TYMSHARE MANUALS

ECAP

ELECTRONIC CIRCUIT ANALYSIS PROGRAM

NOVEMBER 1970

TYMSHARE, INC.
525 UNIVERSITY AVENUE, SUITE 220
PALO ALTO, CALIFORNIA 94301

70, TYMSHARE, INC., Litho in U.S.A.

PREFACE

This manual is intended to supplement any one of the many publications available on ECAP. Most of the emphasis is placed on the editing and file manipulation features unique to Tymshare's ECAP. Please refer to one of the references for material on how to use ECAP as a powerful design tool. A bibliography is provided in the Appendix.

TYMSHARE REFERENCE MANUAL
SYMBOL CONVENTIONS

The symbols used in this manual to indicate Carriage Return, Line Feed, and ALT MODE/ESCAPE are as follows:

Carriage Return: ↩

Line Feed: ↴

ALT MODE/ESCAPE: ⊕

Action At The Terminal

To indicate clearly what is typed by the computer and what is typed by the user, the portion typed by the user is underlined.

Price: $2.00

CONTENTS

Page

LIST OF ILLUSTRATIONS

Page

INTRODUCTION

WHY ECAP?

ECAP is a fast, efficient, convenient replacement for breadboard models of electronic circuits. Instead of actually wiring the hardware, an engineer can describe a circuit configuration to ECAP and ECAP will print the voltages, currents, and other parameters as requested. The major advantages of using ECAP rather than actually breadboarding the circuit are:

- ECAP is faster. A circuit description can be entered into the program much faster than the circuit can be wired. Furthermore, ECAP can immediately calculate parameters which would be extremely difficult to measure in a circuit.
- ECAP eliminates purchasing expensive circuit components which may not be adequate, and reduces the amount of circuit analysis equipment which must be bought and maintained.

WHY TYMSHARE ECAP?

The real value of ECAP depends on its being easy for the engineer to use. Recognizing this, Tymshare has harnessed the full power of the interactive Tymshare system to provide convenient communication with ECAP. The result includes these advantages:

- Immediate accessibility through a Tymshare terminal in your office or laboratory.
- Free form input allows the engineer to concentrate on the accuracy of the data he is entering rather than on the format.
- Circuit definitions can be stored on files always ready for use.
- Circuits can be corrected or modified after they have been entered into ECAP using the extensive editing capabilities built into ECAP.
- Standard circuit components can be retrieved from a file and relocated in a particular circuit to reduce both the amount of typing time required and the possibility of errors.
- ECAP can take commands from a file rather than the terminal, thus making the engineer's working hours more efficient.

THE STRUCTURE OF TYMSHARE ECAP

ECAP is called from the Tymshare EXECUTIVE simply by typing ECAP ↵. When ECAP is ready to receive a command from the terminal, a colon (:) will print. The colon indicates that ECAP is in the Control Mode and is ready to receive any Control Language command. The Control Language sets the input/output devices and provides certain utility functions as discussed in Section 1. The actual ECAP programs are run in the Analysis Mode. Therefore, after the controls have been set in the Control Mode, the program enters the Analysis Mode. All the options set in the Control Mode remain set until the user terminates the analysis, returns to the Control Mode, and resets the controls.

In the Analysis Mode, designated by an asterisk (*), ECAP will accept any Analysis Language command as discussed in Section 2. These commands are used to edit and list circuit descriptions. ECAP programs consist of Analysis Language commands and circuit descriptions.

There are actually three analysis modes: DC analysis, AC analysis, and transient analysis. All three of these modes will accept the same editing commands; however, the circuit descriptions (ECAP programs) differ among the analysis types. Sections 3, 4, and 5 give a brief summary of the formats for describing circuits in DC, AC, and transient analysis.

This manual is intended to be a supplement to one of the several publications available on ECAP[1]. While the Tymshare Control Language and Tymshare Analysis Language are described in detail, the standard ECAP programs are reviewed only briefly in Sections 3, 4, and 5. Extensive material on how ECAP can be used as a tool to solve circuit designers' problems can be found in the references.

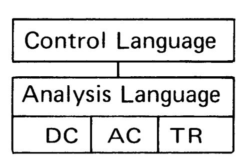

Figure 1 - Structure of ECAP

1 - See Bibliography in the Appendix.

SAMPLE PROBLEM

The procedure for solving a problem in Tymshare ECAP can be demonstrated easily through the following example. Figure 2 shows the circuit for a single stage common emitter amplifier. DC, AC, and transient analyses will be performed on this circuit.

