+4)^2)+18
15575   RPLOT Xx*D,Yy*D
15580  NEXT Xx
15585  RPLOT -8*D,Yy*D,2
15590  RPLOT 0,18*D,1
15595  FOR Xx=0 TO 8 STEP .5
15600   Yy=SQR(16-(Xx-4)^2)+18
```

```
15605  RPLOT Xx*D,Yy*D
15610  NEXT Xx
15615  FOR Xx=8 TO 0 STEP -.5
15620   Yy=-SQR(16-(Xx-4)^2)+18
15625   RPLOT Xx*D,Yy*D
15630  NEXT Xx
15635  RPLOT 0,Yy*D,2
15640  FOR Dd=0 TO 1.6 STEP .2
15645   RPLOT (.68629+Dd)*D,16.34314*D,1
15650   FOR Xx=.68629+Dd TO 4-Dd STEP .5
15655    Yy=SQR((1.65686-Dd)^2-(Xx-2.3431)^2)+16.3431457
15660    RPLOT Xx*D,Yy*D
15665   NEXT Xx
15670   FOR Xx=4-Dd TO .68629+Dd STEP -.5
15675    Yy=-SQR((1.65686-Dd)^2-(Xx-2.3432)^2)+16.3431457
15680    RPLOT Xx*D,Yy*D
15685   NEXT Xx
15690   RPLOT (.68629+Dd)*D,(SQR((1.6569-Dd)^2-((4-Dd)-2.3432)^2)+16.3431457)*D,
2
15695  NEXT Dd
15700  FOR Dd=0 TO 1.6 STEP .2
15705   RPLOT (-.68629-Dd)*D,16.34314*D,1
15710   FOR Xx=-.68629-Dd TO -4+Dd STEP -.5
15715    Yy=SQR((1.65686-Dd)^2-(Xx+2.3430)^2)+16.3431457
15720    RPLOT Xx*D,Yy*D
15725   NEXT Xx
15730   FOR Xx=-4+Dd TO -.68629-Dd STEP .5
15735    Yy=-SQR((1.65686-Dd)^2-(Xx+2.3432)^2)+16.3431457
15740    RPLOT Xx*D,Yy*D
15745   NEXT Xx
15750   RPLOT (-.68629-Dd)*D,16.341457*D,2
15755  NEXT Dd
15760  RPLOT -7.4641*D,20*D,1
15765  RPLOT -14*D,28*D
15770  FOR Xx=-14 TO -14.4721 STEP -.5
15775   Yy=-SQR(20-(Xx+10)^2)+30
15780   RPLOT Xx*D,Yy*D
15785  NEXT Xx
15790  FOR Xx=-14.4721 TO -6 STEP .5
15795   Yy=SQR(20-(Xx+10)^2)+30
15800   RPLOT Xx*D,Yy*D
15805  NEXT Xx
15810  RPLOT -4*D,22*D,2
15815  RPLOT -10*D,27*D,1
15820  RPLOT -8*D,24*D
15825  RPLOT -8*D,28*D,2
15830  RPLOT 7.4641*D,20*D,1
15835  RPLOT 14*D,28*D
15840  FOR Xx=14 TO 14.4721 STEP .5
15845   Yy=-SQR(20-(Xx-10)^2)+30
15850   RPLOT Xx*D,Yy*D
15855  NEXT Xx
15860  FOR Xx=14.4721 TO 6 STEP -.5
```

```
15865   Yy=SQR(20-(Xx-10)^2)+30
15870   RPLOT Xx*D,Yy*D
15875  NEXT Xx
15880  RPLOT 4*D,22*D,2
15885  RPLOT 10*D,27*D,1
15890  RPLOT 8*D,24*D
15895  RPLOT 8*D,28*D,2
15900  RPLOT 8*D,18*D,1
15905  RPLOT 6*D,10*D
15910  RPLOT -6*D,10*D
15915  RPLOT -8*D,18*D,2
15920  RPLOT -3*D,10*D,1
15925  RPLOT -3*D,7*D
15930  RPLOT -.5,7*D
15935  RPLOT -.5,10*D,2
15940  RPLOT .5,10*D,1
15945  RPLOT .5,7*D
15950  RPLOT 3*D,7*D
15955  RPLOT 3*D,10*D,2
15960  RPLOT 6*D,10*D
15965  FOR Xx=6 TO 4 STEP -.5
15970   Yy=SQR(240.25-(Xx-19.5)^2)+2.3842
15975   RPLOT Xx*D,Yy*D
15980  NEXT Xx
15985  FOR Xx=4 TO 10 STEP .5
15990   Yy=-SQR(240.25-(Xx-19.5)^2)+2.3842
15995   RPLOT Xx*D,Yy*D
16000  NEXT Xx
16005  FOR Xx=10 TO 18 STEP .5
16010   Yy=SQR(144*(1-Xx^2/324))-19.8410
16015   RPLOT Xx*D,Yy*D
16020  NEXT Xx
16025  FOR Xx=18 TO -18 STEP -.5
16030   Yy=-SQR(144*(1-Xx^2/324))-19.8410
16035   RPLOT Xx*D,Yy*D
16040  NEXT Xx
16045  FOR Xx=-18 TO -10 STEP .5
16050   Yy=SQR(144*(1-Xx^2/324))-19.8410
16055   RPLOT Xx*D,Yy*D
16060  NEXT Xx
16065  FOR Xx=-10 TO -4 STEP .5
16070   Yy=-SQR(240.25-(Xx+19.5)^2)+2.3842
16075   RPLOT Xx*D,Yy*D
16080  NEXT Xx
16085  FOR Xx=-4 TO -6 STEP -.5
16090   Yy=SQR(240.25-(Xx+19.5)^2)+2.3842
16095   RPLOT Xx*D,Yy*D
16100  NEXT Xx
16105  RPLOT -6*D,10*D,2
16110  RPLOT -1*D,-26*D,1
16115  FOR Xx=-1 TO 1 STEP .5
16120   Yy=SQR(1-Xx^2)-26
16125   RPLOT Xx*D,Yy*D
```

```
16130  NEXT Xx
16135  FOR Xx=1 TO -1 STEP -.5
16140   Yy=-SQR(1-Xx^2)-26
16145   RPLOT Xx*D,Yy*D
16150  NEXT Xx
16155  RPLOT -1*D,-26*D,2
16160        !
16165  RPLOT 0,-40*D,1
16170  FOR Xx=0 TO 6 STEP .5
16175   Yy=SQR(78.2392*(1-(Xx-6)^2/36))-40
16180   RPLOT Xx*D,Yy*D
16185  NEXT Xx
16190  FOR Xx=6 TO 9.5 STEP .5
16195   Yy=SQR(3.4051-((Xx-6)^2)/4)-33
16200   RPLOT Xx*D,Yy*D
16205  NEXT Xx
16210  RPLOT 9.6950*D,-33*D
16215  FOR Xx=9.5 TO 6 STEP -.5
16220   Yy=-SQR(3.4051-((Xx-6)^2)/4)-33
16225   RPLOT Xx*D,Yy*D
16230  NEXT Xx
16235  FOR Xx=6 TO 8 STEP .2
16240   Yy=SQR(1-(Xx-6)^2/4)-35.8452
16245   RPLOT Xx*D,Yy*D
16250  NEXT Xx
16255  FOR Xx=8 TO 6 STEP -.2
16260   Yy=-SQR(1-(Xx-6)^2/4)-35.8452
16265   RPLOT Xx*D,Yy*D
16270  NEXT Xx
16275  FOR Xx=6 TO 8 STEP .2
16280   Yy=SQR(1-(Xx-6)^2/4)-37.8452
16285   RPLOT Xx*D,Yy*D
16290  NEXT Xx
16295  FOR Xx=8 TO 4 STEP -.2
16300   Yy=-SQR(1-(Xx-6)^2/4)-37.8452
16305   RPLOT Xx*D,Yy*D
16310  NEXT Xx
16315  RPLOT 2.5*D,-36*D
16320  RPLOT 0,-40*D,2
16325        !
16330  RPLOT 0,-40*D,1
16335  FOR Xx=0 TO -6 STEP -.5
16340   Yy=SQR(78.2392*(1-(Xx+6)^2/36))-40
16345   RPLOT Xx*D,Yy*D
16350  NEXT Xx
16355  FOR Xx=-6 TO -9.5 STEP -.5
16360   Yy=SQR(3.4051-((Xx+6)^2)/4)-33
16365   RPLOT Xx*D,Yy*D
16370  NEXT Xx
16375  RPLOT -9.6950*D,-33*D
16380  FOR Xx=-9.5 TO -6 STEP .5
16385   Yy=-SQR(3.4051-((Xx+6)^2)/4)-33
16390   RPLOT Xx*D,Yy*D
```

```
16395 NEXT Xx
16400 FOR Xx=-6 TO -8 STEP -.2
16405  Yy=SQR(1-(Xx+6)^2/4)-35.8452
16410  RPLOT Xx*D,Yy*D
16415 NEXT Xx
16420 FOR Xx=-8 TO -6 STEP .2
16425  Yy=-SQR(1-(Xx+6)^2/4)-35.8452
16430  RPLOT Xx*D,Yy*D
16435 NEXT Xx
16440 FOR Xx=-6 TO -8 STEP -.2
16445  Yy=SQR(1-(Xx+6)^2/4)-37.8452
16450  RPLOT Xx*D,Yy*D
16455 NEXT Xx
16460 FOR Xx=-8 TO -4 STEP .2
16465  Yy=-SQR(1-(Xx+6)^2/4)-37.8452
16470  RPLOT Xx*D,Yy*D
16475 NEXT Xx
16480 RPLOT -2.5*D,-36*D
16485 RPLOT 0,-40*D,2
16490       !
16495 RPLOT 4.4*D,6*D,1
16500 RPLOT 8*D,2*D
16505 RPLOT 20*D,22*D,2
16510 RPLOT 17*D,22*D,1
16515 RPLOT 28*D,-20*D
16520 RPLOT 18*D,-2*D
16525 RPLOT 16*D,-8*D
16530 RPLOT 12*D,-4*D
16535 RPLOT 12*D,-8*D
16540 RPLOT 4.25*D,0,2
16545 RPLOT 15*D,10*D,1
16550 RPLOT 13*D,4*D
16555 RPLOT 14*D,-2*D,2
16560 RPLOT 18*D,10*D,1
16565 RPLOT 20*D,4*D
16570 RPLOT 19*D,-2*D,2
16575 RPLOT 10*D,1*D,1
16580 RPLOT 12*D,1*D,2
16585 RPLOT 10*D,-1*D,1
16590 RPLOT 12*D,-1*D,2
16595 RPLOT 16*D,5*D,1
16600 RPLOT 18*D,5*D,2
16605 RPLOT 16*D,3*D,1
16610 RPLOT 18*D,3*D,2
16615 RPLOT 21*D,-3*D,1
16620 RPLOT 23*D,-3*D,2
16625 RPLOT 21*D,-5*D,1
16630 RPLOT 23*D,-5*D,2
16635       !
16640 RPLOT -4.4*D,6*D,1
16645 RPLOT -8*D,2*D
16650 RPLOT -20*D,22*D,2
16655 RPLOT -17*D,22*D,1
```

```
16660  RPLOT -28*D,-20*D
16665  RPLOT -18*D,-2*D
16670  RPLOT -16*D,-8*D
16675  RPLOT -12*D,-4*D
16680  RPLOT -12*D,-8*D
16685  RPLOT -4.25*D,0,2
16690  RPLOT -15*D,10*D,1
16695  RPLOT -13*D,4*D
16700  RPLOT -14*D,-2*D,2
16705  RPLOT -18*D,10*D,1
16710  RPLOT -20*D,4*D
16715  RPLOT -19*D,-2*D,2
16720  RPLOT -10*D,1*D,1
16725  RPLOT -12*D,1*D,2
16730  RPLOT -10*D,-1*D,1
16735  RPLOT -12*D,-1*D,2
16740  RPLOT -16*D,5*D,1
16745  RPLOT -18*D,5*D,2
16750  RPLOT -16*D,3*D,1
16755  RPLOT -18*D,3*D,2
16760  RPLOT -21*D,-3*D,1
16765  RPLOT -23*D,-3*D,2
16770  RPLOT -21*D,-5*D,1
16775  RPLOT -23*D,-5*D,2
16780  RETURN
16785 C14_pin: ! 14-PIN CHIP
16790  D=1
16795  RPLOT -16*D,28*D,1
16800  RPLOT 16*D,28*D
16805  RPLOT 16*D,-28*D
16810  RPLOT -16*D,-28*D
16815  RPLOT -16*D,28*D,2
16820  RPLOT 16*D,24*D,1
16825  RPLOT 20*D,24*D,2
16830  RPLOT 16*D,16*D,1
16835  RPLOT 20*D,16*D,2
16840  RPLOT 16*D,8*D,1
16845  RPLOT 20*D,8*D,2
16850  RPLOT 16*D,0,1
16855  RPLOT 20*D,0,2
16860  RPLOT 16*D,-8*D,1
16865  RPLOT 20*D,-8*D,2
16870  RPLOT 16*D,-16*D,1
16875  RPLOT 20*D,-16*D,2
16880  RPLOT 16*D,-24*D,1
16885  RPLOT 20*D,-24*D,2
16890  RPLOT -16*D,24*D,1
16895  RPLOT -20*D,24*D,2
16900  RPLOT -16*D,16*D,1
16905  RPLOT -20*D,16*D,2
16910  RPLOT -16*D,8*D,1
16915  RPLOT -20*D,8*D,2
16920  RPLOT -16*D,0,1
```

```
16925  RPLOT -20*D,0,2
16930  RPLOT -16*D,-8*D,1
16935  RPLOT -20*D,-8*D,2
16940  RPLOT -16*D,-16*D,1
16945  RPLOT -20*D,-16*D,2
16950  RPLOT -16*D,-24*D,1
16955  RPLOT -20*D,-24*D,2
16960  RETURN
16965 C16_pin:     !16-PIN CHIP
16970  D=1
16975  RPLOT -16*D,32*D,1
16980  RPLOT 16*D,32*D
16985  RPLOT 16*D,-32*D
16990  RPLOT -16*D,-32*D
16995  RPLOT -16*D,32*D,2
17000  RPLOT 16*D,28*D,1
17005  RPLOT 20*D,28*D,2
17010  RPLOT 16*D,20*D,1
17015  RPLOT 20*D,20*D,2
17020  RPLOT 16*D,12*D,1
17025  RPLOT 20*D,12*D,2
17030  RPLOT 16*D,4*D,1
17035  RPLOT 20*D,4*D,2
17040  RPLOT 16*D,-4*D,1
17045  RPLOT 20*D,-4*D,2
17050  RPLOT 16*D,-12*D,1
17055  RPLOT 20*D,-12*D,2
17060  RPLOT 16*D,-20*D,1
17065  RPLOT 20*D,-20*D,2
17070  RPLOT 16*D,-28*D,1
17075  RPLOT 20*D,-28*D,2
17080  RPLOT -16*D,28*D,1
17085  RPLOT -20*D,28*D,2
17090  RPLOT -16*D,20*D,1
17095  RPLOT -20*D,20*D,2
17100  RPLOT -16*D,12*D,1
17105  RPLOT -20*D,12*D,2
17110  RPLOT -16*D,4*D,1
17115  RPLOT -20*D,4*D,2
17120  RPLOT -16*D,-4*D,1
17125  RPLOT -20*D,-4*D,2
17130  RPLOT -16*D,-12*D,1
17135  RPLOT -20*D,-12*D,2
17140  RPLOT -16*D,-20*D,1
17145  RPLOT -20*D,-20*D,2
17150  RPLOT -16*D,-28*D,1
17155  RPLOT -20*D,-28*D,2
17160  RETURN
17165 C24_pin: ! 24-PIN CHIP
17170  D=1
17175  RPLOT -16*D,48*D,1
17180  RPLOT 16*D,48*D
17185  RPLOT 16*D,-48*D
```

```
17190  RPLOT -16*D,-48*D
17195  RPLOT -16*D,48*D,2
17200  RPLOT 16*D,44*D,1
17205  RPLOT 20*D,44*D,2
17210  RPLOT 16*D,36*D,1
17215  RPLOT 20*D,36*D,2
17220  RPLOT 16*D,28*D,1
17225  RPLOT 20*D,28*D,2
17230  RPLOT 16*D,20*D,1
17235  RPLOT 20*D,20*D,2
17240  RPLOT 16*D,12*D,1
17245  RPLOT 20*D,12*D,2
17250  RPLOT 16*D,4*D,1
17255  RPLOT 20*D,4*D,2
17260  RPLOT 16*D,-4*D,1
17265  RPLOT 20*D,-4*D,2
17270  RPLOT [illegible]-12*D,1
17275  RPLOT [illegible]D,-12*D,2
17280  RPLOT 16*D,-20*D,1
17285  RPLOT 20*D,-20*D,2
17290  RPLOT 16*D,-28*D,1
17295  RPLOT 20*D,-28*D,2
17300  RPLOT 16*D,-36*D,1
17305  RPLOT 20*D,-36*D,2
17310  RPLOT 16*D,-44*D,1
17315  RPLOT 20*D,-44*D,2
17320  RPLOT -16*D,44*D,1
17325  RPLOT -20*D,44*D,2
17330  RPLOT -16*D,36*D,1
17335  RPLOT -20*D,36*D,2
17340  RPLOT -16*D,28*D,1
17345  RPLOT -20*D,28*D,2
17350  RPLOT -16*D,20*D,1
17355  RPLOT -20*D,20*D,2
17360  RPLOT -16*D,12*D,1
17365  RPLOT -20*D,12*D,2
17370  RPLOT -16*D,4*D,1
17375  RPLOT -20*D,4*D,2
17380  RPLOT -16*D,-4*D,1
17385  RPLOT -20*D,-4*D,2
17390  RPLOT -16*D,-12*D,1
17395  RPLOT -20*D,-12*D,2
17400  RPLOT -16*D,-20*D,1
17405  RPLOT -20*D,-20*D,2
17410  RPLOT -16*D,-28*D,1
17415  RPLOT -20*D,-28*D,2
17420  RPLOT -16*D,-36*D,1
17425  RPLOT -20*D,-36*D,2
17430  RPLOT -16*D,-44*D,1
17435  RPLOT -20*D,-44*D,2
17440  RETURN
17445 Half_bridge_in:!
17450  OFF ERROR
```

```
17455  !       ********* DC source ***********
17460  Sh=-96
17465  RPLOT Sh*S,Pz,1
17470  RPLOT Sh*S,3.5*S,2
17475  RPLOT (-12+Sh)*S,3.5*S,1
17480  RPLOT (12+Sh)*S,3.5*S,2
17485  RPLOT (-6+Sh)*S,-3.5*S,1
17490  RPLOT (6+Sh)*S,-3.5*S,2
17495  RPLOT Sh*S,-3.5*S,1
17500  RPLOT Sh*S,-Pz,2
17505  RPLOT (-8+Sh)*S,10*S,1
17510  RPLOT (-4+Sh)*S,10*S,2
17515  RPLOT (-6+Sh)*S,12*S,1
17520  RPLOT (-6+Sh)*S,8*S,2
17525  RPLOT (-8+Sh)*S,-10*S,1
17530  RPLOT (-4+Sh)*S,-10*S,2
17535  Sh=-64
17540  Shy=(Pz/2)/S
17545  Top=Pz
17550  Bottom=0
17555 !        ********** Electrolytic **********
17560  Elel=Elel+1
17565  RPLOT Sh*S,Top,1
17570  RPLOT Sh*S,(3.2+Shy)*S,2
17575  RPLOT (-8+Sh)*S,(3.2+Shy)*S,1
17580  RPLOT (8+Sh)*S,((3.2+Shy)*S),2
17585  Xx=(-8)
17590  Yy=SQR(256-Xx^2)-19.2
17595  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
17600  FOR Xx=-7.6 TO 7.6 STEP .4
17605   Yy=SQR(256-Xx^2)-19.2
17610   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
17615  NEXT Xx
17620  Xx=8
17625  Yy=SQR(256-Xx^2)-19.2
17630  RPLOT (Xx+Sh)*S,(Yy+Shy)*S,2
17635  RPLOT Sh*S,(-3.2+Shy)*S,1
17640  RPLOT Sh*S,Bottom,2
17645  IF Elel=2 THEN
17650   Elel=0
17655   GOTO 17695
17660  END IF
17665  !       ************************************
17670  Sh=-64
17675  Shy=(-Pz/2)/S
17680  Top=0
17685  Bottom=-Pz
17690  GOTO 17555
17695 !
17700  Sh=-32
17705  Shy=(Pz/2)/S
17710  Top=Pz
17715  Bottom=0
```

```
17720 !          ********* SWITCH *********
17725  Elel=Elel+1
17730  RPLOT Sh*S,Top,1
17735  RPLOT Sh*S,(Shy+13)*S
17740  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
17745  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
17750  FOR Xx=-1 TO 1 STEP .2
17755   Yy=SQR(1-Xx^2)-11
17760   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
17765  NEXT Xx
17770  FOR Xx=1 TO -1 STEP -.2
17775   Yy=-SQR(1-Xx^2)-11
17780   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
17785  NEXT Xx
17790  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
17795  RPLOT Sh*S,(-12+Shy)*S,1
17800  RPLOT Sh*S,Bottom,2
17805  RPLOT (16+Sh)*S,(-5+Shy)*S,1
17810  FOR Xx=+16 TO +14 STEP -.2
17815   Yy=SQR(1-(Xx-15)^2)-5
17820   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
17825  NEXT Xx
17830  FOR Xx=+14 TO +16 STEP .2
17835   Yy=-SQR(1-(Xx-15)^2)-5
17840   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
17845  NEXT Xx
17850  RPLOT (16+Sh)*S,(-5+Shy)*S,2
17855  IF Elel=2 THEN
17860   Elel=0
17865   GOTO 17900
17870  END IF
17875  Sh=-32
17880  Shy=(-Pz/2)/S
17885  Top=0
17890  Bottom=-Pz
17895  GOTO 17720
17900 ! ************ DOT ON THE RIGHT ************
17905  RPLOT -32*S,8*S,1
17910  FOR Xx=-33 TO -31 STEP .2
17915   Yy=SQR(1-(Xx+32)^2)+8
17920   RPLOT Xx*S,Yy*S
17925  NEXT Xx
17930  FOR Xx=-31 TO -33 STEP -.2
17935   Yy=-SQR(1-(Xx+32)^2)+8
17940   RPLOT Xx*S,Yy*S
17945  NEXT Xx
17950  RPLOT -32*S,8*S,2
17955 !    ********** DOT ON THE LEFT **********
17960  RPLOT -64*S,-8*S,1
17965  FOR Xx=-65 TO -63 STEP .2
17970   Yy=SQR(1-(Xx+64)^2)-8
17975   RPLOT Xx*S,Yy*S
17980  NEXT Xx
```

```
17985  FOR Xx=-63 TO -65 STEP -.2
17990   Yy=-SQR(1-(Xx+64)^2)-8
17995   RPLOT Xx*S,Yy*S
18000  NEXT Xx
18005  RPLOT -64*S,-8*S,2
18010 !          **********************************
18015 !
18020  RPLOT -96*S,Pz,1    ! TOP LINE
18025  RPLOT -32*S,Pz,2
18030  RPLOT -96*S,-Pz,1   ! BOTTOM LINE
18035  RPLOT -32*S,-Pz,2
18040 !
18045  RPLOT -32*S,8*S,1   ! TOP TERMINAL LINE
18050  RPLOT 0,8*S
18055  RPLOT 0,Pz,2
18060  RPLOT -64*S,-8*S,1  ! BOTTOM  TERMINAL LINE
18065  RPLOT 0,-8*S
18070  RPLOT 0,-Pz,2
18075  RETURN
18080 !
18085                !
18090 Full_bridge_1in:   !
18095  OFF ERROR
18100   !       ********* DC_source  *************
18105  Sh=-96
18110  RPLOT Sh*S,Pz,1
18115  RPLOT Sh*S,3.5*S,2
18120  RPLOT (-12+Sh)*S,3.5*S,1
18125  RPLOT (12+Sh)*S,3.5*S,2
18130  RPLOT (-6+Sh)*S,-3.5*S,1
18135  RPLOT (6+Sh)*S,-3.5*S,2
18140  RPLOT Sh*S,-3.5*S,1
18145  RPLOT Sh*S,-Pz,2
18150  RPLOT (-8+Sh)*S,10*S,1
18155  RPLOT (-4+Sh)*S,10*S,2
18160  RPLOT (-6+Sh)*S,12*S,1
18165  RPLOT (-6+Sh)*S,8*S,2
18170  RPLOT (-8+Sh)*S,-10*S,1
18175  RPLOT (-4+Sh)*S,-10*S,2
18180 !
18185  Sh=-64
18190  Shy=(Pz/2)/S
18195  Top=Pz
18200  Bottom=0
18205 !       ********** SWITCH **********
18210  Elel=Elel+1
18215  RPLOT Sh*S,Top,1
18220  RPLOT Sh*S,(Shy+13)*S
18225  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
18230  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
18235  FOR Xx=-1 TO 1 STEP .2
18240   Yy=SQR(1-Xx^2)-11
18245   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
```

```
18250  NEXT Xx
18255  FOR Xx=1 TO -1 STEP -.2
18260   Yy=-SQR(1-Xx^2)-11
18265   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
18270  NEXT Xx
18275  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
18280  RPLOT Sh*S,(-12+Shy)*S,1
18285  RPLOT Sh*S,Bottom,2
18290  RPLOT (16+Sh)*S,(-8+Shy)*S,1
18295  FOR Xx=+16 TO +14 STEP -.2
18300   Yy=SQR(1-(Xx-15)^2)-5
18305   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
18310  NEXT Xx
18315  FOR Xx=+14 TO +16 STEP .2
18320   Yy=-SQR(1-(Xx-15)^2)-5
18325   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
18330  NEXT Xx
18335  RPLOT (16+Sh)*S,(-5+Shy)*S,2
18340  IF Elel=1 THEN 18375
18345  IF Elel=2 THEN 18400
18350  IF Elel=3 THEN 18425
18355  IF Elel=4 THEN
18360   Elel=0
18365   GOTO 18455
18370  END IF
18375  Sh=-64
18380  Shy=(-Pz/2)/S
18385  Top=0
18390  Bottom=-Pz
18395  GOTO 18205
18400  Sh=-32
18405  Shy=(Pz/2)/S
18410  Top=Pz
18415  Bottom=0
18420  GOTO 18205!
18425  Sh=-32
18430  Shy=(-Pz/2)/S
18435  Top=0
18440  Bottom=-Pz
18445  GOTO 18205!
18450 ! ********* DOT ON THE RIGHT **********
18455  RPLOT -32*S,8*S,1
18460  FOR Xx=-33 TO -31 STEP .2
18465   Yy=SQR(1-(Xx+32)^2)+8
18470   RPLOT Xx*S,Yy*S
18475  NEXT Xx
18480  FOR Xx=-31 TO -33 STEP -.2
18485   Yy=-SQR(1-(Xx+32)^2)+8
18490   RPLOT Xx*S,Yy*S
18495  NEXT Xx
18500  RPLOT -32*S,8*S,2
18505 !   ************************************
18510 !   ********** DOT ON THE LEFT **********
```

```
18515  RPLOT -64*S,-8*S,1
18520  FOR Xx=-65 TO -63 STEP .2
18525   Yy=SQR(1-(Xx+64)^2)-8
18530   RPLOT Xx*S,Yy*S
18535  NEXT Xx
18540  FOR Xx=-63 TO -65 STEP -.2
18545   Yy=-SQR(1-(Xx+64)^2)-8
18550   RPLOT Xx*S,Yy*S
18555  NEXT Xx
18560  RPLOT -64*S,-8*S,2
18565 !            **********************************
18570 !
18575  RPLOT -96*S,Pz,1     ! TOP LINE
18580  RPLOT -32*S,Pz,2
18585  RPLOT -96*S,-Pz,1    ! BOTTOM LINE
18590  RPLOT -32*S,-Pz,2
18595 !
18600  RPLOT -32*S,8*S,1    ! TOP TERMINAL LINE
18605  RPLOT 0,8*S
18610  RPLOT 0,Pz,2
18615  RPLOT -64*S,-8*S,1   ! BOTTOM TERMINAL LINE
18620  RPLOT 0,-8*S
18625  RPLOT 0,-Pz,2
18630  RETURN
18635 Full_bridge_3in: !
18640  OFF ERROR
18645   !        ********* DC_source  **********
18650  Sh=-128
18655  RPLOT Sh*S,Pz,1
18660  RPLOT Sh*S,3.5*S,2
18665  RPLOT (-12+Sh)*S,3.5*S,1
18670  RPLOT (12+Sh)*S,3.5*S,2
18675  RPLOT (-6+Sh)*S,-3.5*S,1
18680  RPLOT (6+Sh)*S,-3.5*S,2
18685  RPLOT Sh*S,-3.5*S,1
18690  RPLOT Sh*S,-Pz,2
18695  RPLOT (-8+Sh)*S,10*S,1
18700  RPLOT (-4+Sh)*S,10*S,2
18705  RPLOT (-6+Sh)*S,12*S,1
18710  RPLOT (-6+Sh)*S,8*S,2
18715  RPLOT (-8+Sh)*S,-10*S,1
18720  RPLOT (-4+Sh)*S,-10*S,2
18725 !
18730  Sh=-96
18735  Shy=(Pz/2)/S
18740  Top=Pz
18745  Bottom=0
18750 !          ************ SWITCH ************
18755  Elel=Elel+1
18760  RPLOT Sh*S,Top,1
18765  RPLOT-Sh*S,(Shy+13)*S
18770  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
18775  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
```

```
18780  FOR Xx=-1 TO 1 STEP .2
18785   Yy=SQR(1-Xx^2)-11
18790   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
18795  NEXT Xx
18800  FOR Xx=1 TO -1 STEP -.2
18805   Yy=-SQR(1-Xx^2)-11
18810   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
18815  NEXT Xx
18820  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
18825  RPLOT Sh*S,(-12+Shy)*S,1
18830  RPLOT Sh*S,Bottom,2
18835  RPLOT (16+Sh)*S,(-5+Shy)*S,1
18840  FOR Xx=+16 TO +14 STEP -.2
18845   Yy=SQR(1-(Xx-15)^2)-5
18850   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
18855  NEXT Xx
18860  FOR Xx=+14 TO +16 STEP .2
18865   Yy=-SQR(1-(Xx-15)^2)-5
18870   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
18875  NEXT Xx
18880  RPLOT (16+Sh)*S,(-5+Shy)*S,2
18885  IF Elel=1 THEN 18930
18890  IF Elel=2 THEN 18955
18895  IF Elel=3 THEN 18980
18900  IF Elel=4 THEN 19005
18905  IF Elel=5 THEN 19030
18910  IF Elel=6 THEN
18915   Elel=0
18920   GOTO 19060
18925  END IF
18930  Sh=-96
18935  Shy=(-Pz/2)/S
18940  Top=0
18945  Bottom=-Pz
18950  GOTO 18750
18955  Sh=-64
18960  Shy=(Pz/2)/S
18965  Top=Pz
18970  Bottom=0
18975  GOTO 18750
18980  Sh=-64
18985  Shy=(-Pz/2)/S
18990  Top=0
18995  Bottom=-Pz
19000  GOTO 18750
19005  Sh=-32
19010  Shy=(Pz/2)/S
19015  Top=Pz
19020  Bottom=0
19025  GOTO 18750!
19030  Sh=-32
19035  Shy=(-Pz/2)/S
19040  Top=0
```

```
19045  Bottom=-Pz
19050  GOTO 18750!
19055 ! ************ DOT ON THE LEFT ************
19060  RPLOT -96*S,10*S,1
19065  FOR Xx=-97 TO -95 STEP .2
19070   Yy=SQR(1-(Xx+96)^2)+10
19075   RPLOT Xx*S,Yy*S
19080  NEXT Xx
19085  FOR Xx=-95 TO -97 STEP -.2
19090   Yy=-SQR(1-(Xx+96)^2)+10
19095   RPLOT Xx*S,Yy*S
19100  NEXT Xx
19105  RPLOT -96*S,10*S,2
19110 !   ************************************
19115 !   *********** DOT IN THE MIDDLE  **********
19120  RPLOT -64*S,0,1
19125  FOR Xx=-65 TO -63 STEP .2
19130   Yy=SQR(1-(Xx+64)^2)
19135   RPLOT Xx*S,Yy*S
19140  NEXT Xx
19145  FOR Xx=-63 TO -65 STEP -.2
19150   Yy=-SQR(1-(Xx+64)^2)
19155   RPLOT Xx*S,Yy*S
19160  NEXT Xx
19165  RPLOT -64*S,0,2
19170 !            *********************************
19175 !
19180 !   ************ DOT ON THE BOTTOM  ***********
19185  RPLOT -32*S,-10,1
19190  FOR Xx=-33 TO -31 STEP .2
19195   Yy=SQR(1-(Xx+32)^2)-10
19200   RPLOT Xx*S,Yy*S
19205  NEXT Xx
19210  FOR Xx=-31 TO -33 STEP -.2
19215   Yy=-SQR(1-(Xx+32)^2)-10
19220   RPLOT Xx*S,Yy*S
19225  NEXT Xx
19230  RPLOT -32*S,-10,2
19235 !            *********************************
19240 !
19245  RPLOT -128*S,Pz,1    ! TOP LINE
19250  RPLOT -32*S,Pz,2
19255  RPLOT -128*S,-Pz,1   ! BOTTOM LINE
19260  RPLOT -32*S,-Pz,2
19265 !
19270  RPLOT -96*S,10*S,1   ! TOP TERMINAL LINE
19275  RPLOT 0,10*S
19280  RPLOT 0,Pz,2
19285  RPLOT -64*S,0,1      ! MIDDLE TERMINAL LINE
19290  RPLOT 0,0,2
19295  RPLOT -32*S,-10*S,1  ! BOTTOM  TERMINAL LINE
19300  RPLOT 0,-10*S
19305  RPLOT 0,-Pz,2
```

```
19310  RETURN
19315 Phase1_rect: !
19320  OFF ERROR
19325   !        ********* AC Source  ***********
19330  Sh=-102
19335  Shy=0
19340  Top=Pz
19345  Bottom=-Pz
19350  RPLOT Sh*S,Top/2,1
19355  RPLOT Sh*S,(16+Shy)*S,2
19360  RPLOT (-16+Sh)*S,Shy*S,1
19365  FOR Xx=-16 TO 16 STEP .5
19370   Yy=SQR(256-Xx^2)
19375   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
19380  NEXT Xx
19385  FOR Xx=16 TO -16 STEP -.5
19390   Yy=-SQR(256-Xx^2)
19395   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
19400  NEXT Xx
19405  RPLOT (-16+Sh)*S,(Shy*S),2
19410  RPLOT (Sh*S),(-16+Shy)*S,1
19415  RPLOT Sh*S,Bottom/2,2
19420  RPLOT (-16+Sh)*S,(20+Shy)*S,1
19425  RPLOT (-16+Sh)*S,(12+Shy)*S,2
19430  RPLOT (-20+Sh)*S,(16+Shy)*S,1
19435  RPLOT (-12+Sh)*S,(16+Shy)*S,2
19440 !         ************************************
19445  Sh=-64
19450  Shy=(Pz/2)/S
19455  Top=Pz
19460  Bottom=0
19465 !         ************** SWITCH **************
19470  Elel=Elel+1
19475  RPLOT Sh*S,Top,1
19480  RPLOT Sh*S,(Shy+13)*S
19485  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
19490  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
19495  FOR Xx=-1 TO 1 STEP .2
19500   Yy=SQR(1-Xx^2)-11
19505   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
19510  NEXT Xx
19515  FOR Xx=1 TO -1 STEP -.2
19520   Yy=-SQR(1-Xx^2)-11
19525   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
19530  NEXT Xx
19535  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
19540  RPLOT Sh*S,(-12+Shy)*S,1
19545  RPLOT Sh*S,Bottom,2
19550  RPLOT (16+Sh)*S,(-5+Shy)*S,1
19555  FOR Xx=+16 TO +14 STEP -.2
19560   Yy=SQR(1-(Xx-15)^2)-5
19565   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
19570  NEXT Xx
```

```
19575  FOR Xx=+14 TO +16 STEP .2
19580   Yy=-SQR(1-(Xx-15)^2)-5
19585   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
19590  NEXT Xx
19595  RPLOT (16+Sh)*S,(-5+Shy)*S,2
19600  IF Elel=1 THEN 19635
19605  IF Elel=2 THEN 19660
19610  IF Elel=3 THEN 19685
19615  IF Elel=4 THEN
19620   Elel=0
19625   GOTO 19715
19630  END IF
19635  Sh=-64
19640  Shy=(-Pz/2)/S
19645  Top=0
19650  Bottom=-Pz
19655  GOTO 19465
19660  Sh=-32
19665  Shy=(Pz/2)/S
19670  Top=Pz
19675  Bottom=0
19680  GOTO 19465!
19685  Sh=-32
19690  Shy=(-Pz/2)/S
19695  Top=0
19700  Bottom=-Pz
19705  GOTO 19465!
19710 ! ************ DOT ON THE RIGHT ************
19715  RPLOT -32*S,-10*S,1
19720  FOR Xx=-33 TO -31 STEP .2
19725   Yy=SQR(1-(Xx+32)^2)-10
19730   RPLOT Xx*S,Yy*S
19735  NEXT Xx
19740  FOR Xx=-31 TO -33 STEP -.2
19745   Yy=-SQR(1-(Xx+32)^2)-10
19750   RPLOT Xx*S,Yy*S
19755  NEXT Xx
19760  RPLOT -32*S,-10*S,2
19765 !   ************************************
19770 !   *************** DOT ON THE LEFT***************
19775  RPLOT -64*S,10*S,1
19780  FOR Xx=-65 TO -63 STEP .2
19785   Yy=SQR(1-(Xx+64)^2)+10
19790   RPLOT Xx*S,Yy*S
19795  NEXT Xx
19800  FOR Xx=-63 TO -65 STEP -.2
19805   Yy=-SQR(1-(Xx+64)^2)+10
19810   RPLOT Xx*S,Yy*S
19815  NEXT Xx
19820  RPLOT -64*S,10*S,2
19825 !      ********************************
19830 !
19835  RPLOT -102*S,(Pz/2),1      ! TOP LINE
```