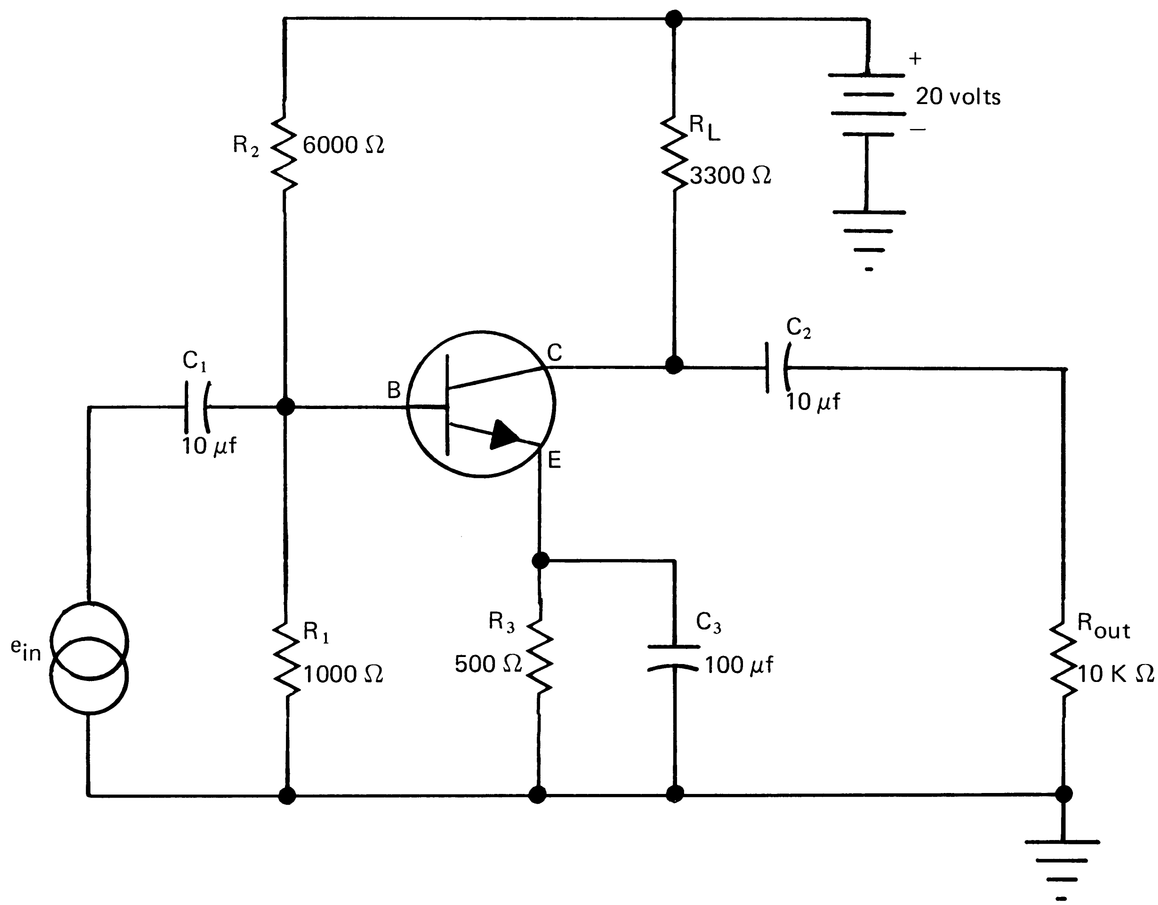

Figure 2 - A Single Stage Common Emitter Amplifier

The DC analysis is started by drawing the DC equivalent circuit as is shown in Figure 3 below. To analyze the performance of any electronic circuit with ECAP, all devices must be replaced with equivalent circuits composed of the circuit elements recognized by ECAP. It is assumed that the user has an introductory understanding of the synthesis of equivalent circuits for common electronic components; therefore, no attempt is made in this manual to present a comprehensive discussion of the subject.

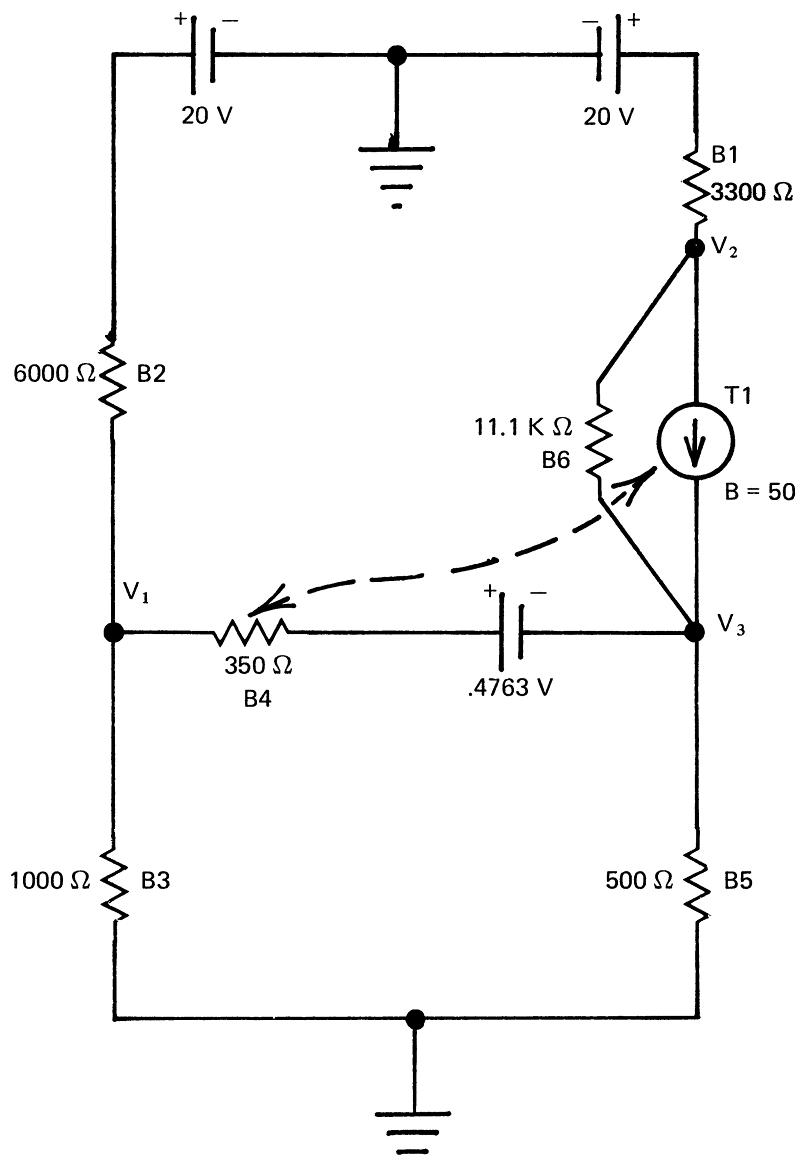

Figure 3 - DC Equivalent Circuit

Creating The Input

After the circuit has been reduced to the proper equivalent circuit shown in Figure 3, numbers are assigned to the nodes and branches of the circuit. The user now describes the equivalent circuit by writing a description of each branch as shown below.

```
        DC ANALYSIS
B1      N (0, 2), R=3300, E=20
B2      N (0, 1), R=6000, E=20
B3      N (0,1), R=1000
B4      N (1,3), R=350, E=-.4763
B5      N (3,0), R=500
B6      N (2,3), R=11.1E3
T1      B (4, 6), BETA=50
        PUNCH, MISCELLANEOUS, VOLTAGES, CURRENTS
```

Those statements which are preceded by the letter B describe a branch of the circuit and are referred to as the branch statements. In this example, Branch 1 connects Nodes 0 and 2 by a resistance of 3300 ohms and has a voltage of 20 volts.

The statement preceded by the letter T describes the current gain of the transistor from Branch 4 to Branch 6.

After the user entered the parameters of the circuit branches and the transistor, he typed the command PUNCH, MISCELLANEOUS, VOLTAGES, CURRENTS. PUNCH indicates that the specified output data (in this case, the MISCELLANEOUS solution data and the solution VOLTAGES and CURRENTS) shall be printed on the output file. The MISCELLANEOUS data includes the nodal impedance matrix and the nodal conductance matrix.

The user continues to describe the equivalent circuit for the DC analysis. He then creates the AC and transient equivalent circuits shown in Figures 4 and 5. He describes the AC and transient equivalent circuit branches in the same way he described the DC equivalent circuit branches. The next two pages show the AC and transient equivalent circuits.

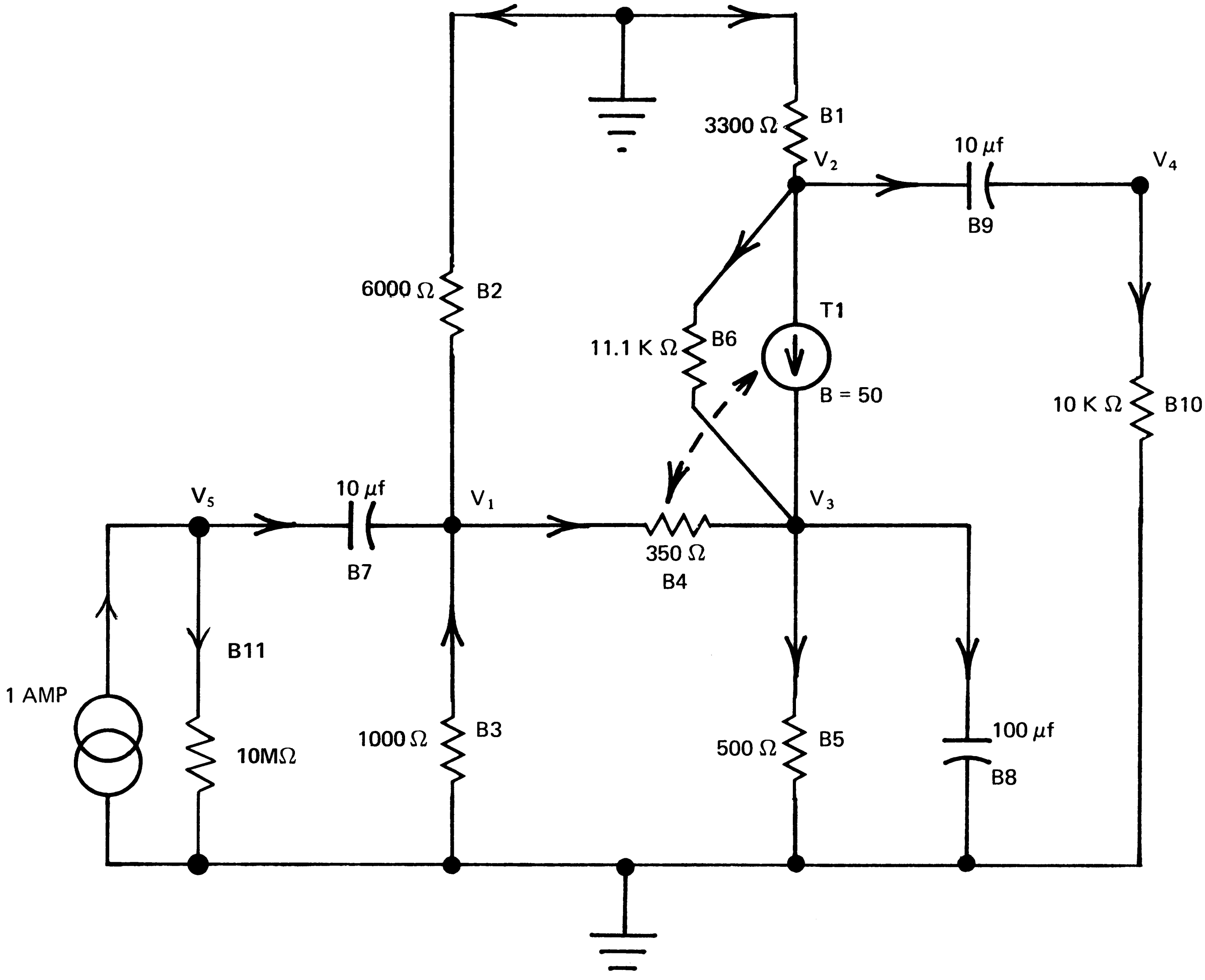

Figure 4 - AC Equivalent Circuit

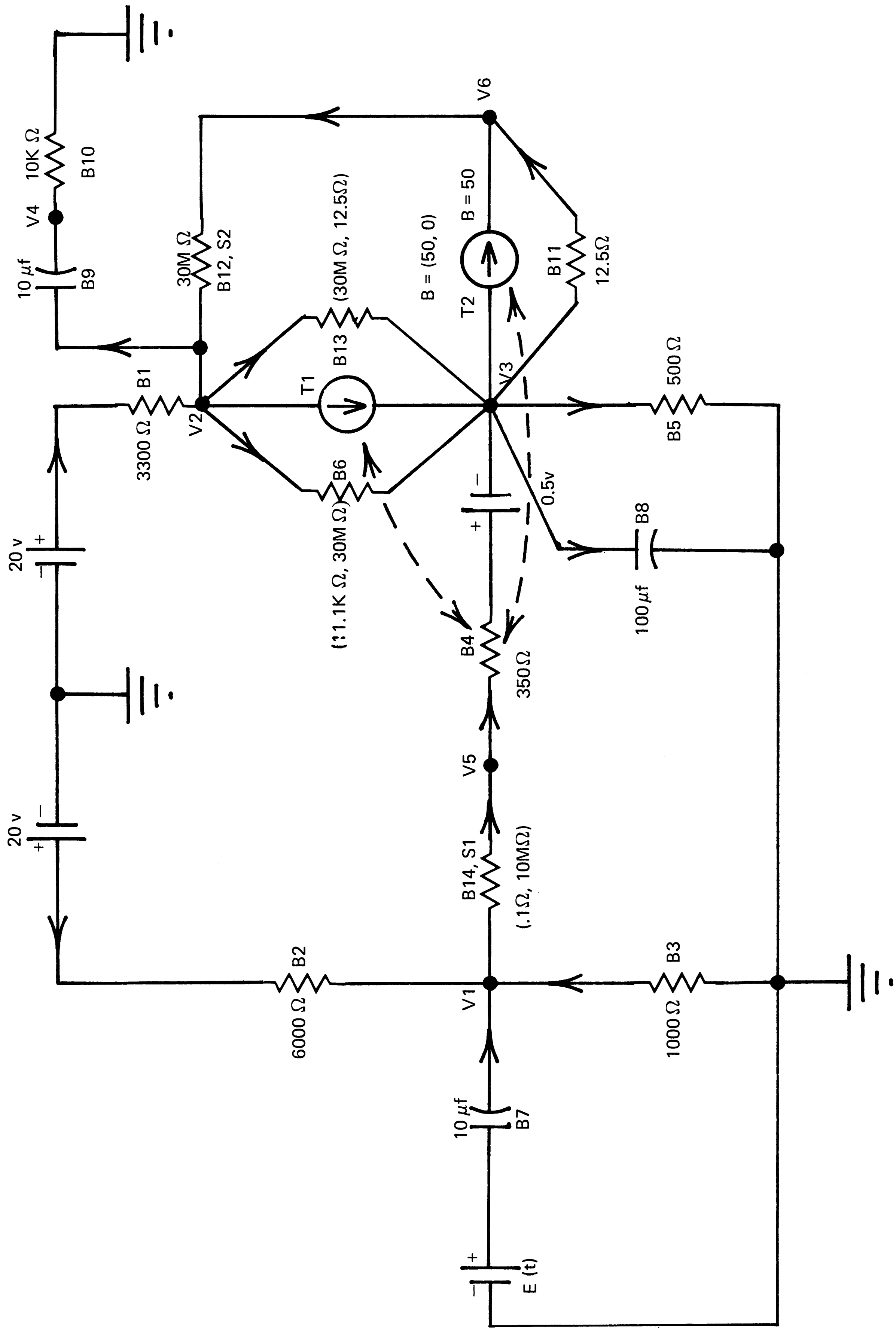

Figure 5 - Transient Equivalent Circuit

Running ECAP

A circuit description can be entered directly into ECAP or from a file. In this sample, the programs were written in EDITOR[1] on the file AMP. Then the user called ECAP and gave the Control Language command RUN. He specified that the output file was GAINSTAGE, that he wanted the output thoroughly annotated, and that ECAP should execute the entire program without pausing for additional input.