```
19840  RPLOT -78*S,(Pz/2)
19845  RPLOT -78*S,10*S
19850  RPLOT -64*S,10*S,2
19855  RPLOT -102*S,-Pz/2,1  ! BOTTOM LINE
19860  RPLOT -78*S,-Pz/2
19865  RPLOT -78*S,-10*S
19870  RPLOT -32*S,-10*S,2
19875  RPLOT -64*S,Pz,1    ! TOP TERMINAL LINE
19880  RPLOT 0,Pz,2
19885  RPLOT -64*S,-Pz,1  ! BOTTOM  TERMINAL LINE
19890  RPLOT 0,-Pz,2
19895  RETURN
19900 Phase3_rect_d:      !
19905  OFF ERROR
19910   !       ****************************************
19915  Xd1=-3*Pz/2+(-152+16/SQR(2))*S
19920  Xd2=-3*Pz/2+(-152-16/SQR(2))*S
19925   !       ********** PHASE C ,AC SOURCE *********
19930  RPLOT (-152*S)-Pz,Pz,1
19935  RPLOT Xd1,Pz/2+16/SQR(2)*S,2
19940  RPLOT -3*Pz/2-(152+16)*S,Pz/2,1
19945  FOR Xx=-16 TO 16 STEP .5
19950   Yy=SQR(256-Xx^2)
19955   RPLOT -3*Pz/2+(Xx-152)*S,Yy*S+Pz/2
19960  NEXT Xx
19965  FOR Xx=16 TO -16 STEP -.5
19970   Yy=-SQR(256-Xx^2)
19975   RPLOT -3*Pz/2+(Xx-152)*S,Yy*S+Pz/2
19980  NEXT Xx
19985  RPLOT -3*Pz/2-(152+16)*S,Pz/2,2
19990  RPLOT Xd2,Pz/2-16/SQR(2)*S,1
19995  RPLOT -2*Pz-152*S,0,2
20000  RPLOT (-3*Pz/2-152*S)-4/SQR(2)*S,Pz/2-(SQR(2)*16+4/SQR(2))*S,1
20005  RPLOT (-3*Pz/2-152*S)+4/SQR(2)*S,Pz/2-(SQR(2)*16-4/SQR(2))*S,2
20010  RPLOT (-3*Pz/2-152*S)-4/SQR(2)*S,Pz/2-(SQR(2)*16-4/SQR(2))*S,1
20015  RPLOT (-3*Pz/2-152*S)+4/SQR(2)*S,Pz/2-(SQR(2)*16+4/SQR(2))*S,2
20020 !         ****************************************
20025   !       ********** PHASE B ,AC SOURCE **********
20030  RPLOT (-152*S-2*Pz),0,1
20035  RPLOT ((-152-16)*S-Pz),0,2
20040  RPLOT ((-152-16)*S-Pz),0,1
20045  FOR Xx=-16 TO 16 STEP .5
20050   Yy=SQR(256-Xx^2)
20055   RPLOT -Pz+(Xx-152)*S,Yy*S
20060  NEXT Xx
20065  FOR Xx=16 TO -16 STEP -.5
20070   Yy=-SQR(256-Xx^2)
20075   RPLOT -Pz+(Xx-152)*S,Yy*S
20080  NEXT Xx
20085  RPLOT ((-152-16)*S-Pz),0,2
20090  RPLOT ((-152+16)*S-Pz),0,1
20095  RPLOT -152*S,0,2
20100  RPLOT -152*S-Pz+12*S,16*S,1
```

```
20105 RPLOT -152*S-Pz+20*S,16*S,2
20110 RPLOT -152*S-Pz+16*S,20*S,1
20115 RPLOT -152*S-Pz+16*S,12*S,2
20120 !        ******************************************
20125 Xd1=-Pz/2+(-152-16/SQR(2))*S
20130 Xd2=-Pz/2+(-152+16/SQR(2))*S
20135 !        ************* PHASE A ,AC SOURCE **************
20140 RPLOT (-152*S)-Pz,Pz,1
20145 RPLOT Xd1,Pz/2+16/SQR(2)*S,2
20150 RPLOT -Pz/2-(152+16)*S,Pz/2,1
20155 FOR Xx=-16 TO 16 STEP .5
20160  Yy=SQR(256-Xx^2)
20165  RPLOT -Pz/2+(Xx-152)*S,Yy*S+Pz/2
20170 NEXT Xx
20175 FOR Xx=16 TO -16 STEP -.5
20180  Yy=-SQR(256-Xx^2)
20185  RPLOT -Pz/2+(Xx-152)*S,Yy*S+Pz/2
20190 NEXT Xx
20195 RPLOT -Pz/2-(152+16)*S,Pz/2,2
20200 RPLOT Xd2,Pz/2-16/SQR(2)*S,1
20205 RPLOT -152*S,0,2
20210 RPLOT (-Pz/2+(-152-(SQR(2)*16-4/SQR(2)))*S)-4/SQR(2)*S,Pz/2+4/SQR(2)*S,1
20215 RPLOT (-Pz/2+(-152-(SQR(2)*16+4/SQR(2)))*S)-4/SQR(2)*S,Pz/2-4/SQR(2)*S,2
20220 RPLOT (-Pz/2+(-152-(SQR(2)*16-4/SQR(2)))*S)-4/SQR(2)*S,Pz/2-4/SQR(2)*S,1
20225 RPLOT (-Pz/2+(-152-(SQR(2)*16+4/SQR(2)))*S)-4/SQR(2)*S,Pz/2+4/SQR(2)*S,2
20230 !        ******************************************
20235 Sh=-96
20240 Shy=(Pz/2)/S
20245 Top=Pz
20250 Bottom=0
20255 !        ************** SWITCH **************
20260 Ele1=Ele1+1
20265 RPLOT Sh*S,Top,1
20270 RPLOT Sh*S,(Shy+13)*S
20275 RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
20280 RPLOT (-1+Sh)*S,(-11+Shy)*S,1
20285 FOR Xx=-1 TO 1 STEP .2
20290  Yy=SQR(1-Xx^2)-11
20295  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
20300 NEXT Xx
20305 FOR Xx=1 TO -1 STEP -.2
20310  Yy=-SQR(1-Xx^2)-11
20315  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
20320 NEXT Xx
20325 RPLOT (-1+Sh)*S,(-11+Shy)*S,2
20330 RPLOT Sh*S,(-12+Shy)*S,1
20335 RPLOT Sh*S,Bottom,2
20340 RPLOT (16+Sh)*S,(-5+Shy)*S,1
20345 FOR Xx=+16 TO +14 STEP -.2
20350  Yy=SQR(1-(Xx-15)^2)-5
20355  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
20360 NEXT Xx
20365 FOR Xx=+14 TO +16 STEP .2
```

```
20370   Yy=-SQR(1-(Xx-15)^2)-5
20375   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
20380  NEXT Xx
20385  RPLOT (16+Sh)*S,(-5+Shy)*S,2
20390  IF Elel=1 THEN 20435
20395  IF Elel=2 THEN 20460
20400  IF Elel=3 THEN 20485
20405  IF Elel=4 THEN 20510
20410  IF Elel=5 THEN 20535
20415  IF Elel=6 THEN
20420   Elel=0
20425   GOTO 20565
20430  END IF
20435  Sh=-96
20440  Shy=(-Pz/2)/S
20445  Top=0
20450  Bottom=-Pz
20455  GOTO 20255
20460  Sh=-64
20465  Shy=(Pz/2)/S
20470  Top=Pz
20475  Bottom=0
20480  GOTO 20255
20485  Sh=-64
20490  Shy=(-Pz/2)/S
20495  Top=0
20500  Bottom=-Pz
20505  GOTO 20255
20510  Sh=-32
20515  Shy=(Pz/2)/S
20520  Top=Pz
20525  Bottom=0
20530  GOTO 20255!
20535  Sh=-32
20540  Shy=(-Pz/2)/S
20545  Top=0
20550  Bottom=-Pz
20555  GOTO 20255!
20560 ! ************ DOT ON THE LEFT ************
20565  RPLOT -96*S,10*S,1
20570  FOR Xx=-97 TO -95 STEP .2
20575   Yy=SQR(1-(Xx+96)^2)+10
20580   RPLOT Xx*S,Yy*S
20585  NEXT Xx
20590  FOR Xx=-95 TO -97 STEP -.2
20595   Yy=-SQR(1-(Xx+96)^2)+10
20600   RPLOT Xx*S,Yy*S
20605  NEXT Xx
20610  RPLOT -96*S,10*S,2
20615 !   ************************************
20620 !   ************* DOT IN THE MIDDLE ***********
20625  RPLOT -64*S,0,1
20630  FOR Xx=-65 TO -63 STEP .2
```

```
20635   Yy=SQR(1-(Xx+64)^2)
20640   RPLOT Xx*S,Yy*S
20645  NEXT Xx
20650  FOR Xx=-63 TO -65 STEP -.2
20655   Yy=-SQR(1-(Xx+64)^2)
20660   RPLOT Xx*S,Yy*S
20665  NEXT Xx
20670  RPLOT -64*S,0,2
20675 !            **********************************
20680 !
20685 !   ************ DOT ON THE BOTTOM  **********
20690  RPLOT -32*S,-10,1
20695  FOR Xx=-33 TO -31 STEP .2
20700   Yy=SQR(1-(Xx+32)^2)-10
20705   RPLOT Xx*S,Yy*S
20710  NEXT Xx
20715  FOR Xx=-31 TO -33 STEP -.2
20720   Yy=-SQR(1-(Xx+32)^2)-10
20725   RPLOT Xx*S,Yy*S
20730  NEXT Xx
20735  RPLOT -32*S,-10,2
20740 !            **********************************
20745 !
20750  RPLOT -96*S,Pz,1    ! TOP LINE
20755  RPLOT 0,Pz,2
20760  RPLOT -96*S,-Pz,1   ! BOTTOM LINE
20765  RPLOT 0,-Pz,2
20770 !
20775  RPLOT (-152*S)-Pz,Pz,1   ! TOP TERMINAL LINE
20780  RPLOT -128*S,Pz
20785  RPLOT -128*S,10*S
20790  RPLOT -96*S,10*S,2
20795  RPLOT -152*S,0,1      ! MIDDLE TERMINAL LINE
20800  RPLOT -64*S,0,2
20805  RPLOT (-152*S)-2*Pz,0,1  ! BOTTOM  TERMINAL LINE
20810  RPLOT (-152*S)-2*Pz,-Pz
20815  RPLOT -128*S,-Pz
20820  RPLOT -128*S,-10*S
20825  RPLOT -32*S,-10*S,2
20830  RETURN
20835 Phase3_rect_y:                    !
20840  OFF ERROR
20845   !       ********************************************
20850  Xd1=(-152-32)*S-Pz-16*S/SQR(2)
20855  Xd2=(-152-32)*S-Pz+16*S/SQR(2)
20860   !       ************ PHASE A ,AC SOURCE ************
20865  RPLOT -Pz+(-152-64)*S,64*S,1
20870  RPLOT Xd1,(32+16/SQR(2))*S,2
20875  RPLOT (-152-48)*S-Pz,32*S,1
20880  FOR Xx=-16 TO 16 STEP .5
20885   Yy=SQR(256-Xx^2)
20890   RPLOT -Pz+(Xx-152-32)*S,(Yy+32)*S
20895  NEXT Xx
```

```
20900  FOR Xx=16 TO -16 STEP -.5
20905   Yy=-SQR(256-Xx^2)
20910   RPLOT -Pz+(Xx-152-32)*S,(Yy+32)*S
20915  NEXT Xx
20920  RPLOT (-152-48)*S-Pz,32*S,2
20925  RPLOT Xd2,(32-16/SQR(2))*S,1
20930  RPLOT -152*S-Pz,0,2
20935  RPLOT (-152-32-32/SQR(2)-4)*S-Pz,(32+4)*S,1
20940  RPLOT (-152-32-32/SQR(2)+4)*S-Pz,(32-4)*S,2
20945  RPLOT (-152-32-32/SQR(2)-4)*S-Pz,(32-4)*S,1
20950  RPLOT (-152-32-32/SQR(2)+4)*S-Pz,(32+4)*S,2
20955 !         ****************************************
20960  Xd1=(-152+32)*S-Pz+16*S/SQR(2)
20965  Xd2=(-152+32)*S-Pz-16*S/SQR(2)
20970   !         ************ PHASE B ,AC SOURCE **********
20975  RPLOT -Pz+(-152+64)*S,64*S,1
20980  RPLOT Xd1,(32+16/SQR(2))*S,2
20985  RPLOT (-152+32-16)*S-Pz,32*S,1
20990  FOR Xx=-16 TO 16 STEP .5
20995   Yy=SQR(256-Xx^2)
21000   RPLOT (-152+32+Xx)*S-Pz,(32+Yy)*S
21005  NEXT Xx
21010  FOR Xx=16 TO -16 STEP -.5
21015   Yy=-SQR(256-Xx^2)
21020   RPLOT (-152+32+Xx)*S-Pz,(32+Yy)*S
21025  NEXT Xx
21030  RPLOT (-152+32-16)*S-Pz,32*S,2
21035  RPLOT (-152+16)*S-Pz,32*S,2
21040  RPLOT Xd2,(32-16/SQR(2))*S,1
21045  RPLOT (-152*S)-Pz,0,2
21050  RPLOT (-152+32-4)*S-Pz,(32+32/SQR(2)+4)*S,1
21055  RPLOT (-152+32+4)*S-Pz,(32+32/SQR(2)-4)*S,2
21060  RPLOT (-152+32-4)*S-Pz,(32+32/SQR(2)-4)*S,1
21065  RPLOT (-152+32+4)*S-Pz,(32+32/SQR(2)+4)*S,2
21070 !         ****************************************
21075   !         ************ PHASE C ,AC SOURCE *********
21080  RPLOT -152*S-Pz,0,1
21085  RPLOT -152*S-Pz,-(32*SQR(2)-16)*S,2
21090  RPLOT -Pz-(152+16)*S,-(32*SQR(2))*S,1
21095  FOR Xx=-16 TO 16 STEP .5
21100   Yy=SQR(256-Xx^2)
21105   RPLOT -Pz-(152-Xx)*S,-(32*SQR(2)-Yy)*S
21110  NEXT Xx
21115  FOR Xx=16 TO -16 STEP -.5
21120   Yy=-SQR(256-Xx^2)
21125   RPLOT -Pz-(152-Xx)*S,-(32*SQR(2)-Yy)*S
21130  NEXT Xx
21135  RPLOT -Pz-(152+16)*S,-(32*SQR(2))*S,2
21140  RPLOT -Pz-(152)*S,-(32*SQR(2)+16)*S,1
21145  RPLOT -Pz-(152)*S,-Pz,2
21150  RPLOT -152*S-Pz+(16-4)*S,-(32*SQR(2)+16)*S,1
21155  RPLOT -152*S-Pz+(16+4)*S,-(32*SQR(2)+16)*S,2
21160  RPLOT -152*S-Pz+16*S,-(32*SQR(2)+16-4)*S,1
```

```
21165 RPLOT -152*S-Pz+16*S,-(32*SQR(2)+16+4)*S,2
21170 !          ************************************
21175 Eb:!
21180 Sh=-96
21185 Shy=(Pz/2)/S
21190 Top=Pz
21195 Bottom=0
21200 !          ************** SWITCH **************
21205 Elel=Elel+1
21210 RPLOT Sh*S,Top,1
21215 RPLOT Sh*S,(Shy+13)*S
21220 RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
21225 RPLOT (-1+Sh)*S,(-11+Shy)*S,1
21230 FOR Xx=-1 TO 1 STEP .2
21235  Yy=SQR(1-Xx^2)-11
21240  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
21245 NEXT Xx
21250 FOR Xx=1 TO -1 STEP -.2
21255  Yy=-SQR(1-Xx^2)-11
21260  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
21265 NEXT Xx
21270 RPLOT (-1+Sh)*S,(-11+Shy)*S,2
21275 RPLOT Sh*S,(-12+Shy)*S,1
21280 RPLOT Sh*S,Bottom,2
21285 RPLOT (16+Sh)*S,(-5+Shy)*S,1
21290 FOR Xx=+16 TO +14 STEP -.2
21295  Yy=SQR(1-(Xx-15)^2)-5
21300  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
21305 NEXT Xx
21310 FOR Xx=+14 TO +16 STEP .2
21315  Yy=-SQR(1-(Xx-15)^2)-5
21320  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
21325 NEXT Xx
21330 RPLOT (16+Sh)*S,(-5+Shy)*S,2
21335 IF Elel=1 THEN 21380
21340 IF Elel=2 THEN 21405
21345 IF Elel=3 THEN 21430
21350 IF Elel=4 THEN 21455
21355 IF Elel=5 THEN 21480
21360 IF Elel=6 THEN
21365  Elel=0
21370  GOTO 21510
21375 END IF
21380 Sh=-96
21385 Shy=(-Pz/2)/S
21390 Top=0
21395 Bottom=-Pz
21400 GOTO 21200
21405 Sh=-64
21410 Shy=(Pz/2)/S
21415 Top=Pz
21420 Bottom=0
21425 GOTO 21200
```

```
21430  Sh=-64
21435  Shy=(-Pz/2)/S
21440  Top=0
21445  Bottom=-Pz
21450  GOTO 21200
21455  Sh=-32
21460  Shy=(Pz/2)/S
21465  Top=Pz
21470  Bottom=0
21475  GOTO 21200!
21480  Sh=-32
21485  Shy=(-Pz/2)/S
21490  Top=0
21495  Bottom=-Pz
21500  GOTO 21200!
21505 ! ************ DOT ON THE LEFT ************
21510  RPLOT -96*S,10*S,1
21515  FOR Xx=-97 TO -95 STEP .2
21520   Yy=SQR(1-(Xx+96)^2)+10
21525   RPLOT Xx*S,Yy*S
21530  NEXT Xx
21535  FOR Xx=-95 TO -97 STEP -.2
21540   Yy=-SQR(1-(Xx+96)^2)+10
21545   RPLOT Xx*S,Yy*S
21550  NEXT Xx
21555  RPLOT -96*S,10*S,2
21560 !   ************************************
21565 !   ************** DOT IN THE MIDDLE **************
21570  RPLOT -64*S,0,1
21575  FOR Xx=-65 TO -63 STEP .2
21580   Yy=SQR(1-(Xx+64)^2)
21585   RPLOT Xx*S,Yy*S
21590  NEXT Xx
21595  FOR Xx=-63 TO -65 STEP -.2
21600   Yy=-SQR(1-(Xx+64)^2)
21605   RPLOT Xx*S,Yy*S
21610  NEXT Xx
21615  RPLOT -64*S,0,2
21620 !          **********************************
21625 !
21630 !   ************** DOT ON THE BOTTOM **************
21635  RPLOT -32*S,-10,1
21640  FOR Xx=-33 TO -31 STEP .2
21645   Yy=SQR(1-(Xx+32)^2)-10
21650   RPLOT Xx*S,Yy*S
21655  NEXT Xx
21660  FOR Xx=-31 TO -33 STEP -.2
21665   Yy=-SQR(1-(Xx+32)^2)-10
21670   RPLOT Xx*S,Yy*S
21675  NEXT Xx
21680  RPLOT -32*S,-10,2
21685 !          **********************************
21690 !
```

```
21695  RPLOT -96*S,Pz,1    ! TOP LINE
21700  RPLOT 0,Pz,2
21705  RPLOT -96*S,-Pz,1   ! BOTTOM LINE
21710  RPLOT 0,-Pz,2
21715 !
21720  RPLOT (-152-64)*S-Pz,64*S,1   ! TOP TERMINAL LINE
21725  RPLOT (-152-64)*S-Pz,Pz
21730  RPLOT -128*S,Pz
21735  RPLOT -128*S,10*S
21740  RPLOT -96*S,10*S,2
21745  RPLOT (-152+64)*S-Pz,64*S,1  ! MIDDLE TERMINAL LINE
21750  RPLOT -152*S,64*S
21755  RPLOT -152*S,0
21760  RPLOT -64*S,0,2
21765  RPLOT (-152*S)-Pz,-Pz,1  ! BOTTOM TERMINAL LINE
21770  RPLOT (-128*S),-Pz
21775  RPLOT -128*S,-10*S
21780  RPLOT -32*S,-10*S,2
21785  RETURN
21790          !
21795 Err1:ALPHA OFF
21800  BEEP
21805  RETURN
21810 En:  !
21815  GINIT
21820  DEG
21825  OFF KEY
21830  OFF KBD
21835  WINDOW 0,131,0,100
21840  MOVE 60,50
21845  FOR I=30 TO 0 STEP -1
21850   Pb=I*2
21855   PIVOT Pb
21860   PEN 1
21865   POLYGON I+2,8
21870   PEN -1
21875   PIVOT Pb
21880   POLYGON I+2,8
21885  NEXT I
21890  PIVOT 0
21895  MOVE 0,0
21900  RECTANGLE 131,100,FILL
21905  MOVE 55,50
21910  CSIZE 10,.2
21915  PEN -1
21920  LABEL "GOOD-BYE"
21925  FOR I=81.38 TO 5126.94 STEP 81.38
21930   BEEP I,.007
21935  NEXT I
21940 En1:GCLEAR
21945  OUTPUT 2;S11$&"LOAD KEY"&S22$;
21950      ! END OF CIRCUIT DRAWING PROGRAM
21955  END
```

APPENDIX 2

TREE-LINK PROGRAM

```
5 !  ********************************************************
10!  *                                                      *
15!  *              TREE-LINK   PROGRAM                     *
20!  *USING BASIC 2.0 OR 3.0 WITH EXTENDED GRAPHICS *
25!  *                                                      *
30!  ********************************************************
35!  PROGRAM CHECKS WHICH HP BASIC VERSION IS BEING USED
40     IF VAL(SYSTEM$("VERSION:BASIC"))=3 THEN Bas3=1
45     OUTPUT 2;CHR$(255)&"K";
50     DIM Zx$(8)[10],Cat1$(50)[80],Ab$[80],Hihi$[30]
55     INTEGER E(70,200),Ss,Li,Hi,Xx1(40),Yy1(40)
60     INTEGER Aa1(1000),E2(70,200)
65     INTEGER Screen1(12480),Screen2(12480)
70     INTEGER Screen3(12480),Screen4(12480)
75     S11$=CHR$(255)&CHR$(80)&CHR$(255)&CHR$(88)
80     S22$=CHR$(255)&CHR$(88)&CHR$(255)&CHR$(67)
85     OUTPUT 2;S11$&"SCRATCH KEY"&S22$; ! SCRATCH KEYS
90     GOTO 100
95 Startprogram1:   !
100    GINIT
105    Voltage=1
110    Aa=1
115    DEG
120    WINDOW 0,800,0,611
125    GRAPHICS OFF
130    ALPHA OFF
135    ON ERROR GOTO Extendg
140    RECTANGLE 0,0
145    OFF ERROR
150    GCLEAR
155    GOTO Startpro
160!   IF EXTENDED GRAPHICS IS NOT LOADED
165!   THE PROGRAM IS ABORTED
170 Extendg:GRAPHICS ON
175    CSIZE 6,.3
180    MOVE 20,70
185    LABEL "SORRY, YOU NEED EXTENDED GRAPHICS"
190    MOVE 20,60
195    LABEL "FOR THIS PROGRAM"
200    MOVE 40,10
205    LABEL "* * * P R O G R A M  A B O R T E D * * * *"
210    WAIT 15
215    GOTO End
220 Startpro:  ! START PROGRAM
225    LORG 4
230    OFF KEY
235    GCLEAR
240    GRAPHICS ON
245    PEN 1
250    F=0
255    Cat1$(1)=""
260!   DISINGUISH BETWEEN THE TWO DRIVES
```

```
265!  ONE BEING THE PROGRAM DRIVE AND THE OTHER
270!  THE DATA DRIVE
275    CAT TO Cat1$(*)
280    Hihi$=SYSTEM$("MSI")&"   "
285    IF Cat1$(1)[9,13]="PROGR" AND Hihi$[15,15]="1" THEN
290     MASS STORAGE IS ":INTERNAL,4,1"
295    ELSE
300     IF Cat1$(1)[9,13]="PROGR" AND Hihi$[15,15]<>"1" THEN 315
305     GOTO 350
310    END IF
315    Hihi$=""
320    Hihi$=SYSTEM$("MSI")&"   "
325    IF Hihi$[15,15]="1" THEN
330     MASS STORAGE IS ":INTERNAL,4,0"
335    ELSE
340     MASS STORAGE IS ":INTERNAL,4,1"
345    END IF
350    CSIZE 5,.4
355    MOVE 400,400
360    LABEL "CHECKING FOR CIRCUIT FILES"
365!  LIST OF CIRCUIT FILES STORED ON THE DATA
370!  FLOPPY DISK ARE DISPLAYED ON THE SCREEN
375    MAT Cat1$= ("")
380    CAT TO Cat1$(*);SELECT "_",COUNT Nn,NO HEADER
385    IF Nn=0 THEN
390     MOVE 400,350
395     LABEL "SORRY, NO CIRCUIT FILES FOUND"
400     WAIT 10
405     GOTO En1
410    END IF
415    GCLEAR
420    CSIZE 4,.4
425    MOVE 500,100
430    LABEL "PLEASE ENTER FILE NAME"
435    CSIZE 2.6,.8
440    MOVE 145,565
445    LABEL "NO.  FILE NAME"
450    MOVE 154,545
455    FOR I=0 TO Nn
460     IF Cat1$(I)="" THEN 475
465     Nn1=Nn1+1
470     LABEL I+1;"...";Cat1$(I)[2,10]
475    NEXT I
480    ALLOCATE Arr(7,2)
485    READ Arr(*)
490    DATA  -32,8,1,   0,8,1,    0,16,1,   32,0,1,
495    DATA  0,-16,1   0,-8,1,   -32,-8,1,   -32,8,2
500    RESTORE
505    OFF KNOB
510    OFF ERROR
515    GRAPHICS ON
520    CLIP 8,792,16,586
525    Ppy=554
```

```
530!  ****************************
535!  * ROUTINE TO LOAD CIRCUIT  *
540!  *       INTO MEMORY        *
545!  ****************************
550     Ppx=40
555 Beg2:     !
560   ' CSIZE 3,.5
565     AREA PEN 1
570!  A LARGE ARROW IS DRAWN BESIDE THE
575!  COLUMN OF FILE NAMES
580     MOVE Ppx,Ppy
585!  THE KNOB IS USED TO MOVE THE ARROW
590!  VERTICALLY POINTING AT FILE NAMES
595     SYMBOL Arr(*),FILL
600     ON KNOB .04 GOTO Xy2
605 Xy2: !
610     ALPHA OFF
615     Ppy2=Ppy
620     Ppx2=Ppx
625     OFF ERROR
630     ON KBD GOTO 650
635!  THE NUMERICAL KEYS ARE USED TO MOVE THE
640!  ARROW IN JUMPS FROM FILE NUMBER 1 TO 10 FOR
645!  QUICK SELECTION OF FILES
650     Ab$=KBD$
655     IF Ab$="1" OR Ab$="2" OR Ab$="3" OR Ab$="4" OR Ab$="5" OR Ab$="6" OR Ab$=
"7" OR Ab$="8" OR Ab$="9" OR Ab$="0" THEN
660       Ppy=554-((VAL(Ab$)-1)*16)
665       IF Ab$="0" THEN Ppy=410
670       OFF KBD
675       GOTO 1080
680     END IF
685!  IN CONJUNCTION WITH THE SHIFT KEY THE NUMERICAL KEYS
690!  ARE USED TO MOVE THE ARROW IN JUMPS FROM FILE NUMBER 11 TO 20 FOR
695!  QUICK SELECTION OF FILES
700     IF Ab$="!" OR Ab$="@" OR Ab$="#" OR Ab$="$" OR Ab$="%" OR Ab$="^" OR Ab$=
"&" OR Ab$="*" OR Ab$="(" OR Ab$=")" THEN
705       IF Ab$="!" THEN Ppy=394
710       IF Ab$="@" THEN Ppy=378
715       IF Ab$="#" THEN Ppy=362
720       IF Ab$="$" THEN Ppy=346
725       IF Ab$="%" THEN Ppy=330
730       IF Ab$="^" THEN Ppy=314
735       IF Ab$="&" THEN Ppy=298
740       IF Ab$="*" THEN Ppy=282
745       IF Ab$="(" THEN Ppy=266
750       IF Ab$=")" THEN Ppy=250
755       OFF KBD
760       GOTO 1080
765     END IF
770!  PRESSING THE CONTINUE KEY WILL FORCE
775!  THE PROGRAM TO IGNORE THIS ROUTINE
780!  REDRAW THE PREVIOUS CIRCUIT
```

```
785!  AND GO BACK TO BEG
790    IF Ab$=CHR$(255)&"C" THEN En1
795!  PRESSING THE SHIFT AND UP ARROW KEY MOVES THE
800!  LARGE ARROW TO THE TOP OF THE SCREEN
805    IF Ab$=CHR$(255)&"T" THEN
810     Ppy=38
815     OFF KBD
820     GOTO 1080
825    END IF
830!  PRESSING THE SHIFT AND DOWN ARROW KEY
835!  WILL MOVE THE ARROW TO THE BOTTOM OF THE SCREEN
840    IF Ab$=CHR$(255)&"W" THEN
845     Ppy=554
850     OFF KBD
855     GOTO 1080
860    END IF
865!  UP ARROW KEY IS USED TO MOVE THE
870!  ARROW UP ONE FILE NAME AT A TIME
875    IF Ab$=CHR$(255)&"^" THEN
880     OFF KBD
885     Y=Y+16
890     IF Ppy2+Y<38 OR Ppy2+Y>554 THEN Beg2
895     OFF KBD
900     GOTO 1080
905    END IF
910!  DOWN ARROW KEY IS USED TO MOVE THE
915!  ARROW DOWN ONE FILE NAME AT A TIME
920    IF Ab$=CHR$(255)&"V" THEN
925     Y=Y-16
930     IF Ppy2+Y<38 OR Ppy2+Y>554 THEN Beg2
935     OFF KBD
940     GOTO 1080
945    END IF
950!  PRESSING THE ENTER KEY WILL LOAD THE
955!  CIRCUIT FILE INDICATED GRAPHICALLY
960!  BY THE LARGE ARROW
965    IF Ab$=CHR$(255)&"E" THEN
970     CSIZE 4,.4
975     IF (570-Ppy)/16>Nn THEN
980      IF F=1 THEN En1
985      PEN -1
990      MOVE 500,100
995      LABEL "PLEASE ENTER FILE NAME"
1000     PEN 1
1005     MOVE 500,100
1010     LABEL "SELECT A FILE OR PRESS CONTINUE"
1015     F=1
1020     GOTO Py2
1025    END IF
1030    Z$=Cat1$((554-Ppy)/16)[1,10]
1035    MOVE 510,80
1040    LABEL "THANK-YOU"
1045    WAIT .5
```

```
1050    GOTO Rec
1055   END IF
1060 Py2: !
1065   Y=KNOBY*(-16)
1070   IF Y=0 THEN Beg2
1075   IF Ppy2+Y<38 OR Ppy2+Y>554 THEN Beg2
1080   IF Y>=0 THEN
1085    AREA PEN -1
1090    PEN -1
1095    MOVE Ppx2,Ppy2
1100    SYMBOL Arr(*),FILL
1105    Ppy=Ppy+Y
1110    MOVE Ppx,Ppy
1115    AREA PEN 1
1120    PEN 1
1125    SYMBOL Arr(*),FILL
1130    GOTO Beg2
1135   ELSE
1140    AREA PEN -1
1145    PEN -1
1150    MOVE Ppx2,Ppy2
1155    SYMBOL Arr(*),FILL
1160    AREA PEN 1
1165    PEN 1
1170    Ppy=Ppy+Y
1175    MOVE Ppx,Ppy
1180    PEN 1
1185    SYMBOL Arr(*),FILL
1190   END IF
1195   GOTO Beg2
1200 Rec:    ! CHECK DATA FLOPPY DISK
1205! THE PROGRAM CHECKS 4 OTHER CIRCUIT
1210! DATA FILES PRODUCED FROM PREVIOUS RUNS
1215! IF ANY
1220   GCLEAR
1225   MOVE 400,400
1230   LABEL "LOADING CIRCUIT FILE ";Z$[2,9]
1235   Data1=0 ! FILE "1FILE-NAME"; PWM WAVEFORMS
1240   Data2=0 ! FILE "2FILE-NAME"; PWM WAVEFORMS
1245   Data3=0 ! FILE "3FILE-NAME"; PARAMETERS ex. BASE_FREQ
1250   Data4=0 ! FILE "4FILE_NAME"; TREE CIRCUIT
1255 !
1260! THE 4 DATA FILES HAVE AN IDENTIFICATION
1265! CHARACTER APPENDED TO THE CIRCUIT FILE NAME
1270   CAT TO Cat1$(*);SELECT "1"&Z$[2,9],COUNT Nn,NO HEADER
1275   IF Nn=1 THEN Data1=1
1280   CAT TO Cat1$(*);SELECT "2"&Z$[2,9];COUNT Nn,NO HEADER
1285   IF Nn=1 THEN Data2=1
1290   CAT TO Cat1$(*);SELECT "3"&Z$[2,9],COUNT Nn,NO HEADER
1295   IF Nn=1 THEN Data3=1
1300   CAT TO Cat1$(*);SELECT "4"&Z$[2,9],COUNT Nn,NO HEADER
1305   IF Nn=1 THEN Data4=1
1310   MAT B= (0)
```

```
1315   OFF KNOB
1320   GCLEAR
1325   ASSIGN @Path TO Z$
1330   ENTER @Path;Li
1335   ALLOCATE INTEGER El(Li-1)
1340   ENTER @Path;El(*)
1345   ASSIGN @Path TO *
1350! TRANSFORMING THE CIRCUIT ELEMENT SIZE
1355! BACK TO A REAL NUMBER
1360   S=El(0)/100
1365   K=0
1370   !
1375   Converter=0
1380! CREATING THE E ARRAY
1385   Zzc=1
1390   FOR I=1 TO Li-1 STEP 5
1395    IF K=El(I) THEN
1400     J=J+4
1405    ELSE
1410     J=4
1415     Conv=Conv+1
1420    END IF
1425    Zzc=Zzc+1      !   ESTIMATE # OF "Con"
1430    K=El(I)
1435    E(El(I),0)=E(El(I),0)+1
1440    E(El(I),J-3)=El(I+1)
1445    E(El(I),J-2)=El(I+2)
1450    E(El(I),J-1)=El(I+3)
1455    E(El(I),J)=El(I+4)
1460   NEXT I
1465   Zzc=Zzc*2
1470   GCLEAR
1475   DEALLOCATE El(*)
1480   DEALLOCATE Arr(*)
1485   ALLOCATE Values(Zzc),Elem_label$(Zzc)[9],Wave$(Zzc)[9]
1490   ALLOCATE INTEGER Element2(Zzc),X1(Zzc),Y1(Zzc),X2(Zzc),Y2(Zzc)
1495   !
1500! THIS ROUTINE
1505! LOADS THE NUMBER 3 CIRCUIT DATA FILE
1510! IF IT EXIST
1515    IF Data3=1 THEN
1520     ASSIGN @Path TO "3"&Z$[2,9]
1525     ENTER @Path;Con;Npoints;Limit;Dt;Base_freq
1530     ENTER @Path;X1(*);Y1(*);X2(*);Y2(*);Element2(*);Values(*);Elem_label$(*)
,Wave$(*),Gating_wave$,Carrier_wave,Mod_wave
1535     REM  OUTPUT @Path;Con;Npoints;Limit;Dt;X1(*);Y1(*);X2(*);Y2(*);Element2(
*);Values(*);Elem_label$(*);Wave$(*)
1540     ASSIGN @Path TO *
1545     Gate$=Gating_wave$
1550     Carr=Carrier_wave
1555     Mod_w=Mod_wave
1560     Voltage=1
1565     Tcycles=Dt*Base_freq*Npoints
```

```
1570    IF Values(0)<>0 THEN Voltage=Values(0)
1575    END IF
1580    !
1585 Edit_program: !
1590    !
1595 Out1:GOSUB Redraw
1600    GOTO Startprogram
1605!
1610! ***********************************************
1615! *  FROM NOW UNTIL Startprogram SUBROUTINES     *
1620! * Rect_wave, Rect_pwm, AND Invert_pwm EXIST    *
1625! ***********************************************
1630!
1635! *****************************************************
1640! *OUTPUT WAVEFORMS FOR 1-PHASE OR 3-PHASE RECTIFIER   *
1645! *****************************************************
1650!
1655! ROUTINE TO CREATE OUTPUT WAVEFORMS OF
1660! A SINGLE AND 3 PHASE RECTIFIER
1665 Rect_wave:     ! FOR RECTIFIERS
1670    Gating_wave$="SQR"
1675    Carrier_wave=0
1680    Mod_wave=0
1685    ALLOCATE Vn1(Npoints)
1690    IF Converter_type=0 THEN
1695     FOR I=0 TO Npoints
1700      Az1=ABS(SIN(360*Base_freq*I*Dt))
1705      Vn1(I)=Az1
1710     NEXT I
1715    END IF
1720    IF Converter_type=1 OR Converter_type=2 THEN
1725     FOR I=0 TO Npoints
1730      Az1=ABS(SIN(360*Base_freq*I*Dt))
1735      M=Az1
1740      Bz1=ABS(SIN(360*Base_freq*I*Dt-120))
1745      IF Bz1>M THEN M=Bz1
1750      Cz1=ABS(SIN(360*Base_freq*I*Dt-240))
1755      IF Cz1>M THEN M=Cz1
1760      Vn1(I)=M
1765     NEXT I
1770    END IF
1775! ***********************************************
1780! *    STORE WAVEFORM IN FILE "1FILE-NAME"       *
1785! ***********************************************
1790    ON ERROR GOTO 1805
1795    Hihi$="1"&Z$[2,9]
1800    PURGE Hihi$
1805    OFF ERROR
1810    CREATE BDAT Hihi$,3,(Npoints*8+8)/3
1815    ASSIGN @Path TO Hihi$
1820    OUTPUT @Path;Vn1(*)
1825    ASSIGN @Path TO *
1830    DEALLOCATE Vn1(*)
```