ECAP proceeded until it encountered an error in the program (an omitted comma). The user was allowed to correct the error and processing continued. When ECAP finished, the user left ECAP and listed the output file using the EXECUTIVE command COPY.

```
-EDITOR↩                          The user calls EDITOR and creates the ECAP program.
*APPEND↩
C                                 Each line the user types in the APPEND mode is
DC                                terminated by a Carriage Return.
C
B1 N(0,2),R=3300,E=20
B2 N(0,1),R=6000,E=20
B3 N(0,1),R=1E3
B4 N(13),R=350.,E=-.4763          The user made an error here, but he did not notice it.
B5 N(3,0),R=5E2
B6 N(2,3),R=11.1E3
T1 B(4,6),BETA=50
PUNCH,VOLTAGES,CURRENTS
EXECUTE
MODIFY
B1 R=5000
EX
C
AC
C
B1 N(0,2),R=3300
B2 N(0,1),R=6E3
B3 N(0,1),R=1000
B4 N(1,3),R=350
B5 N(3,0),R=500
B6 N(2,3),R=11.1E3
B7 N(5,1),C=10E-6
B8 N(3,0),C=100E-6
B9 N(2,4),C=10E-6
B10 N(4,0),R=10E3
B11 N(5,0),R=10E6,I=1
T1 B(4,6),BETA=50
FREQ=1000
PU,VO,CU                          The user requests the voltages and currents to be
EX                                written on a file.
MO
FR=20(10)20E3                     The analysis is of a series of frequencies, starting at
EX                                20 and increasing by a factor of 10 until 20,000 Hz
C                                 is reached (20, 200, 2,000, 20,000).
C
TR
C
```

1 - Refer to the Tymshare *EDITOR Manual, Reference Series,* for creating and editing files in EDITOR.

```
B1 N(0,2),R=3300,E=20
B2 N(0,1),R=6000,E=20
B3 N(0,1),R=1000
B4 N(5,3),R=350,E=-.4763
B5 N(3,0),R=500
B6 N(2,3),R=(11.1E3,30E6)
B7 N(0,1),C=10E-6
B8 N(3,0),C=100E-6
B9 N(2,4),C=10E-6
B10 N(4,0),R=10E3
B11 N(3,6),R=12.4763
B12 N(6,2),R=30E6
B13 N(2,3),R=(30E6,12.4763)
B14 N(1,5),R=(.1,10E6)
T1 B(4,6),BETA=(50,0)
T2 B(4,11),BETA=50
S1 B=14,(4,6,14),ON
S2 B=12,(4,6,13),OFF
SHORT=.1
OPEN=100E6
```

Sets the resistances substituted for inductance and capacitance in initial and steady state solutions.

```
TIME STEP=.01E-3
OUTPUT INTERVAL=5
```

Specifies time step between calculations and number of time steps between output.

```
FINISH TIME=1.4E-3
PUNCH,VOLTAGES,CURRENTS
EQUILIBRIUM
```

The user requests only a steady state solution.

```
EX
*WRITE AMP
 NEW FILE
955 CHARACTERS
```

He writes the ECAP program on a file named AMP.

```
*QUIT

-ECAP

:RUN
```

The user now calls ECAP and gives the RUN command.

```
ENTER INPUT-OUTPUT MEDIA [DISK FILE(S) OR TERMINAL]

INPUT FROM: AMP
 OUTPUT TO: GAINSTAGE
 NEW FILE
```

The solution is written on a file named GAINSTAGE.

```
PRINT INPUT CIRCUIT WITH SOLUTION? Y

PAUSE AFTER EACH OPERATION? N

DO YOU WANT A HEADING WITH THE SOLUTION? Y

ILLEGAL CHARACTER -- PLEASE EDIT
 B4 N(13),R=350.,E=-.4763
        ↑
?B4 N(1<,>3),R=350.,E=-.4763
```

ECAP detects the error and asks the user to correct the line. The user could have simply retyped the line; instead, he uses the ECAP text editing capabilities as described in Section 2 by typing Z^C1E^C,E^CD^C.

```
PROCEEDING WITH THE TERMINAL AS INPUT FILE

*EN

-COPY GAINSTAGE TO T
```

The user lists the contents of the solution file.

```
        TYMSHARE ECAP     JUL.21,1970   16:28     PROGRAM:   AMP



     DC
C
B1   N(0,2),R=3300,E=20
B2   N(0,1),R=6000,E=20
B3   N(0,1),R=1E3
B4   N(1,3),R=350.,E=-.4763
B5   N(3,0),R=5E2
B6   N(2,3),R=11.1E3
T1   B(4,6),BETA=50
     PUNCH,VOLTAGES,CURRENTS
     EXECUTE


 NODES                  NODE VOLTAGES
 1- 3      .27850044E  1    .52346973E  1    .22792478E  1

 BRANCHES               ELEMENT CURRENTS
 1- 4      .44743341E- 2    .28691659E- 2   -.27850044E- 2    .84161548E- 4
 5- 6      .45584957E- 2    .44743341E- 2


EOJ        CPU  0:00:01        TERMINAL  0:00:36



     MODIFY
B1   R=5000
     EX


 NODES                  NODE VOLTAGES
 1- 3      .27743369E  1   -.21592093E  1    .22642244E  1

 BRANCHES               ELEMENT CURRENTS
 1- 4      .44318419E- 2    .28709439E- 2   -.27743369E- 2    .96606988E- 4
 5- 6      .45284488E- 2    .44318418E- 2


EOJ        CPU  0:00:00        TERMINAL  0:00:10
```

```
        TYMSHARE ECAP    JUL.21,1970  16:28    PROGRAM:  AMP

      AC
C
B1    N(0,2),R=3300
B2    N(0,1),R=6E3
B3    N(0,1),R=1000
B4    N(1,3),R=350
B5    N(3,0),R=500
B6    N(2,3),R=11.1E3
B7    N(5,1),C=10E-6
B8    N(3,0),C=100E-6
B9    N(2,4),C=10E-6
B10   N(4,0),R=10E3
B11   N(5,0),R=10E6,I=1
T1    B(4,6),BETA=50
      FREQ=1000
      PR,VO,CU
      EX

FREQ =    .10000000E  4

   NODES                    NODE VOLTAGES
MAG  1- 4  .25270270E  3  .71871982E  5  .47229629E  2  .71871891E  5
PHA       -.76127359E  1 -.17685260E  3 -.86647861E  2 -.17676141E  3
MAG  5- 5  .25529898E  3
PHA       -.11155274E  2

   BRANCHES                 ELEMENT CURRENTS
MAG  1- 4  .21779388E  2  .42117117E- 1  .25270270E  0  .70883075E  0
PHA        .31473983E  1  .17238726E  3  .17238726E  3  .31590121E  1
MAG  5- 8  .94459258E- 1  .28966571E  2  .99997495E  0  .29675251E  2
PHA       -.86647861E  2  .31700243E  1  .28299355E- 3  .33521388E  1
MAG  9-11  .71871891E  1  .71871891E  1  .25529898E- 4
PHA       -.17676141E  3 -.17676141E  3 -.11155274E  2


EOJ        CPU  0:00:03        TERMINAL  0:01:49


      MO
      FR=20(10)20E3
      EX


FREQ =    .20000000E  2

   NODES                    NODE VOLTAGES
MAG  1- 4  .78210043E  3  .23589929E  5  .76443265E  3  .23515589E  5
PHA       -.12950822E  2 -.11899840E  3 -.18842481E  2 -.11444853E  3
MAG  5- 5  .12344657E  4
PHA       -.51866161E  2
```

```
    BRANCHES               ELEMENT CURRENTS
MAG  1- 4  .71484633E  1  .13035007E  0  .78210043E  0  .23261160E  0
PHA        .61001602E  2  .16704918E  3  .16704918E  3  .61586624E  2
MAG  5- 8  .15288653E  1  .94944444E  1  .99992378E  0  .96061435E  1
PHA       -.18842481E  2  .62127389E  2  .55638100E- 2  .71157519E  2
MAG  9-11  .23515589E  1  .23515589E  1  .12344657E- 3
PHA       -.11444853E  3 -.11444853E  3 -.51866161E  2


FREQ =    .20000000E  3

    NODES                  NODE VOLTAGES
MAG  1- 4  .33205387E  3  .69102737E  5  .22701877E  3  .69100549E  5
PHA       -.27766385E  2 -.16470015E  3 -.73676529E  2 -.16424422E  3
MAG  5- 5  .37578255E  3
PHA       -.38565906E  2

    BRANCHES               ELEMENT CURRENTS
MAG  1- 4  .20940223E  2  .55342312E- 1  .33205387E  0  .68151828E  0
PHA        .15299848E  2  .15223362E  3  .15223362E  3  .15357921E  2
MAG  5- 8  .45403754E  0  .27850114E  2  .99997062E  0  .28528019E  2
PHA       -.73676529E  2  .15412972E  2  .13422932E- 2  .16323471E  2
MAG  9-11  .69100549E  1  .69100549E  1  .37578255E- 4
PHA       -.16424422E  3 -.16424422E  3 -.38565906E  2


FREQ =    .20000000E  4

    NODES                  NODE VOLTAGES
MAG  1- 4  .24957087E  3  .71963994E  5  .23645147E  2  .71963971E  5
PHA       -.38555919E  1 -.17842487E  3 -.88322495E  2 -.17837927E  3
MAG  5- 5  .25023196E  3
PHA       -.56738183E  1

    BRANCHES               ELEMENT CURRENTS
MAG  1- 4  .21807271E  2  .41595145E- 1  .24957087E  0  .70973825E  0
PHA        .15751341E  1  .17614441E  3  .17614441E  3  .15809411E  1
MAG  5- 8  .47290294E- 1  .29003666E  2  .99997510E  0  .29713367E  2
PHA       -.88322495E  2  .15864472E  1  .14172474E- 3  .16775047E  1
MAG  9-11  .71963971E  1  .71963971E  1  .25023196E- 4
PHA       -.17837927E  3 -.17837927E  3 -.56738183E  1


FREQ =    .20000000E  5

    NODES                  NODE VOLTAGES
MAG  1- 4  .24852513E  3  .71994436E  5  .23655182E  1  .71994436E  5
PHA       -.38722101E  0 -.17984244E  3 -.89832202E  2 -.17983788E  3
MAG  5- 5  .24853178E  3
PHA       -.57066812E  0
```

```
    BRANCHES                 ELEMENT CURRENTS
MAG  1- 4  .21816496E  2  .41420855E- 1  .24852513E  0  .71003849E  0
PHA        .15755969E  0  .17961278E  3  .17961278E  3  .15814137E  0
MAG  5- 8  .47310365E- 2  .29015939E  2  .99997515E  0  .29725977E  2
PHA       -.89832202E  2  .15869221E  0  .14014890E- 4  .16779796E  0
MAG  9-11  .71994436E  1  .71994436E  1  .24853178E- 4
PHA       -.17983788E  3 -.17983788E  3 -.57066812E  0


EOJ        CPU  0:00:07        TERMINAL  0:05:30



        TYMSHARE ECAP    JUL.21,1970  16:35    PROGRAM:  AMP


      TR
C
B1    N(0,2),R=3300,E=20
B2    N(0,1),R=6000,E=20
B3    N(0,1),R=1000
B4    N(5,3),R=350,E=-.4763
B5    N(3,0),R=500
B6    N(2,3),R=(11.1E3,30E6)
B7    N(0,1),C=10E-6
B8    N(3,0),C=100E-6
B9    N(2,4),C=10E-6
B10   N(4,0),R=10E3
B11   N(3,6),R=12.4763
B12   N(6,2),R=30E6
B13   N(2,3),R=(30E6,12.4763)
B14   N(1,5),R=(.1,10E6)
T1    B(4,6),BETA=(50,0)
T2    B(4,11),BETA=50
S1    B=14,(4,6,14),ON
S2    B=12,(4,6,13),OFF
      SHORT=.1
      OPEN=100E6
      TIMESTEP=.01E-3
      OUTPUTINTERVAL=5
      FINISHTIME=1.4E-3
      PUNCH,VOLTAGES,CURRENTS
      EQUILIBRIUM
      EX
```

STEADY STATE SOLUTION

NODES	NODE VOLTAGES			
1- 4	.27849842E 1	.52346137E 1	.22792208E 1	.52340903E- 3
5- 6	.27849758E 1	.23317205E 1		

BRANCHES	ELEMENT CURRENTS			
1- 4	.44743595E- 2	.28691693E- 2	-.27849842E- 2	.84157205E- 4
5- 8	.45584416E- 2	.44741119E- 2	-.27849842E- 7	.22792208E- 7
9-12	.52340903E- 7	.52340903E- 7	-.96771629E- 7	-.96763105E- 7
13-14	.98513097E- 7	.84157218E- 4		

EOJ CPU 0:00:02 TERMINAL 0:00:16

-

SECTION 1
THE CONTROL LANGUAGE

Tymshare ECAP allows extensive control of input files, output files, and output format. An ECAP program can be entered from one or more files, from the terminal, or a combination of both. Likewise, the output can be written on a file and/or the terminal. Besides the requested calculated values, the output can include the circuit description and information such as the time and date. Furthermore, the user can direct ECAP to write the updated circuit description on a file every time an analysis is executed. This feature is especially useful when the circuit description is entered into ECAP from the terminal.