```
1835   RETURN
1840   !
1845! ****************************************************
1850! *FINISHED WAVEFORMS FOR 1-PHASE OR 3-PHASE RECTIFIER*
1855! ****************************************************
1860   !
1865! ************************************************
1870! *PWM WAVEFORMS FOR 1-PHASE OR 3-PHASE RECTIFIERS*
1875! ************************************************
1880 Rect_pwm:      ! RECTIFIER PWM
1885   ALLOCATE P(300),Inter(300),V111(Period)
1890   ALLOCATE V112(Period),I11(Period*Tcycles)
1895   ALLOCATE Sw1(Period),Sw3(Period)
1900   ALLOCATE Sw4(Period),Sw2(Period)
1905   DEG
1910   PRINT CHR$(12)
1915   MAT Inter= (1000)
1920   M=1.15
1925   PRINT "Select modulation scheme"
1930   ON KEY 1 LABEL "SPWM" GOTO Spwm1
1935   ON KEY 2 LABEL "GRANI" GOTO Grant1
1940   ON KEY 3 LABEL "JAPP" GOTO Jappl
1945   ON KEY 4 LABEL "ADV_JAPP" GOTO Adv_jappl
1950   ON KEY 0 LABEL "ADV_GRANI" GOTO Adv_grantl
1955   GOTO 1930
1960 Spwm1:Scheme=1
1965   Gating_wave$="SPWM"
1970   GOTO 2030
1975 Grant1:Scheme=2
1980   Gating_wave$="GRANI"
1985   GOTO 2030
1990 Jappl:Scheme=3
1995   Gating_wave$="JAPP"
2000   GOTO 2030
2005 Adv_grantl:Scheme=4
2010   Gating_wave$="ADV GRANI"
2015   GOTO 2030
2020 Adv_jappl:Scheme=5
2025   Gating_wave$="ADV JAPP"
2030   OFF KEY
2035   PRINT CHR$(12)
2040   PRINT "Select carrier frequency."
2045   ON KEY 6 LABEL "Carr_Fre=9" GOTO Carrfre9
2050   ON KEY 7 LABEL "Carr_Fre=15" GOTO Carrfre15
2055   ON KEY 8 LABEL "Carr_Fre=21" GOTO Carrfre21
2060   ON KEY 9 LABEL "Carr_Fre=27" GOTO Carrfre27
2065   GOTO 2045
2070 Carrfre9:Triangles=9
2075   GOTO 2105
2080 Carrfre15:Triangles=15
2085   GOTO 2105
2090 Carrfre21:Triangles=21
2095   GOTO 2105
```

```
2100 Carrfre27:Triangles=27
2105   OFF KEY
2110   Carrier_wave=Triangles
2115   IF Scheme=5 THEN Triangles=Triangles+6
2120   PRINT CHR$(12)
2125   INPUT "Select the modulation index ",Mod
2130   PRINT CHR$(12)
2135   Mod_wave=Mod
2140   DEG
2145   LDIR 90
2150   CSIZE 2.9
2155   FOR A1=0 TO 120 STEP 20
2160    IF A1=0 THEN
2165     IF Scheme=1 OR Scheme=2 OR Scheme=4 THEN Triangles
2170     IF Scheme=3 OR Scheme=5 THEN Unilateral
2175     !
2180    END IF
2185   !*****************************************
2190    IF A1=20 THEN
2195     Factor=1
2200     X=0
2205     FOR N=0 TO Period
2210      IF N/(Period/360)>Inter(X) THEN
2215       IF Scheme=5 THEN
2220        Tt1=0
2225        IF (Inter(X)<60 AND Inter(X+1)=60) OR (Inter(X)<240 AND Inter(X+1)=2
40) THEN Tt1=1
2230        IF Tt1 OR (Inter(X)<180 AND Inter(X+1)=180) OR (Inter(X)<360 AND Int
er(X+1)=360) THEN
2235         X=X+2
2240         GOTO 2265
2245        END IF
2250       END IF
2255       Factor=-Factor
2260       X=X+1
2265      END IF
2270      V111(N)=Factor
2275      V112((N+(Period/3)) MOD (Period))=Factor
2280     NEXT N
2285     V112(Period)=1
2290     FOR N=0 TO Period
2295      IF V111(N)>0 THEN Sw1(N)=1
2300     NEXT N
2305     FOR N=0 TO Period
2310      V112(N)=(V111(N)-V112(N))/2
2315      IF Scheme=3 OR Scheme=5 THEN
2320       IF N/(Period/360)<180 AND V112(N)<0 THEN V112(N)=0
2325       IF N/(Period/360)>180 AND V112(N)>0 THEN V112(N)=0
2330      END IF
2335     NEXT N
2340     FOR N=0 TO Period
2345      Sw3(N)=Sw1((N+(Period/3*2)) MOD (Period))
2350     NEXT N
```

```
2355 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2360    IF Scheme<>5 THEN GOTO 2490
2365    Factor=1
2370    X=0
2375    FOR N=0 TO Period
2380     IF N/(Period/360)>Inter(X) THEN
2385      Factor=-Factor
2390      X=X+1
2395     END IF
2400     V111(N)=Factor
2405     V112((N+(Period/3)) MOD (Period))=Factor
2410    NEXT N
2415    V112(Period)=1
2420    FOR N=0 TO Period
2425     IF V111(N)>0 THEN Sw1(N)=1
2430    NEXT N
2435    FOR N=0 TO Period
2440     V112(N)=(V111(N)-V112(N))/2
2445     IF Scheme=3 OR Scheme=5 THEN
2450      IF N/(Period/360)<180 AND V112(N)<0 THEN V112(N)=0
2455      IF N/(Period/360)>180 AND V112(N)>0 THEN V112(N)=0
2460     END IF
2465    NEXT N
2470    FOR N=0 TO Period
2475     Sw3(N)=Sw1((N+(Period/3*2)) MOD (Period))
2480    NEXT N
2485   END IF
2490 !
2495 !*************************************************************
2500   IF A1=100 THEN
2505    FOR I=0 TO Period*1/6
2510     IF Scheme=1 OR Scheme=2 OR Scheme=4 THEN
2515      I11(I)=SIN(I/(Period/360))*V111(I)+SIN(I/(Period/360)+(240))*V111(I+(
Period/3*2))+SIN(I/(Period/360)+(120))*V111(I+(Period/3))
2520       IF I11(I)<0 THEN I11(I)=0
2525     END IF
2530     IF Scheme=3 OR Scheme=5 THEN
2535      I11(I)=SIN(I/(Period/360))*V112(I)+SIN(I/(Period/360)+(240))*V112(I+(
Period/3*2))+SIN(I/(Period/360)+(120))*V112(I+(Period/3))
2540       IF I11(I)<0 THEN I11(I)=0
2545     END IF
2550     I11(I)=I11(I)/SQR(3)*Mod
2555     IF Scheme=3 OR Scheme=5 THEN I11(I)=I11(I)/SQR(3)*2
2560    NEXT I
2565 ! VIEWPORT 0,132,0,100
2570 ! WINDOW 0,Period*Tcycles,-2,2
2575 ! AXES
2580    FOR I=0 TO Period*Tcycles
2585     I11(I)=I11(I MOD (Period/6))
2590    NEXT I
2595 ! FOR I=0 TO Period*Tcycles
2600 !   DRAW I,I11(I)
2605 ! NEXT I
```

```
2610! STORE WAVEFORM ON FLOPPY DISK
2615     ON ERROR GOTO 2630
2620     Hihi$="1"&Z$[2,9]
2625     PURGE Hihi$
2630     OFF ERROR
2635     CREATE BDAT Hihi$,3,(Npoints*8+8)/3
2640     ASSIGN @Path TO Hihi$
2645     OUTPUT @Path;I11(*)
2650     ASSIGN @Path TO *
2655! DEALLOCATE ARRAYS NOT BEING USED
2660     DEALLOCATE P(*),Inter(*),V111(*),V112(*)
2665     DEALLOCATE Sw1(*),Sw3(*),Sw4(*),Sw2(*)
2670     DEALLOCATE I11(*)
2675     RETURN
2680    END IF
2685  !
2690   NEXT A1
2695  !
2700   STOP
2705 Triangles:!
2710   MOVE 0,0
2715   Polarity=1
2720   FOR N=0 TO Triangles*2
2725    M=Polarity*Triangles*4/360
2730    Bb=M*(360/2/Triangles)*N
2735    IF N<=INT(Triangles/2) THEN Intersections
2740    IF Triangles*2=N THEN
2745     IF Flag=1 THEN GOTO 2750
2750     GOTO 2770
2755    END IF
2760    IF Flag=1 THEN GOTO 2765
2765    Polarity=-Polarity
2770   NEXT N
2775   IF Flag=1 THEN GOTO 24164
2780   GOTO 2175
2785 Intersections:  !
2790   P(0)=0              !STEP 0
2795   P(1)=180
2800   I=2                 !STEP 1
2805   SELECT Scheme
2810   CASE 1
2815    F1=Mod*(SIN(P(I-1)))-(M*P(I-1)+Bb)
2820    F2=Mod*(SIN(P(I-2)))-(M*P(I-2)+Bb)
2825   CASE 2
2830    F2=Mod*1.15*(SIN(P(I-2))+1/6*SIN(P(I-2)*3)-0.*SIN(P(I-2)*9))-(M*P(I-2)+B
b)
2835    F1=Mod*1.15*(SIN(P(I-1))+1/6*SIN(P(I-1)*3)-0.*SIN(P(I-1)*9))-(M*P(I-1)+B
b)
2840   CASE 3
2845    F1=Mod*1*(SIN(P(I-1)))-(M*P(I-1)+Bb)
2850    F2=Mod*1*(SIN(P(I-2)))-(M*P(I-2)+Bb)
2855   CASE 4
2860    F1=Mod*1.1545*(SIN(P(I-1))+.24*SIN(P(I-1)*3)-.025*SIN(P(I-1)*9))-(M*P(I-
```

```
1)+Bb)
2865    F2=Mod*1.1545*(SIN(P(I-2))+.24*SIN(P(I-2)*3)-.025*SIN(P(I-2)*9))-(M*P(I-
2)+Bb)
2870    CASE 5
2875    F1=Mod*1*(SIN(P(I-1)))-(M*P(I-1)+Bb)
2880    F2=Mod*1*(SIN(P(I-2)))-(M*P(I-2)+Bb)
2885    END SELECT
2890    P(I)=P(I-1)-(F1*(P(I-1)-P(I-2))/(F1-F2))
2895    IF ABS(P(I)-P(I-1))<.00001 THEN GOTO 2930
2900    IF I>100 THEN
2905     PRINT "THERE IS A PROBLEM"
2910     STOP
2915    END IF
2920    I=I+1
2925    GOTO 2805
2930    Inter(N)=P(I)
2935    IF Scheme=3 OR Scheme=5 THEN GOTO 3060
2940    Inter(N+Triangles)=P(I)+180
2945    Inter(Triangles-N)=180-P(I)
2950    Inter(2*Triangles-N)=360-P(I)
2955    GOTO 2740
2960 Unilateral:    !JAPPANESE
2965    Offset=0
2970    Factor=1
2975    MOVE 0,0
2980    FOR N=1 TO Triangles/3*2
2985     IF Flag=1 THEN GOTO 2990
2990     IF N=Triangles/3 THEN
2995      Offset=Offset+60
3000      IF Flag=1 THEN GOTO 3005
3005     END IF
3010    NEXT N
3015    IF Offset=240 THEN GOTO 3035
3020    Factor=-1
3025    Offset=180
3030    GOTO 2980
3035    Factor=1
3040    FOR N=1 TO Triangles*1/3 STEP 1
3045     M=Factor*Triangles/180
3050     Bb=(INT(N/2)*360/Triangles)*(-M)
3055     GOTO Intersections
3060     Factor=-Factor
3065     Inter(N-1)=Inter(N)
3070    NEXT N
3075    Inter(Triangles/3)=60
3080    Inter(Triangles/3+1)=120
3085    Inter(Triangles*2)=360
3090    Z=0
3095    FOR N=1 TO Triangles/3-1
3100     IF N=1 THEN GOTO 3110
3105     IF (N-1) MOD (2)=0 THEN Z=Z+2
3110     Inter(Triangles-2-Z-N)=180-Inter(N)
3115     Inter(Triangles/3+3+Z+N)=120+Inter(N)
```

```
3120   NEXT N
3125   FOR N=0 TO Triangles
3130     Inter(Triangles+N)=Inter(N)+180
3135   NEXT N
3140   IF Flag=1 THEN GOTO 24164
3145   GOTO 2175
3150   RETURN
3155   !
3160! ****************************************
3165! *FINISHED PWM WAVEFORMS FOR  RECTIFIERS *
3170! ****************************************
3175   !
3180 Invert_wave: !
3185              !
3190              !
3195              !
3200   RETURN
3205! ********************************
3210! *INVERTER PWM WAVEFORMS FOR     *
3215! * 1-PHASE AND 3-PHASE INVERTERS *
3220! ********************************
3225   !
3230 Invert_pwm: !
3235   ALLOCATE P(300),Inter(300),V111(Period*Tcycles)
3240   ALLOCATE V112(Period*Tcycles)
3245   ALLOCATE Sw1(Period),Sw3(Period)
3250   ALLOCATE Sw4(Period),Sw2(Period)
3255   DEG
3260   PRINT CHR$(12)
3265   MAT Inter= (1000)
3270   M=1.15
3275   PRINT "Select modulation scheme"
3280   ON KEY 1 LABEL "SPWM" GOTO Spwm
3285   ON KEY 2 LABEL "GRANT" GOTO Grant
3290   ON KEY 3 LABEL "JAPP" GOTO Japp
3295   ON KEY 4 LABEL "ADV_JAPP" GOTO Adv_japp
3300   ON KEY 0 LABEL "ADV_GRANT" GOTO Adv_grant
3305   GOTO 3280
3310 Spwm:Scheme=1
3315   Gating_wave$="SPWM"
3320   GOTO 3380
3325 Grant:Scheme=2
3330   Gating_wave$="GRANT"
3335   GOTO 3380
3340 Japp:Scheme=3
3345   Gating_wave$="JAPP"
3350   GOTO 3380
3355 Adv_grant:Scheme=4
3360   Gating_wave$="ADV GRANT"
3365   GOTO 3380
3370 Adv_japp:Scheme=5
3375   Gating_wave$="ADV JAPP"
3380   OFF KEY
```

```
3385   PRINT CHR$(12)
3390   PRINT "Select carrier frequency."
3395   ON KEY 6 LABEL "Carr_Fre=9" GOTO Carrfr9
3400   ON KEY 7 LABEL "Carr_Fre=15" GOTO Carrfr15
3405   ON KEY 8 LABEL "Carr_Fre=21" GOTO Carrfr21
3410   ON KEY 9 LABEL "Carr_Fre=27" GOTO Carrfr27
3415   GOTO 3395
3420 Carrfr9:Triangle1=9
3425   GOTO 3455
3430 Carrfr15:Triangle1=15
3435   GOTO 3455
3440 Carrfr21:Triangle1=21
3445   GOTO 3455
3450 Carrfr27:Triangle1=27
3455   OFF KEY
3460   Carrier_wave=Triangle1
3465   IF Scheme=5 THEN Triangle1=Triangle1+6
3470   PRINT CHR$(12)
3475   INPUT "Select the modulation index for the chosen scheme",Mod
3480   PRINT CHR$(12)
3485   Mod_wave=Mod
3490 !GRAPHICS ON
3495   DEG
3500   LDIR 90
3505   CSIZE 2.9
3510   FOR A1=0 TO 40 STEP 20
3515    IF A1=0 THEN
3520     IF Scheme=1 OR Scheme=2 OR Scheme=4 THEN Triangle1
3525     IF Scheme=3 OR Scheme=5 THEN Unilateral1
3530      !
3535    END IF
3540  !*****************************************
3545    IF A1=20 THEN
3550     Factor=1
3555     X=0
3560     FOR N=0 TO Period
3565      IF N/(Period/360)>Inter(X) THEN
3570       IF Scheme=5 THEN
3575        Tt1=0
3580        IF (Inter(X)<60 AND Inter(X+1)=60) OR (Inter(X)<240 AND Inter(X+1)=2
40) THEN Tt1=1
3585        IF Tt1=1 OR (Inter(X)<180 AND Inter(X+1)=180) OR (Inter(X)<360 AND I
nter(X+1)=360) THEN
3590         X=X+2
3595         GOTO 3620
3600        END IF
3605       END IF
3610       Factor=-Factor
3615       X=X+1
3620      END IF
3625      V11(N)=Factor
3630      V12((N+(Period/3)) MOD (Period))=Factor
3635     NEXT N
```

```
3640    V112(Period)=1
3645    FOR N=0 TO Period
3650     IF V111(N)>0 THEN Sw1(N)=1
3655    NEXT N
3660    FOR N=0 TO Period
3665     V112(N)=(V111(N)-V112(N))/2
3670     IF Scheme=3 OR Scheme=5 THEN
3675      IF N/(Period/360)<180 AND V112(N)<0 THEN V112(N)=0
3680      IF N/(Period/360)>180 AND V112(N)>0 THEN V112(N)=0
3685     END IF
3690    NEXT N
3695    FOR N=0 TO Period
3700     Sw3(N)=Sw1((N+(Period/3*2)) MOD (Period))
3705    NEXT N
3710 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
3715    IF Scheme<>5 THEN GOTO 3845
3720    Factor=1
3725    X=0
3730    FOR N=0 TO Period
3735     IF N/(Period/360)>Inter(X) THEN
3740      Factor=-Factor
3745      X=X+1
3750     END IF
3755     V111(N)=Factor
3760     V112((N+(Period/3)) MOD (Period))=Factor
3765    NEXT N
3770    V112(Period)=1
3775    FOR N=0 TO Period
3780     IF V111(N)>0 THEN Sw1(N)=1
3785    NEXT N
3790    FOR N=0 TO Period
3795     V112(N)=(V111(N)-V112(N))/2
3800     IF Scheme=3 OR Scheme=5 THEN
3805      IF N/(Period/360)<180 AND V112(N)<0 THEN V112(N)=0
3810      IF N/(Period/360)>180 AND V112(N)>0 THEN V112(N)=0
3815     END IF
3820    NEXT N
3825    FOR N=0 TO Period
3830     Sw3(N)=Sw1((N+(Period/3*2)) MOD (Period))
3835    NEXT N
3840   END IF
3845 !
3850!   WAVEFORMS 1 AND 2
3855 !
3860   IF A1=40 THEN
3865    FOR I=0 TO Period
3870     V111((I+(Period/3)) MOD (Period))=V112(I)
3875    NEXT I
3880    V111(Period)=V111(Period-1)
3885    FOR I=Period TO Period*Tcycles
3890     V112(I)=V112(I-Period)
3895     V111(I)=V111(I-Period)
3900    NEXT I
```

```
3905      !
3910!     DRAW WAVEFORMS
3915!
3920      ! VIEWPORT 0,132,50,100
3925      ! WINDOW 0,Period*Tcycles,-2,2
3930      ! AXES
3935      ! MOVE 0,0
3940      ! FOR I=0 TO Period*Tcycles
3945      !   DRAW I,V112(I)
3950      ! NEXT I
3955      ! VIEWPORT 0,132,0,50
3960      ! WINDOW 0,Period*Tcycles,-2,2
3965      ! MOVE 0,V111(0)
3970      ! FOR I=0 TO Period*Tcycles
3975      !   DRAW I,V111(I)
3980      ! NEXT I
3985        !
3990! ************************************************
3995! * STORE WAVEFORMS (1FILE-NAME) , (2FILE-NAME) *
4000! ************************************************
4005      ON ERROR GOTO 4020
4010      Hihi$="1"&Z$[2,9]
4015      PURGE Hihi$
4020      OFF ERROR
4025      CREATE BDAT Hihi$,3,((Npoints)*8+8)/3
4030      ASSIGN @Path TO Hihi$
4035      OUTPUT @Path;V112(*)
4040      ASSIGN @Path TO *
4045      ON ERROR GOTO 4060
4050      Hihi$="2"&Z$[2,9]
4055      PURGE Hihi$
4060      OFF ERROR
4065      CREATE BDAT Hihi$,3,((Npoints)*8+8)/3
4070      ASSIGN @Path TO Hihi$
4075      OUTPUT @Path;V111(*)
4080      ASSIGN @Path TO *
4085      DEALLOCATE Sw1(*),Sw3(*),Sw4(*),Sw2(*)
4090      DEALLOCATE P(*),Inter(*),V111(*),V112(*)
4095 !
4100      RETURN  ! RETURN BACK TO MAIN PROGRAM (FINISHED STORING WAVEFORMS)
4105     END IF
4110    NEXT A1
4115!   SUBROUTINE TRIANGLE1
4120 !
4125 Triangle1:!
4130    MOVE 0,0
4135    Polarity=1
4140    FOR N=0 TO Triangle1*2
4145     M=Polarity*Triangle1*4/360
4150     Bb=M*(360/2/Triangle1)*N
4155     IF N<=INT(Triangle1/2) THEN Intersection1
4160     IF Triangle1*2=N THEN
4165      IF Flag=1 THEN GOTO 4170
```

```
4170     GOTO 4190
4175    END IF
4180    IF Flag=1 THEN GOTO 4185
4185    Polarity=-Polarity
4190   NEXT N
4195   IF Flag=1 THEN GOTO 17805
4200   GOTO 3530
4205 Intersection1:                                    !
4210   P(0)=0                                          !STEP 0
4215   P(1)=180
4220   I=2                                             !STEP 1
4225   SELECT Scheme
4230   CASE 1
4235    F1=Mod*(SIN(P(I-1)))-(M*P(I-1)+Bb)
4240    F2=Mod*(SIN(P(I-2)))-(M*P(I-2)+Bb)
4245   CASE 2
4250    F2=Mod*1.15*(SIN(P(I-2))+1/6*SIN(P(I-2)*3)-0.*SIN(P(I-2)*9))-(M*P(I-2)+B
b)
4255    F1=Mod*1.15*(SIN(P(I-1))+1/6*SIN(P(I-1)*3)-0.*SIN(P(I-1)*9))-(M*P(I-1)+B
b)
4260   CASE 3
4265    F1=Mod*1*(SIN(P(I-1)))-(M*P(I-1)+Bb)
4270    F2=Mod*1*(SIN(P(I-2)))-(M*P(I-2)+Bb)
4275   CASE 4
4280    F1=Mod*1.1545*(SIN(P(I-1))+.24*SIN(P(I-1)*3)-.025*SIN(P(I-1)*9))-(M*P(I-
1)+Bb)
4285    F2=Mod*1.1545*(SIN(P(I-2))+.24*SIN(P(I-2)*3)-.025*SIN(P(I-2)*9))-(M*P(I-
2)+Bb)
4290   CASE 5
4295    F1=Mod*1*(SIN(P(I-1)))-(M*P(I-1)+Bb)
4300    F2=Mod*1*(SIN(P(I-2)))-(M*P(I-2)+Bb)
4305   END SELECT
4310   P(I)=P(I-1)-(F1*(P(I-1)-P(I-2))/(F1-F2))
4315   IF ABS(P(I)-P(I-1))<.00001 THEN GOTO 4350
4320   IF I>100 THEN
4325    PRINT "THERE IS A PROBLEM"
4330    STOP
4335   END IF
4340   I=I+1
4345   GOTO 4225
4350   Inter(N)=P(I)
4355   IF Scheme=3 OR Scheme=5 THEN GOTO 4490
4360   Inter(N+Triangle1)=P(I)+180
4365   Inter(Triangle1-N)=180-P(I)
4370   Inter(2*Triangle1-N)=360-P(I)
4375   GOTO 4160
4380 !
4385!  ROUTINE UNILATERAL
4390 Unilateral1:        !JAPPANESE
4395   Offset=0
4400   Factor=1
4405   MOVE 0,0
4410   FOR N=1 TO Triangle1/3*2
```

```
4415    IF Flag=1 THEN GOTO 4420
4420    IF N=Triangle1/3 THEN
4425     Offset=Offset+60
4430     IF Flag=1 THEN GOTO 4435
4435    END IF
4440   NEXT N
4445   IF Offset=240 THEN GOTO 4465
4450   Factor=-1
4455   Offset=180
4460   GOTO 4410
4465   Factor=1
4470   FOR N=1 TO Triangle1*1/3 STEP 1
4475    M=Factor*Triangle1/180
4480    Bb=(INT(N/2)*360/Triangle1)*(-M)
4485    GOTO Intersection1
4490    Factor=-Factor
4495    Inter(N-1)=Inter(N)
4500   NEXT N
4505   Inter(Triangle1/3)=60
4510   Inter(Triangle1/3+1)=120
4515   Inter(Triangle1*2)=360
4520   Z=0
4525   FOR N=1 TO Triangle1/3-1
4530    IF N=1 THEN GOTO 4540
4535    IF (N-1) MOD (2)=0 THEN Z=Z+2
4540    Inter(Triangle1-2-Z-N)=180-Inter(N)
4545    Inter(Triangle1/3+3+Z+N)=120+Inter(N)
4550   NEXT N
4555   FOR N=0 TO Triangle1
4560    Inter(Triangle1+N)=Inter(N)+180
4565   NEXT N
4570   IF Flag=1 THEN GOTO 17805
4575   GOTO 3530
4580   RETURN
4585  !
4590! ****************************************
4595! *FINISHED PWM WAVEFORMS FOR INVERTERS  *
4600! ****************************************
4605     !
4610 Startprogram: !
4615     !
4620   GSTORE Screen1(*)
4625! ********************************************
4630! * ENTER DEVICE VALUES AND IDENTIFY TYPE OF *
4635! *                CONVERTER                 *
4640! ********************************************
4645  !
4650   ON ERROR GOTO 4660
4655   GOTO 4680
4660   PRINT CHR$(12)
4665   PRINT "*** YOU ARE GOING TO FAST PLEASE, SLOW DOWN ***"
4670   WAIT 1
4675   PRINT CHR$(12)
```

```
4680 !
4685  Converter_type=0
4690    !
4695  IF Data3=0 THEN MAT Values= (0)
4700  Zz$=""
4705  MAT X1= (0)
4710  MAT Y1= (0)
4715  MAT X2= (0)
4720  MAT Y2= (0)
4725  MAT Element2= (0)
4730  Con=0
4735  IF Pass=0 THEN Code_8=E(8,0)
4740  IF Pass=0 THEN
4745   Code_81=Code_8
4750  END IF
4755  N_c=0
4760  Pass=1
4765  IF Converter=1 AND Data3=1 THEN
4770   IF E(62,0)<>0 THEN N_c=2
4775  ELSE
4780   N_c=1
4785  END IF
4790  FOR I=3 TO 18
4795   IF I=9 OR I=10 OR I=11 OR I=12 OR I>18 THEN 5080
4800   IF E(I,0)=0 THEN 5080
4805   FOR J=4 TO E(I,0)*4 STEP 4
4810    IF I=8 AND J>Code_8*4 THEN 5080
4815    Con=Con+1
4820    Ee=I
4825    Px1=E(I,J-3)
4830    Py1=E(I,J-2)
4835    Px2=E(I,J-1)
4840    Py2=E(I,J)
4845    Aa=1
4850    PRINT CHR$(12)
4855    GOSUB Draw_box
4860! INPUT ELEMENT FILES NAMES OR ELEMENT VALUES
4865    ! ON ERROR GOTO 4410
4870    Xx$=VAL$(Values(Con+N_c))
4875    IF Values(Con+N_c)=9999 THEN Xx$=Elem_label$(Con+N_c)
4880    SELECT I
4885    CASE 3,4
4890     IF Values(Con+N_c)=9999 THEN PRINT "DEFAULT FILE IS CALLED ";Xx$
4895     IF Values(Con+N_c)<>9999 THEN PRINT "DEFAULT VALUE IS ";Xx$;" VOLTS"
4900     INPUT "ENTER RMS VOLTAGE IN VOLTS",Xx$
4905    CASE 5,6
4910     IF Values(Con+N_c)=9999 THEN PRINT "DEFAULT FILE IS CALLED ";Xx$
4915     IF Values(Con+N_c)<>9999 THEN PRINT "DEFAULT VALUE IS ";Xx$;" AMPS"
4920     INPUT "ENTER CURRENT IN AMPS",Xx$
4925    CASE 7,8
4930     IF Values(Con+N_c)=9999 THEN PRINT "DEFAULT FILE IS CALLED ";Xx$
4935     IF Values(Con+N_c)<>9999 THEN PRINT "DEFAULT VALUE IS ";Xx$;" VOLTS"
4940     INPUT "ENTER DC VOLTAGE IN VOLTS",Xx$
```

```
4945      CASE 13
4950       IF Values(Con+N_c)=9999 THEN PRINT "DEFAULT FILE IS CALLED ";Xx$
4955       IF Values(Con+N_c)<>9999 THEN PRINT "DEFAULT VALUE IS ";Xx$;" OHMS"
4960       INPUT "ENTER RESISTANCE IN OHMS",Xx$
4965      CASE 14,15
4970       IF Values(Con+N_c)=9999 THEN PRINT "DEFAULT FILE IS CALLED ";Xx$
4975       IF Values(Con+N_c)<>9999 THEN PRINT "DEFAULT VALUE IS ";Xx$;" FARADS"
4980       INPUT "ENTER CAPACITANCE IN FARADS",Xx$
4985      CASE 16,17
4990       IF Values(Con+N_c)=9999 THEN PRINT "DEFAULT FILE IS CALLED ";Xx$
4995       IF Values(Con+N_c)<>9999 THEN PRINT "DEFAULT VALUE IS ";Xx$;" HENRIES"
5000       INPUT "ENTER INDUCTANCE IN HENRIES",Xx$
5005      END SELECT
5010      X1(Con)=Px1
5015      Y1(Con)=Py1
5020      X2(Con)=Px2
5025      Y2(Con)=Py2
5030      Element2(Con)=I
5035      ON ERROR GOTO 5045
5040       IF VAL(Xx$)=0 THEN
5045        OFF ERROR
5050        Elem_label$(Con)=Xx$
5055        Xx$="9999"
5060       END IF
5065       Values(Con)=VAL(Xx$)
5070       GOSUB Draw_box
5075     NEXT J
5080    NEXT I
5085    !
5090     Ch_voltage=Voltage
5095    !
5100     IF Converter=0 THEN 5620
5105     FOR I=60 TO 65
5110      IF E(I,0)=0 THEN 5610
5115       FOR J=4 TO E(I,0)*4 STEP 4
5120        E(8,0)=E(8,0)+1
5125        Ee=I
5130        Px1=E(I,J-3)
5135        Py1=E(I,J-2)
5140        Px2=E(I,J-1)
5145        Py2=E(I,J)
5150! ********************************************************
5155! *              IDENTIFY CONVERTER TYPE                 *
5160! * 1-0 RECTIFIER CODE=1, 3-0 DELTA RECTIFIER CODE=2     *
5165! *3-0 Y RECTIFIER CODE=3, HALF BRIDGE INVERTER CODE =4*
5170! *  1-0  INVERTER CODE =5     3-0 INVERTER CODE=6      *
5175! * ENTER ELEMENT VALUES, BASE FREQUENCY, NUMBER OF      *
5180! *POINTS, NUMBER OF CYCLES,AND CONVERTER SCHEME IF ANY*
5185! *                                                      *
5190! *   THIS PART ALSO ENTERS APPROPRIATE SOURCES          *
5195! *              TRANSPARENT TO THE USER                 *
5200! ********************************************************
5205       !
```

```
5210    SELECT I
5215    CASE 60
5220     Converter_type=4
5225     Voltage=Voltage*2
5230    CASE 61
5235     Converter_type=5
5240     Voltage=Voltage
5245    CASE 62
5250     Converter_type=6
5255     Voltage=Voltage
5260    CASE 63
5265     Converter_type=1
5270     Voltage=Voltage/SQR(2)
5275    CASE 64
5280     Converter_type=2
5285     Voltage=(Voltage/SQR(2))*SQR(3)
5290    CASE 65
5295     Converter_type=3
5300     Voltage=(Voltage/SQR(2))*SQR(3)
5305    END SELECT
5310     !
5315    SELECT I
5320    CASE 63,64,65
5325     !
5330     PRINT CHR$(12)
5335     PRINT "VOLTAGE DEFAULT VALUE IS ";Voltage;" VOLTS"
5340     INPUT "ENTER RMS VOLTAGE IN VOLTS",Voltage
5345     IF Converter_type=2 THEN Voltage=(Voltage/SQR(3))*SQR(2)
5350     IF Converter_type=1 THEN Voltage=(Voltage*SQR(2))
5355     E(8,E(8,0)*4-3)=Px1
5360     E(8,E(8,0)*4-2)=Py1
5365     E(8,E(8,0)*4-1)=Px2
5370     E(8,E(8,0)*4)=Py2
5375     GOTO 5555
5380    CASE 60,61
5385     PRINT CHR$(12)
5390     PRINT "VOLTAGE DEFAULT VALUE IS ";Voltage;" VOLTS"
5395     INPUT "ENTER DC VOLTAGE IN VOLTS ",Voltage
5400     IF I=60 THEN Voltage=Voltage/2
5405     E(8,E(8,0)*4-3)=Px1
5410     E(8,E(8,0)*4-2)=Py1
5415     E(8,E(8,0)*4-1)=Px2
5420     E(8,E(8,0)*4)=Py2
5425     GOTO 5555
5430    CASE 62
5435     PRINT CHR$(12)
5440     PRINT "VOLTAGE DEFAULT VALUE IS ";Voltage;" VOLTS"
5445     INPUT "ENTER DC VOLTAGE IN VOLTS ",Voltage
5450     Converter_type=5
5455     FOR Ik=0 TO Con-1
5460      Ii=Con-Ik
5465      Element2(Ii+1)=Element2(Ii)
5470      Elem_label$(Ii+1)=Elem_label$(Ii)
```

```
5475        Values(Ii+1)=Values(Ii)
5480      NEXT Ik
5485      Con=Con+1
5490      E(8,E(8,0)*4-3)=Px1
5495      E(8,E(8,0)*4-2)=Py2-((Py2-Py1)/2)
5500      E(8,E(8,0)*4-1)=Px2
5505      E(8,E(8,0)*4)=Py2
5510      Element2(1)=8
5515      Elem_label$(1)="2"&Z$[2,9]
5520      Values(1)=9999
5525      E(8,0)=E(8,0)+1
5530      E(8,E(8,0)*4-3)=Px1
5535      E(8,E(8,0)*4-2)=Py1
5540      E(8,E(8,0)*4-1)=Px2
5545      E(8,E(8,0)*4)=Py2-((Py2-Py1)/2)
5550    END SELECT
5555    FOR Ik=0 TO Con-1
5560      Ii=Con-Ik
5565      Element2(Ii+1)=Element2(Ii)
5570      Elem_label$(Ii+1)=Elem_label$(Ii)
5575      Values(Ii+1)=Values(Ii)
5580    NEXT Ik
5585    Con=Con+1
5590    Element2(1)=8
5595    Elem_label$(1)="1"&Z$[2,9]
5600    Values(1)=9999
5605   NEXT J
5610  NEXT I
5615  OFF ERROR
5620  PRINT CHR$(12)
5625  PRINT "DEFAULT IS NO"
5630  INPUT "WOULD YOU LIKE TO REPEAT [Y/N] ",Zz$[1,1]
5635  IF Zz$="Y" OR Zz$="y" THEN
5640   E(8,0)=Code_81
5645   Code_8=Code_81
5650   Pass=1
5655   GOTO Startprogram
5660  END IF
5665  !
5670  IF Voltage<>Ch_voltage THEN Changed=1! IF THERE IS A CHANGE
5675  Nn=0
5680  !
5685  !
5690  IF Base_freq=0 THEN Base_freq=60
5695  !
5700  !
5705  PRINT CHR$(12)
5710  PRINT "FUNDAMENTAL FREQUENCY DEFAULT VALUE IS ";Base_freq;" HZ"
5715  INPUT "INPUT FUNDAMENTAL FREQUENCY ",Nn
5720  IF Nn<0 OR (Nn<.01 AND Nn>0) OR Nn>10000 THEN
5725   BEEP
5730   PRINT CHR$(12)
5735   PRINT "The fundamental frequency you have selected is ";Nn
```

```
5740    PRINT "The allowable range is .01 - 10000 hz."
5745    WAIT 1
5750    GOTO 5705
5755   END IF
5760   IF Nn<>0 THEN
5765    IF Changed=1 THEN
5770     Changed=3
5775    ELSE
5780     Changed=2
5785    END IF
5790    Base_freq=Nn
5795   END IF
5800   !
5805   Nn=0
5810   IF Npoints=0 THEN Npoints=300
5815   !
5820   PRINT CHR$(12)
5825   PRINT "NUMBER OF POINTS DEFAULT VALUE IS ";Npoints;" POINTS"
5830   INPUT "INPUT NUMBER OF POINTS ",Nn
5835   IF (Nn<10 OR Nn>10000) AND Nn<>0 THEN
5840    BEEP
5845    PRINT CHR$(12)
5850    PRINT "The number of points  you have selected is ";Nn
5855    PRINT "The allowable range is 10 - 10000  points."
5860    WAIT 1
5865    GOTO 5820
5870   END IF
5875   IF Nn<>0 THEN
5880    IF Changed=1 THEN
5885     Changed=3
5890    ELSE
5895     Changed=2
5900    END IF
5905    Npoints=Nn
5910   END IF
5915   !
5920   Nn=0
5925   IF Tcycles=0 THEN Tcycles=3
5930   !
5935   PRINT CHR$(12)
5940   PRINT "NUMBER OF CYCLES DEFAULT VALUE IS ";Tcycles;" CYCLES"
5945   INPUT "INPUT NUMBER OF CYCLES ",Nn
5950   IF (Nn<1 OR Nn>100) AND Nn<>0 THEN
5955    BEEP
5960    PRINT CHR$(12)
5965    PRINT "The cycles you have selected is ";Nn
5970    PRINT "The allowable range is 1 - 100  cycles."
5975    WAIT 1
5980    GOTO 5935
5985   END IF
5990   IF Nn<>0 THEN
5995    IF Changed=1 THEN
6000     Changed=3
```