All of these file controls are set while ECAP is in the Control Mode. ECAP indicates that it is in the Control Mode awaiting a Control Language command by displaying a colon (:) on the terminal. Any Control Language command can be typed after a colon.

The Control Language also includes utility commands which transfer control to a command file, provide instructions about using ECAP, convert free form ECAP programs to a form suitable for use with a card oriented processor, and catalog the contents of the retrievable model files discussed in Section 6.

To make the Control Language easy to use, several convenient features have been incorporated. Any command can be abbreviated to the first three letters. In fact, ECAP only looks at the first three letters; therefore, any misspelling beyond that slips by unnoticed. For example,

: LISP ↵

is interpreted as the LIST command. If the Control Mode does not understand a command, a question mark will be displayed and then the Control Mode colon.

If the user wishes to abort a command before he has hit the Carriage Return, he can hit the ALT MODE/ESCAPE key. This will return him to the Control Mode. If the user realizes before he has typed the Carriage Return that he has typed a command incorrectly, there are two editing characters available in the Control Mode to correct the error. Control A (A^C) deletes one preceding character for each time it is struck. A back arrow (←) will print for each A^C struck. For example,

SAVE FIL←←←OUTFILE ↵ *A^C typed three times.*

is equivalent to

SAVE OUTFILE ↵

Control Q (Q^C) deletes the entire line, prints an up arrow (↑), returns the carriage, and then prints the colon.

RUN/BATCH

The Control Language has commands that set each of the file and format controls individually. In general, however, it is easier to use one of the single, multi-option commands, RUN or BATCH. The RUN command will guide the user step-by-step in setting the options. The BATCH command requires the user to set a minimum of options and assumes conditions for the others. Both of these commands will carry ECAP into the Analysis Mode. A third command to transfer to the Analysis Mode is EXECUTE. The EXECUTE command is used in conjunction with the option commands.

RUN

A user can set most of the file and format controls with the one command, RUN. After the user types RUN and a Carriage Return, ECAP will respond with a series of questions. These questions can be answered affirmatively by Y (or YES); or negatively by N, NO, or Carriage Return.

Example

```
: RUN ↵

ENTER INPUT-OUTPUT MEDIA [DISK FILE(S)
                          OR TERMINAL]

INPUT FROM: T ↵
 OUTPUT TO: OUT ↵
 NEW FILE ↵
```

PRINT INPUT CIRCUIT WITH SOLUTION? ↩

PAUSE AFTER EACH OPERATION? ↩

DO YOU WANT A HEADING WITH THE SOLUTION? Y ↩

ANALYSIS: DC ↩

*

The first item requested is the name of the file (or files) that contain the ECAP program (or programs). If input is from the terminal, type T. Several input files may be given if they are separated by commas. When control is transferred to the files, ECAP will read all of the first file, then the next, and so forth.

INPUT FROM: FILE1, 2NDFILE ↩
OUTPUT TO: OUT ↩
OLD FILE ↩

Then ECAP requests the output file. Either a file name or T (for terminal) may be entered. Even though a file name is entered, the user can select the terminal as the output device when he is in the Analysis Mode; likewise, if the terminal is specified, the user can select a file in the Analysis Mode. If a file name is entered, ECAP responds with NEW FILE or OLD FILE. The file name is confirmed by typing a Carriage Return, or it can be aborted by typing ALT MODE. *NOTE: If an old file is confirmed to be the output file, its contents will be erased. For more detailed discussion of the input and output files, refer to READ and WRITE under* **Option Commands**, *Page 17.*

The next question asked by ECAP is

LIST CIRCUIT WITH SOLUTION?

An affirmative answer will cause a circuit description in standard ECAP card format to be listed with the solution. In other words, an affirmative answer is the same as invoking the LIST and FORMAT options.

The next option set in the RUN sequence is the PAUSE option

PAUSE AFTER EACH OPERATION?

An affirmative answer has the same effect as invoking the PAUSE option.

The next question,

DO YOU WANT A HEADING WITH THE SOLUTION?

is a combination of the DATE and TIME options. An affirmative answer will annotate the run. The date and input file name will be written on the output device as a heading for the solution, and the CPU seconds and terminal time will be written at the end of each execution.

After all of these questions have been answered, the RUN sequence transfers control to the Analysis Mode and accepts input from the input device just as if the EXECUTE command were given. If input is from the terminal, ECAP requests the analysis type.

ANALYSIS:

The user answers AC, DC, or TR plus a Carriage Return. ECAP is then in the corresponding Analysis Mode and displays an asterisk on the terminal.

If the ECAP program is stored on a file, the first line of the file must contain the analysis type. ECAP reads the first line of the file and transfers to the corresponding analysis mode. If no pause option is set, ECAP continues reading the file and executes any commands contained on the file until the entire file is read. After the file (or files) is read, control is transferred back to the terminal and the following message is displayed.

PROCEEDING WITH THE TERMINAL AS INPUT FILE

*

EXPERT, NOEXPERT

The EXPERT command shortens the text of the RUN sequence. The same information is requested in the same order; only the format of the questions is changed. The NOEXPERT command cancels the expert option. Below is a sample EXPERT-RUN sequence with the equivalent option commands shown on the right.

– ECAP ↩	*Option Command*
: EXPERT ↩	
OK.	
: RUN ↩	
INPUT: CIRCUITIN ↩	*READ*
OUTPUT: COUT ↩	*WRITE*
OLD FILE ↩	

```
CIRCUIT? Y ↵          LIST + FORMAT
PAUSE? ↵              PAUSE
HEADING? Y ↵          DATE + TIME
                      EXECUTE
PROCEEDING WITH TERMINAL AS INPUT
                               FILE

*
```

BATCH

The BATCH command sets the controls for annotated output with a minimum of typing. The user enters only the names of the input and output devices. ECAP automatically sets the TIME, DATE, FORMAT, and LIST options and then executes the ECAP program. The sequence for the BATCH command is

```
: BATCH ↵
READ FILE(S): CIN ↵
PUNCH TO: OUTFILE ↵
 NEW FILE ↵

PROCEEDING WITH THE TERMINAL AS
                     INPUT FILE

*
```

OPTION COMMANDS

The RUN and BATCH commands have been designed to allow the user to set the controls with a minimum knowledge of the Control Language. As a user becomes more familiar with Tymshare's ECAP, he may decide it is quicker to set the controls directly by individual option commands and then to use the EXECUTE command to transfer to the Analysis Mode and begin execution.