```
6005    ELSE
6010     Changed=2
6015    END IF
6020    Tcycles=Nn
6025   END IF
6030   !
6035   !
6040! ******************************
6045! *PERIOD AND SAMPLING TIME ARE*
6050! *     CALACULATED HERE       *
6055! ******************************
6060   Period=Npoints/Tcycles
6065   Limit=1/Base_freq*Tcycles
6070   Dt=1/Base_freq*Tcycles/Npoints
6075   !
6080   IF Converter=0 THEN GOTO 6370
6085! ******************************************************
6090! * 1-0 RECTIFIER CODE=0, 3-0 DELTA RECTIFIER CODE=1    *
6095! *3-0 Y RECTIFIER CODE=2, HALF BRIDGE INVERTER CODE =3*
6100! *  1-0  INVERTER CODE =4       3-0 INVERTER CODE=5    *
6105! ******************************************************
6110   !
6115   IF Changed=2 OR Changed=3 THEN Gating_wave$=""
6120   IF Converter_type=0 OR Converter_type<3 THEN
6125    PRINT CHR$(12)
6130    IF (Changed=2 OR Changed=3) AND Data3=1 THEN
6135     PRINT "YOU MUST RE-SELECT THE SCHEME"
6140    END IF
6145    IF Data3=1 THEN
6150     IF Gating_wave$="SQR" THEN PRINT "DEFAULT IS SQUARE WAVE SCHEME"
6155     IF Gating_wave$<>"SQR" AND Gating_wave$<>"" THEN
6160      PRINT "DEFAULT IS ";Gating_wave$;" SCHEME WITH A CARRIER FREQUENCY OF"
;Carrier_wave;" TRIANGLES"
6165      PRINT "WITH A MODULATION INDEX OF ";Mod_wave
6170     END IF
6175    END IF
6180   !
6185    INPUT "SQUARE WAVE [SQR] OR PWM SCHEME [PWM]",Ww$
6190    IF Ww$[1,3]="PWM" OR Ww$[1,3]="pwm" THEN GOSUB Rect_pwm
6195    IF Ww$[1,3]="SQR" OR Ww$[1,3]="sqr" THEN GOSUB Rect_wave
6200    IF (Ww$[1,3]<>"PWM" OR Ww$[1,3]<>"pwm" OR Ww$[1,3]="SQR" OR Ww$[1,3]="sq
r" OR Ww$[1,3]="NO") AND Gating_wave$="" THEN
6205     PRINT CHR$(12)
6210     PRINT "PLEASE, ENTER EITHER 'PWM' OR 'SQR'"
6215     GOTO 6185
6220    END IF
6225   END IF
6230   !
6235   IF Converter_type=3 OR Converter_type=4 OR Converter_type=5 THEN
6240    PRINT CHR$(12)
6245    IF (Changed=2 OR Changed=3) AND Data3=1 THEN
6250     PRINT "YOU MUST RE-SELECT THE SCHEME"
6255    END IF
```

```
6260    IF Data3=1 THEN
6265     IF Gating_wave$="SQR" THEN PRINT "DEFAULT IS SQUARE WAVE SCHEME"
6270     IF Gating_wave$<>"SQR" AND Gating_wave$<>"" THEN
6275      PRINT "DEFAULT IS ";Gating_wave$;" SCHEME WITH A CARRIER FREQUENCY OF"
;Carrier_wave;" TRIANGLES"
6280      PRINT "WITH A MODULATION INDEX OF ";Mod_wave
6285     END IF
6290    END IF
6295  !
6300    INPUT "SQUARE WAVE [SQR] OR PWM SCHEME [PWM]",Ww$
6305    IF Ww$[1,3]="PWM" OR Ww$[1,3]="pwm" THEN GOSUB Invert_pwm
6310    IF Ww$[1,3]="SQR" OR Ww$[1,3]="sqr" THEN GOSUB Invert_wave
6315    IF (Ww$[1,3]<>"PWM" OR Ww$[1,3]<>"pwm" OR Ww$[1,3]="SQR" OR Ww$[1,3]="sq
r" OR Ww$[1,3]="NO") AND Gating_wave$="" THEN
6320     PRINT CHR$(12)
6325     PRINT "PLEASE, ENTER EITHER 'PWM' OR 'SQR'"
6330    END IF
6335   END IF
6340! ******************************************
6345! *END OF ROUTINE TO ENTER ELEMENT VALUES, *
6350! *BASE FREQUENCY, NUMBER OF POINTS, NUMBER*
6355! *OF CYCLES, AND CONVERTER SCHEME (IF ANY)*
6360! ******************************************
6365  !
6370  !  START
6375  !
6380! ******************************************
6385! *ROUTINE TO REMOVE ALL CONNECTION LINES  *
6390! ******************************************
6395   MAT E2= E
6400   OFF ERROR
6405         ! ELIMINATE THE LINES
6410   IF E2(2,0)=0 THEN 6600
6415   FOR Jj=4 TO E2(2,0)*4 STEP 4
6420    IF E2(2,Jj-3)>E2(2,Jj-1) THEN
6425     Ppx1=E2(2,Jj-1)
6430     Ppx2=E2(2,Jj-3)
6435    ELSE
6440     Ppx1=E2(2,Jj-3)
6445     Ppx2=E2(2,Jj-1)
6450    END IF
6455    IF E2(2,Jj-2)>E2(2,Jj) THEN
6460     Ppy1=E2(2,Jj)
6465     Ppy2=E2(2,Jj-2)
6470    ELSE
6475     Ppy1=E2(2,Jj-2)
6480     Ppy2=E2(2,Jj)
6485    END IF
6490    FOR I=2 TO 18
6495     IF I=9 OR I=10 OR I=11 OR I=12 OR I>18 THEN 6595
6500     IF E2(I,0)=0 THEN 6595
6505     FOR J=4 TO E2(I,0)*4 STEP 4
6510      IF E2(I,J-3)=Ppx2 AND E2(I,J-2)>=Ppy1 AND E2(I,J-2)<=Ppy2 THEN
```

```
6515        E2(I,J-3)=Ppx1
6520        E2(I,J-2)=Ppy1
6525       END IF
6530       IF E2(I,J-1)=Ppx2 AND E2(I,J)>=Ppy1 AND E2(I,J)<=Ppy2 THEN
6535        E2(I,J-1)=Ppx1
6540        E2(I,J)=Ppy1
6545       END IF
6550       IF E2(I,J-2)=Ppy2 AND E2(I,J-3)>=Ppx1 AND E2(I,J-3)<=Ppx2 THEN
6555        E2(I,J-3)=Ppx1
6560        E2(I,J-2)=Ppy1
6565       END IF
6570       IF E2(I,J)=Ppy2 AND E2(I,J-1)>=Ppx1 AND E2(I,J-1)<=Ppx2 THEN
6575        E2(I,J-1)=Ppx1
6580        E2(I,J)=Ppy1
6585       END IF
6590      NEXT J
6595     NEXT I
6600    NEXT Jj
6605    Con=0
6610    FOR I=3 TO 18
6615     IF E2(I,0)=0 THEN 6655
6620     FOR J=4 TO E2(I,0)*4 STEP 4
6625      Con=Con+1
6630      X1(Con)=E2(I,J-3)
6635      Y1(Con)=E2(I,J-2)
6640      X2(Con)=E2(I,J-1)
6645      Y2(Con)=E2(I,J)
6650     NEXT J
6655    NEXT I
6660! ****************************
6665! * END OF ROUTINE TO REMOVE *
6670! * CONNECTING LINES         *
6675! ****************************
6680 !
6685! ****************************************
6690! * ROUTINE TO DETERMINE IF THE CIRCUIT *
6695! * HAS ANY CAPACITOR LOOPS OR INDUCTOR *
6700! * CUT-SETS                            *
6705! ****************************************
6710 !          TREE DESIGN - CHECK
6715 !
6720  ALLOCATE Linkk(Con),Xp(Con),Yp(Con)
6725  FOR I=1 TO Con
6730   IF Element2(I)=5 OR Element2(I)=6 THEN Linkk(I)=1
6735   IF Element2(I)=16 OR Element2(I)=17 THEN Linkk(I)=1
6740  NEXT I
6745  FOR I=1 TO Con
6750   IF Element2(I)<>13 THEN GOTO 6965
6755   IF Linkk(I)=1 THEN GOTO 6965
6760   FOR J=1 TO Con
6765    Xp(J)=0
6770    Yp(J)=0
6775   NEXT J
```

```
6780   Cnx=X1(I)
6785   Cny=Y1(I)
6790   Xp(1)=X2(I)
6795   Yp(1)=Y2(I)
6800   Aa=2
6805   FOR Kk=1 TO Con
6810    Aa_crit=Aa
6815    FOR K=1 TO Aa_crit-1
6820     FOR Test=1 TO Con
6825      IF Xp(Test)=Cnx AND Yp(Test)=Cny THEN
6830       BEEP
6835       Linkk(I)=1
6840       GOTO 6965
6845      END IF
6850     NEXT Test
6855    FOR J=1 TO Con
6860     IF Xp(K)=X1(J) AND Yp(K)=Y1(J) AND Linkk(J)=0 AND J<>I THEN
6865      FOR Ll1=1 TO Con
6870       IF Xp(Ll1)=X2(J) AND Yp(Ll1)=Y2(J) THEN GOTO 6950
6875      NEXT Ll1
6880      Xp(Aa)=X2(J)
6885      Yp(Aa)=Y2(J)
6890      Aa=Aa+1
6895      GOTO 6950
6900     END IF
6905     IF Xp(K)=X2(J) AND Yp(K)=Y2(J) AND Linkk(J)=0 AND J<>I THEN
6910      FOR Ll1=1 TO Con
6915       IF Xp(Ll1)=X1(J) AND Yp(Ll1)=Y1(J) THEN GOTO 6950
6920      NEXT Ll1
6925      Xp(Aa)=X1(J)
6930      Yp(Aa)=Y1(J)
6935      Aa=Aa+1
6940      GOTO 6950
6945     END IF
6950    NEXT J
6955   NEXT K
6960  NEXT Kk
6965 NEXT I
6970  !                          FINDING CAPACITOR LOOPS
6975 FOR I=1 TO Con
6980  IF Element2(I)<>14 AND Element2(I)<>15 THEN GOTO 7205
6985  IF Linkk(I)=1 THEN GOTO 7205
6990  FOR J=1 TO Con
6995   Xp(J)=0
7000   Yp(J)=0
7005  NEXT J
7010  Cnx=X1(I)
7015  Cny=Y1(I)
7020  Xp(1)=X2(I)
7025  Yp(1)=Y2(I)
7030  Aa=2
7035  FOR Kk=1 TO Con
7040   Aa_crit=Aa
```

```
7045   FOR K=1 TO Aa_crit-1
7050    FOR Test=1 TO Con
7055     IF Xp(Test)=Cnx AND Yp(Test)=Cny THEN
7060      BEEP
7065      MAT E= E2
7070      GOTO No_outputs
7075     END IF
7080    NEXT Test
7085    FOR J=1 TO Con
7090     IF Element2(J)=14 OR Element2(J)=15 OR Element2(J)=7 OR Element2(J)=8
 THEN
7095      IF Xp(K)=X1(J) AND Yp(K)=Y1(J) AND Linkk(J)=0 AND J<>I THEN
7100       FOR Lll=1 TO Con
7105        IF Xp(Lll)=X2(J) AND Yp(Lll)=Y2(J) THEN GOTO 7190
7110       NEXT Lll
7115       Xp(Aa)=X2(J)
7120       Yp(Aa)=Y2(J)
7125       Aa=Aa+1
7130       GOTO 7190
7135      END IF
7140      IF Xp(K)=X2(J) AND Yp(K)=Y2(J) AND Linkk(J)=0 AND J<>I THEN
7145       FOR Lll=1 TO Con
7150        IF Xp(Lll)=X1(J) AND Yp(Lll)=Y1(J) THEN GOTO 7190
7155       NEXT Lll
7160       Xp(Aa)=X1(J)
7165       Yp(Aa)=Y1(J)
7170       Aa=Aa+1
7175       GOTO 7190
7180      END IF
7185     END IF
7190    NEXT J
7195   NEXT K
7200  NEXT Kk
7205 NEXT I
7210  !   FINDING INDUCTOR CUT SETS
7215 FOR I=1 TO Con
7220  IF Element2(I)<>16 AND Element2(I)<>17 THEN GOTO 7465
7225  IF Linkk(I)=0 THEN GOTO 7465
7230  FOR J=1 TO Con
7235   Xp(J)=0
7240   Yp(J)=0
7245  NEXT J
7250  Cnx=X1(I)
7255  Cny=Y1(I)
7260  Xp(1)=X2(I)
7265  Yp(1)=Y2(I)
7270  Aa=2
7275  FOR Kk=1 TO Con
7280   Aa_crit=Aa
7285   FOR K=1 TO Aa_crit-1
7290    FOR J=1 TO Con
7295     IF Xp(K)=X1(J) AND Yp(K)=Y1(J) AND Linkk(J)=0 AND J<>I THEN
7300      FOR Lll=1 TO Con
```

```
7305          IF Xp(Lll)=X2(J) AND Yp(Lll)=Y2(J) THEN GOTO 7385
7310         NEXT Lll
7315         Xp(Aa)=X2(J)
7320         Yp(Aa)=Y2(J)
7325         Aa=Aa+1
7330         GOTO 7385
7335        END IF
7340        IF Xp(K)=X2(J) AND Yp(K)=Y2(J) AND Linkk(J)=0 AND J<>I THEN
7345         FOR Lll=1 TO Con
7350          IF Xp(Lll)=X1(J) AND Yp(Lll)=Y1(J) THEN GOTO 7385
7355         NEXT Lll
7360         Xp(Aa)=X1(J)
7365         Yp(Aa)=Y1(J)
7370         Aa=Aa+1
7375         GOTO 7385
7380        END IF
7385       NEXT J
7390      NEXT K
7395     NEXT Kk
7400     FOR Test=1 TO Con
7405      IF Xp(Test)=Cnx AND Yp(Test)=Cny THEN
7410       GOTO 7460
7415      END IF
7420     NEXT Test
7425     BEEP
7430     MAT E= E2
7435! IF THERE IS AT LEAST ONE CAPACITOR LOOP
7440! OR INDUCTOR CUT-SET THE PROGRAM JUMPS
7445! OVER THE OUTPUT SELECTION ROUTINES
7450     GOTO No_outputs
7455     Linkk(I)=0
7460           !   LABEL TO LEAVE A LOOP
7465    NEXT I
7470! END OF ROUTINE
7475 !
7480 ! ***** OUTPUT SELECTION ROUTINES *****
7485 !
7490! ******************************
7495! *GRAPHICAL ROUTINE TO SELECT*
7500! *OUTPUT CURRENT WAVEFORMS   *
7505! ******************************
7510     !
7515    Aa1(0)=0
7520    FOR I=1 TO 18
7525     IF I=9 OR I=10 OR I=11 OR I=12 OR I>18 THEN 7660
7530     IF E(I,0)=0 THEN 7660
7535     FOR J=4 TO E(I,0)*4 STEP 4
7540      Flag=0
7545      E2(I,0)=E(I,0)
7550      E2(I,J-3)=E(I,J-3)
7555      E2(I,J-2)=E(I,J-2)
7560      E2(I,J-1)=E(I,J-1)
7565      E2(I,J)=E(I,J)
```

```
7570     FOR K=2 TO Aa1(0) STEP 2
7575      IF Flag=3 THEN 7655
7580      IF E2(I,J-1)=Aa1(K-1) AND E2(I,J)=Aa1(K) AND Flag=0 THEN Flag=1
7585      IF E2(I,J-1)=Aa1(K-1) AND E2(I,J)=Aa1(K) AND Flag=2 THEN Flag=3
7590      IF E2(I,J-3)=Aa1(K-1) AND E2(I,J-2)=Aa1(K) AND Flag=0 THEN Flag=2
7595      IF E2(I,J-3)=Aa1(K-1) AND E2(I,J-2)=Aa1(K) AND Flag=1 THEN Flag=3
7600     NEXT K
7605     IF Flag<>3 AND Flag<>2 THEN
7610      Aa1(0)=Aa1(0)+2
7615      Aa1(Aa1(0)-1)=E(I,J-3)
7620      Aa1(Aa1(0))=E(I,J-2)
7625     END IF
7630     IF Flag<>1 AND Flag<>3 THEN
7635      Aa1(0)=Aa1(0)+2
7640      Aa1(Aa1(0)-1)=E(I,J-1)
7645      Aa1(Aa1(0))=E(I,J)
7650     END IF
7655    NEXT J
7660   NEXT I
7665   Flag=0
7670         !
7675   PRINT CHR$(12)
7680   WINDOW 0,800,0,611
7685   LDIR 0
7690      !
7695   OFF ERROR
7700   Aww=1
7705   PEN 1
7710   CSIZE 3.2,.5
7715   MOVE 400,570
7720   LABEL "PLEASE SELECT YOUR  OUTPUT CURRENT  WAVEFORMS"
7725   Www=Code_8
7730! ARROW KEYS AND KNOB ARE USED
7735! TO SELECT THE BRANCH
7740      !
7745   FOR I=3 TO 18
7750    IF E2(I,0)<>0 THEN
7755     IF I=8 AND Code_8=0 THEN 7800
7760     Px1=E2(I,1)
7765     Py1=E2(I,2)
7770     Px2=E2(I,3)
7775     Py2=E2(I,4)
7780     GOSUB Draw_box
7785     J=4
7790     GOTO Keyy
7795    END IF
7800   NEXT I
7805 Keyy:     !
7810   ON KBD GOTO Key1
7815 Key1: !
7820   Ab$=KBD$
7825   IF Ab$=CHR$(255)&">" THEN
7830    OFF KBD
```

```
7835   GOSUB Draw_box
7840   J=J+4
7845   IF I=8 AND J>Code_8 THEN 7855
7850   IF J>E2(I,0)*4 THEN
7855    I=I+1
7860    IF I>18 THEN 7745
7865    IF E2(I,0)=0 THEN 7855
7870    J=4
7875   END IF
7880   Px1=E2(I,J-3)
7885   Py1=E2(I,J-2)
7890   Px2=E2(I,J-1)
7895   Py2=E2(I,J)
7900   GOSUB Draw_box
7905   ON KBD GOTO Keyy
7910  END IF
7915  IF Ab$=CHR$(255)&"<" THEN
7920   OFF KBD
7925   GOSUB Draw_box
7930   J=J-4
7935   IF J=0 THEN
7940    I=I-1
7945    IF I=3 THEN I=18
7950    IF E2(I,0)=0 THEN 7940
7955    IF I=8 AND Code_8=0 THEN 7940
7960    J=E2(I,0)*4
7965    IF I=8 THEN J=Code_8*4
7970   END IF
7975   Px1=E2(I,J-3)
7980   Py1=E2(I,J-2)
7985   Px2=E2(I,J-1)
7990   Py2=E2(I,J)
7995   GOSUB Draw_box
8000   ON KBD GOTO Keyy
8005  END IF
8010! PRESSING THE CONTINUE KEY
8015! WILL EXIT THIS ROUTINE
8020! AND CONTINUE TO THE NEXT
8025  IF Ab$=CHR$(255)&"C" THEN
8030   GOSUB Draw_box
8035   GOTO Outt
8040  END IF
8045! PRESSING THE ENTER KEY WILL
8050! SELECT THE BRANCH AS AN OUTPUT
8055  IF Ab$=CHR$(255)&"E" THEN
8060   Fflag=0
8065   OFF ERROR
8070   OFF KBD
8075   Ccc=Ccc+1
8080   Wave$(Ccc)[1,1]="I"
8085   Wave$(Ccc)[2,3]=VAL$(Ccc)
8090   FOR Kk=3 TO 18
8095    IF E(Kk,0)=0 THEN 8150
```

```
8100     FOR Ll=4 TO E(Kk,0)*4 STEP 4
8105      IF E(Kk,Ll-3)=Px1 AND E(Kk,Ll-1)=Px2 AND E(Kk,Ll-2)=Py1 AND E(Kk,Ll)=P
y2 THEN
8110       Fflag=1
8115       GOTO 8155
8120      END IF
8125      IF E(Kk,Ll-3)=Px2 AND E(Kk,Ll-1)=Px1 AND E(Kk,Ll-2)=Py2 AND E(Kk,Ll)=P
y1 THEN
8130       Fflag=2
8135       GOTO 8155
8140      END IF
8145     NEXT Ll
8150    NEXT Kk
8155    IF Px1=Px2 THEN
8160     E(8,E(8,0)*4+1)=Px1
8165     E(8,E(8,0)*4+3)=Px2
8170     Aaa=30
8175     IF Py1>Py2 THEN Aaa=-30
8180     IF Fflag=1 THEN
8185      E(Kk,Ll-2)=E(Kk,Ll-2)+Aaa
8190     ELSE
8195      E(Kk,Ll)=E(Kk,Ll)+Aaa
8200     END IF
8205     E(8,E(8,0)*4+2)=Py1
8210     E(8,E(8,0)*4+4)=Py1+Aaa
8215    ELSE
8220     Aaa=30
8225     IF Px1>Px2 THEN Aaa=-30
8230     E(8,E(8,0)*4+2)=Py1
8235     E(8,E(8,0)*4+4)=Py2
8240     IF Fflag=1 THEN
8245      E(Kk,Ll-3)=E(Kk,Ll-3)+Aaa
8250     ELSE
8255      E(Kk,Ll-1)=E(Kk,Ll-1)+Aaa
8260     END IF
8265     E(8,E(8,0)*4+1)=Px1
8270     E(8,E(8,0)*4+3)=Px1+Aaa
8275    END IF
8280    E(8,0)=E(8,0)+1
8285    BEEP 3000,.01
8290    Ab$=""
8295    GOTO 7835
8300   END IF
8305! ******************************************
8310! * END OF OUTPUT CURRENT WAVEFORM ROUTINE *
8315! ******************************************
8320         !
8325   GOTO Keyy
8330 Outt:  !
8335        !
8340        !
8345! ********************************
8350! * GRAPHICAL ROUTINE TO SELECT *
```

```
8355! *   OUTPUT VOLTAGE WAVEFORMS  *
8360! *********************************
8365        !
8370   Element2(0)=1        ! IF ELEMENT2(0)=1 THEN OUTPUTS WERE SELECTED
8375        !                 ELSE  ELEMENT2(0)=0 NO OUTPUTS WERE SELECTED
8380   PEN -1
8385   CSIZE 3.2,.5
8390   MOVE 400,570
8395   LABEL "PLEASE SELECT YOUR  OUTPUT CURRENT  WAVEFORMS"
8400   PEN 1
8405   MOVE 400,570
8410   LABEL "PLEASE SELECT YOUR  OUTPUT VOLTAGE  WAVEFORMS"
8415   Www=E(6,0)
8420     !
8425   Flag=0
8430   K=2
8435   PEN 0
8440   Flag=0
8445 Kkeyy:   !
8450   ON KBD GOTO Kkey1
8455 Kkey1:  !
8460   Ab$=KBD$
8465! THE ARROW KEYS AND KNOB ARE
8470! USED TO SELECT THE NODES
8475   SET ECHO Aa1(K-1),Aa1(K)
8480   IF Ab$=CHR$(255)&">" THEN
8485    OFF KBD
8490    SET ECHO Aa1(K-1),Aa1(K)
8495    K=K+2
8500    IF K>Aa1(0) THEN K=2
8505    ON KBD GOTO Kkey1
8510   END IF
8515     !
8520   IF Ab$=CHR$(255)&"<" THEN
8525    OFF KBD
8530    SET ECHO Aa1(K-1),Aa1(K)
8535    K=K-2
8540    IF K=0 THEN K=Aa1(0)
8545    ON KBD GOTO Kkey1
8550   END IF
8555! PRESSING THE CONTINUE KEY WILL
8560! EXIT THIS ROUTINE
8565   IF Ab$=CHR$(255)&"C" THEN
8570    GOTO Ooutt
8575   END IF
8580! THE ENTER KEY MUST BE PRESSED TWICE
8585! TO SELECT THE TWO NODES TO SELECT THE
8590! OUTPUT VOLTAGE WAVEFORMS
8595   IF Ab$=CHR$(255)&"E" THEN
8600    OFF KBD
8605    IF Flag=0 THEN
8610     BEEP 2000,.01
8615     Xxx=Aa1(K-1)
```

```
8620     Yyy=Aa1(K)
8625     PEN 1
8630     MOVE 200,100
8635     LABEL "PLEASE, SELECT SECOND TERMINAL"
8640     Flag=1
8645     PEN 0
8650     ON KBD GOTO Kkey1
8655     GOTO Kkey1
8660    END IF
8665    IF Flag=1 THEN
8670     BEEP 4000,.01
8675     PEN -1
8680     MOVE 200,100
8685     LABEL "PLEASE, SELECT SECOND TERMINAL"
8690     Ccc1=Ccc1+1
8695     Wave$(Ccc+Ccc1)[1,1]="V"
8700     Wave$(Ccc+Ccc1)[2,4]=VAL$(Ccc1)
8705     E(6,0)=E(6,0)+1
8710     E(6,E(6,0)*4-3)=Aa1(K-1)
8715     E(6,E(6,0)*4-2)=Aa1(K)
8720     E(6,E(6,0)*4-1)=Xxx
8725     E(6,E(6,0)*4)=Yyy
8730     PEN 0
8735     Flag=0
8740     ON KBD GOTO Kkey1
8745     GOTO Kkey1
8750    END IF
8755   END IF
8760         !
8765   GOTO Kkeyy
8770         !
8775! ********************************************
8780! *  END OF OUTPUT VOLTAGE WAVEFORM ROUTINE *
8785! ********************************************
8790         !
8795! ******* END OF OUTPUT ROUTINES *******
8800         !
8805! ARRANGING THE ARRAYS
8810 Coutt:        !
8815   OFF KBD
8820   DEG
8825   IDIR 0
8830   CSIZE 4.2
8835   PEN 1
8840   MOVE 400,100
8845   LABEL "SOLVING YOUR OUTPUT WAVEFORMS"
8850         !
8855           !
8860   Ppx1=0
8865   Ppx2=0
8870   Ppy1=0
8875   Ppy2=0
8880   IF E(2,0)=0 THEN 9070
```

```
8885  FOR Jj=4 TO E(2,0)*4 STEP 4
8890   IF E(2,Jj-3)>E(2,Jj-1) THEN
8895    Ppx1=E(2,Jj-1)
8900    Ppx2=E(2,Jj-3)
8905   ELSE
8910    Ppx1=E(2,Jj-3)
8915    Ppx2=E(2,Jj-1)
8920   END IF
8925   IF E(2,Jj-2)>E(2,Jj) THEN
8930    Ppy1=E(2,Jj)
8935    Ppy2=E(2,Jj-2)
8940   ELSE
8945    Ppy1=E(2,Jj-2)
8950    Ppy2=E(2,Jj)
8955   END IF
8960   FOR I=2 TO 18
8965    IF I=9 OR I=10 OR I=11 OR I=12 OR I>18 THEN 9065
8970    IF E(I,0)=0 THEN 9065
8975    FOR J=4 TO E(I,0)*4 STEP 4
8980     IF E(I,J-3)=Ppx2 AND E(I,J-2)>=Ppy1 AND E(I,J-2)<=Ppy2 THEN
8985      E(I,J-3)=Ppx1
8990      E(I,J-2)=Ppy1
8995     END IF
9000     IF E(I,J-1)=Ppx2 AND E(I,J)>=Ppy1 AND E(I,J)<=Ppy2 THEN
9005      E(I,J-1)=Ppx1
9010      E(I,J)=Ppy1
9015     END IF
9020     IF E(I,J-2)=Ppy2 AND E(I,J-3)>=Ppx1 AND E(I,J-3)<=Ppx2 THEN
9025      E(I,J-3)=Ppx1
9030      E(I,J-2)=Ppy1
9035     END IF
9040     IF E(I,J)=Ppy2 AND E(I,J-1)>=Ppx1 AND E(I,J-1)<=Ppx2 THEN
9045      E(I,J-1)=Ppx1
9050      E(I,J)=Ppy1
9055     END IF
9060    NEXT J
9065   NEXT I
9070  NEXT Jj
9075  E(8,0)=E(8,0)-Ccc
9080  E(6,0)=E(6,0)-Ccc1
9085  Con=0
9090  FOR I=3 TO 18
9095   IF E(I,0)=0 THEN 9135
9100   FOR J=4 TO E(I,0)*4 STEP 4
9105    Con=Con+1
9110    X1(Con)=E(I,J-3)
9115    Y1(Con)=E(I,J-2)
9120    X2(Con)=E(I,J-1)
9125    Y2(Con)=E(I,J)
9130   NEXT J
9135  NEXT I
9140  IF Ccc=0 THEN 9190
9145  FOR J=(E(8,0)+1)*4 TO (E(8,0)+Ccc)*4 STEP 4
```

```
9150    Con=Con+1
9155    Element2(Con)=8
9160    Values(Con)=0
9165    X1(Con)=E(8,J-3)
9170    Y1(Con)=E(8,J-2)
9175    X2(Con)=E(8,J-1)
9180    Y2(Con)=E(8,J)
9185   NEXT J
9190   IF Ccc1=0 THEN 9240
9195   FOR J=(E(6,0)+1)*4 TO (E(6,0)+Ccc1)*4 STEP 4
9200    Con=Con+1
9205    Element2(Con)=6
9210    Values(Con)=0
9215    X1(Con)=E(6,J-3)
9220    Y1(Con)=E(6,J-2)
9225    X2(Con)=E(6,J-1)
9230    Y2(Con)=E(6,J)
9235   NEXT J
9240   E(2,0)=0
9245   Aa=1
9250   Dflag=0
9255   !
9260 No_outputs:                    ! NO OUTPUT
9265   OFF KBD
9270   CSIZE 4.2
9275   LDIR 0
9280   PEN 1
9285   MOVE 400,100
9290   LABEL "SOLVING YOUR OUTPUT WAVEFORMS"
9295   !
9300   IF Changed=0 AND Data3=1 THEN
9305    GOTO 9370
9310   ELSE
9315    ON ERROR GOTO 9325
9320    PURGE "3"&Z$[2,9]
9325    OFF ERROR !
9330! STORING DATA IN FILE 3
9335    Values(0)=Voltage
9340    CREATE BDAT "3"&Z$[2,9],1,(6*(Con+1)*8+(5*10*(Con+1))+(8+5*8))
9345    ASSIGN @Path TO "3"&Z$[2,9]
9350    OUTPUT @Path;Con;Npoints;Limit;Dt;Base_freq;X1(*);Y1(*);X2(*);Y2(*);Elem
ent2(*);Values(*);Elem_label$(*);Wave$(*);
9355    OUTPUT @Path;Gating_wave$;Carrier_wave;Mod_wave
9360    ASSIGN @Path TO *
9365   END IF
9370   !
9375 Continue1:  !
9380   FOR I=1 TO Con
9385    PRINT I,Element2(I),Values(I),X1(I),Y1(I),X2(I),Y2(I)
9390   NEXT I
9395   !
9400      !  SO FAR THE ARRAYS VALUES(*) ,  ELEMENT2(*) ,X1(*),
9405      !  X2(*) , Y1(*)  AND  Y2(*)  WERE  GENERATED
```

```
9410      !            REAL :   VALUES(*)
9415      !         INTEGER: ELEMENT2(*),X1(*),X2(*),Y1(*) AND Y2(*)
9420!
9425! CALL TIME DOMAIN ROUTINE
9430  GOSUB Timedomain ! (Con,Values(*),X1(*),X2(*),Y1(*),Y2(*),Element2(*),Ele
m_label$(*),Wave$(*))
9435      !
9440  GOTO En1
9445 Draw_box:    ! DRAWING BOX ROUTINE
9450  OFF KEY
9455  PEN Aa
9460  Angle=0
9465  ON ERROR GOTO Pangle2
9470  Angle=ATN((Py2-Py1)/(Px2-Px1))
9475  GOTO Pstart2
9480 Pangle2:Angle=90
9485  OFF ERROR
9490 Pstart2:                    !
9495  M1=(Px1+Px2)/2
9500  M2=(Py1+Py2)/2
9505  IF Py2-Py1=0 AND Px2<Px1 THEN Angle=180
9510  IF Py2<Py1 AND Px2-Px1=0 THEN Angle=270
9515  IF Py2-Py1<>0 AND Px1-Px2<>0 THEN
9520   IF Px1>Px2 AND Py1>Py2 THEN Angle=Angle+180
9525   IF Px1>Px2 AND Py1<Py2 THEN Angle=Angle+180
9530  END IF
9535  SELECT Ee
9540  CASE =6
9545   Ff=20
9550  CASE =10
9555   Ff=16/(S+1.E-39)
9560  CASE ELSE
9565   Ff=16
9570  END SELECT
9575  Aaa=Ff*(Px2-Px1)/((Px2-Px1)^2+(Py2-Py1)^2)^.5
9580  Bbb=Ff*(Py2-Py1)/((Px2-Px1)^2+(Py2-Py1)^2)^.5
9585  Bbb=Bbb*S
9590  Aaa=Aaa*S
9595  Pz=SQR((Px2-Px1)^2+(Py2-Py1)^2)/2
9600  MOVE M1+2,M2+2
9605  PIVOT Angle-90
9610  GOSUB Device_box
9615  PIVOT 0
9620  RETURN
9625! DRAWING ELEMENT ROUTINE
9630 Redraw:  !
9635  IF Dflag=1 THEN
9640   GINIT
9645   VIEWPORT 0,131,0,100
9650   WINDOW 0,800,0,611
9655   GCLEAR
9660   GRAPHICS ON
9665   ALPHA OFF
```

```
9670    CLIP 8,792,8,600
9675   END IF
9680   FOR I=1 TO 70
9685    IF E(I,0)=0 THEN 9785
9690    FOR J=4 TO E(I,0)*4 STEP 4
9695     IF E(I,J-3)=0 THEN 9780
9700     IF I>59 AND I<66 THEN Converter=1 ! A CONVERTER  0: NO , 1: YES
9705     Ee=I
9710     Px1=E(I,J-3)
9715     Py1=E(I,J-2)
9720     Px2=E(I,J-1)
9725     Py2=E(I,J)
9730     Flag=0
9735     SELECT Ee
9740     CASE =1
9745      GOSUB Pdo
9750     CASE =2
9755      GOSUB Pli
9760     CASE ELSE
9765      GOSUB Pdraw1
9770     END SELECT
9775     !
9780    NEXT J
9785   NEXT I
9790   Flag=0
9795   PLOTTER IS 3,"INTERNAL"
9800   RETURN
9805 Pdo:     ! DRAW DOTS
9810   PEN Aa
9815   MOVE Px1+2,Py1+2
9820   POLYGON 2,30
9825   POLYGON 1,30
9830   POLYGON .5,30
9835   OFF KEY
9840   RETURN
9845     !
9850 Pli:     ! DRAW LINE
9855   PEN Aa
9860   MOVE Px1+2,Py1+2
9865   DRAW Px2+2,Py2+2
9870   OFF KEY
9875   RETURN
9880 Pdraw1:     !
9885   GOSUB Error_control
9890   OFF KEY
9895   PEN Aa
9900   Angle=0
9905   ON ERROR GOTO Pangle1
9910   Angle=ATN((Py2-Py1)/(Px2-Px1))
9915   GOTO Pstart1
9920 Pangle1:Angle=90
9925 Pstart1:                    !
9930   M1=(Px1+Px2)/2
```

```
9935  M2=(Py1+Py2)/2
9940  IF Py2-Py1=0 AND Px2<Px1 THEN Angle=180
9945  IF Py2<Py1 AND Px2-Px1=0 THEN Angle=270
9950  IF Py2-Py1<>0 AND Px1-Px2<>0 THEN
9955   IF Px1>Px2 AND Py1>Py2 THEN Angle=Angle+180
9960   IF Px1>Px2 AND Py1<Py2 THEN Angle=Angle+180
9965  END IF
9970  SELECT Ee
9975  CASE =6
9980   Ff=20
9985  CASE =10
9990   Ff=16/(S+1.E-39)
9995  CASE ELSE
10000  Ff=16
10005 END SELECT
10010 Aaa=Ff*(Px2-Px1)/((Px2-Px1)^2+(Py2-Py1)^2)^.5
10015 Bbb=Ff*(Py2-Py1)/((Px2-Px1)^2+(Py2-Py1)^2)^.5
10020 Bbb=Bbb*S
10025 Aaa=Aaa*S
10030 Pz=SQR((Px2-Px1)^2+(Py2-Py1)^2)/2
10035 MOVE M1+2,M2+2
10040 PIVOT Angle-90
10045 SELECT Ee
10050 CASE =3
10055  GOSUB Ac_volt
10060 CASE =4
10065  GOSUB A_volt
10070 CASE =5
10075  GOSUB Cirrent_sour
10080 CASE =6
10085  GOSUB Current
10090 CASE =7
10095  GOSUB Battery
10100 CASE =8
10105  GOSUB Dc_source
10110 CASE =9
10115  GOSUB Arrow
10120 CASE =13
10125  GOSUB Resistor
10130 CASE =14
10135  GOSUB Capacitor
10140 CASE =15
10145  GOSUB Electrolytic
10150 CASE =16
10155  GOSUB Inductor
10160 CASE =17
10165  GOSUB Choke
10170 CASE =60
10175  GOSUB Half_bridge_in
10180 CASE =61
10185  GOSUB Full_bridge_1in
10190 CASE =62
10195  GOSUB Full_bridge_3in
```

```
10200  CASE =63
10205   GOSUB Phase1_rect
10210  CASE =64
10215   GOSUB Phase3_rect_d
10220  CASE =65
10225   GOSUB Phase3_rect_y
10230  END SELECT
10235  PIVOT 0
10240  RETURN
10245 Error_control:        !
10250  RETURN
10255 !
10260! CIRCUIT ELEMENTS
10265 !
10270 Device_box:  !
10275  PEN 0
10280  RPLOT 0,Pz,1
10285  RPLOT -24*S,20*S
10290  RPLOT -24*S,-20*S
10295  RPLOT 0,-Pz
10300  RPLOT 24*S,-20*S
10305  RPLOT 24*S,20*S
10310  RPLOT 0,Pz,2
10315  IF Aww=1 THEN
10320   RPLOT -32*S,20*S,1
10325   RPLOT -32*S,-20*S
10330   RPLOT -28*S,-16*S,2
10335   RPLOT -32*S,-20*S,1
10340   RPLOT -36*S,-16*S,2
10345  END IF
10350  RETURN
10355  !
10360 Arrow: !
10365  RPLOT 0,-Pz,1
10370  RPLOT 0,Pz
10375  FOR K=0 TO 2 STEP .5
10380   RPLOT -3+K,Pz-5+K
10385   RPLOT 3-K,Pz-5+K
10390   RPLOT 0,Pz-K
10395  NEXT K
10400  RETURN
10405  !
10410 Current:   !
10415  RPLOT 0,Pz,1
10420  RPLOT 0,20*S,2
10425  RPLOT -12*S,8*S,1
10430  FOR Xx=-12 TO 12 STEP .5
10435   Yy=SQR(144-Xx^2)+8
10440   RPLOT Xx*S,Yy*S
10445  NEXT Xx
10450  FOR Xx=12 TO -12 STEP -.5
10455   Yy=-SQR(144-Xx^2)+8
10460   RPLOT Xx*S,Yy*S
```

```
10465  NEXT Xx
10470  RPLOT -12*S,8*S,2
10475  RPLOT -12*S,-8*S,1
10480  FOR Xx=-12 TO 12 STEP .5
10485   Yy=SQR(144-Xx^2)-8
10490   RPLOT Xx*S,Yy*S
10495  NEXT Xx
10500  FOR Xx=12 TO -12 STEP -.5
10505   Yy=-SQR(144-Xx^2)-8
10510   RPLOT Xx*S,Yy*S
10515  NEXT Xx
10520  RPLOT -12*S,-8*S,2
10525  RPLOT -20*S,12*S,1
10530  RPLOT -20*S,-12*S
10535  RPLOT -24*S,-8*S
10540  RPLOT -20*S,-12*S
10545  RPLOT -16*S,-8*S,2
10550  RPLOT 0,-20*S,1
10555  RPLOT 0,-Pz,2
10560  RETURN
10565  !
10570 Current_sour:    !
10575  RPLOT 0,Pz,1
10580  RPLOT 0,16*S,2
10585  RPLOT -16*S,0,1
10590  FOR Xx=-16 TO 16 STEP .5
10595   Yy=SQR(256-Xx^2)
10600   RPLOT Xx*S,Yy*S
10605  NEXT Xx
10610  FOR Xx=16 TO -16 STEP -.5
10615   Yy=-SQR(256-Xx^2)
10620   RPLOT Xx*S,Yy*S
10625  NEXT Xx
10630  RPLOT -16*S,Yy*S,2
10635  RPLOT 0,-16*S,1
10640  RPLOT 0,-Pz,2
10645  RPLOT -16*S,0,2
10650  RPLOT 0,12*S,1
10655  RPLOT 0,-12*S,2
10660  RPLOT 0,12*S,1
10665  RPLOT -4*S,8*S,2
10670  RPLOT 0,12*S,1
10675  RPLOT 4*S,8*S,2
10680  RETURN
10685  !
10690 Dc_source:    !
10695  RPLOT 0,Pz,1
10700  RPLOT 0,3.5*S,2
10705  RPLOT -12*S,3.5*S,1
10710  RPLOT 12*S,3.5*S,2
10715  RPLOT -6*S,-3.5*S,1
10720  RPLOT 6*S,-3.5*S,2
10725  RPLOT 0,-3.5*S,1
```

```
10730 RPLOT 0,-Pz,2
10735 RPLOT -8*S,10*S,1
10740 RPLOT -4*S,10*S,2
10745 RPLOT -6*S,12*S,1
10750 RPLOT -6*S,8*S,2
10755 RPLOT -8*S,-10*S,1
10760 RPLOT -4*S,-10*S,2
10765 RETURN
10770 !
10775 Battery:      !
10780 RPLOT 0,Pz,1
10785 RPLOT 0,12*S,2
10790 RPLOT -12*S,12*S
10795 RPLOT 12*S,12*S,2
10800 RPLOT -6*S,4*S,1
10805 RPLOT 6*S,4*S,2
10810 RPLOT -12*S,-4*S,1
10815 RPLOT 12*S,-4*S,2
10820 RPLOT -6*S,-12*S,1
10825 RPLOT 6*S,-12*S,2
10830 RPLOT 0,-12*S,1
10835 RPLOT 0,-Pz,2
10840 RETURN
10845 !
10850 A volt:   !
10855 RPLOT 0,Pz,1
10860 RPLOT 0,16*S,2
10865 RPLOT -16*S,0,1
10870 FOR Xx=-16 TO 16 STEP .5
10875  Yy=SQR(256-Xx^2)
10880  RPLOT Xx*S,Yy*S
10885 NEXT Xx
10890 FOR Xx=16 TO -16 STEP -.5
10895  Yy=-SQR(256-Xx^2)
10900  RPLOT Xx*S,Yy*S
10905 NEXT Xx
10910 RPLOT -16*S,0,2
10915 MOVE M1+2,M2+2
10920 FOR Xx=9 TO -9 STEP -.5
10925  Yy=-SIN(Xx*20)
10930  RPLOT Xx*S,(Yy*5)*S
10935 NEXT Xx
10940 RPLOT -9*S,Yy*S,2
10945 RPLOT 0,-16*S,1
10950 RPLOT 0,-Pz,2
10955 RETURN
10960 !
10965 Ac volt:   !
10970 RPLOT 0,Pz,1
10975 RPLOT 0,16*S,2
10980 RPLOT -16*S,0,1
10985 FOR Xx=-16 TO 16 STEP .5
10990  Yy=SQR(256-Xx^2)
```

```
10995   RPLOT Xx*S,Yy*S
11000  NEXT Xx
11005  FOR Xx=16 TO -16 STEP -.5
11010   Yy=-SQR(256-Xx^2)
11015   RPLOT Xx*S,Yy*S
11020  NEXT Xx
11025  RPLOT -16*S,0,2
11030  MOVE M1+2,M2+2
11035  FOR Xx=9 TO -9 STEP -.5
11040   Yy=-SIN(Xx*20)
11045   RPLOT Xx*S,(Yy*5)*S
11050  NEXT Xx
11055  RPLOT -9*S,Yy*S,2
11060  RPLOT 0,-16*S,1
11065  RPLOT 0,-Pz,2
11070  RPLOT -16*S,20*S,1
11075  RPLOT -16*S,12*S,2
11080  RPLOT -20*S,16*S,1
11085  RPLOT -12*S,16*S,2
11090  RETURN
11095    !
11100 Resistor:    !
11105  RPLOT 0,Pz,1
11110  RPLOT 0,16*S
11115  RPLOT -8*S,14*S
11120  RPLOT 8*S,10*S
11125  RPLOT -8*S,6*S
11130  RPLOT 8*S,2*S
11135  RPLOT -8*S,-2*S
11140  RPLOT 8*S,-6*S
11145  RPLOT -8*S,-10*S
11150  RPLOT 8*S,-14*S
11155  RPLOT 0,-16*S
11160  RPLOT 0,-Pz,2
11165  RETURN
11170  !
11175 Inductor:    !
11180  RPLOT 0,Pz,1
11185  RPLOT 0,16*S
11190  RPLOT -4*S,16*S
11195  FOR Yy=16 TO 8 STEP -.5
11200   Xx=-SQR(16-(Yy-12)^2)-4
11205   RPLOT Xx*S,Yy*S
11210  NEXT Yy
11215  RPLOT 0,8*S,2
11220  RPLOT -4*S,8*S,1
11225  FOR Yy=8 TO 0 STEP -.5
11230   Xx=-SQR(16-(Yy-4)^2)-4
11235   RPLOT Xx*S,Yy*S
11240  NEXT Yy
11245  RPLOT 0,0,2
11250  RPLOT -4*S,0,1
11255  FOR Yy=0 TO -8 STEP -.5
```

```
11260   Xx=-SQR(16-(Yy+4)^2)-4
11265   RPLOT Xx*S,Yy*S
11270  NEXT Yy
11275  RPLOT 0,-8*S,2
11280  RPLOT -4*S,-8*S,1
11285  FOR Yy=-8 TO -16 STEP -.5
11290   Xx=-SQR(16-(Yy+12)^2)-4
11295   RPLOT Xx*S,Yy*S
11300  NEXT Yy
11305  RPLOT 0,-16*S
11310  RPLOT 0,-Pz,2
11315  RETURN
11320    !
11325 Choke:   !
11330  RPLOT 0,Pz,1
11335  RPLOT 0,16*S
11340  RPLOT -4*S,16*S
11345  FOR Yy=16 TO 8 STEP -.5
11350   Xx=-SQR(16-(Yy-12)^2)-4
11355   RPLOT Xx*S,Yy*S
11360  NEXT Yy
11365  RPLOT 0,8*S,2
11370  RPLOT -4*S,8*S,1
11375  FOR Yy=8 TO 0 STEP -.5
11380   Xx=-SQR(16-(Yy-4)^2)-4
11385   RPLOT Xx*S,Yy*S
11390  NEXT Yy
11395  RPLOT 0,0,2
11400  RPLOT -4*S,0,1
11405  FOR Yy=0 TO -8 STEP -.5
11410   Xx=-SQR(16-(Yy+4)^2)-4
11415   RPLOT Xx*S,Yy*S
11420  NEXT Yy
11425  RPLOT 0,-8*S,2
11430  RPLOT -4*S,-8*S,1
11435  FOR Yy=-8 TO -16 STEP -.5
11440   Xx=-SQR(16-(Yy+12)^2)-4
11445   RPLOT Xx*S,Yy*S
11450  NEXT Yy
11455  RPLOT 0,-16*S
11460  RPLOT 0,-Pz,2
11465  RPLOT -10*S,16*S,1
11470  RPLOT -10*S,-16*S,2
11475  RPLOT -14*S,16*S,1
11480  RPLOT -14*S,-16*S,2
11485  RETURN
11490  !
11495 Capacitor:    !
11500  RPLOT 0,Pz,1
11505  RPLOT 0,3.2*S,2
11510  RPLOT -8*S,3.2*S,1
11515  RPLOT 8*S,3.2*S,2
11520  RPLOT -8*S,-3.2*S,1
```

```
11525  RPLOT 8*S,-3.2*S,2
11530  RPLOT 0,-3.2*S,1
11535  RPLOT 0,-Pz,2
11540  RETURN
11545  !
11550 Electrolytic:!
11555  RPLOT 0,Pz,1
11560  RPLOT 0,3.2*S,2
11565  RPLOT -8*S,3.2*S,1
11570  RPLOT 8*S,3.2*S,2
11575  Xx=-8
11580  Yy=SQR(256-Xx^2)-19.2
11585  RPLOT Xx*S,Yy*S
11590  FOR Xx=-7.6 TO 7.6 STEP .4
11595   Yy=SQR(256-Xx^2)-19.2
11600   RPLOT Xx*S,Yy*S
11605  NEXT Xx
11610  Xx=8
11615  Yy=SQR(256-Xx^2)-19.2
11620  RPLOT Xx*S,Yy*S,2
11625  RPLOT 0,-3.2*S,1
11630  RPLOT 0,-Pz,2
11635  RETURN
11640 Err1:ALPHA OFF
11645  BEEP
11650  RETURN
11655      !
11660 Half_bridge_in:!
11665  OFF ERROR
11670   !        ********** DC_source ***********************
11675  Sh=-96
11680  RPLOT Sh*S,Pz,1
11685  RPLOT Sh*S,3.5*S,2
11690  RPLOT (-12+Sh)*S,3.5*S,1
11695  RPLOT (12+Sh)*S,3.5*S,2
11700  RPLOT (-6+Sh)*S,-3.5*S,1
11705  RPLOT (6+Sh)*S,-3.5*S,2
11710  RPLOT Sh*S,-3.5*S,1
11715  RPLOT Sh*S,-Pz,2
11720  RPLOT (-8+Sh)*S,10*S,1
11725  RPLOT (-4+Sh)*S,10*S,2
11730  RPLOT (-6+Sh)*S,12*S,1
11735  RPLOT (-6+Sh)*S,8*S,2
11740  RPLOT (-8+Sh)*S,-10*S,1
11745  RPLOT (-4+Sh)*S,-10*S,2
11750  Sh=-64
11755  Shy=(Pz/2)/S
11760  Top=Pz
11765  Bottom=0
11770  !        ********************** Electrolytic ***************
11775  Elel=Elel+1
11780  RPLOT Sh*S,Top,1
11785  RPLOT Sh*S,(3.2+Shy)*S,2
```

```
11790  RPLOT (-8+Sh)*S,(3.2+Shy)*S,1
11795  RPLOT (8+Sh)*S,((3.2+Shy)*S),2
11800  Xx=(-8)
11805  Yy=SQR(256-Xx^2)-19.2
11810  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
11815  FOR Xx=-7.6 TO 7.6 STEP .4
11820   Yy=SQR(256-Xx^2)-19.2
11825   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
11830  NEXT Xx
11835  Xx=8
11840  Yy=SQR(256-Xx^2)-19.2
11845  RPLOT (Xx+Sh)*S,(Yy+Shy)*S,2
11850  RPLOT Sh*S,(-3.2+Shy)*S,1
11855  RPLOT Sh*S,Bottom,2
11860  IF Elel=2 THEN
11865   Elel=0
11870   GOTO 11910
11875  END IF
11880  !      ***********************************************
11885  Sh=-64
11890  Shy=(-Pz/2)/S
11895  Top=0
11900  Bottom=-Pz
11905  GOTO 11770
11910  !
11915  Sh=-32
11920  Shy=(Pz/2)/S
11925  Top=Pz
11930  Bottom=0
11935  !          ***************       SWITCH ***************
11940  Elel=Elel+1
11945  RPLOT Sh*S,Top,1
11950  RPLOT Sh*S,(Shy+13)*S
11955  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
11960  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
11965  FOR Xx=-1 TO 1 STEP .2
11970   Yy=SQR(1-Xx^2)-11
11975   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
11980  NEXT Xx
11985  FOR Xx=1 TO -1 STEP -.2
11990   Yy=-SQR(1-Xx^2)-11
11995   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
12000  NEXT Xx
12005  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
12010  RPLOT Sh*S,(-12+Shy)*S,1
12015  RPLOT Sh*S,Bottom,2
12020  RPLOT (16+Sh)*S,(-5+Shy)*S,1
12025  FOR Xx=+16 TO +14 STEP -.2
12030   Yy=SQR(1-(Xx-15)^2)-5
12035   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
12040  NEXT Xx
12045  FOR Xx=+14 TO +16 STEP .2
12050   Yy=-SQR(1-(Xx-15)^2)-5
```

```
12055   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
12060  NEXT Xx
12065  RPLOT (16+Sh)*S,(-5+Shy)*S,2
12070  IF Elel=2 THEN
12075   Elel=0
12080   GOTO 12115
12085  END IF
12090  Sh=-32
12095  Shy=(-Pz/2)/S
12100  Top=0
12105  Bottom=-Pz
12110  GOTO 11935
12115 ! ************     DOT ON THE RIGHT ************
12120  RPLOT -32*S,8*S,1
12125  FOR Xx=-33 TO -31 STEP .2
12130   Yy=SQR(1-(Xx+32)^2)+8
12135   RPLOT Xx*S,Yy*S
12140  NEXT Xx
12145  FOR Xx=-31 TO -33 STEP -.2
12150   Yy=-SQR(1-(Xx+32)^2)+8
12155   RPLOT Xx*S,Yy*S
12160  NEXT Xx
12165  RPLOT -32*S,8*S,2
12170 !   ************************************
12175 !   ****************** DOT ON THE LEFT ***************
12180  RPLOT -64*S,-8*S,1
12185  FOR Xx=-65 TO -63 STEP .2
12190   Yy=SQR(1-(Xx+64)^2)-8
12195   RPLOT Xx*S,Yy*S
12200  NEXT Xx
12205  FOR Xx=-63 TO -65 STEP -.2
12210   Yy=-SQR(1-(Xx+64)^2)-8
12215   RPLOT Xx*S,Yy*S
12220  NEXT Xx
12225  RPLOT -64*S,-8*S,2
12230 !           **********************************
12235 !
12240  RPLOT -96*S,Pz,1     ! TOP LINE
12245  RPLOT -32*S,Pz,2
12250  RPLOT -96*S,-Pz,1    ! BOTTOM LINE
12255  RPLOT -32*S,-Pz,2
12260 !
12265  RPLOT -32*S,8*S,1    ! TOP TERMINAL LINE
12270  RPLOT 0,8*S
12275  RPLOT 0,Pz,2
12280  RPLOT -64*S,-8*S,1   ! BOTTOM  TERMINAL LINE
12285  RPLOT 0,-8*S
12290  RPLOT 0,-Pz,2
12295  RETURN
12300 !
12305                !
12310 Full_bridge_lin:              !
12315  OFF ERROR
```

```
12320  !        ********* DC_source **********************
12325  Sh=-96
12330  RPLOT Sh*S,Pz,1
12335  RPLOT Sh*S,3.5*S,2
12340  RPLOT (-12+Sh)*S,3.5*S,1
12345  RPLOT (12+Sh)*S,3.5*S,2
12350  RPLOT (-6+Sh)*S,-3.5*S,1
12355  RPLOT (6+Sh)*S,-3.5*S,2
12360  RPLOT Sh*S,-3.5*S,1
12365  RPLOT Sh*S,-Pz,2
12370  RPLOT (-8+Sh)*S,10*S,1
12375  RPLOT (-4+Sh)*S,10*S,2
12380  RPLOT (-6+Sh)*S,12*S,1
12385  RPLOT (-6+Sh)*S,8*S,2
12390  RPLOT (-8+Sh)*S,-10*S,1
12395  RPLOT (-4+Sh)*S,-10*S,2
12400  !
12405  Sh=-64
12410  Shy=(Pz/2)/S
12415  Top=Pz
12420  Bottom=0
12425  !          **************      SWITCH **************
12430  Elel=Elel+1
12435  RPLOT Sh*S,Top,1
12440  RPLOT Sh*S,(Shy+13)*S
12445  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
12450  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
12455  FOR Xx=-1 TO 1 STEP .2
12460   Yy=SQR(1-Xx^2)-11
12465   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
12470  NEXT Xx
12475  FOR Xx=1 TO -1 STEP -.2
12480   Yy=-SQR(1-Xx^2)-11
12485   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
12490  NEXT Xx
12495  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
12500  RPLOT Sh*S,(-12+Shy)*S,1
12505  RPLOT Sh*S,Bottom,2
12510  RPLOT (16+Sh)*S,(-5+Shy)*S,1
12515  FOR Xx=+16 TO +14 STEP -.2
12520   Yy=SQR(1-(Xx-15)^2)-5
12525   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
12530  NEXT Xx
12535  FOR Xx=+14 TO +16 STEP .2
12540   Yy=-SQR(1-(Xx-15)^2)-5
12545   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
12550  NEXT Xx
12555  RPLOT (16+Sh)*S,(-5+Shy)*S,2
12560  IF Elel=1 THEN 12595
12565  IF Elel=2 THEN 12620
12570  IF Elel=3 THEN 12645
12575  IF Elel=4 THEN
12580   Elel=0
```

```
12585   GOTO 12675
12590  END IF
12595  Sh=-64
12600  Shy=(-Pz/2)/S
12605  Top=0
12610  Bottom=-Pz
12615  GOTO 12425
12620  Sh=-32
12625  Shy=(Pz/2)/S
12630  Top=Pz
12635  Bottom=0
12640  GOTO 12425!
12645  Sh=-32
12650  Shy=(-Pz/2)/S
12655  Top=0
12660  Bottom=-Pz
12665  GOTO 12425!
12670 ! ************     DOT ON THE RIGHT ************
12675  RPLOT -32*S,8*S,1
12680  FOR Xx=-33 TO -31 STEP .2
12685    Yy=SQR(1-(Xx+32)^2)+8
12690    RPLOT Xx*S,Yy*S
12695  NEXT Xx
12700  FOR Xx=-31 TO -33 STEP -.2
12705    Yy=-SQR(1-(Xx+32)^2)+8
12710    RPLOT Xx*S,Yy*S
12715  NEXT Xx
12720  RPLOT -32*S,8*S,2
12725 !    ************************************
12730 !    ******************** DOT ON THE LEFT ***************
12735  RPLOT -64*S,-8*S,1
12740  FOR Xx=-65 TO -63 STEP .2
12745    Yy=SQR(1-(Xx+64)^2)-8
12750    RPLOT Xx*S,Yy*S
12755  NEXT Xx
12760  FOR Xx=-63 TO -65 STEP -.2
12765    Yy=-SQR(1-(Xx+64)^2)-8
12770    RPLOT Xx*S,Yy*S
12775  NEXT Xx
12780  RPLOT -64*S,-8*S,2
12785 !              *********************************
12790 !
12795  RPLOT -96*S,Pz,1     ! TOP LINE
12800  RPLOT -32*S,Pz,2
12805  RPLOT -96*S,-Pz,1    ! BOTTOM LINE
12810  RPLOT -32*S,-Pz,2
12815 !
12820  RPLOT -32*S,8*S,1    ! TOP TERMINAL LINE
12825  RPLOT 0,8*S
12830  RPLOT 0,Pz,2
12835  RPLOT -64*S,-8*S,1   ! BOTTOM  TERMINAL LINE
12840  RPLOT 0,-8*S
12845  RPLOT 0,-Pz,2
```

```
12850  RETURN
12855 Full_bridge_3in:                !
12860  OFF ERROR
12865   !        ********* DC_source  **********************
12870  Sh=-128
12875  RPLOT Sh*S,Pz,1
12880  RPLOT Sh*S,3.5*S,2
12885  RPLOT (-12+Sh)*S,3.5*S,1
12890  RPLOT (12+Sh)*S,3.5*S,2
12895  RPLOT (-6+Sh)*S,-3.5*S,1
12900  RPLOT (6+Sh)*S,-3.5*S,2
12905  RPLOT Sh*S,-3.5*S,1
12910  RPLOT Sh*S,-Pz,2
12915  RPLOT (-8+Sh)*S,10*S,1
12920  RPLOT (-4+Sh)*S,10*S,2
12925  RPLOT (-6+Sh)*S,12*S,1
12930  RPLOT (-6+Sh)*S,8*S,2
12935  RPLOT (-8+Sh)*S,-10*S,1
12940  RPLOT (-4+Sh)*S,-10*S,2
12945 !
12950  Sh=-96
12955  Shy=(Pz/2)/S
12960  Top=Pz
12965  Bottom=0
12970 !          **************       SWITCH **************
12975  Elel=Elel+1
12980  RPLOT Sh*S,Top,1
12985  RPLOT Sh*S,(Shy+13)*S
12990  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
12995  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
13000  FOR Xx=-1 TO 1 STEP .2
13005   Yy=SQR(1-Xx^2)-11
13010   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13015  NEXT Xx
13020  FOR Xx=1 TO -1 STEP -.2
13025   Yy=-SQR(1-Xx^2)-11
13030   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13035  NEXT Xx
13040  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
13045  RPLOT Sh*S,(-12+Shy)*S,1
13050  RPLOT Sh*S,Bottom,2
13055  RPLOT (16+Sh)*S,(-5+Shy)*S,1
13060  FOR Xx=+16 TO +14 STEP -.2
13065   Yy=SQR(1-(Xx-15)^2)-5
13070   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13075  NEXT Xx
13080  FOR Xx=+14 TO +16 STEP .2
13085   Yy=-SQR(1-(Xx-15)^2)-5
13090   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13095  NEXT Xx
13100  RPLOT (16+Sh)*S,(-5+Shy)*S,2
13105  IF Elel=1 THEN 13150
13110  IF Elel=2 THEN 13175
```

```
13115  IF Elel=3 THEN 13200
13120  IF Elel=4 THEN 13225
13125  IF Elel=5 THEN 13250
13130  IF Elel=6 THEN
13135   Elel=0
13140   GOTO 13280
13145  END IF
13150  Sh=-96
13155  Shy=(-Pz/2)/S
13160  Top=0
13165  Bottom=-Pz
13170  GOTO 12970
13175  Sh=-64
13180  Shy=(Pz/2)/S
13185  Top=Pz
13190  Bottom=0
13195  GOTO 12970
13200  Sh=-64
13205  Shy=(-Pz/2)/S
13210  Top=0
13215  Bottom=-Pz
13220  GOTO 12970
13225  Sh=-32
13230  Shy=(Pz/2)/S
13235  Top=Pz
13240  Bottom=0
13245  GOTO 12970!
13250  Sh=-32
13255  Shy=(-Pz/2)/S
13260  Top=0
13265  Bottom=-Pz
13270  GOTO 12970!
13275  ! ************     DOT ON THE LEFT ************
13280  RPLOT -96*S,10*S,1
13285  FOR Xx=-97 TO -95 STEP .2
13290   Yy=SQR(1-(Xx+96)^2)+10
13295   RPLOT Xx*S,Yy*S
13300  NEXT Xx
13305  FOR Xx=-95 TO -97 STEP -.2
13310   Yy=-SQR(1-(Xx+96)^2)+10
13315   RPLOT Xx*S,Yy*S
13320  NEXT Xx
13325  RPLOT -96*S,10*S,2
13330  !   ************************************
13335  !   ******************** DOT IN THE MIDDLE  **************
13340  RPLOT -64*S,0,1
13345  FOR Xx=-65 TO -63 STEP .2
13350   Yy=SQR(1-(Xx+64)^2)
13355   RPLOT Xx*S,Yy*S
13360  NEXT Xx
13365  FOR Xx=-63 TO -65 STEP -.2
13370   Yy=-SQR(1-(Xx+64)^2)
13375   RPLOT Xx*S,Yy*S
```

```
13380  NEXT Xx
13385  RPLOT -64*S,0,2
13390 !           **********************************
13395 !
13400 !   ******************** DOT ON THE BOTTOM  ***************
13405  RPLOT -32*S,-10,1
13410  FOR Xx=-33 TO -31 STEP .2
13415   Yy=SQR(1-(Xx+32)^2)-10
13420   RPLOT Xx*S,Yy*S
13425  NEXT Xx
13430  FOR Xx=-31 TO -33 STEP -.2
13435   Yy=-SQR(1-(Xx+32)^2)-10
13440   RPLOT Xx*S,Yy*S
13445  NEXT Xx
13450  RPLOT -32*S,-10,2
13455 !           **********************************
13460 !
13465  RPLOT -128*S,Pz,1    ! TOP LINE
13470  RPLOT -32*S,Pz,2
13475  RPLOT -128*S,-Pz,1   ! BOTTOM LINE
13480  RPLOT -32*S,-Pz,2
13485 !
13490  RPLOT -96*S,10*S,1   ! TOP TERMINAL LINE
13495  RPLOT 0,10*S
13500  RPLOT 0,Pz,2
13505  RPLOT -64*S,0,1      ! MIDDLE TERMINAL LINE
13510  RPLOT 0,0;2
13515  RPLOT -32*S,-10*S,1  ! BOTTOM  TERMINAL LINE
13520  RPLOT 0,-10*S
13525  RPLOT 0,-Pz,2
13530  RETURN
13535 Phase1_rect:                          !
13540  OFF ERROR
13545   !        ********* AC Source  **********************
13550  Sh=-102
13555  Shy=0
13560  Top=Pz
13565  Bottom=-Pz
13570  RPLOT Sh*S,Top/2,1
13575  RPLOT Sh*S,(16+Shy)*S,2
13580  RPLOT (-16+Sh)*S,Shy*S,1
13585  FOR Xx=-16 TO 16 STEP .5
13590   Yy=SQR(256-Xx^2)
13595   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13600  NEXT Xx
13605  FOR Xx=16 TO -16 STEP -.5
13610   Yy=-SQR(256-Xx^2)
13615   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13620  NEXT Xx
13625  RPLOT (-16+Sh)*S,(Shy*S),2
13630  RPLOT (Sh*S),(-16+Shy)*S,1
13635  RPLOT Sh*S,Bottom/2,2
13640  RPLOT (-16+Sh)*S,(20+Shy)*S,1
```

```
13645  RPLOT (-16+Sh)*S,(12+Shy)*S,2
13650  RPLOT (-20+Sh)*S,(16+Shy)*S,1
13655  RPLOT (-12+Sh)*S,(16+Shy)*S,2
13660  !       ****************************************
13665  Sh=-64
13670  Shy=(Pz/2)/S
13675  Top=Pz
13680  Bottom=0
13685  !       **************      SWITCH **************
13690  Elel=Elel+1
13695  RPLOT Sh*S,Top,1
13700  RPLOT Sh*S,(Shy+13)*S
13705  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
13710  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
13715  FOR Xx=-1 TO 1 STEP .2
13720   Yy=SQR(1-Xx^2)-11
13725   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13730  NEXT Xx
13735  FOR Xx=1 TO -1 STEP -.2
13740   Yy=-SQR(1-Xx^2)-11
13745   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13750  NEXT Xx
13755  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
13760  RPLOT Sh*S,(-12+Shy)*S,1
13765  RPLOT Sh*S,Bottom,2
13770  RPLOT (16+Sh)*S,(-5+Shy)*S,1
13775  FOR Xx=+16 TO +14 STEP -.2
13780   Yy=SQR(1-(Xx-15)^2)-5
13785   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13790  NEXT Xx
13795  FOR Xx=+14 TO +16 STEP .2
13800   Yy=-SQR(1-(Xx-15)^2)-5
13805   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
13810  NEXT Xx
13815  RPLOT (16+Sh)*S,(-5+Shy)*S,2
13820  IF Elel=1 THEN 13855
13825  IF Elel=2 THEN 13880
13830  IF Elel=3 THEN 13905
13835  IF Elel=4 THEN
13840   Elel=0
13845   GOTO 13935
13850  END IF
13855  Sh=-64
13860  Shy=(-Pz/2)/S
13865  Top=0
13870  Bottom=-Pz
13875  GOTO 13685
13880  Sh=-32
13885  Shy=(Pz/2)/S
13890  Top=Pz
13895  Bottom=0
13900  GOTO 13685!
13905  Sh=-32
```

```
13910  Shy=(-Pz/2)/S
13915  Top=0
13920  Bottom=-Pz
13925  GOTO 13685!
13930 ! ************     DOT ON THE RIGHT************
13935  RPLOT -32*S,-10*S,1
13940  FOR Xx=-33 TO -31 STEP .2
13945   Yy=SQR(1-(Xx+32)^2)-10
13950   RPLOT Xx*S,Yy*S
13955  NEXT Xx
13960  FOR Xx=-31 TO -33 STEP -.2
13965   Yy=-SQR(1-(Xx+32)^2)-10
13970   RPLOT Xx*S,Yy*S
13975  NEXT Xx
13980  RPLOT -32*S,-10*S,2
13985 !    ************************************
13990 !    ******************** DOT ON THE LEFT**************
13995  RPLOT -64*S,10*S,1
14000  FOR Xx=-65 TO -63 STEP .2
14005   Yy=SQR(1-(Xx+64)^2)+10
14010   RPLOT Xx*S,Yy*S
14015  NEXT Xx
14020  FOR Xx=-63 TO -65 STEP -.2
14025   Yy=-SQR(1-(Xx+64)^2)+10
14030   RPLOT Xx*S,Yy*S
14035  NEXT Xx
14040  RPLOT -64*S,10*S,2
14045 !             **********************************
14050 !
14055  RPLOT -102*S,(Pz/2),1       ! TOP LINE
14060  RPLOT -78*S,(Pz/2)
14065  RPLOT -78*S,10*S
14070  RPLOT -64*S,10*S,2
14075  RPLOT -102*S,-Pz/2,1   ! BOTTOM LINE
14080  RPLOT -78*S,-Pz/2
14085  RPLOT -78*S,-10*S
14090  RPLOT -32*S,-10*S,2
14095  RPLOT -64*S,Pz,1     ! TOP TERMINAL LINE
14100  RPLOT 0,Pz,2
14105  RPLOT -64*S,-Pz,1   ! BOTTOM TERMINAL LINE
14110  RPLOT 0,-Pz,2
14115  RETURN
14120 Phase3_rect_d:         !
14125  OFF ERROR
14130   !        *****************************************
14135  Xd1=-3*Pz/2+(-152+16/SQR(2))*S
14140  Xd2=-3*Pz/2+(-152-16/SQR(2))*S
14145   !        ************* PHASE C OF THE AC SOURCE **************
14150  RPLOT (-152*S),-Pz,Pz,1
14155  RPLOT Xd1,Pz/2+16/SQR(2)*S,2
14160  RPLOT -3*Pz/2-(152+16)*S,Pz/2,1
14165  FOR Xx=-16 TO 16 STEP .5
14170   Yy=SQR(256-Xx^2)
```

```
14175  RPLOT -3*Pz/2+(Xx-152)*S,Yy*S+Pz/2
14180  NEXT Xx
14185  FOR Xx=16 TO -16 STEP -.5
14190   Yy=-SQR(256-Xx^2)
14195   RPLOT -3*Pz/2+(Xx-152)*S,Yy*S+Pz/2
14200  NEXT Xx
14205  RPLOT -3*Pz/2-(152+16)*S,Pz/2,2
14210  RPLOT Xd2,Pz/2-16/SQR(2)*S,1
14215  RPLOT -2*Pz-152*S,0,2
14220  RPLOT (-3*Pz/2-152*S)-4/SQR(2)*S,Pz/2-(SQR(2)*16+4/SQR(2))*S,1
14225  RPLOT (-3*Pz/2-152*S)+4/SQR(2)*S,Pz/2-(SQR(2)*16-4/SQR(2))*S,2
14230  RPLOT (-3*Pz/2-152*S)-4/SQR(2)*S,Pz/2-(SQR(2)*16-4/SQR(2))*S,1
14235  RPLOT (-3*Pz/2-152*S)+4/SQR(2)*S,Pz/2-(SQR(2)*16+4/SQR(2))*S,2
14240  !        ******************************************
14245   !       ************** PHASE B ,AC SOURCE **************
14250  RPLOT (-152*S-2*Pz),0,1
14255  RPLOT ((-152-16)*S-Pz),0,2
14260  RPLOT ((-152-16)*S-Pz),0,1
14265  FOR Xx=-16 TO 16 STEP .5
14270   Yy=SQR(256-Xx^2)
14275   RPLOT -Pz+(Xx-152)*S,Yy*S
14280  NEXT Xx
14285  FOR Xx=16 TO -16 STEP -.5
14290   Yy=-SQR(256-Xx^2)
14295   RPLOT -Pz+(Xx-152)*S,Yy*S
14300  NEXT Xx
14305  RPLOT ((-152-16)*S-Pz),0,2
14310  RPLOT ((-152+16)*S-Pz),0,1
14315  RPLOT -152*S,0,2
14320  RPLOT -152*S-Pz+12*S,16*S,1
14325  RPLOT -152*S-Pz+20*S,16*S,2
14330  RPLOT -152*S-Pz+16*S,20*S,1
14335  RPLOT -152*S-Pz+16*S,12*S,2
14340  !        ******************************************
14345  Xd1=-Pz/2+(-152-16/SQR(2))*S
14350  Xd2=-Pz/2+(-152+16/SQR(2))*S
14355   !       ************** PHASE A ,AC SOURCE **************
14360  RPLOT (-152*S)-Pz,Pz,1
14365  RPLOT Xd1,Pz/2+16/SQR(2)*S,2
14370  RPLOT -Pz/2-(152+16)*S,Pz/2,1
14375  FOR Xx=-16 TO 16 STEP .5
14380   Yy=SQR(256-Xx^2)
14385   RPLOT -Pz/2+(Xx-152)*S,Yy*S+Pz/2
14390  NEXT Xx
14395  FOR Xx=16 TO -16 STEP -.5
14400   Yy=-SQR(256-Xx^2)
14405   RPLOT -Pz/2+(Xx-152)*S,Yy*S+Pz/2
14410  NEXT Xx
14415  RPLOT -Pz/2-(152+16)*S,Pz/2,2
14420  RPLOT Xd2,Pz/2-16/SQR(2)*S,1
14425  RPLOT -152*S,0,2
14430  RPLOT (-Pz/2+(-152-(SQR(2)*16-4/SQR(2)))*S)-4/SQR(2)*S,Pz/2+4/SQR(2)*S,1
14435  RPLOT (-Pz/2+(-152-(SQR(2)*16+4/SQR(2)))*S)-4/SQR(2)*S,Pz/2-4/SQR(2)*S,2
```

```
14440  RPLOT (-Pz/2+(-152-(SQR(2)*16-4/SQR(2)))*S)-4/SQR(2)*S,Pz/2-4/SQR(2)*S,1
14445  RPLOT (-Pz/2+(-152-(SQR(2)*16+4/SQR(2)))*S)-4/SQR(2)*S,Pz/2+4/SQR(2)*S,2
14450 !        ******************************************
14455  Sh=-96
14460  Shy=(Pz/2)/S
14465  Top=Pz
14470  Bottom=0
14475 !        ************** SWITCH **************
14480  Elel=Elel+1
14485  RPLOT Sh*S,Top,1
14490  RPLOT Sh*S,(Shy+13)*S
14495  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
14500  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
14505  FOR Xx=-1 TO 1 STEP .2
14510   Yy=SQR(1-Xx^2)-11
14515   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
14520  NEXT Xx
14525  FOR Xx=1 TO -1 STEP -.2
14530   Yy=-SQR(1-Xx^2)-11
14535   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
14540  NEXT Xx
14545  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
14550  RPLOT Sh*S,(-12+Shy)*S,1
14555  RPLOT Sh*S,Bottom,2
14560  RPLOT (16+Sh)*S,(-5+Shy)*S,1
14565  FOR Xx=+16 TO +14 STEP -.2
14570   Yy=SQR(1-(Xx-15)^2)-5
14575   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
14580  NEXT Xx
14585  FOR Xx=+14 TO +16 STEP .2
14590   Yy=-SQR(1-(Xx-15)^2)-5
14595   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
14600  NEXT Xx
14605  RPLOT (16+Sh)*S,(-5+Shy)*S,2
14610  IF Elel=1 THEN 14655
14615  IF Elel=2 THEN 14680
14620  IF Elel=3 THEN 14705
14625  IF Elel=4 THEN 14730
14630  IF Elel=5 THEN 14755
14635  IF Elel=6 THEN
14640   Elel=0
14645   GOTO 14785
14650  END IF
14655  Sh=-96
14660  Shy=(-Pz/2)/S
14665  Top=0
14670  Bottom=-Pz
14675  GOTO 14475
14680  Sh=-64
14685  Shy=(Pz/2)/S
14690  Top=Pz
14695  Bottom=0
14700  GOTO 14475
```