EXECUTE

The EXECUTE command initiates the Analysis Mode by transferring control to the input device. If no input file is open (for example, by a READ command), ECAP assumes that the ECAP program will be entered from the terminal. When input is from the terminal, the analysis type should accompany the EXECUTE command.

EXECUTE DC

EXECUTE AC

EXECUTE TR

An asterisk will be displayed on the terminal to indicate that ECAP is in the appropriate Analysis Mode awaiting input.

If the analysis commands are to come from a file, the EXECUTE command starts ECAP reading the file. ECAP will read the file line-by-line and act accordingly until either the file is read completely, or a QUIT or END is encountered. When ECAP finishes with the file, control is either transferred to the next file if several were specified, or returned to the terminal.

Actually, the input file is opened when the READ command is given. The EXECUTE command commences reading the file. If a file has just been opened, reading will begin at the beginning of the file, but if the file has been only partially read (for example, reading was interrupted by an Analysis Language QUIT command), reading will continue after the last statement read. ECAP closes the input file after the entire file has been read.

READ, LOAD

READ (or LOAD) is used to specify one or more files containing ECAP programs. The form of the command is

READ FILE1

or

LOAD THISFILE,THATFILE,THOSEFILES

When the command is given, any input file previously opened is closed and the first file listed is opened. The files are not read, however, until the EXECUTE command is given. The files listed in the READ (or LOAD) command must contain only Analysis Commands and ECAP circuit descriptions.

The circuit description will be assumed to come from the terminal if this command is not used.

WRITE

WRITE specifies the solution output file. The solution will be written on this file whenever the PUNCH command is encountered in the ECAP program. The form of the command is

: <u>WRITE OUTFILE</u> ↵
NEW FILE ↵

After the command is given, ECAP responds with either OLD FILE or NEW FILE depending on whether or not the file already exists. The WRITE command is confirmed by typing a Carriage Return after OLD FILE/NEW FILE. The command can be aborted by typing an ALT MODE/ESCAPE. If the command is confirmed, any previous output file is closed and the specified file is opened for output. Opening an old file for output erases the entire contents of the file.

It is unnecessary to use the WRITE command. If there is no output file opened when the Analysis Command PUNCH is encountered, ECAP will request an output file name. Either a file name or T (for the terminal) can be entered. There is one major difference, however, between these two methods of specifying an output file. If the file is specified at execution time **the file will not be erased.** The solution will be appended to any previous contents of the file; whereas the output file is emptied when the WRITE command is given. *NOTE: The Analysis Command PUNCH causes the solution to be written at the end of the output file; the file is not erased first. Therefore, if several Analysis PUNCH commands are given, all of the solutions will be written on the file in sequence.*

LIST, NOLIST

LIST sets the controls so that the solution output is accompanied by the circuit description. The form of the command is

: <u>LIST</u> ↵
OK.

When the Analysis Command EXECUTE is given, the circuit block entered since the previous execution is written on the same device as the solution. The LIST option remains in effect until either the NOLIST or the RESET command is given.

The circuit block will be listed in the compact Tymshare format unless the FORMAT option is used.

SAVE, NOSAVE

The SAVE command sets the controls so that whenever the Analysis Command EXECUTE is encountered, the circuit block entered since the previous EXECUTE will be written on the specified file. The circuit description is written in a form that allows the file to be used as an input file in a subsequent Tymshare ECAP run. If the FORMAT option is used, the circuit is written in a form suitable for input to any other version of ECAP.

: <u>SAVE CIRCUITFILE</u> ↵
OLD FILE ↵

When a file is specified, the Control Mode responds with OLD FILE or NEW FILE. The command is confirmed by typing a Carriage Return or aborted by hitting the ALT MODE/ESCAPE key. When the SAVE command is confirmed, the SAVE file is erased and opened, and any SAVE file previously opened is closed.

The primary use of the SAVE command is to preserve on a file circuit descriptions typed directly into ECAP. If the SAVE option is set, all of the commands required to produce the solution will be appended to the SAVE file each time the EXECUTE command is encountered. Therefore, a later ECAP run with this file as the input file would create the same solution output. Although the user may not want the same solution twice, the circuit description stored on a file is available for editing in EDITOR, or, with the PAUSE control set, for editing in ECAP[1]. Therefore, small changes can be made and the problem rerun without retyping the entire circuit description.

The SAVE option is useful when input is from a file if there are syntax errors in the circuit description on the input file. The Analysis Mode will allow the user to edit these errors. Then, when the circuit is executed, the SAVE option will write the corrected circuit description on the SAVE file. Likewise, if the user is operating in the PAUSE mode and creating a new program by editing an old program, the new program will be saved.

NOSAVE negates the SAVE command.

FORMAT, NOFORMAT

Unless the FORMAT option is set, circuit descriptions are written in the compact Tymshare ECAP format. If it is necessary to have the circuit description created in a form suitable for card oriented ECAP, the FORMAT command can be given.

: <u>FORMAT</u> ↵
OK.

NOTE: The Control Command CONVERT can be used to convert a Tymshare free form ECAP program into a card oriented ECAP program (see Page 20).

NOFORMAT negates the FORMAT command.

1 - See the discussion of the PAUSE command on Page 19.

DATE, NODATE

The DATE command causes a heading to be written at the beginning of each solution output. This heading includes the date, the time, and the ECAP program name (the input device). For example,

TYMSHARE ECAP JUN.24,1970 17:09
PROGRAM: CIRCUITIN

NODATE negates the DATE command.

TIME, NOTIME

The TIME command causes the CPU and terminal time required for execution to be written at the end of each analysis. For example,

EOJ CPU 0:00:00 TERMINAL 0:00:08

The times are given in the form

hours:minutes:seconds

NOTIME negates the TIME command.

PAUSE, NOPAUSE

The PAUSE command causes processing to stop at each logical junction in the ECAP program. PAUSE is intended to be used when the ECAP program is stored on a file. Control returns to the terminal each time a block of data is read and after each compute is executed. The form of the command is

: PAUSE ↩
OK.

The purpose of PAUSE is to allow the user to edit in ECAP an already existing program and to run the new program. For example, in a previous run, an ECAP program was saved[1] on a file named CIRCUIT. CIRCUIT contains, in order, the analysis type, the circuit description, the output specifications, and the EXECUTE command. Now the user wishes to modify the circuit slightly and re-execute the program. If he reads the file into ECAP without the PAUSE control set, ECAP will rerun the program he executed earlier before he could modify the program. With PAUSE set, ECAP will read the file and then return control to the terminal. The user can then employ the extensive editing capabilities of the Analysis Mode to change the circuit description. He then types GO, and the new program will be executed. This is the easiest way to alter an ECAP program. The following sequence might be printed on the terminal.

–ECAP ↩

: PAUSE ↩
OK.