```
14705  Sh=-64
14710  Shy=(-Pz/2)/S
14715  Top=0
14720  Bottom=-Pz
14725  GOTO 14475
14730  Sh=-32
14735  Shy=(Pz/2)/S
14740  Top=Pz
14745  Bottom=0
14750  GOTO 14475!
14755  Sh=-32
14760  Shy=(-Pz/2)/S
14765  Top=0
14770  Bottom=-Pz
14775  GOTO 14475!
14780 ! ************     DOT ON THE LEFT ************
14785  RPLOT -96*S,10*S,1
14790  FOR Xx=-97 TO -95 STEP .2
14795   Yy=SQR(1-(Xx+96)^2)+10
14800   RPLOT Xx*S,Yy*S
14805  NEXT Xx
14810  FOR Xx=-95 TO -97 STEP -.2
14815   Yy=-SQR(1-(Xx+96)^2)+10
14820   RPLOT Xx*S,Yy*S
14825  NEXT Xx
14830  RPLOT -96*S,10*S,2
14835 !   ************************************
14840 !   ******************** DOT IN THE MIDDLE  **************
14845  RPLOT -64*S,0,1
14850  FOR Xx=-65 TO -63 STEP .2
14855   Yy=SQR(1-(Xx+64)^2)
14860   RPLOT Xx*S,Yy*S
14865  NEXT Xx
14870  FOR Xx=-63 TO -65 STEP -.2
14875   Yy=-SQR(1-(Xx+64)^2)
14880   RPLOT Xx*S,Yy*S
14885  NEXT Xx
14890  RPLOT -64*S,0,2
14895 !           **********************************
14900 !
14905 !   ******************** DOT ON THE BOTTOM  **************
14910  RPLOT -32*S,-10,1
14915  FOR Xx=-33 TO -31 STEP .2
14920   Yy=SQR(1-(Xx+32)^2)-10
14925   RPLOT Xx*S,Yy*S
14930  NEXT Xx
14935  FOR Xx=-31 TO -33 STEP -.2
14940   Yy=-SQR(1-(Xx+32)^2)-10
14945   RPLOT Xx*S,Yy*S
14950  NEXT Xx
14955  RPLOT -32*S,-10,2
14960 !           **********************************
14965 !
```

```
14970  RPLOT -96*S,Pz,1    ! TOP LINE
14975  RPLOT 0,Pz,2
14980  RPLOT -96*S,-Pz,1   ! BOTTOM LINE
14985  RPLOT 0,-Pz,2
14990  !
14995  RPLOT (-152*S)-Pz,Pz,1   ! TOP TERMINAL LINE
15000  RPLOT -128*S,Pz
15005  RPLOT -128*S,10*S
15010  RPLOT -96*S,10*S,2
15015  RPLOT -152*S,0,1     ! MIDDLE TERMINAL LINE
15020  RPLOT -64*S,0,2
15025  RPLOT (-152*S)-2*Pz,0,1  ! BOTTOM  TERMINAL LINE
15030  RPLOT (-152*S)-2*Pz,-Pz
15035  RPLOT -128*S,-Pz
15040  RPLOT -128*S,-10*S
15045  RPLOT -32*S,-10*S,2
15050  RETURN
15055 Phase3 rect_y:                    !
15060  OFF ERROR
15065   !       ****************************************
15070  Xd1=(-152-32)*S-Pz-16*S/SQR(2)
15075  Xd2=(-152-32)*S-Pz+16*S/SQR(2)
15080   !       ************* PHASE A ,AC SOURCE **************
15085  RPLOT -Pz+(-152-64)*S,64*S,1
15090  RPLOT Xd1,(32+16/SQR(2))*S,2
15095  RPLOT (-152-48)*S-Pz,32*S,1
15100  FOR Xx=-16 TO 16 STEP .5
15105   Yy=SQR(256-Xx^2)
15110   RPLOT -Pz+(Xx-152-32)*S,(Yy+32)*S
15115  NEXT Xx
15120  FOR Xx=16 TO -16 STEP -.5
15125   Yy=-SQR(256-Xx^2)
15130   RPLOT -Pz+(Xx-152-32)*S,(Yy+32)*S
15135  NEXT Xx
15140  RPLOT (-152-48)*S-Pz,32*S,2
15145  RPLOT Xd2,(32-16/SQR(2))*S,1
15150  RPLOT -152*S-Pz,0,2
15155  RPLOT (-152-32-32/SQR(2)-4)*S-Pz,(32+4)*S,1
15160  RPLOT (-152-32-32/SQR(2)+4)*S-Pz,(32-4)*S,2
15165  RPLOT (-152-32-32/SQR(2)-4)*S-Pz,(32-4)*S,1
15170  RPLOT (-152-32-32/SQR(2)+4)*S-Pz,(32+4)*S,2
15175  !       ****************************************
15180  Xd1=(-152+32)*S-Pz+16*S/SQR(2)
15185  Xd2=(-152+32)*S-Pz-16*S/SQR(2)
15190   !       ************* PHASE B ,AC SOURCE **************
15195  RPLOT -Pz+(-152+64)*S,64*S,1
15200  RPLOT Xd1,(32+16/SQR(2))*S,2
15205  RPLOT (-152+32-16)*S-Pz,32*S,1
15210  FOR Xx=-16 TO 16 STEP .5
15215   Yy=SQR(256-Xx^2)
15220   RPLOT (-152+32+Xx)*S-Pz,(32+Yy)*S
15225  NEXT Xx
15230  FOR Xx=16 TO -16 STEP -.5
```

```
15235   Yy=-SQR(256-Xx^2)
15240   RPLOT (-152+32+Xx)*S-Pz,(32+Yy)*S
15245  NEXT Xx
15250  RPLOT (-152+32-16)*S-Pz,32*S,2
15255  RPLOT (-152+16)*S-Pz,32*S,2
15260  RPLOT Xd2,(32-16/SQR(2))*S,1
15265  RPLOT (-152*S)-Pz,0,2
15270  RPLOT (-152+32-4)*S-Pz,(32+32/SQR(2)+4)*S,1
15275  RPLOT (-152+32+4)*S-Pz,(32+32/SQR(2)-4)*S,2
15280  RPLOT (-152+32-4)*S-Pz,(32+32/SQR(2)-4)*S,1
15285  RPLOT (-152+32+4)*S-Pz,(32+32/SQR(2)+4)*S,2
15290 !          ******************************************
15295   !        ************* PHASE C ,AC SOURCE *********
15300  RPLOT -152*S-Pz,0,1
15305  RPLOT -152*S-Pz,-(32*SQR(2)-16)*S,2
15310  RPLOT -Pz-(152+16)*S,-(32*SQR(2))*S,1
15315  FOR Xx=-16 TO 16 STEP .5
15320   Yy=SQR(256-Xx^2)
15325   RPLOT -Pz-(152-Xx)*S,-(32*SQR(2)-Yy)*S
15330  NEXT Xx
15335  FOR Xx=16 TO -16 STEP -.5
15340   Yy=-SQR(256-Xx^2)
15345   RPLOT -Pz-(152-Xx)*S,-(32*SQR(2)-Yy)*S
15350  NEXT Xx
15355  RPLOT -Pz-(152+16)*S,-(32*SQR(2))*S,2
15360  RPLOT -Pz-(152)*S,-(32*SQR(2)+16)*S,1
15365  RPLOT -Pz-(152)*S,-Pz,2
15370  RPLOT -152*S-Pz+(16-4)*S,-(32*SQR(2)+16)*S,1
15375  RPLOT -152*S-Pz+(16+4)*S,-(32*SQR(2)+16)*S,2
15380  RPLOT -152*S-Pz+16*S,-(32*SQR(2)+16-4)*S,1
15385  RPLOT -152*S-Pz+16*S,-(32*SQR(2)+16+4)*S,2
15390 !          ************************************
15395 Eb:!
15400  Sh=-96
15405  Shy=(Pz/2)/S
15410  Top=Pz
15415  Bottom=0
15420 !          *************** SWITCH ***************
15425  Elel=Elel+1
15430  RPLOT Sh*S,Top,1
15435  RPLOT Sh*S,(Shy+13)*S
15440  RPLOT (+14.3+Sh)*S,(-4+Shy)*S,2
15445  RPLOT (-1+Sh)*S,(-11+Shy)*S,1
15450  FOR Xx=-1 TO 1 STEP .2
15455   Yy=SQR(1-Xx^2)-11
15460   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
15465  NEXT Xx
15470  FOR Xx=1 TO -1 STEP -.2
15475   Yy=-SQR(1-Xx^2)-11
15480   RPLOT (Xx+Sh)*S,(Yy+Shy)*S
15485  NEXT Xx
15490  RPLOT (-1+Sh)*S,(-11+Shy)*S,2
15495  RPLOT Sh*S,(-12+Shy)*S,1
```

```
15500 RPLOT Sh*S,Bottom,2
15505 RPLOT (16+Sh)*S,(-5+Shy)*S,1
15510 FOR Xx=+16 TO +14 STEP -.2
15515  Yy=SQR(1-(Xx-15)^2)-5
15520  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
15525 NEXT Xx
15530 FOR Xx=+14 TO +16 STEP .2
15535  Yy=-SQR(1-(Xx-15)^2)-5
15540  RPLOT (Xx+Sh)*S,(Yy+Shy)*S
15545 NEXT Xx
15550 RPLOT (16+Sh)*S,(-5+Shy)*S,2
15555 IF Elel=1 THEN 15600
15560 IF Elel=2 THEN 15625
15565 IF Elel=3 THEN 15650
15570 IF Elel=4 THEN 15675
15575 IF Elel=5 THEN 15700
15580 IF Elel=6 THEN
15585  Elel=0
15590  GOTO 15730
15595 END IF
15600 Sh=-96
15605 Shy=(-Pz/2)/S
15610 Top=0
15615 Bottom=-Pz
15620 GOTO 15420
15625 Sh=-64
15630 Shy=(Pz/2)/S
15635 Top=Pz
15640 Bottom=0
15645 GOTO 15420
15650 Sh=-64
15655 Shy=(-Pz/2)/S
15660 Top=0
15665 Bottom=-Pz
15670 GOTO 15420
15675 Sh=-32
15680 Shy=(Pz/2)/S
15685 Top=Pz
15690 Bottom=0
15695 GOTO 15420!
15700 Sh=-32
15705 Shy=(-Pz/2)/S
15710 Top=0
15715 Bottom=-Pz
15720 GOTO 15420!
15725 ! ********** DOT ON THE LEFT ***********
15730 RPLOT -96*S,10*S,1
15735 FOR Xx=-97 TO -95 STEP .2
15740  Yy=SQR(1-(Xx+96)^2)+10
15745  RPLOT Xx*S,Yy*S
15750 NEXT Xx
15755 FOR Xx=-95 TO -97 STEP -.2
15760  Yy=-SQR(1-(Xx+96)^2)+10
```

```
15765   RPLOT Xx*S,Yy*S
15770  NEXT Xx
15775  RPLOT -96*S,10*S,2
15780 !   ************************************
15785 !   ******* DOT IN THE MIDDLE  **********
15790  RPLOT -64*S,0,1
15795  FOR Xx=-65 TO -63 STEP .2
15800   Yy=SQR(1-(Xx+64)^2)
15805   RPLOT Xx*S,Yy*S
15810  NEXT Xx
15815  FOR Xx=-63 TO -65 STEP -.2
15820   Yy=-SQR(1-(Xx+64)^2)
15825   RPLOT Xx*S,Yy*S
15830  NEXT Xx
15835  RPLOT -64*S,0,2
15840 !   ************************************
15845 !   ******** DOT ON THE BOTTOM  *********
15850  RPLOT -32*S,-10,1
15855  FOR Xx=-33 TO -31 STEP .2
15860   Yy=SQR(1-(Xx+32)^2)-10
15865   RPLOT Xx*S,Yy*S
15870  NEXT Xx
15875  FOR Xx=-31 TO -33 STEP -.2
15880   Yy=-SQR(1-(Xx+32)^2)-10
15885   RPLOT Xx*S,Yy*S
15890  NEXT Xx
15895  RPLOT -32*S,-10,2
15900 !   ************************************
15905 !
15910  RPLOT -96*S,Pz,1     ! TOP LINE
15915  RPLOT 0,Pz,2
15920  RPLOT -96*S,-Pz,1    ! BOTTOM LINE
15925  RPLOT 0,-Pz,2
15930 !
15935  RPLOT (-152-64)*S-Pz,64*S,1    ! TOP TERMINAL LINE
15940  RPLOT (-152-64)*S-Pz,Pz
15945  RPLOT -128*S,Pz
15950  RPLOT -128*S,10*S
15955  RPLOT -96*S,10*S,2
15960  RPLOT (-152+64)*S-Pz,64*S,1      ! MIDDLE TERMINAL LINE
15965  RPLOT -152*S,64*S
15970  RPLOT -152*S,0
15975  RPLOT -64*S,0,2
15980  RPLOT (-152*S)-Pz,-Pz,1  ! BOTTOM  TERMINAL LINE
15985  RPLOT (-128*S),-Pz
15990  RPLOT -128*S,-10*S
15995  RPLOT -32*S,-10*S,2
16000  RETURN
16005! END OF ELEMENT ROUTINES
16010      !
16015! ****************************
16020! *  TIME DOMAIN ROUTINE     *
16025! ****************************
```

```
16030!
16035 Timedomain:     !        TIME DOMAIN ROUTINE
16040 ! OPTION BASE 1
16045  U1=TIMEDATE
16050  N=Con ! THE NUMBER  OF ELEMENTS
16055      ! THE NUMBER OF TWIGS=M (SOLVED LATER)
16060  Limit=Npoints ! NO OF POINTS IN SAMPLING INTERVAL
16065 ! Dt=.01
16070  ALLOCATE INTEGER Graph1(1:12480),B$(1:50)[10],Bd$[10]
16075 ! Values= ELEMENT PARAMETERS
16080 ! Elem_array=CODE FOR ELEMENTS
16085 ! Link=>  0=twig  1=link
16090 !
16095  Eto=0
16100  Et=0
16105  L1=0
16110  Aaa=0
16115  Bbb=0
16120  Ccc=0
16125  Ddd=0
16130 !
16135  ALLOCATE INTEGER Elem_array(1:N)
16140  FOR I=1 TO N
16145   Elem_array(I)=Element2(I)
16150  NEXT I
16155  ALLOCATE INTEGER Flags(1:N),Elementtt(1:N)
16160  ALLOCATE Elemvalue(1:N),Xppp(1:N),Yppp(1:N)
16165  ALLOCATE INTEGER Node_a_x(1:N),Node_a_y(1:N),Node_b_x(1:N),Node_b_y(1:N)
16170 !************************************
16175! CHECKING TO SEE IF THE TREE DESIGN EXISTS
16180! CIRCUIT DATA FILE 4
16185 !          TREE DESIGN
16190  IF Data4=5 THEN
16195   ALLOCATE INTEGER Link(1:N)
16200   ASSIGN @Path TO "4"&Z$[2,9]
16205   ENTER @Path;Link(*)
16210   ASSIGN @Path TO *
16215   GOTO Jump
16220  ELSE
16225   ALLOCATE INTEGER Link(1:N)
16230  END IF
16235 !
16240  FOR I=1 TO N
16245   IF Elem_array(I)=5 OR Elem_array(I)=6 THEN Link(I)=1
16250   IF Elem_array(I)=16 OR Elem_array(I)=17 THEN Link(I)=1
16255  NEXT I
16260  FOR I=1 TO N
16265   IF Elem_array(I)<>13 THEN GOTO 16480
16270   IF Link(I)=1 THEN GOTO 16480
16275   FOR J=1 TO N
16280    Xppp(J)=0
16285    Yppp(J)=0
16290   NEXT J
```

```
16295   Cnx=X1(I)
16300   Cny=Y1(I)
16305   Xppp(1)=X2(I)
16310   Yppp(1)=Y2(I)
16315   Aa=2
16320   FOR Kk=1 TO N
16325    Aa_crit=Aa
16330    FOR K=1 TO Aa_crit-1
16335     FOR Test=1 TO N
16340      IF Xppp(Test)=Cnx AND Yppp(Test)=Cny THEN
16345       BEEP
16350       Link(I)=1
16355       GOTO 16480
16360      END IF
16365     NEXT Test
16370     FOR J=1 TO N
16375      IF Xppp(K)=X1(J) AND Yppp(K)=Y1(J) AND Link(J)=0 AND J<>I THEN
16380       FOR L11=1 TO N
16385        IF Xppp(L11)=X2(J) AND Yppp(L11)=Y2(J) THEN GOTO 16465
16390       NEXT L11
16395       Xppp(Aa)=X2(J)
16400       Yppp(Aa)=Y2(J)
16405       Aa=Aa+1
16410       GOTO 16465
16415      END IF
16420      IF Xppp(K)=X2(J) AND Yppp(K)=Y2(J) AND Link(J)=0 AND J<>I THEN
16425       FOR L11=1 TO N
16430        IF Xppp(L11)=X1(J) AND Yppp(L11)=Y1(J) THEN GOTO 16465
16435       NEXT L11
16440       Xppp(Aa)=X1(J)
16445       Yppp(Aa)=Y1(J)
16450       Aa=Aa+1
16455       GOTO 16465
16460      END IF
16465     NEXT J
16470    NEXT K
16475   NEXT Kk
16480  NEXT I
16485   !                    FINDING CAPACITOR LOOPS
16490  FOR I=1 TO N
16495   IF Elem_array(I)<>14 AND Elem_array(I)<>15 THEN GOTO 16720
16500   IF Link(I)=1 THEN GOTO 16720
16505   FOR J=1 TO N
16510    Xppp(J)=0
16515    Yppp(J)=0
16520   NEXT J
16525   Cnx=X1(I)
16530   Cny=Y1(I)
16535   Xppp(1)=X2(I)
16540   Yppp(1)=Y2(I)
16545   Aa=2
16550   FOR Kk=1 TO N
16555    Aa_crit=Aa
```

```
16560    FOR K=1 TO Aa_crit-1
16565     FOR Test=1 TO N
16570      IF Xppp(Test)=Cnx AND Yppp(Test)=Cny THEN
16575       BEEP
16580       Link(I)=1
16585       GOTO 16720
16590      END IF
16595     NEXT Test
16600     FOR J=1 TO N
16605      IF Elem_array(J)=14 OR Elem_array(J)=15 OR Elem_array(J)=7 OR Elem_ar
ray(J)=8 THEN
16610       IF Xppp(K)=X1(J) AND Yppp(K)=Y1(J) AND Link(J)=0 AND J<>I THEN
16615        FOR Lll=1 TO N
16620         IF Xppp(Lll)=X2(J) AND Yppp(Lll)=Y2(J) THEN GOTO 16705
16625        NEXT Lll
16630        Xppp(Aa)=X2(J)
16635        Yppp(Aa)=Y2(J)
16640        Aa=Aa+1
16645        GOTO 16705
16650       END IF
16655       IF Xppp(K)=X2(J) AND Yppp(K)=Y2(J) AND Link(J)=0 AND J<>I THEN
16660        FOR Lll=1 TO N
16665         IF Xppp(Lll)=X1(J) AND Yppp(Lll)=Y1(J) THEN GOTO 16705
16670        NEXT Lll
16675        Xppp(Aa)=X1(J)
16680        Yppp(Aa)=Y1(J)
16685        Aa=Aa+1
16690        GOTO 16705
16695       END IF
16700      END IF
16705     NEXT J
16710    NEXT K
16715   NEXT Kk
16720  NEXT I
16725   !   FINDING INDUCTOR CUT-SETS
16730  FOR I=1 TO N
16735   IF Elem_array(I)<>16 AND Elem_array(I)<>17 THEN GOTO 16955
16740   IF Link(I)=0 THEN GOTO 16955
16745   FOR J=1 TO N
16750    Xppp(J)=0
16755    Yppp(J)=0
16760   NEXT J
16765   Cnx=X1(I)
16770   Cny=Y1(I)
16775   Xppp(1)=X2(I)
16780   Yppp(1)=Y2(I)
16785   Aa=2
16790   FOR Kk=1 TO N
16795    Aa_crit=Aa
16800    FOR K=1 TO Aa_crit-1
16805     FOR J=1 TO N
16810      IF Xppp(K)=X1(J) AND Yppp(K)=Y1(J) AND Link(J)=0 AND J<>I THEN
16815       FOR Lll=1 TO N
```

```
16820         IF Xppp(Lll)=X2(J) AND Yppp(Lll)=Y2(J) THEN GOTO 16900
16825       NEXT Lll
16830       Xppp(Aa)=X2(J)
16835       Yppp(Aa)=Y2(J)
16840       Aa=Aa+1
16845       GOTO 16900
16850      END IF
16855      IF Xppp(K)=X2(J) AND Yppp(K)=Y2(J) AND Link(J)=0 AND J<>I THEN
16860       FOR Lll=1 TO N
16865        IF Xppp(Lll)=X1(J) AND Yppp(Lll)=Y1(J) THEN GOTO 16900
16870       NEXT Lll
16875       Xppp(Aa)=X1(J)
16880       Yppp(Aa)=Y1(J)
16885       Aa=Aa+1
16890       GOTO 16900
16895      END IF
16900     NEXT J
16905    NEXT K
16910    NEXT Kk
16915    FOR Test=1 TO N
16920     IF Xppp(Test)=Cnx AND Yppp(Test)=Cny THEN
16925      GOTO 16950
16930     END IF
16935    NEXT Test
16940    BEEP
16945    Link(I)=0
16950          !   LABEL TO LEAVE A LOOP
16955  NEXT I
16960          !   TREE SOLVED
16965!
16970! IF THE TREE DESIGN DOES NOT EXIST
16975! ON FLOPPY DISK IT IS STORED HERE
16980  ON ERROR GOTO 16990
16985  PURGE "4"&Z$[2,9]
16990  OFF ERROR
16995  CREATE BDAT "4"&Z$[2,9],1,N*2+8
17000  ASSIGN @Path TO "4"&Z$[2,9]
17005  OUTPUT @Path;Link(*)
17010  ASSIGN @Path TO *
17015!
17020! ARRANGING THE ARRAYS
17025!
17030 Jump:    !
17035  M=0
17040  FOR I=1 TO N
17045   IF Link(I)=0 THEN M=M+1
17050  NEXT I
17055  ALLOCATE INTEGER Cn_y(1:M),Cn_x(1:M),Q(1:M,1:N-M),T(1:M)
17060 !*******************************************
17065  FOR I=1 TO N
17070   IF Values(I)=9999 THEN Time_sources=Time_sources+1
17075  NEXT I
17080  IF Time_sources<>0 THEN
```

```
17085   ALLOCATE Control(1:Time_sources,1:Limit),Control_var(1:Time_sources),Vec
t(1:Time_sources)
17090   FOR I=2 TO Time_sources
17095    Vect(I)=I-1
17100   NEXT I
17105   Vect(1)=Time_sources
17110  END IF
17115 !*****************************************
17120  M1=1
17125  FOR J=1 TO 10
17130   FOR I=1 TO N
17135    SELECT J
17140    CASE 1
17145     IF (Elem_array(I)=7 OR Elem_array(I)=8) AND Values(I)=0 THEN
17150      Elementtt(M1)=1
17155      Elemvalue(M1)=0
17160      Flags(M1)=0
17165      Eto=Eto+1
17170      Et=Et+1
17175      Node_a_x(M1)=X1(I)
17180      Node_a_y(M1)=Y1(I)
17185      Node_b_x(M1)=X2(I)
17190      Node_b_y(M1)=Y2(I)
17195      M1=M1+1
17200     END IF
17205    CASE 2
17210     IF (Elem_array(I)=7 OR Elem_array(I)=8) AND Values(I)<>0 THEN
17215      Elementtt(M1)=1
17220      Elemvalue(M1)=Values(I)
17225      Flags(M1)=0
17230      Et=Et+1
17235      Node_a_x(M1)=X1(I)
17240      Node_a_y(M1)=Y1(I)
17245      Node_b_x(M1)=X2(I)
17250      Node_b_y(M1)=Y2(I)
17255      !
17260! LOADING SOURCE FILES
17265      IF Values(I)=9999 THEN
17270       MAT REORDER Control BY Vect
17275       Var_source=Var_source+1
17280       Control_var(Var_source)=M1
17285       ASSIGN @Path TO Elem_label$(I)
17290       ON ERROR GOTO 17305
17295       ENTER @Path;Enotneeded
17300       ENTER @Path;Control(*)
17305       OFF ERROR
17310      END IF
17315      M1=M1+1
17320     END IF
17325    CASE 3
17330     IF (Elem_array(I)=14 OR Elem_array(I)=15) AND Link(I)=0 THEN
17335      Elementtt(M1)=2
17340      Elemvalue(M1)=Values(I)
```

```
17345      Flags(M1)=0
17350      Ct=Ct+1
17355      Node_a_x(M1)=X1(I)
17360      Node_a_y(M1)=Y1(I)
17365      Node_b_x(M1)=X2(I)
17370      Node_b_y(M1)=Y2(I)
17375      M1=M1+1
17380     END IF
17385    CASE 4
17390     IF Elem_array(I)=13 AND Link(I)=0 THEN
17395      Elementtt(M1)=4
17400      Elemvalue(M1)=Values(I)
17405      Flags(M1)=0
17410      Rt=Rt+1
17415      Node_a_x(M1)=X1(I)
17420      Node_a_y(M1)=Y1(I)
17425      Node_b_x(M1)=X2(I)
17430      Node_b_y(M1)=Y2(I)
17435      M1=M1+1
17440     END IF
17445    CASE 5
17450     IF (Elem_array(I)=16 OR Elem_array(I)=17) AND Link(I)=0 THEN
17455      Elementtt(M1)=3
17460      Elemvalue(M1)=Values(I)
17465      Flags(M1)=0
17470      It=It+1
17475      Node_a_x(M1)=X1(I)
17480      Node_a_y(M1)=Y1(I)
17485      Node_b_x(M1)=X2(I)
17490      Node_b_y(M1)=Y2(I)
17495      M1=M1+1
17500     END IF
17505    CASE 6
17510     IF (Elem_array(I)=14 OR Elem_array(I)=15) AND Link(I)=1 THEN
17515      Elementtt(M1)=2
17520      Elemvalue(M1)=Values(I)
17525      Flags(M1)=1
17530      Cl=Cl+1
17535      Node_a_x(M1)=X1(I)
17540      Node_a_y(M1)=Y1(I)
17545      Node_b_x(M1)=X2(I)
17550      Node_b_y(M1)=Y2(I)
17555      M1=M1+1
17560     END IF
17565    CASE 7
17570     IF Elem_array(I)=13 AND Link(I)=1 THEN
17575      Elementtt(M1)=4
17580      Elemvalue(M1)=Values(I)
17585      Flags(M1)=1
17590      Rl=Rl+1
17595      Node_a_x(M1)=X1(I)
17600      Node_a_y(M1)=Y1(I)
17605      Node_b_x(M1)=X2(I)
```

```
17610       Node_b_y(M1)=Y2(I)
17615       M1=M1+1
17620      END IF
17625     CASE 8
17630      IF (Elem_array(I)=16 OR Elem_array(I)=17) AND Link(I)=1 THEN
17635       Elementttt(M1)=3
17640       Elemvalue(M1)=Values(I)
17645       Flags(M1)=1
17650       Ll=Ll+1
17655       Node_a_x(M1)=X1(I)
17660       Node_a_y(M1)=Y1(I)
17665       Node_b_x(M1)=X2(I)
17670       Node_b_y(M1)=Y2(I)
17675       M1=M1+1
17680      END IF
17685     CASE 9
17690      IF (Elem_array(I)=5 OR Elem_array(I)=6) AND Values(I)=0 THEN
17695       Elementttt(M1)=5
17700       Elemvalue(M1)=0
17705       Flags(M1)=1
17710       Ilo=Ilo+1
17715       Il=Il+1
17720       Node_a_x(M1)=X1(I)
17725       Node_a_y(M1)=Y1(I)
17730       Node_b_x(M1)=X2(I)
17735       Node_b_y(M1)=Y2(I)
17740       M1=M1+1
17745      END IF
17750     CASE 10
17755      IF (Elem_array(I)=5 OR Elem_array(I)=6) AND Values(I)<>0 THEN
17760       Elementttt(M1)=5
17765       Elemvalue(M1)=Values(I)
17770       Flags(M1)=1
17775       Il=Il+1
17780       Node_a_x(M1)=X1(I)
17785       Node_a_y(M1)=Y1(I)
17790       Node_b_x(M1)=X2(I)
17795       Node_b_y(M1)=Y2(I)
17800       IF Values(I)=9999 THEN
17805        MAT REORDER Control BY Vect
17810        Var_source=Var_source+1
17815        Control_var(Var_source)=M1-(Ct+Rt+It+Cl+Rl+Ll)
17820        ASSIGN @Path TO Elem_label$(I)
17825        ON ERROR GOTO 17840
17830        ENTER @Path;Enotneeded
17835        ENTER @Path;Control(*)
17840        OFF ERROR
17845       END IF
17850       M1=M1+1
17855      END IF
17860     END SELECT
17865    NEXT I
17870   NEXT J
```

```
17875  Occ1=1
17880  IF Eto=0 AND Ilc=0 THEN
17885   FOR I=1 TO N
17890    IF (Elem_array(I)=14 OR Elem_array(I)=15) AND Link(I)=0 THEN
17895     Wave$(Occ1)[1,1]="V"
17900     Wave$(Occ1)[2,4]=VAL$(Occ1)
17905     Occ1=Occ1+1
17910    END IF
17915   NEXT I
17920   Occ2=Occ1-1
17925   Occ1=1
17930   FOR I=1 TO N
17935    IF (Elem_array(I)=16 OR Elem_array(I)=17) AND Link(I)=1 THEN
17940     Wave$(Occ1+Occ2)[1,1]="I"
17945     Wave$(Occ1+Occ2)[2,4]=VAL$(Occ1)
17950     Occ1=Occ1+1
17955    END IF
17960   NEXT I
17965  END IF
17970        !
17975 !*****************************************
17980 !            END OF INPUTS              *
17985 !*****************************************
17990! ARRANGING MATRICES
17995  IF C1<>0 THEN
18000   ALLOCATE Clink(1:C1,1:C1)
18005   FOR I=1 TO C1
18010    Clink(I,I)=Elemvalue(I+Et+Ct+Rt+It)
18015   NEXT I
18020  END IF
18025 !-----------------------------------------
18030  IF R1<>0 THEN
18035   ALLOCATE Rlink(1:R1,1:R1),Glink(1:R1,1:R1)
18040   FOR I=1 TO R1
18045    Rlink(I,I)=Elemvalue(I+Et+Ct+Rt+It+C1)
18050    Glink(I,I)=1/Rlink(I,I)
18055   NEXT I
18060  END IF
18065 !-----------------------------------------
18070  IF L1<>0 THEN
18075   ALLOCATE Llink(1:L1,1:L1)
18080   FOR I=1 TO L1
18085    Llink(I,I)=Elemvalue(I+Et+Ct+Rt+It+C1+R1)
18090   NEXT I
18095  END IF
18100 !-----------------------------------------
18105  IF Et<>0 THEN
18110   IF C1<>0 THEN ALLOCATE INTEGER Qec(1:Et,1:C1)
18115   IF R1<>0 THEN ALLOCATE INTEGER Qer(1:Et,1:R1)
18120   IF L1<>0 THEN ALLOCATE INTEGER Qel(1:Et,1:L1)
18125   IF I1<>0 THEN ALLOCATE INTEGER Qei(1:Et,1:I1)
18130  END IF
18135 !-----------------------------------------
```

```
18140  IF Ct<>0 THEN
18145   ALLOCATE Ctwig(1:Ct,1:Ct)
18150   FOR I=1 TO Ct
18155    Ctwig(I,I)=Elemvalue(I+Et)
18160   NEXT I
18165   IF Cl<>0 THEN ALLOCATE INTEGER Qcc(1:Ct,1:Cl)
18170   IF Rl<>0 THEN ALLOCATE INTEGER Qcr(1:Ct,1:Rl)
18175   IF Ll<>0 THEN ALLOCATE INTEGER Qcl(1:Ct,1:Ll)
18180   IF Il<>0 THEN ALLOCATE INTEGER Qci(1:Ct,1:Il)
18185  END IF
18190 !-------------------------------------------------
18195  IF Rt<>0 THEN
18200   ALLOCATE Rtwig(1:Rt,1:Rt),Gtwig(1:Rt,1:Rt)
18205   FOR I=1 TO Rt
18210    Rtwig(I,I)=Elemvalue(I+Et+Ct)
18215    Gtwig(I,I)=1/Rtwig(I,I)
18220   NEXT I
18225   IF Cl<>0 THEN ALLOCATE INTEGER Qrc(1:Rt,1:Cl)
18230   IF Rl<>0 THEN ALLOCATE INTEGER Qrr(1:Rt,1:Rl)
18235   IF Ll<>0 THEN ALLOCATE INTEGER Qrl(1:Rt,1:Ll)
18240   IF Il<>0 THEN ALLOCATE INTEGER Qri(1:Rt,1:Il)
18245  END IF
18250 !-------------------------------------------------
18255  IF It<>0 THEN
18260   ALLOCATE Itwig(1:It,1:It)
18265   FOR I=1 TO It
18270    Itwig(I,I)=Elemvalue(I+Et+Ct+Rt)
18275   NEXT I
18280   IF Cl<>0 THEN ALLOCATE INTEGER Qlc(1:It,1:Cl)
18285   IF Rl<>0 THEN ALLOCATE INTEGER Qlr(1:It,1:Rl)
18290   IF Ll<>0 THEN ALLOCATE INTEGER Qll(1:It,1:Ll)
18295   IF Il<>0 THEN ALLOCATE INTEGER Qli(1:It,1:Il)
18300  END IF
18305 !*******************************************************
18310  FOR Element1=1 TO N
18315   IF Flags(Element1)=1 THEN GOTO 18560
18320   Aa=0
18325   Bb=0
18330   FOR I=1 TO N
18335    IF I=Element1 THEN GOTO 18355
18340    IF Flags(I)=1 THEN GOTO 18355
18345    IF Node_a_x(I)=Node_a_x(Element1) AND Node_a_y(I)=Node_a_y(Element1) OR
 Node_b_x(I)=Node_a_x(Element1) AND Node_b_y(I)=Node_a_y(Element1) THEN Aa=Aa+1
18350    IF Node_a_x(I)=Node_b_x(Element1) AND Node_a_y(I)=Node_b_y(Element1) OR
 Node_b_x(I)=Node_b_x(Element1) AND Node_b_y(I)=Node_b_y(Element1) THEN Bb=Bb+1
18355   NEXT I
18360   IF Aa<=Bb THEN
18365    On_x(Element1)=Node_a_x(Element1)
18370    On_y(Element1)=Node_a_y(Element1)
18375    Factor=1
18380   ELSE
18385    On_x(Element1)=Node_b_x(Element1)
18390    On_y(Element1)=Node_b_y(Element1)
```

```
18395     Factor=-1
18400    END IF
18405    T(Element1)=1
18410  Find_twigs:!
18415    FOR I=1 TO M
18420     IF T(I)<>0 THEN GOTO Sort_q
18425    NEXT I
18430    GOTO 18560
18435  Sort_q:!
18440    FOR J=1 TO N
18445     IF I=J THEN GOTO 18545
18450     IF Node_a_x(J)=Cn_x(I) AND Node_a_y(J)=Cn_y(I) THEN
18455      IF Flags(J)=1 THEN
18460       Q(Element1,J-M)=Q(Element1,J-M)-Factor
18465      ELSE
18470       T(J)=-1
18475       Cn_x(J)=Node_b_x(J)
18480       Cn_y(J)=Node_b_y(J)
18485      END IF
18490      GOTO 18545
18495     END IF
18500     IF Node_b_x(J)=Cn_x(I) AND Node_b_y(J)=Cn_y(I) THEN
18505      IF Flags(J)=1 THEN
18510       Q(Element1,J-M)=Q(Element1,J-M)+Factor
18515      ELSE
18520       T(J)=1
18525       Cn_x(J)=Node_a_x(J)
18530       Cn_y(J)=Node_a_y(J)
18535      END IF
18540     END IF
18545    NEXT J
18550    T(I)=0
18555    GOTO Find_twigs
18560   NEXT Element1
18565   MAT Q= Q*(-1)
18570  !*************************************************************
18575   FOR I=1 TO Et+Ct+Rt+Lt
18580    FOR J=1 TO Cl+Rl+Ll+Il
18585     IF J>Cl+Rl+Ll THEN
18590      IF I>Et+Ct+Rt THEN Qli(I-Et-Ct-Rt,J-Cl-Rl-Ll)=Q(I,J)
18595      IF I>Et+Ct AND I<=Et+Ct+Rt THEN Qri(I-Et-Ct,J-Cl-Rl-Ll)=Q(I,J)
18600      IF I>Et AND I<=Et+Ct THEN Qci(I-Et,J-Cl-Rl-Ll)=Q(I,J)
18605      IF I<=Et THEN Qei(I,J-Cl-Rl-Ll)=Q(I,J)
18610     END IF
18615     IF J>Cl+Rl AND J<=Cl+Rl+Ll THEN
18620      IF I>Et+Ct+Rt THEN Qll(I-Et-Ct-Rt,J-Cl-Rl)=Q(I,J)
18625      IF I>Et+Ct AND I<=Et+Ct+Rt THEN Qrl(I-Et-Ct,J-Cl-Rl)=Q(I,J)
18630      IF I>Et AND I<=Et+Ct THEN Qcl(I-Et,J-Cl-Rl)=Q(I,J)
18635      IF I<=Et THEN Qel(I,J-Cl-Rl)=Q(I,J)
18640     END IF
18645     IF J>Cl AND J<=Cl+Rl THEN
18650      IF I>Et+Ct+Rt THEN Qlr(I-Et-Ct-Rt,J-Cl)=Q(I,J)
18655      IF I>Et+Ct AND I<=Et+Ct+Rt THEN Qrr(I-Et-Ct,J-Cl)=Q(I,J)
```

```
18660     IF D>Et AND I<=Et+Ct THEN Qcr(I-Et,J-Cl)=Q(I,J)
18665     IF I<=Et THEN Qer(I,J-Cl)=Q(I,J)
18670    END IF
18675    IF J<=Cl THEN
18680     IF D>Et+Ct+Rt THEN Qlc(I-Et-Ct-Rt,J)=Q(I,J)
18685     IF D>Et+Ct AND I<=Et+Ct+Rt THEN Qrc(I-Et-Ct,J)=Q(I,J)
18690     IF D>Et AND I<=Et+Ct THEN Qcc(I-Et,J)=Q(I,J)
18695     IF I<=Et THEN Qec(I,J)=Q(I,J)
18700    END IF
18705   NEXT J
18710  NEXT I
18715 !*********************************************
18720  IF Ct<>0 THEN
18725   ALLOCATE C(1:Ct,1:Ct),Dummy1(1:Ct,1:Ct)
18730   IF Cl<>0 THEN
18735    ALLOCATE Dummy2(1:Cl,1:Ct),Dummy3(1:Ct,1:Cl)
18740    MAT Dummy3= Qcc*Clink
18745    IF Et<>0 THEN
18750     ALLOCATE Dummy4(1:Cl,1:Et),C_(1:Ct,1:Et)
18755     MAT Dummy4= TRN(Qec)
18760     MAT C_= Dummy3*Dummy4
18765     MAT C_= C_*(-1)
18770     DEALLOCATE Dummy4(*)
18775    END IF
18780    MAT Dummy2= TRN(Qcc)
18785    MAT Dummy1= Dummy3*Dummy2
18790    DEALLOCATE Dummy2(*),Dummy3(*)
18795   END IF
18800   MAT C= Ctwig+Dummy1
18805   DEALLOCATE Dummy1(*)
18810  END IF
18815 !-----------------------------------------
18820  IF Ll<>0 THEN
18825   ALLOCATE L(1:Ll,1:Ll),Dummy1(1:Ll,1:Ll)
18830   IF It<>0 THEN
18835    ALLOCATE Dummy2(1:Ll,1:It),Dummy3(1:Ll,1:It)
18840    MAT Dummy2= TRN(Qll)
18845    MAT Dummy3= Dummy2*Itwig
18850    IF Il<>0 THEN
18855     ALLOCATE L_(1:Ll,1:Il)
18860     MAT L_= Dummy3*Qli
18865     MAT L_= L_*(-1)
18870    END IF
18875    MAT Dummy1= Dummy3*Qll
18880    DEALLOCATE Dummy2(*),Dummy3(*)
18885   END IF
18890   MAT L= Llink+Dummy1
18895   DEALLOCATE Dummy1(*)
18900  END IF
18905 !-----------------------------------------
18910  IF Rl<>0 THEN
18915   ALLOCATE R(1:Rl,1:Rl),Dummy1(1:Rl,1:Rl)
18920   IF Rt<>0 THEN
```

```
18925    ALLOCATE Dummy2(1:Rl,1:Rt),Dummy3(1:Rl,1:Rt)
18930    MAT Dummy2= TRN(Qrr)
18935    MAT Dummy3= Dummy2*Rtwig
18940    MAT Dummy1= Dummy3*Qrr
18945    DEALLOCATE Dummy2(*),Dummy3(*)
18950   END IF
18955   MAT R= Rlink+Dummy1
18960   DEALLOCATE Dummy1(*)
18965  END IF
18970 !-----------------------------------------
18975  IF Rt<>0 THEN
18980   ALLOCATE G(1:Rt,1:Rt),Dummy1(1:Rt,1:Rt)
18985   IF Rl<>0 THEN
18990    ALLOCATE Dummy2(1:Rl,1:Rt),Dummy3(1:Rt,1:Rl)
18995    MAT Dummy2= TRN(Qrr)
19000    MAT Dummy3= Qrr*Glink
19005    MAT Dummy1= Dummy3*Dummy2
19010    DEALLOCATE Dummy3(*),Dummy2(*)
19015   END IF
19020   MAT G= Gtwig+Dummy1
19025   DEALLOCATE Dummy1(*)
19030  END IF
19035 !*****************************************
19040  IF Rl<>0 THEN
19045   ALLOCATE Dummy1(1:Rl,1:Rl)
19050   MAT Dummy1= INV(R)
19055   IF Ct<>0 THEN
19060    ALLOCATE Dummy2(1:Ct,1:Rl),Yz(1:Ct,1:Ct),Dummy3(1:Rl,1:Ct)
19065    MAT Dummy2= Qcr*Dummy1
19070    MAT Dummy3= TRN(Qcr)
19075    MAT Yz= Dummy2*Dummy3
19080    DEALLOCATE Dummy3(*)
19085    IF Et<>0 THEN
19090     ALLOCATE Dummy3(1:Rl,1:Et),Y_(1:Ct,1:Et)
19095     MAT Dummy3= TRN(Qer)
19100     MAT Y_= Dummy2*Dummy3
19105     DEALLOCATE Dummy3(*)
19110    END IF
19115   END IF
19120   DEALLOCATE Dummy1(*)
19125  END IF
19130  IF Ll<>0 AND Ct<>0 THEN
19135   ALLOCATE H(1:Ct,1:Ll),Dummy4(1:Ct,1:Ll)
19140   IF Rt<>0 AND Rl<>0 THEN
19145    ALLOCATE Dummy5(1:Rl,1:Rt),Dummy6(1:Ct,1:Rt),Dummy3(1:Ct,1:Rt)
19150    MAT Dummy5= TRN(Qrr)
19155    MAT Dummy3= Dummy2*Dummy5
19160    MAT Dummy6= Dummy3*Rtwig
19165    MAT Dummy4= Dummy6*Qrl
19170    DEALLOCATE Dummy6(*),Dummy3(*),Dummy5(*)
19175   END IF
19180   MAT H= Dummy4-Qcl
19185   DEALLOCATE Dummy4(*)
```

```
19190  END IF
19195  IF Il<>0 AND Ct<>0 THEN
19200   ALLOCATE H_(1:Ct,1:Il),Dummy4(1:Ct,1:Il)
19205   IF Rt<>0 AND Rl<>0 THEN
19210    ALLOCATE Dummy5(1:Rl,1:Rt),Dummy6(1:Ct,1:Rt),Dummy3(1:Ct,1:Rt)
19215    MAT Dummy5= TRN(Qrr)
19220    MAT Dummy3= Dummy2*Dummy5
19225    MAT Dummy6= Dummy3*Rtwig
19230    MAT Dummy4= Dummy6*Qri
19235    DEALLOCATE Dummy6(*),Dummy3(*),Dummy5(*)
19240   END IF
19245   MAT H_ = Dummy4-Qci
19250   DEALLOCATE Dummy4(*)
19255  END IF
19260  IF Ct<>0 AND Rl<>0 THEN DEALLOCATE Dummy2(*)
19265  !-----------------------------------------------
19270  IF Rt<>0 THEN
19275   ALLOCATE Dummy1(1:Rt,1:Rt)
19280   MAT Dummy1= INV(G)
19285   IF Ll<>0 THEN
19290    ALLOCATE Dummy2(1:Ll,1:Rt),Dummy3(1:Ll,1:Rt),Ffz(1:Ll,1:Ll)
19295    MAT Dummy2= TRN(Qrl)
19300    MAT Dummy3= Dummy2*Dummy1
19305    MAT Ffz= Dummy3*Qrl
19310    DEALLOCATE Dummy2(*)
19315    IF Il<>0 THEN
19320     ALLOCATE F_(1:Ll,1:Il)
19325     MAT F_ = Dummy3*Qri
19330    END IF
19335   END IF
19340   DEALLOCATE Dummy1(*)
19345  END IF
19350  IF Ct<>0 AND Ll<>0 THEN
19355   ALLOCATE Sttttd(1:Ll,1:Ct),Dummy2(1:Ll,1:Ct),Dummy7(1:Ll,1:Ct)
19360   IF Rl<>0 AND Rt<>0 THEN
19365    ALLOCATE Dummy4(1:Rl,1:Ct),Dummy5(1:Ll,1:Rl),Dummy6(1:Ll,1:Rl)
19370    MAT Dummy4= TRN(Qcr)
19375    MAT Dummy5= Dummy3*Qrr
19380    MAT Dummy6= Dummy5*Glink
19385    MAT Dummy2= Dummy6*Dummy4
19390    DEALLOCATE Dummy4(*),Dummy5(*),Dummy6(*)
19395   END IF
19400   MAT Dummy7= TRN(Qcl)
19405   MAT Sttttd= Dummy7-Dummy2
19410   DEALLOCATE Dummy7(*),Dummy2(*)
19415  END IF
19420  IF Et<>0 AND Ll<>0 THEN
19425   ALLOCATE S_(1:Ll,1:Et),Dummy2(1:Ll,1:Et),Dummy7(1:Ll,1:Et)
19430   IF Rl<>0 AND Rt<>0 THEN
19435    ALLOCATE Dummy4(1:Rl,1:Et),Dummy5(1:Ll,1:Rl),Dummy6(1:Ll,1:Rl)
19440    MAT Dummy4= TRN(Qer)
19445    MAT Dummy5= Dummy3*Qrr
19450    MAT Dummy6= Dummy5*Glink
```

```
19455     MAT Dummy2= Dummy6*Dummy4
19460     DEALLOCATE Dummy4(*),Dummy5(*),Dummy6(*)
19465    END IF
19470    MAT Dummy7= IRN(Qel)
19475    MAT S = Dummy7-Dummy2
19480    DEALLOCATE Dummy7(*),Dummy2(*)
19485   END IF
19490   IF Rt<>0 AND Ll<>0 THEN DEALLOCATE Dummy3(*)
19495   !*********************************************
19500   IF Ct<>0 AND Ll<>0 THEN
19505    ALLOCATE Dummy1(1:Ct+Ll,1:Ct+Ll),Dummy2(1:Ct,1:Ct),Dummy3(1:Ll,1:Ll)
19510   ELSE
19515    IF Ct<>0 THEN
19520     ALLOCATE Dummy1(1:Ct,1:Ct),Dummy2(1:Ct,1:Ct)
19525    ELSE
19530     ALLOCATE Dummy1(1:Ll,1:Ll),Dummy3(1:Ll,1:Ll)
19535    END IF
19540   END IF
19545   IF Ct<>0 THEN
19550    MAT Dummy2= INV(C)
19555    FOR I=1 TO Ct
19560     FOR J=1 TO Ct
19565      Dummy1(I,J)=Dummy2(I,J)
19570     NEXT J
19575    NEXT I
19580    DEALLOCATE Dummy2(*)
19585   END IF
19590   IF Ll<>0 THEN
19595    MAT Dummy3= INV(L)
19600    FOR I=1 TO Ll
19605     FOR J=1 TO Ll
19610      Dummy1(I+Ct,J+Ct)=Dummy3(I,J)
19615     NEXT J
19620    NEXT I
19625    DEALLOCATE Dummy3(*)
19630   END IF
19635   !
19640   ALLOCATE Dummy2(1:Ct+Ll,1:Ct+Ll),A(1:Ct+Ll,1:Ct+Ll)
19645   IF Rl<>0 AND Ct<>0 THEN
19650    FOR I=1 TO Ct
19655     FOR J=1 TO Ct
19660      Dummy2(I,J)=-Yz(I,J)
19665     NEXT J
19670    NEXT I
19675   END IF
19680   IF Ct<>0 AND Ll<>0 THEN
19685    FOR I=1 TO Ct
19690     FOR J=1 TO Ll
19695      Dummy2(I,J+Ct)=H(I,J)
19700     NEXT J
19705    NEXT I
19710   END IF
19715   IF Rt<>0 AND Ll<>0 THEN
```

```
19720   FOR I=1 TO Ll
19725    FOR J=1 TO Ll
19730     Dummy2(I+Ct,J+Ct)=Ffz(I,J)
19735    NEXT J
19740   NEXT I
19745  END IF
19750  IF Ll<>0 AND Ct<>0 THEN
19755   FOR I=1 TO Ll
19760    FOR J=1 TO Ct
19765     Dummy2(I+Ct,J)=Stttta(I,J)
19770    NEXT J
19775   NEXT I
19780  END IF
19785  MAT A= Dummy1*Dummy2
19790  DEALLOCATE Dummy2(*)
19795 !*****************************************
19800  IF Et<>0 OR Il<>0 THEN
19805   ALLOCATE Dummy2(1:Ct+Ll,1:Et+Il),B1(1:Ct+Ll,1:Et+Il),B(1:Ct+Ll,1:Et+Il)
19810   IF Ct<>0 AND Et<>0 AND Rl<>0 THEN
19815    FOR I=1 TO Ct
19820     FOR J=1 TO Et
19825      Dummy2(I,J)=-V_(I,J)
19830     NEXT J
19835    NEXT I
19840   END IF
19845   IF Ct<>0 AND Il<>0 THEN
19850    FOR I=1 TO Ct
19855     FOR J=1 TO Il
19860      Dummy2(I,J+Et)=H_(I,J)
19865     NEXT J
19870    NEXT I
19875   END IF
19880   IF Ll<>0 AND Rt<>0 AND Il<>0 THEN
19885    FOR I=1 TO Ll
19890     FOR J=1 TO Il
19895      Dummy2(I+Ct,J+Et)=-F_(I,J)
19900     NEXT J
19905    NEXT I
19910   END IF
19915   IF Et<>0 AND Ll<>0 THEN
19920    FOR I=1 TO Ll
19925     FOR J=1 TO Et
19930      Dummy2(I+Ct,J)=S_(I,J)
19935     NEXT J
19940    NEXT I
19945   END IF
19950   MAT B1= Dummy1*Dummy2
19955   DEALLOCATE Dummy2(*)
19960 !*****************************************
19965   ALLOCATE Dummy2(1:Ct+Ll,1:Et+Il),B2(1:Ct+Ll,1:Et+Il)
19970   IF Cl<>0 AND Et<>0 THEN
19975    FOR I=1 TO Ct
19980     FOR J=1 TO Et
```

```
19985       Dummy2(I,J)=C_(I,J)
19990      NEXT J
19995     NEXT I
20000    END IF
20005    IF Lt<>0 AND Il<>0 THEN
20010     FOR I=1 TO Ll
20015      FOR J=1 TO Il
20020       Dummy2(I+Ct,J+Et)=L_(I,J)
20025      NEXT J
20030     NEXT I
20035    END IF
20040    MAT B2= Dummy1*Dummy2
20045    DEALLOCATE Dummy1(*),Dummy2(*)
20050    ALLOCATE Dummy1(1:Ct+Ll,1:Et+Il)
20055    MAT Dummy1= A*B2
20060    MAT B= B1+Dummy1
20065    DEALLOCATE Dummy1(*)
20070   END IF
20075! FINISHED MATRICES A AND B
20080 ! PRINT "MATRIX A"
20085 ! PRINT A(*),"
20090 ! PRINT "MATRIX B"
20095 ! PRINT B(*),"
20100 !*******************************************
20105  ALLOCATE Dummy1(1:Ct+Ll,1:Ct+Ll),Dummy2(1:Ct+Ll,1:Ct+Ll),Dummy3(1:Ct+Ll,1
:Ct+Ll)
20110  ALLOCATE Dummy4(1:Ct+Ll,1:Ct+Ll),A_dt(1:Ct+Ll,1:Ct+Ll),Dummy5(1:Ct+Ll,1:C
t+Ll)
20115  MAT Dummy1= IDN
20120  MAT Dummy4= IDN
20125  FOR K=0 TO 6
20130   IF K=0 THEN GOTO 20180
20135   FOR I=1 TO Ct+Ll
20140    FOR J=1 TO Ct+Ll
20145     IF ABS(Dummy1(I,J))<=1.E-30 THEN Dummy1(I,J)=0
20150     IF ABS(Dummy1(I,J))>=1.E+50 THEN GOTO 20190
20155    NEXT J
20160   NEXT I
20165   MAT Dummy3= Dummy1*A
20170   MAT Dummy1= Dummy3*(Dt/K)
20175   MAT Dummy4= Dummy1/(K+1)
20180   MAT Dummy2= Dummy1+Dummy2
20185   MAT Dummy5= Dummy5+Dummy4
20190  NEXT K
20195  MAT A_dt= Dummy2
20200  DEALLOCATE Dummy1(*),Dummy2(*),Dummy3(*),Dummy4(*)
20205  IF Il<>0 OR Et<>0 THEN
20210   ALLOCATE B_dt(1:Ct+Ll,1:Et+Il)
20215   MAT B_dt= Dummy5*B
20220   MAT B_dt= B_dt*(Dt)
20225  END IF
20230  DEALLOCATE Dummy5(*)
20235! ************************
```

```
20240! * MATRICES ARE SET UP   *
20245! ************************
20250 ! PRINT "A_dt"
20255 ! PRINT A_dt(*),"
20260 ! PRINT "B_dt"
20265 ! PRINT B_dt(*)
20270 !*******************************************
20275 !*              OUTPUTS                    *
20280 !*******************************************
20285 !
20290 !-------------------------------------------------------------------
20295 ! IF ETO AND ITO =0 (CAPACITOR LOOP OR INDUCTOR CUT-SET EXISTS)
20300 !-------------------------------------------------------------------
20305 !
20310  IF Eto=0 AND Ilo=0 THEN
20315   ALLOCATE C_dt(1:Ct+Ll,1:Ct+Ll)
20320   MAT C_dt= IDN
20325   IF Et<>0 OR Il<>0 THEN
20330    ALLOCATE D_dt(1:Ct+Ll,1:Et+Il)
20335    MAT D_dt= C_dt*B2
20340   END IF
20345   GOTO 21810
20350  END IF
20355 !
20360 !-------------------------------------------
20365 !-          FINDING Eto and Ilo            -
20370 !-------------------------------------------
20375  IF Eto<>0 THEN
20380   IF Cl<>0 THEN
20385    REDIM Qec(1:Eto,1:Cl)
20390    ALLOCATE Qecc(1:Eto,1:Cl)
20395    MAT Qecc= Qec
20400    REDIM Qec(1:Et,1:Cl)
20405   END IF
20410   IF Rl<>0 THEN
20415    REDIM Qer(1:Eto,1:Rl)
20420    ALLOCATE Qecr(1:Eto,1:Rl)
20425    MAT Qecr= Qer
20430    REDIM Qer(1:Et,1:Rl)
20435   END IF
20440   IF Ll<>0 THEN
20445    REDIM Qel(1:Eto,1:Ll)
20450    ALLOCATE Qecl(1:Eto,1:Ll)
20455    MAT Qecl= Qel
20460    REDIM Qel(1:Et,1:Ll)
20465   END IF
20470   IF Il<>0 THEN
20475    REDIM Qei(1:Eto,1:Il)
20480    ALLOCATE Qeci(1:Eto,1:Il)
20485    MAT Qeci= Qei
20490    REDIM Qei(1:Et,1:Il)
20495   END IF
20500  END IF
```

```
20505  IF Ilo<>0 THEN
20510   IF Et<>0 THEN
20515    ALLOCATE Qeio(1:Et,1:Ilo),Dummy15(1:Il,1:Et)
20520    MAT Dummy15= TRN(Qei)
20525    REDIM Dummy15(1:Ilo,1:Et)
20530    MAT Qeio= TRN(Dummy15)
20535    DEALLOCATE Dummy15(*)
20540   END IF
20545   IF Ct<>0 THEN
20550    ALLOCATE Qcio(1:Ct,1:Ilo),Dummy15(1:Il,1:Ct)
20555    MAT Dummy15= TRN(Qci)
20560    REDIM Dummy15(1:Ilo,1:Ct)
20565    MAT Qcio= TRN(Dummy15)
20570    DEALLOCATE Dummy15(*)
20575   END IF
20580   IF Rt<>0 THEN
20585    ALLOCATE Qrio(1:Rt,1:Ilo),Dummy15(1:Il,1:Rt)
20590    MAT Dummy15= TRN(Qri)
20595    REDIM Dummy15(1:Ilo,1:Rt)
20600    MAT Qrio= TRN(Dummy15)
20605    DEALLOCATE Dummy15(*)
20610   END IF
20615   IF It<>0 THEN
20620    ALLOCATE Qlio(1:It,1:Ilo),Dummy15(1:Il,1:It)
20625    MAT Dummy15= TRN(Qli)
20630    REDIM Dummy15(1:Ilo,1:It)
20635    MAT Qlio= TRN(Dummy15)
20640    DEALLOCATE Dummy15(*)
20645   END IF
20650  END IF
20655  !
20660  !-------------------------------------------------------------
20665  !-              SOLVING SUBMATRICES                          -
20670  !-------------------------------------------------------------
20675  IF (Eto<>0 AND (Il<>0 OR (Ct<>0 AND (Cl<>0 OR Rl<>0)))) OR (Ilo<>0 AND (C
t<>0 OR (Il<>0 AND (It<>0 OR Rt<>0)))) THEN ALLOCATE C_dt(1:Eto+Ilo,1:Ct+Il)
20680  IF (Eto<>0 AND (Il<>0 OR Rl<>0 OR (Cl<>0 AND Ct<>0))) OR (Ilo<>0 AND (Et<
>0 OR Rt<>0 OR (It<>0 AND Il<>0))) THEN ALLOCATE D_dt(1:Eto+Ilo,1:Et+Il)
20685  IF (Eto<>0 AND Cl<>0 AND Ct<>0) OR (Ilo<>0 AND It<>0 AND Il<>0) THEN ALLO
CATE Dummy20(1:Eto+Ilo,1:Ct+Il)
20690  IF Eto<>0 AND Cl<>0 AND Ct<>0 THEN
20695   ALLOCATE Dummy22(1:Cl,1:Ct),Dummy23(1:Cl,1:Ct),Dummy24(1:Eto,1:Ct)
20700   MAT Dummy22= TRN(Qcc)
20705   MAT Dummy23= Clink*Dummy22
20710   MAT Dummy24= Qeoc*Dummy23
20715   FOR I=1 TO Eto
20720    FOR J=1 TO Ct
20725     Dummy20(I,J)=Dummy24(I,J)
20730    NEXT J
20735   NEXT I
20740   DEALLOCATE Dummy22(*),Dummy23(*),Dummy24(*)
20745  END IF
20750  IF Ilo<>0 AND It<>0 AND Il<>0 THEN
```

```
20755  ALLOCATE Dummy22(1:Ilo,1:Ll),Dummy23(1:It,1:Ll),Dummy24(1:Ilo,1:Ll)
20760  MAT Dummy22= TRN(Qlio)
20765  MAT Dummy23= Itwig*Qll
20770  MAT Dummy24= Dummy22*Dummy23
20775  FOR I=1 TO Ilo
20780   FOR J=1 TO Ll
20785    Dummy20(I+Eto,J+Ct)=Dummy24(I,J)
20790   NEXT J
20795  NEXT I
20800  DEALLOCATE Dummy22(*),Dummy23(*),Dummy24(*)
20805 END IF
20810 IF (Eto<>0 AND (Ll<>0 OR (Ct<>0 AND Rl<>0))) OR (Ilo<>0 AND (Ct<>0 OR (Ll
<>0 AND Rt<>0))) THEN ALLOCATE Dummy30(1:Eto+Ilo,1:Ct+Ll)
20815 IF (Eto<>0 AND (Il<>0 OR Rl<>0)) OR (Ilo<>0 AND (Et<>0 OR Rt<>0)) THEN AL
LOCATE Dummy40(1:Eto+Ilo,1:Et+Il)
20820 IF Rl<>0 THEN
20825  ALLOCATE Dummy31(1:Rl,1:Rl)
20830  MAT Dummy31= INV(R)
20835  IF Eto<>0 THEN
20840   ALLOCATE Dummy32(1:Eto,1:Rl),Dummy33(1:Rl,1:Et),Xx5(1:Eto,1:Et)
20845   MAT Dummy33= TRN(Qer)
20850   MAT Dummy32= Qeor*Dummy31
20855   MAT Xx5= Dummy32*Dummy33
20860   FOR I=1 TO Eto
20865    FOR J=1 TO Et
20870     Dummy40(I,J)=Xx5(I,J)
20875    NEXT J
20880   NEXT I
20885   DEALLOCATE Dummy33(*)
20890   IF Ct<>0 THEN
20895    ALLOCATE Dummy33(1:Rl,1:Ct),Xx100(1:Eto,1:Ct)
20900    MAT Dummy33= TRN(Qcr)
20905    MAT Xx100= Dummy32*Dummy33
20910    FOR I=1 TO Eto
20915     FOR J=1 TO Ct
20920      Dummy30(I,J)=Xx100(I,J)
20925     NEXT J
20930    NEXT I
20935    DEALLOCATE Dummy33(*)
20940   END IF
20945  END IF
20950  DEALLOCATE Dummy31(*)
20955 END IF
20960 IF Eto<>0 AND Ll<>0 THEN
20965  ALLOCATE Xx2(1:Eto,1:Ll),Dummy34(1:Eto,1:Ll)
20970  IF Rt<>0 AND Rl<>0 THEN
20975   ALLOCATE Dummy35(1:Rl,1:Rt),Dummy36(1:Eto,1:Rt),Dummy33(1:Eto,1:Rt)
20980   MAT Dummy35= TRN(Qrr)
20985   MAT Dummy33= Dummy32*Dummy35
20990   MAT Dummy36= Dummy33*Rtwig
20995   MAT Dummy34= Dummy36*Qrl
21000   DEALLOCATE Dummy36(*),Dummy33(*),Dummy35(*)
21005  END IF
```

```
21010  MAT Xx2= Dummy34-Qeol
21015  FOR I=1 TO Eto
21020   FOR J=1 TO Ll
21025    Dummy30(I,J+Ct)=Xx2(I,J)
21030   NEXT J
21035  NEXT I
21040  DEALLOCATE Dummy34(*)
21045 END IF
21050 IF Eto<>0 AND Il<>0 THEN
21055  ALLOCATE Xx6(1:Eto,1:Il),Dummy34(1:Eto,1:Il)
21060  IF Rt<>0 AND Rl<>0 THEN
21065   ALLOCATE Dummy35(1:Rl,1:Rt),Dummy36(1:Eto,1:Rt),Dummy33(1:Eto,1:Rt)
21070   MAT Dummy35= TRN(Qrr)
21075   MAT Dummy36= Dummy32*Dummy35
21080   MAT Dummy33= Dummy36*Rtwig
21085   MAT Dummy34= Dummy33*Qri
21090   DEALLOCATE Dummy36(*),Dummy35(*),Dummy33(*)
21095  END IF
21100  MAT Xx6= Dummy34-Qeoi
21105  FOR I=1 TO Eto
21110   FOR J=1 TO Il
21115    Dummy40(I,J+Et)=Xx6(I,J)
21120   NEXT J
21125  NEXT I
21130  DEALLOCATE Dummy34(*)
21135 END IF
21140 IF Eto<>0 AND Rl<>0 THEN DEALLOCATE Dummy32(*)
21145 IF Rt<>0 THEN
21150  ALLOCATE Dummy41(1:Rt,1:Rt)
21155  MAT Dummy41= INV(G)
21160  IF Ilo<>0 THEN
21165   ALLOCATE Dummy42(1:Ilo,1:Rt),Dummy43(1:Ilo,1:Rt),Xx8(1:Ilo,1:Il)
21170   MAT Dummy42= TRN(Qrio)
21175   MAT Dummy43= Dummy42*Dummy41
21180   MAT Xx8= Dummy43*Qri
21185   FOR I=1 TO Ilo
21190    FOR J=1 TO Il
21195     Dummy40(I+Eto,J+Et)=Xx8(I,J)
21200    NEXT J
21205   NEXT I
21210   DEALLOCATE Dummy42(*)
21215   IF Il<>0 THEN
21220    ALLOCATE Xx4(1:Ilo,1:Ll)
21225    MAT Xx4= Dummy43*Qrl
21230    FOR I=1 TO Ilo
21235     FOR J=1 TO Ll
21240      Dummy30(I+Eto,J+Ct)=Xx4(I,J)
21245     NEXT J
21250    NEXT I
21255   END IF
21260  END IF
21265  DEALLOCATE Dummy41(*)
21270 END IF
```

```
21275  IF Ct<>0 AND Ilo<>0 THEN
21280   ALLOCATE Xx3(1:Ilo,1:Ct),Dummy42(1:Ilo,1:Ct),Dummy47(1:Ilo,1:Ct)
21285   IF Rl<>0 AND Rt<>0 THEN
21290    ALLOCATE Dummy44(1:Rl,1:Ct),Dummy45(1:Ilo,1:Rl),Dummy46(1:Ilo,1:Rl)
21295    MAT Dummy44= TRN(Qcr)
21300    MAT Dummy45= Dummy43*Qrr
21305    MAT Dummy46= Dummy45*Glink
21310    MAT Dummy42= Dummy46*Dummy44
21315    DEALLOCATE Dummy44(*),Dummy45(*),Dummy46(*)
21320   END IF
21325   MAT Dummy47= TRN(Qcio)
21330   MAT Xx3= Dummy47-Dummy42
21335   FOR I=1 TO Ilo
21340    FOR J=1 TO Ct
21345     Dummy30(I+Eto,J)=Xx3(I,J)
21350    NEXT J
21355   NEXT I
21360   DEALLOCATE Dummy47(*),Dummy42(*)
21365  END IF
21370  IF Et<>0 AND Ilo<>0 THEN
21375   ALLOCATE Xx7(1:Ilo,1:Et),Dummy42(1:Ilo,1:Et),Dummy47(1:Ilo,1:Et)
21380   IF Rl<>0 AND Rt<>0 THEN
21385    ALLOCATE Dummy44(1:Rl,1:Et),Dummy45(1:Ilo,1:Rl),Dummy46(1:Ilo,1:Rl)
21390    MAT Dummy44= TRN(Qer)
21395    MAT Dummy45= Dummy43*Qrr
21400    MAT Dummy46= Dummy45*Glink
21405    MAT Dummy42= Dummy46*Dummy44
21410    DEALLOCATE Dummy44(*),Dummy45(*),Dummy46(*)
21415   END IF
21420   MAT Dummy47= TRN(Qeio)
21425   MAT Xx7= Dummy47-Dummy42
21430   FOR I=1 TO Ilo
21435    FOR J=1 TO Et
21440     Dummy40(I+Eto,J)=Xx7(I,J)
21445    NEXT J
21450   NEXT I
21455   DEALLOCATE Dummy47(*),Dummy42(*)
21460  END IF
21465  IF Rt<>0 AND Ilo<>0 THEN DEALLOCATE Dummy43(*)
21470  IF (Eto<>0 AND Cl<>0 AND Ct<>0) OR (Ilo<>0 AND It<>0 AND Ll<>0) THEN
21475   MAT C_dt= Dummy20*A
21480   MAT D_dt= Dummy20*B
21485  END IF
21490  IF (Eto<>0 AND (Ll<>0 OR (Ct<>0 AND Rl<>0))) OR (Ilo<>0 AND (Ct<>0 OR (Ll
<>0 AND Rt<>0))) THEN MAT C_dt= C_dt+Dummy30
21495  IF (Eto<>0 AND (Ll<>0 OR Rl<>0)) OR (Ilo<>0 AND (Et<>0 OR Rt<>0)) THEN MA
T D_dt= D_dt+Dummy40
21500  IF (Eto<>0 AND (Ll<>0 OR (Ct<>0 AND (Cl<>0 OR Rl<>0)))) OR (Ilo<>0 AND (C
t<>0 OR (Ll<>0 AND (It<>0 OR Rt<>0)))) THEN
21505   ALLOCATE Dummy49(1:Eto+Ilo,1:Et+Il)
21510   MAT Dummy49= C_dt*B2
21515   IF (Eto<>0 AND (Il<>0 OR Rl<>0 OR (Cl<>0 AND Ct<>0))) OR (Ilo<>0 AND (Et
<>0 OR Rt<>0 OR (It<>0 AND Ll<>0))) THEN
```

```
21520    IF (Et<>0 AND Ct<>0 AND Cl<>0) OR (Il AND Ll<>0 AND It<>0) THEN
21525     MAT D_dt= D_dt+Dummy49
21530    END IF
21535   END IF
21540   DEALLOCATE Dummy49(*)
21545  END IF
21550! ******************************************************
21555! * ROUTINE SOLVES THE OUTPUTS IF NO CAPACITOR LOOPS *
21560! *      AND NO VOLTAGE CUT-SETS WERE FOUND           *
21565! ******************************************************
21570 !
21575  ALLOCATE State(1:Ct+Ll,1:1),Dummy50(1:Ct+Ll,1:1),Result(1:Eto+Ilo,1:Limit
),Output(1:Eto+Ilo,1:1)
21580  IF Et<>Eto OR Il<>Ilo THEN
21585   ALLOCATE Dummy51(1:Ct+Ll,1:1)
21590   ALLOCATE Input(1:Et+Il,1:1)
21595   FOR I=1 TO Et
21600    Input(I,1)=Elemvalue(I)
21605   NEXT I
21610   FOR I=1 TO Il
21615    Input(I+Et,1)=Elemvalue(Et+Ct+Rt+It+Cl+Rl+Ll+I)
21620   NEXT I
21625   MAT Dummy51= B2*Input
21630   MAT State= State+Dummy51
21635   DEALLOCATE Dummy51(*)
21640  END IF
21645  ALLOCATE Dummy51(1:Eto+Ilo,1:1)
21650  FOR I=1 TO Limit
21655   IF Time_sources<>0 THEN
21660    FOR J=1 TO Time_sources
21665     Input(Control_var(J),1)=Control(J,I)*Voltage
21670    NEXT J
21675   END IF
21680   IF (Eto<>0 AND (Ll<>0 OR (Ct<>0 AND (Cl<>0 OR Rl<>0)))) OR (Ilo<>0 AND (
Ct<>0 OR (Ll<>0 AND (It<>0 OR Rt<>0)))) THEN
21685    MAT Output= C_dt*State
21690   ELSE
21695    MAT Output= (0)
21700   END IF
21705   IF ((Eto<>0 AND (Il<>0 OR Rl<>0 OR (Cl<>0 AND Ct<>0))) OR (Ilo<>0 AND (E
t<>0 OR Rt<>0 OR (It<>0 AND Ll<>0)))) THEN
21710    IF Eto<>Et OR Ilo<>Il THEN
21715     MAT Dummy51= D_dt*Input
21720     MAT Output= Output+Dummy51
21725    END IF
21730   END IF
21735   FOR J=1 TO Eto+Ilo
21740    Result(J,I)=Output(J,1)
21745   NEXT J
21750   MAT Dummy50= A_dt*State
21755   MAT State= Dummy50
21760   IF Et<>Eto OR Il<>Ilo THEN
21765    MAT Dummy50= B_dt*Input
```

```
21770    MAT State= State+Dummy50
21775   END IF
21780  NEXT I
21785  U2=TIMEDATE
21790  Time=U2-U1
21795  PRINT "TIME=";Time
21800  GOTO 22030
21805! ********************************************
21810! * OUTPUT SOLVING ROUTINE IF A CAPACITOR LOOP*
21815! * OR AN INDUCTOR CUT-SET WAS FOUND          *
21820! ********************************************
21825  ALLOCATE State(1:Ct+Ll,1:1),Dummy50(1:Ct+Ll,1:1),Result(1:Ct+Ll,1:Limit),
Output(1:Ct+Ll,1:1)
21830  IF Et<>0 OR Il<>0 THEN
21835   ALLOCATE Dummy51(1:Ct+Ll,1:1)
21840   ALLOCATE Input(1:Et+Il,1:1)
21845   FOR I=1 TO Et
21850    Input(I,1)=Elemvalue(I)
21855   NEXT I
21860   FOR I=1 TO Il
21865    Input(I+Et,1)=Elemvalue(Et+Ct+Rt+Lt+Cl+Rl+Ll+I)
21870   NEXT I
21875   MAT Dummy51= B2*Input
21880   MAT State= State+Dummy51
21885   DEALLOCATE Dummy51(*)
21890  END IF
21895  ALLOCATE Dummy51(1:Ct+Ll,1:1)
21900  FOR I=1 TO Limit
21905   IF Time_sources<>0 THEN
21910    FOR J=1 TO Time_sources
21915     Input(Control_var(J),1)=Control(J,I)*Voltage
21920    NEXT J
21925   END IF
21930   MAT Output= C_dt*State
21935   IF 0<>Et OR 0<>Il THEN
21940    MAT Dummy51= D_dt*Input
21945    MAT Output= Output+Dummy51
21950   END IF
21955   FOR J=1 TO Ct+Ll
21960    Result(J,I)=Output(J,1)
21965   NEXT J
21970   MAT Dummy50= A_dt*State
21975   MAT State= Dummy50
21980   IF Et<>0 OR Il<>0 THEN
21985    MAT Dummy50= B_dt*Input
21990    MAT State= State+Dummy50
21995   END IF
22000  NEXT I
22005  U2=TIMEDATE
22010  Time=U2-U1
22015  PRINT "TIME=";Time
22020! END OF TIME DOMAIN ROUTINE
22025  !
```

```
22030 !*************************************************************
22035 !*                    PLOTTING ROUTINE                       *
22040 !*************************************************************
22045 !
22050 Wave_no=Eto+Ilo
22055 IF Wave_no=0 THEN Wave_no=Ct+Ll
22060 Plotting: !
22065  OFF KEY
22070  OUTPUT 2 USING "#,B";255,75
22075  Plotflag=0                          ! INITIALIZE HORIZONTAL PLOTTING
22080  FOR N=1 TO 10 STEP 1                ! LOOP TO INITIALIZE
22085   Plot=0                             ! ALL PLOTTING VARIABLES
22090   Wind=0
22095   Axe(N)=0
22100   Manual(N)=0
22105   Imax(N)=0
22110   Imin(N)=0
22115   X_off(N)=0
22120   Y_off(N)=0
22125   X_scale(N)=1
22130   Y_scale(N)=1
22135   Buf(N)=0
22140  NEXT N
22145   !
22150 Adjustments: !
22155 ! ON KEY 10 GOTO PLOT1
22160  ON KEY 11 GOTO Plot2
22165  ON KEY 12 GOTO Plot3
22170  ON KEY 13 GOTO Plot4
22175  ON KEY 14 GOTO Plot5
22180  ON KEY 15 GOTO Plot6
22185  ON KEY 16 GOTO Plot7
22190  ON KEY 17 GOTO Plot8
22195  ON KEY 18 GOTO Plot9
22200  ON KEY 19 GOTO Plot10
22205  ON KEY 0 LABEL "WAVE / EDIT" GOTO Wave_change
22210  ON KEY 10 GOTO Edit_program
22215  ON KEY 1 LABEL "X_offset   / 1" GOTO X_offset
22220  ON KEY 2 LABEL "Y_offset   / 2" GOTO Y_offset
22225  ON KEY 6 LABEL "X_scale    / 6" GOTO X_scale
22230  ON KEY 7 LABEL "Y_scale    / 7" GOTO Y_scale
22235  ON KEY 5 LABEL "WINDOW     / 5" GOTO Window_change
22240  ON KEY 4 LABEL "EXIT       / 4" GOTO Fin
22245  ON KEY 9 LABEL "HARDCOPY   / 9" GOTO Hardcopy
22250  ON KEY 3 LABEL "VER<>HOR   / 3" GOTO Ver_hor
22255  ON KEY 8 LABEL "EFFECTS    / 8" GOTO Effects
22260  GOTO Adjustments
22265 !*************************************************************
22270 Effects:                          ! SELECTION OF USER SPECIAL PLOTTING EFFECTS
22275  IF Buf(Wind)=0 THEN GOTO Adjustments
22280  OFF KEY
22285  ON KEY 0 LABEL "MANUAL X-AXIS" GOTO X_axis
22290  GOTO 22285
```

```
22295 X_axis:                  ! OVER-RIDE AUTO X-AXIS
22300  OFF KEY
22305  PRINT CHR$(12)
22310  GOSUB Wavepurge         ! DELETE WAVE IN EXISTING WINDOW
22315  GRAPHICS OFF
22320  PRINT "The Y plane intersection of the X-AXIS is ";Axe(Wind)
22325  INPUT "Input the desired Y plane intersection of the  X-AXIS",Z
22330  Axe(Wind)=Z             ! NEW INTERSECTION
22335  Manual(Wind)=1          ! MANUAL X-AXIS FLAG SET
22340  PRINT CHR$(12)
22345  GLOAD Graph1(*)         ! SAVE GRAPHICS DISPLAY EXISTING WINDOW
22350  GRAPHICS ON
22355  Adjust=1
22360  N=Wind
22365  GOTO 22545              ! GO TO DRAW NEW WAVEFORM
22370 !*****************************************************
22375 Fin:GCLEAR               ! EXIT FROM PLOTTING ROUTINE
22380  OUTPUT 2 USING "#,B";255,75
22385  PRINT CHR$(12)
22390  GOTO Re_start
22395 !*****************************************************
22400 Plot1:Plot=1             ! DIVIDING THE SCREEN INTO
22405  GOTO 22495              ! THE SELECTED NUMBER OF PLOTS
22410 Plot2:Plot=2
22415  GOTO 22495
22420 Plot3:Plot=3
22425  GOTO 22495
22430 Plot4:Plot=4
22435  GOTO 22495
22440 Plot5:Plot=5
22445  GOTO 22495
22450 Plot6:Plot=6
22455  GOTO 22495
22460 Plot7:Plot=7
22465  GOTO 22495
22470 Plot8:Plot=8
22475  GOTO 22495
22480 Plot9:Plot=9
22485  GOTO 22495
22490 Plot10:Plot=10
22495  OFF KEY
22500  GINIT
22505  OUTPUT 2 USING "#,B";255,75
22510  GRAPHICS ON
22515!
22520! PLOTTING THE WAVEFORMS SELECTED
22525!
22530  FOR N=Plot TO 2 STEP -1      ! PLOTTING OF THE WAVEFORMS
22535   PEN 1
22540   IF Buf(N)=0 THEN GOTO 22970
22545   I=Buf(N)
22550   Aaa=FROUND(X_off(N)*(Limit-1),0)
22555   Bbb=FROUND((Limit-Aaa)*X_scale(N)+Aaa,0)
```

```
22560   IF Aaa=Bbb THEN Bbb=Aaa+1
22565   Ccc=Lmin(N)+(Lmax(N)-Lmin(N))*Y_off(N)
22570   Ddd=(Lmax(N)-Ccc)*Y_scale(N)+Ccc
22575   IF Ccc=0 AND Ddd=0 THEN
22580    Ccc=-1
22585    Ddd=1
22590   END IF
22595   IF Aaa=0 AND Bbb=0 THEN
22600    OUTPUT 2 USING "#,B";255,75
22605    PRINT "WAVEFORM ERROR— 0 POINTS"
22610    BEEP 555,2
22615    WAIT 3
22620    GOTO Fin
22625   END IF
22630   IF Manual(N)=1 THEN                ! IS THE MANUAL X-AXIS OVER-RIDE SET?
22635    IF Axe(N)>Ddd THEN Ddd=Axe(N)      ! YES
22640    IF Axe(N)<Ccc THEN Ccc=Axe(N)
22645    GOTO 22665
22650   END IF
22655   Axe(N)=(Ddd+Ccc)/2                 ! NO
22660   IF 0<=Ddd AND 0>=Ccc THEN Axe(N)=0
22665   DEG
22670   CSIZE 2.4
22675   CLIP ON
22680   SELECT Plotflag                    ! HORIZONTAL(0) OR VERTICAL(1) PLOTTING
22685   CASE 0
22690    VIEWPORT 10,125,(100*(N-1)+20)/Plot,(100*N-20)/Plot
22695    LDIR 0
22700    WINDOW Aaa,Bbb,Ccc,Ddd
22705    AXES (Bbb-Aaa)/10,(Ddd-Ccc)/5,Aaa,Axe(N)
22710    MOVE Aaa,Result(I,Aaa+1)
22715   CASE 1
22720    VIEWPORT 131-(131*N-23)/Plot,131-(131*(N-1)+23)/Plot,10,95
22725    LDIR 90
22730    WINDOW Ddd,Ccc,Aaa,Bbb
22735    AXES (Ddd-Ccc)/5,(Bbb-Aaa)/10,Axe(N),Aaa
22740    MOVE Result(I,Aaa+1),Aaa
22745   END SELECT
22750   LORG 1                             ! PLOTTING OF THE WAVEFORM
22755   Last_point=Bbb
22760   IF Last_point>Limit-1 THEN Last_point=Limit-1
22765   FOR J=Aaa+2 TO Last_point          ! WINDOW
22770    IF Plotflag=0 THEN
22775     DRAW J,Result(I,J)
22780    ELSE
22785     DRAW Result(I,J),J
22790    END IF
22795   NEXT J
22800   CLIP OFF
22805   LORG 1
22810   IF Plotflag=1 THEN
22815    MOVE Ddd,Aaa
22820   ELSE
```

```
22825    MOVE Aaa,Ddd
22830   END IF
22835   LABEL Wave$(Buf(N))[1,6]              ! THE WAVE LABEL
22840   FOR J=(Bbb-Aaa)/10+Aaa TO Bbb STEP (Bbb-Aaa)/5
22845    IF Plotflag=0 THEN
22850     MOVE J,Axe(N)
22855     LORG 6
22860    ELSE
22865     MOVE Axe(N),J
22870     LORG 6
22875    END IF
22880    LABEL DROUND(J*Dt,3)
22885   NEXT J
22890   FOR J=Ccc TO Ddd STEP (Ddd-Ccc)/5   ! Y-AXIS NUMBERING
22895    IF Plotflag=0 THEN
22900     MOVE Aaa,J
22905     LORG 8
22910    ELSE
22915     MOVE J,Aaa
22920     LORG 8
22925    END IF
22930    IF ABS(J)<.0001 THEN J=0
22935    LABEL DROUND(J,3)
22940   NEXT J
22945   CLIP ON
22950   IF Adjust=1 THEN
22955    Adjust=0
22960    GOTO Adjustments
22965   END IF
22970  NEXT N
22975!
22980  IF Hardcopy=1 THEN
22985   Hardcopy=0
22990   PEN Up
22995   GOTO Adjustments
23000  END IF
23005  Wind=2
23010  SELECT Plotflag
23015  CASE 0
23020   VIEWPORT 0,131,100/Plot*(Wind-1),100/Plot*Wind
23025  CASE 1
23030   VIEWPORT 131-131*Wind/Plot,131-131*(Wind-1)/Plot,0,100
23035  END SELECT
23040  FRAME
23045  GOTO Adjustments
23050 !*****************************************************
23055 Ver_hor:                  ! CHANGE OUR VERTICAL PICTURE TO HORIZONTAL
23060  OFF KEY                  ! OR VICE-VERSA
23065  IF Plotflag=1 THEN
23070   Plotflag=0
23075  ELSE
23080   Plotflag=1
23085  END IF
```

```
23090  IF Wind=0 THEN
23095   GOTO Adjustments
23100  ELSE
23105   GOTO 22500
23110  END IF
23115 !*******************************************************
23120 Wave_change:            ! KILL WAVE IN EXISTING WINDOW AND SELECT
23125  OFF KEY                ! REPLACEMENT WAVE
23130  IF Wind=0 THEN GOTO Adjustments
23135  GOSUB Wavepurge        ! KILL EXISTING WAVE
23140  Buf(Wind)=0
23145  X_off(Wind)=0          ! RE-INITIALIZING VARIABLES FOR NEW WAVE
23150  Y_off(Wind)=0
23155  X_scale(Wind)=1
23160  Y_scale(Wind)=1
23165  Manual(Wind)=0
23170  GOSUB Prepare          ! SELECTION OF THE NEW WAVE
23175  Buf(Wind)=I
23180  N=Wind
23185  Imin(N)=Result(I,1)    ! MAX and MIN POINTS OF THE WAVE
23190  Imax(N)=Result(I,1)
23195  FOR J=1 TO Limit
23200   IF Imax(N)<Result(I,J) THEN Imax(N)=Result(I,J)
23205   IF Imin(N)>Result(I,J) THEN Imin(N)=Result(I,J)
23210  NEXT J
23215  IF Imax(N)=Imin(N) THEN
23220   Imin(N)=Imin(N)-ABS(Imin(N)*.1)
23225   Imax(N)=Imax(N)+ABS(Imax(N)*.1)
23230  END IF
23235  GRAPHICS ON
23240  GLOAD Graph1(*)
23245  Adjust=1
23250  GOTO 22545
23255 !*******************************************************
23260 Window_change:          ! CHANGING THE WINDOW AND VIEWPORT
23265  PLOTTER IS 3,"INTERNAL"
23270  OFF KEY
23275  IF Wind=0 THEN GOTO Adjustments
23280  IF Plotflag=0 THEN
23285   VIEWPORT 0,131,100*(Wind-1)/Plot,100*Wind/Plot
23290  ELSE
23295   VIEWPORT 131-131*(Wind)/Plot,131-131*(Wind-1)/Plot,0,100
23300  END IF
23305  PEN -1                       ! KILL EXISTING FRAME
23310  FRAME
23315  PEN 1                        ! CREATE NEW FRAME
23320  Wind=Wind+1
23325  IF Wind>Plot THEN Wind=2
23330  IF Plotflag=0 THEN
23335   VIEWPORT 0,131,100*(Wind-1)/Plot,100*Wind/Plot
23340  ELSE
23345   VIEWPORT 131-131*(Wind)/Plot,131-131*(Wind-1)/Plot,0,100
23350  END IF
```

```
23355  FRAME
23360  GOTO Adjustments
23365 !*************************************************************************
23370 Hardcopy:  !   ROUTINE TO SET VARIABLES FOR THE HARDCOPY
23375  FOR Fgh=2 TO Plot
23380   IF Buf(Fgh)<>0 THEN GOTO 23395
23385  NEXT Fgh
23390  GOTO Adjustments
23395  PLOTTER IS 705,"HPGL"
23400  Hardcopy=1
23405  GOTO 22530
23410 !*************************************************************************
23415 X_scale: ! ADJUST THE X SCALE
23420  IF Buf(Wind)=0 THEN GOTO Adjustments
23425  OFF KEY
23430  GOSUB Wavepurge
23435  GRAPHICS OFF
23440  PRINT "The  X  scale is now ";1/X_scale(Wind)
23445  INPUT "Input the new desired X scale",Z
23450  IF Z=0 THEN                    ! CHECK TO SEE IF USER INPUT IS WITHIN BOUNDS
23455   PRINT CHR$(12)
23460   PRINT "The value ";Z;" is not acceptable"
23465   BEEP
23470   WAIT 2
23475   PRINT CHR$(12)
23480   GOTO 23440
23485  END IF
23490  GLOAD Graph1(*)
23495  GRAPHICS ON
23500  X_scale(Wind)=1/Z
23505  Adjust=1
23510  PRINT CHR$(12)
23515  N=Wind
23520  GOTO 22545!GO TO START OF THE PLOTTING ROUTINE TO REPLOT WITH NEW SCALE
23525  !*********************************************************
23530 Y_scale:                                  ! ADJUST THE Y SCALE
23535  IF Buf(Wind)=0 THEN GOTO Adjustments
23540  OFF KEY
23545  GOSUB Wavepurge
23550  GRAPHICS OFF
23555  PRINT "The  Y  scale is now ";1/Y_scale(Wind)
23560  INPUT "Input the desired scale",Z
23565  IF Z=0 THEN            ! CHECK TO SEE IF USER VALUE IS WITHIN BOUNDS
23570   PRINT CHR$(12)
23575   PRINT "The value ";Z;" is not acceptable."
23580   BEEP
23585   WAIT 2
23590   PRINT CHR$(12)
23595   GOTO 23555
23600  END IF
23605  Y_scale(Wind)=1/Z
23610  GLOAD Graph1(*)
23615  GRAPHICS ON
```

```
23620  PRINT CHR$(12)
23625  Adjust=1
23630  N=Wind
23635  GOTO 22545!GO TO START OF THE PLOTTING ROUTINE TO REPLOT WITH NEW SCALE
23640  !*****************************************************************
23645 X_offset:                          !  ADJUST THE X OFFSET
23650  IF Buf(Wind)=0 THEN GOTO Adjustments
23655  OFF KEY
23660  GOSUB Wavepurge
23665  GRAPHICS OFF
23670  PRINT "The X offset is now ";X_off(Wind)
23675  INPUT "Input the new desired X offset, 0 <= X offset < 1.",Z
23680  IF Z<0 OR Z>=1 THEN  ! CHECK TO SEE IF USER INPUT IS WITHIN BOUNDS
23685   PRINT CHR$(12)
23690   PRINT "The value ";Z;" is not acceptable."
23695   PRINT "                                   0 <= X offset < 1"
23700   BEEP
23705   WAIT 3
23710   PRINT CHR$(12)
23715   GOTO 23670
23720  END IF
23725  X_off(Wind)=Z
23730  GLOAD Graph1(*)
23735  GRAPHICS ON
23740  PRINT CHR$(12)
23745  Adjust=1
23750  N=Wind
23755  GOTO 22545!GO TO START OF THE PLOTTING ROUTINE TO REPLOT WITH NEW OFFSET
23760  !*****************************************************************
23765 Y_offset:                          ! ADJUST THE Y OFFSET
23770  IF Buf(Wind)=0 THEN GOTO Adjustments
23775  OFF KEY
23780  GOSUB Wavepurge
23785  GRAPHICS OFF
23790  PRINT "The Y offset is now";Y_off(Wind)
23795  PRINT "Input the new desired Y offset,     0  <= Y offset <=  1 "
23800  INPUT Z
23805  IF Z<=0 OR Z>=1 THEN    !CHECK TO SEE IF THE USER INPUT IS WITHIN BOUNDS
23810   PRINT CHR$(12)
23815   PRINT "The value";Z;"is not acceptable"
23820   PRINT "                                  0  <= Y offset <=   1"
23825   BEEP
23830   WAIT 3
23835   PRINT CHR$(12)
23840   GOTO 23790
23845  END IF
23850  Y_off(Wind)=Z
23855  PRINT CHR$(12)
23860  GLOAD Graph1(*)
23865  GRAPHICS ON
23870  Adjust=1
23875  N=Wind
23880  GOTO 22545!GO TO START OF THE PLOTTING ROUTINE TO REPLOT WITH NEW OFFSET
```

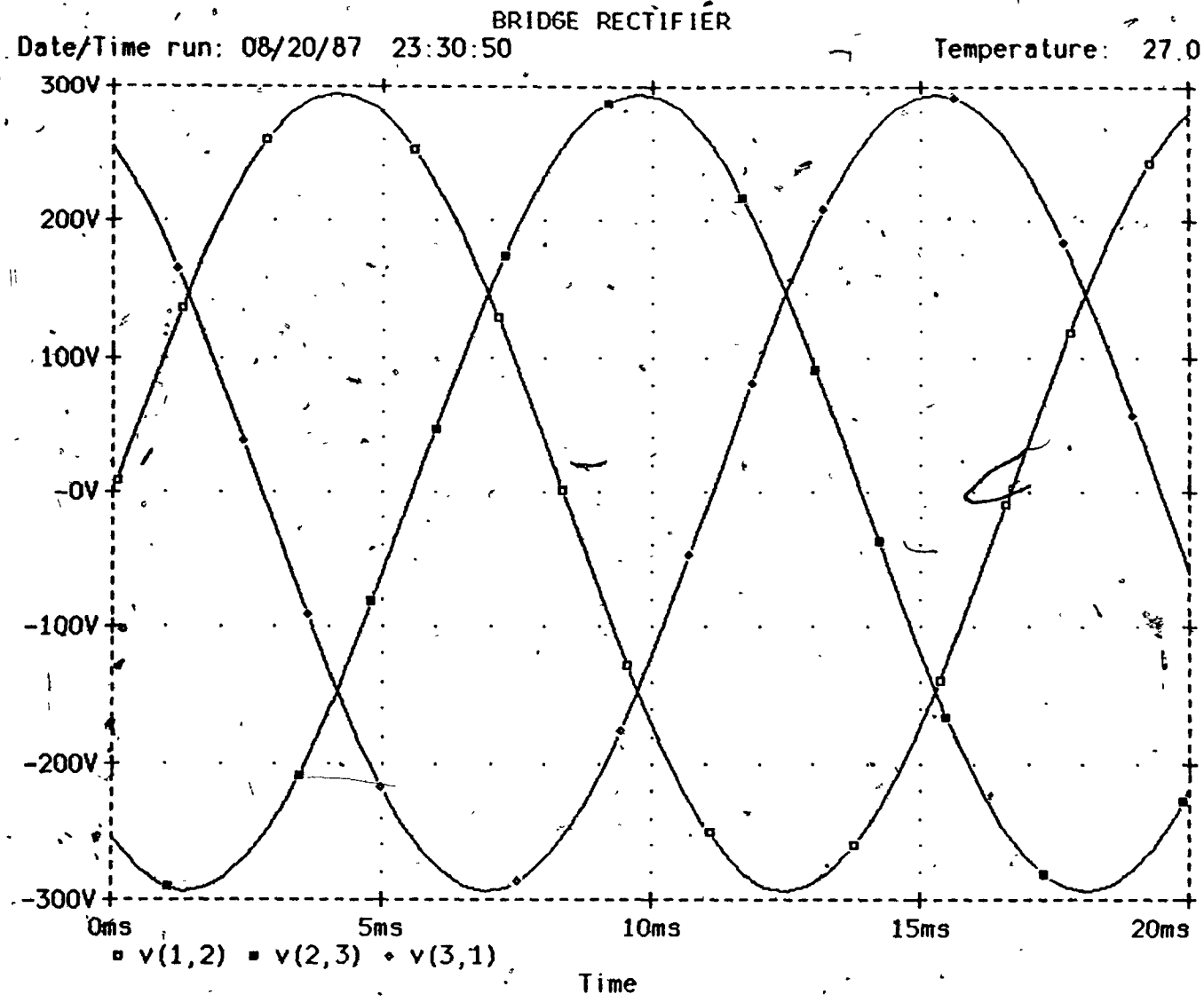
```
23885 !*************************************************************************
23890 Wavepurge: !
23895  AREA PEN -1
23900  IF Plotflag=0 THEN
23905   VIEWPORT 0,131,100*(Wind-1)/Plot,100*Wind/Plot
23910   WINDOW 0,131,100*(Wind-1)/Plot,100*Wind/Plot
23915   MOVE 0,100*(Wind-1)/Plot
23920   RECTANGLE 131,100/Plot,FILL,EDGE
23925   IF Spectrum=1 THEN RECTANGLE 131,50,FILL
23930  ELSE
23935   VIEWPORT 131-131*(Wind)/Plot,131-131*(Wind-1)/Plot,0,100
23940   WINDOW 131-131*(Wind)/Plot,131-131*(Wind-1)/Plot,0,100
23945   MOVE 131-131/Plot*Wind,0
23950   RECTANGLE 131/Plot,100,FILL,EDGE
23955  END IF
23960  GSTORE Graph1(*)
23965  RETURN
23970 Buffer_selec:      ! THIS ROUTINE DISPLAYS THE
23975  J=0               ! BUFFER SELECTION ON THE
23980  Max=Wave_no-1     ! SCREEN AND LETS THE USER CHOOSE
23985  DEG               ! THE DESIRED BUFFER
23990  GINIT
23995  GRAPHICS ON
24000  WINDOW -100,100,0,200
24005  MOVE -10,180
24010  CSIZE (3.6)
24015  LABEL "Rotate knob to select buffer"
24020  MOVE -10,170
24025  LABEL "After selection press ENTER"
24030  CSIZE (-.1*(Max)+7)
24035  IF (-.1*(Max)+7)>5 THEN CSIZE (5)
24040  OUTPUT 2 USING "#,B";255,75
24045  FOR Lll=0 TO Max
24050   MOVE -50,J
24055   IF Max=0 THEN GOTO 24065
24060   J=J+180/(Max)
24065   LABEL B$(Lll+1)[1,6]
24070  NEXT Lll
24075  ON KNOB .2 GOTO Choice
24080 Choice:ON KBD GOTO Choice1
24085 Choice1:IF KBD$=CHR$(255)&"E" THEN GOTO Out
24090  PEN -1
24095  MOVE -55,J
24100  DRAW -60,J
24105  MOVE -56,J+1.5
24110  DRAW -55,J
24115  MOVE -56,J-1.5
24120  DRAW -55,J
24125  IF Max=0 THEN
24130   Lll=3
24135   GOTO 24160
24140  END IF
24145  Lll=(INT(KNOBX/10)*(180/Max)+Lll)
```

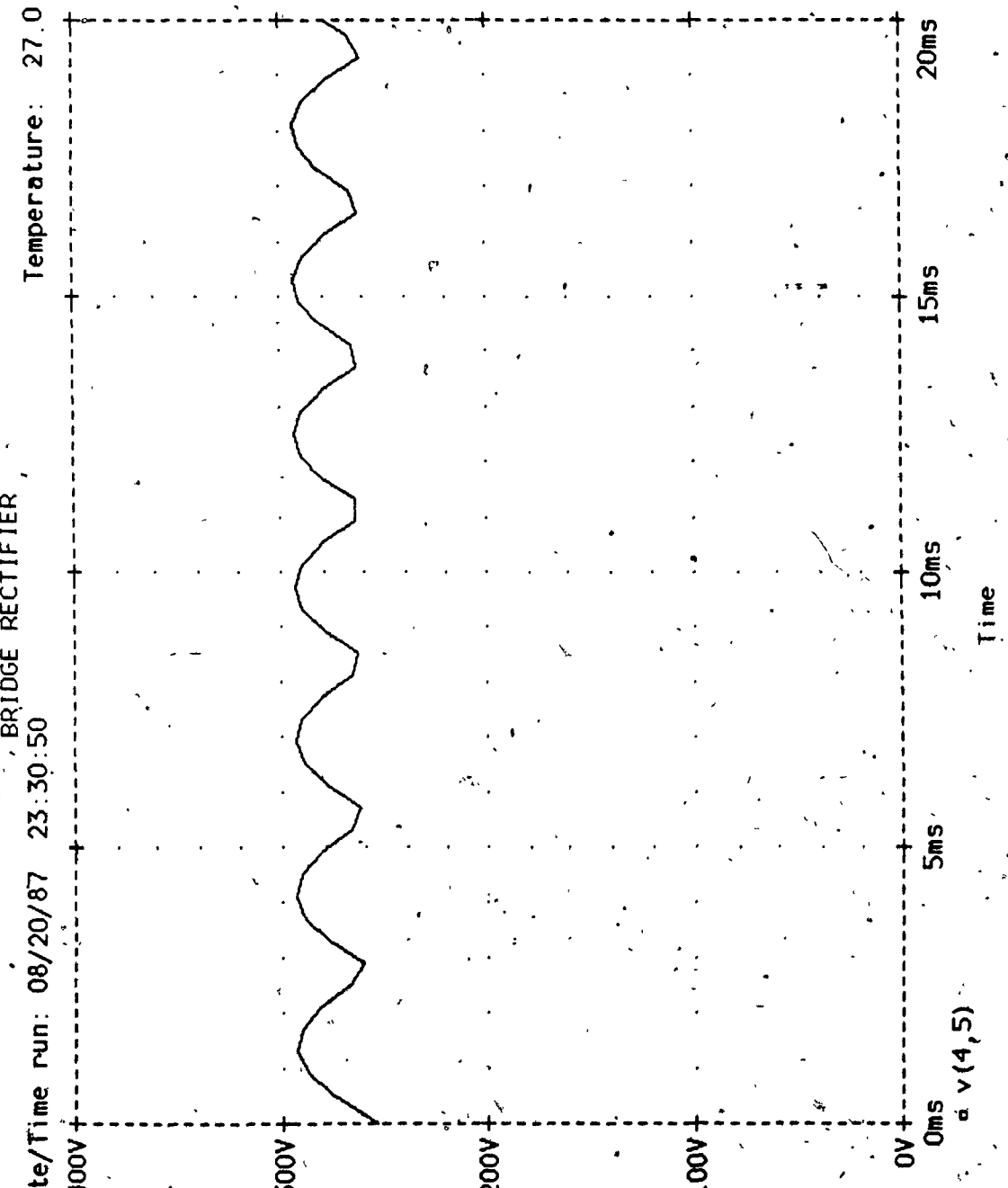
```
24150  IF Lll>183 THEN Lll=183
24155  IF Lll<3 THEN Lll=3
24160  J=Lll
24165  PEN 1
24170  MOVE -55,J
24175  DRAW -60,J
24180  MOVE -56,J-1.5
24185  DRAW -55,J
24190  MOVE -56,J+1.5
24195  DRAW -55,J
24200  WAIT .07
24205  GOTO Choice
24210 Out:        !
24215  GRAPHICS OFF
24220  OFF KBD
24225  OFF KNOB
24230  OUTPUT 2 USING "#,B";255,75
24235  PEN 1
24240 Re_start:!
24245  RETURN
24250   !*****************************************************************
24255 Prepare:            ! BUFFER_SELEC ROUTINE
24260  FOR I=1 TO Wave_no ! SET LABELS IN AN ARRAY
24265   B$(I)=Wave$(I)
24270  NEXT I
24275  GOSUB Buffer_selec
24280  Bd$=B$((J)*Max/180+1)  ! THIS IS THE SELECTED WAVEFORM
24285  FOR I=1 TO Wave_no
24290   IF Bd$=Wave$(I) THEN RETURN   ! FIND THE WAVEFORM SELECTED IN THE
24295  NEXT I                         ! ORIGINAL WAVEFORM LABEL ARRAY Wave$
24300! END OF PLOTTING ROUTINE
24305      !
24310! ****************************
24315! * END OF TREE-LINK PROGRAM  *
24320! ****************************
24325  RETURN
24330 End:GCLEAR
24335  OUTPUT 2;"LOAD KEY"&" X";
24340  END
```

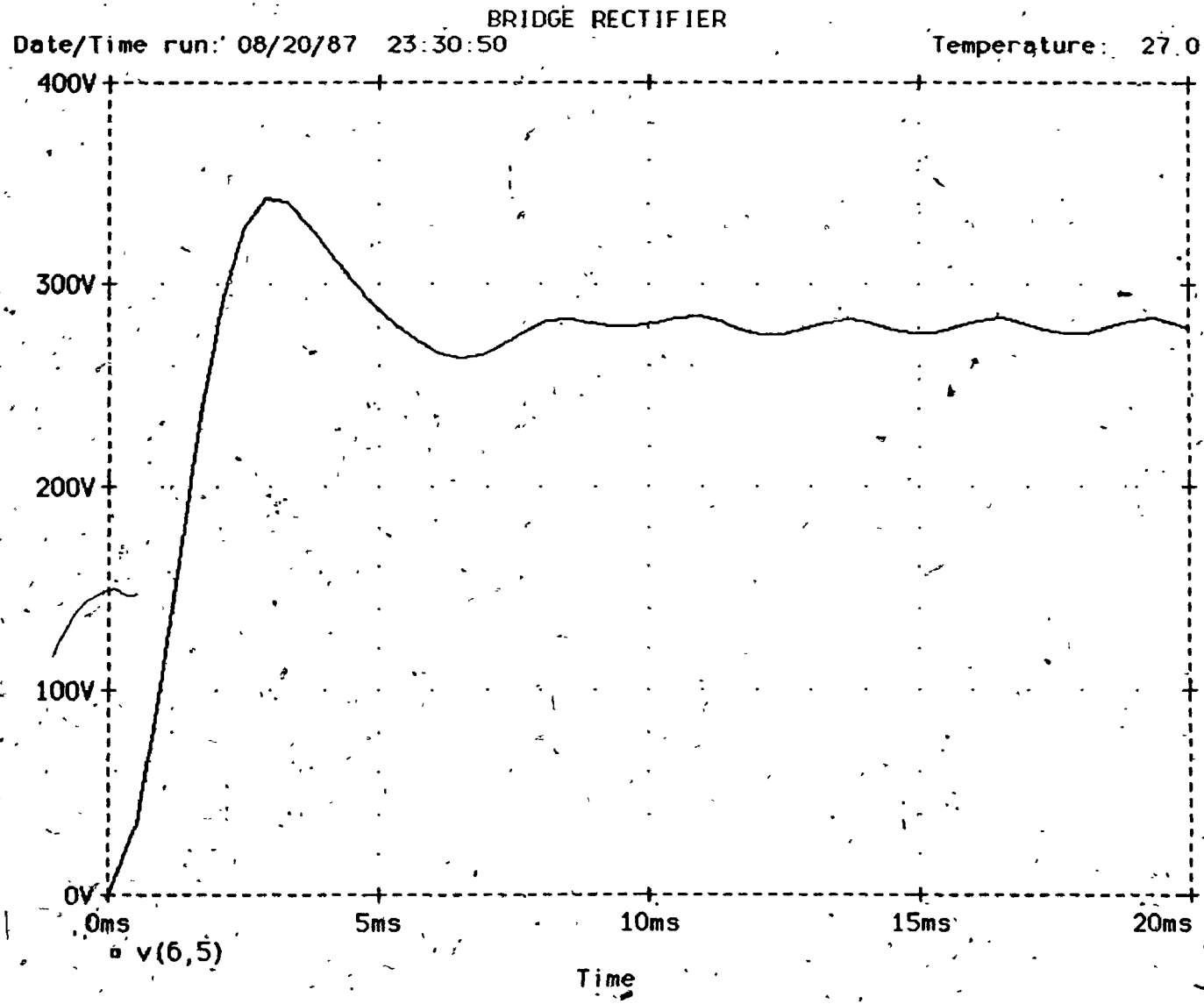
APPENDIX 3

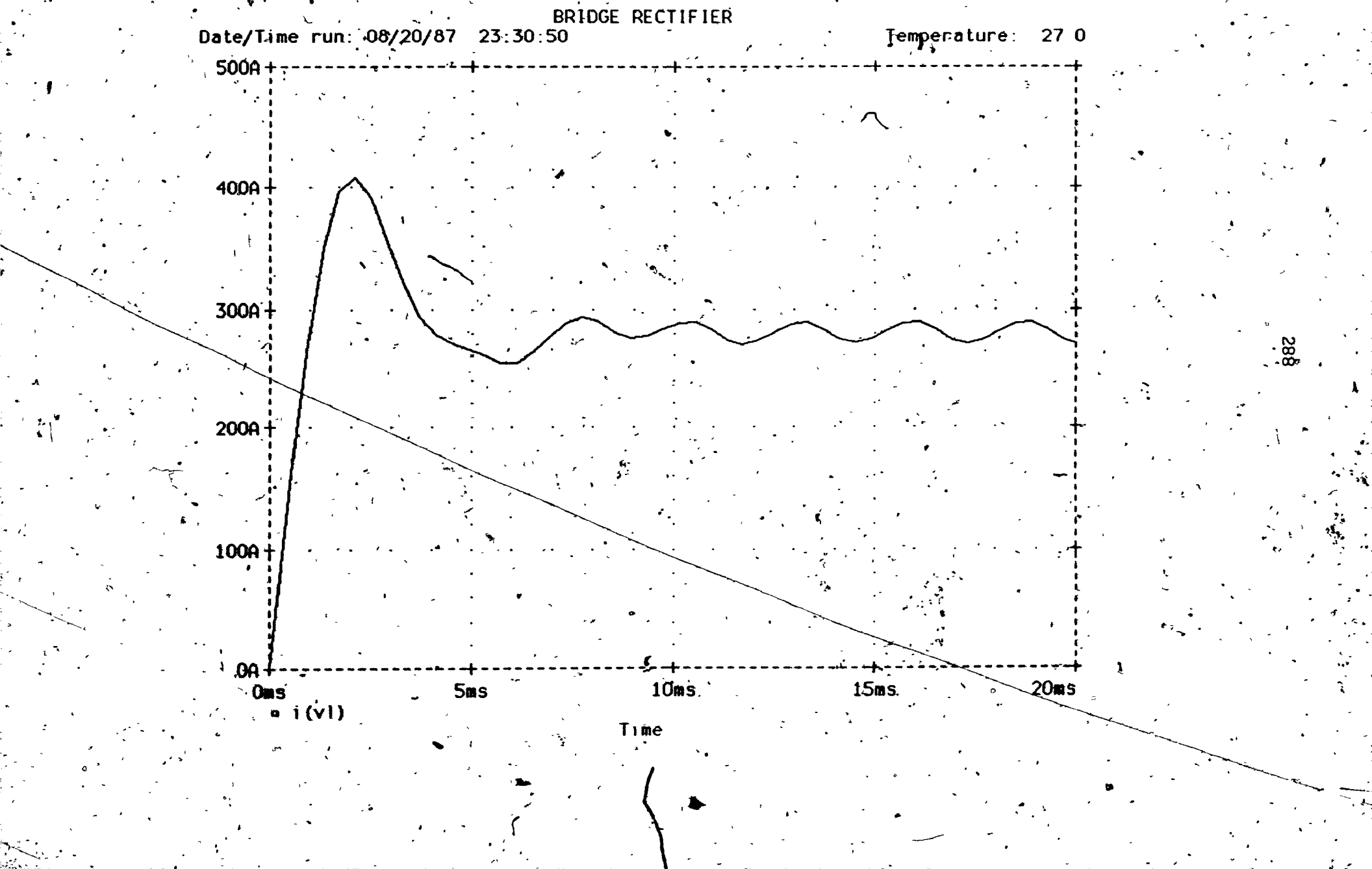
PSPICE OUTPUT

```
BRIDGE RECTIFIER
.OPTIONS NOMOD RELTOL=.15 LIMPTS=9000 ITL5=99000
*****************************************************
V1 1 0 SIN(0 169.8313 60HZ 0 0 -30)
V2 2 0 SIN(0 169.8313 60HZ 0 0 -150)
V3 3 0 SIN(0 169.8313 60HZ 0 0 +90)
VL 4 7 0
*****************************************************
.MODEL DM D(IS=.001)
*****************************************************
.SUBCKT DD 1 2 3
DP 1 2 DM
DN 3 1 DM
.ENDS
.SUBCKT LC 1 2 3
LF 1 2 .8MH IC=0
CF 2 3 1000UF IC=0
.ENDS
XD1 1 4 5 DD
XD2 2 4 5 DD
XD3 3 4 5 DD
XF 7 6 5 LC
RL 6 5 1
*************************
.TRAN .00001MS 20MS UIC
*************************
.PROBE
.END
```


BRIDGE RECTIFIER
Date/Time run: 08/20/87 23:30:50
Temperature: 27.0
300V
200V
100V
0V
-100V
-200V
-300V
0ms
5ms
10ms
15ms
20ms
v(1,2) v(2,3) v(3,1)
Time


Temperature: 27.0
Date/Time run: 08/20/87 23:30:50
BRIDGE RECTIFIER
400V
300V
200V
100V
0V
0ms
5ms
10ms
15ms
20ms
Time
v(4,5)


BRIDGE RECTIFIER
Date/Time run: 08/20/87 23:30:50
Temperature: 27.0
400V
300V
200V
100V
0V
0ms
5ms
10ms
15ms
20ms
v(6,5)
Time


BRIDGE RECTIFIER
Date/Time run: 08/20/87 23:30:50
Temperature: 27 0
500A
400A
300A
200A
100A
0A
0ms
5ms
10ms
15ms
20ms
i(V1)
Time