: READ CIRCUIT ↩

: WRITE SOLUTION ↩
OLD FILE ↩

: EXECUTE ↩

PAUSE AFTER INPUT

*** ED B15 ↩**

B15 N(8,12),R=285
B15 N(8,12),R=300 ↩ *The user retyped the line to change the value of the resistor.*

*** GO ↩**

NOPAUSE negates the PAUSE command.

RESET

The RESET command negates all previous Control Language commands. All controls are reset and all files are closed. RESET allows the user to continue as if he had just entered ECAP from the EXECUTIVE.

UTILITY COMMANDS

The Control Language has several commands to facilitate the use of ECAP. Two file commands, CATALOG and CONVERT, perform independent file functions. COMMANDS allows control of ECAP to be transferred to a command file. In addition, there are six commands to help the beginning user become better acquainted with ECAP; and, finally, QUIT returns to the EXECUTIVE.

CATALOG

The CATALOG command directs ECAP to produce a catalog of all the retrievable models stored on the specified file or files.

: CATALOG ↩
FILE(S): MOD1, MOD2 ↩
OUTPUT TO: T ↩

1 - See the SAVE command on Page 18.

The catalog is in paged format, 11 inches long if output is to the terminal and 7½ inches long for the line printer if output is to a file. The catalog begins with a heading giving the date and time, then the name of the first file given and, for each model on that file, the model identification and a listing of the model. For example,

```
    TYMSHARE ECAP MODEL CATALOG -
          JUN.24,1970 19:40 PAGE 1

FILE:   MOD1

INDEX:

         MINIMOD

MODEL: MINIMOD  THIS IS A DEMO
B1 N(1,2),R=350,E=-.3
B2 N(3,2),R=10E3
T1 B(1,2),BETA=5
```

For a discussion of retrievable models, see Section 6.

CONVERT

The CONVERT command converts Tymshare free form ECAP programs to a form suitable for use with a card oriented ECAP program. For example,

: <u>CONVERT</u> ↩
FILE(S): <u>CIRCUIT</u> ↩
OUTPUT TO: <u>FORMFILE</u> ↩
  OLD FILE ↩

writes the contents of CIRCUIT on FORMFILE with the text starting in the seventh column as appropriate.

COMMANDS

COMMANDS transfers control from the terminal to a file. ECAP will accept commands from the file just as if they were entered from the terminal. For example,

: <u>COMMANDS RUNECAP</u> ↩

causes ECAP to open the file RUNECAP and read and execute the commands one by one until the file is exhausted or the EXECUTE command is encountered. The command file capability is intended for the Control Mode only, since ECAP program files are in fact command files for the Analysis Mode.

CAPABILITIES and INSTRUCTIONS

CAPABILITIES prints a summary of Tymshare ECAP capabilities. The total printed text is about 120 lines.

INSTRUCTIONS prints instructions for using ECAP. The total printed text is about 400 lines.

HELP and SUMMARY

HELP prints a list of all of the Control Commands together with a brief description of each command.

SUMMARY prints a list of all the Control Commands without any description.

QUIT

QUIT returns to the EXECUTIVE. The CONTINUE command cannot be used to return to ECAP. All files are closed.

SECTION 2
THE ANALYSIS LANGUAGE
DIRECT COMMANDS AND EDITING FEATURES

ECAP programs are written in the Analysis Language. The Analysis Language includes the editing commands discussed in this section and the standard circuit descriptions and output specifications for DC, AC, and transient analyses discussed in Sections 3 through 6. Conventional ECAP card oriented programs are comprised of the statements discussed in Sections 3 through 5. Section 6 discusses extensions to standard ECAP. The editing commands discussed in this section are unique to Tymshare. Some of these commands can be included in the file with the ECAP program, but they are direct commands rather than indirect commands. This means that the editing commands are executed as soon as they are entered into ECAP, whereas most ECAP statements are specifications that are stored and not operated upon until the EXECUTE command is given. In card oriented ECAP, EXECUTE, MODIFY, DC, AC, and TR are the only direct commands.

ECAP programs are divided into blocks; each block is terminated by an EXECUTE command. Editing can be done only on the current block. Once the EXECUTE command is given, the block is solidified and the circuit stored in ECAP cannot be changed by direct editing. Therefore, when the circuit is entered from a file, it may be advantageous to omit the EXECUTE.

NOTE: The MODIFY command can be used to modify the previous block and re-execute. MODIFY is discussed on Page 26.

Statements can be entered into ECAP in free form; that is, no special spacing or columniation is required. Several statements can be entered on the same line separated by semicolons (;); however, the Analysis Language will assign a separate physical line to each statement. For example, if

B1 N(0,1),R=30;B2 N(1,2),C=10E-3

is entered, the two statements will be placed in the text block as

B1 N(0,1),R=30
B2 N(1,2),C=10E-3

To facilitate entering text into the Analysis Language, ECAP allows the user to correct typing errors through the use of control characters. The A^C and Q^C can be used in the same way as they are in the Control Language discussed on Page 15. For example,

* B15 N(8,7)A^C←A^C←5),R=100 ↵ — *The first A^C deleted the) and the second deleted the 7.*

* LI B15 ↵ — *The line is listed.*
B15 N(8,5),R=100

*

A^C, Q^C, W^C, and I^C can be used as described in the chart below. I^C can be used for spacing columns.

Control Character	Symbol Printed	Action
A^C	←	Deletes preceding character.
W^C	\	Deletes preceding word.
Q^C	↑	Deletes entire line.
Other		
I^C		Spaces to next tab stop.

ECAP reads only the first two characters of all Analysis Commands; therefore, all commands can be shortened to the first two letters, except INSERT which must contain at least INS.

LINE ADDRESSING

To use the editing commands, the user must be able to specify the lines to be edited. There are several ways to address lines in ECAP: by physical line numbers, by labels, by the special characters . and $, and by address arithmetic. A line range is specified by the address of the first line and the address of the last line separated by a comma.

Physical line numbers are integer numbers denoting the actual line position within a block, such as

* LIST 1 ↵
AC

The physical line number 1 is always the first line of the entire text stored in ECAP regardless of the beginning of the current block. Therefore, line 1 is always DC, AC, or TR and cannot be edited or deleted. Since it is not meaningful to edit a block that has already been executed, caution must be taken with the editing commands to keep the line addresses within the current block. With the LIST command it is useful to be able to address the entire stored text.

Line labels are the first two or three characters of a B, T, S, M, E, or I statement. For example,

* **LIST B8,B10** ↲

lists statements B8 through B10.

The dollar sign specifies the last line in the block and the period signifies the current line in the block. Address arithmetic can be used to refer to lines by adding (+) or subtracting (-) lines from a line address. For example, if the fourth line has the label B3, then

* **LIST B3+2** ↲

will list the sixth (4+2) line. Line arithmetic is especially useful for lines without labels.

LIST

The LIST command displays on the terminal the line or lines specified in the line range. The form of the command is

LIST range

The LIST command given without a line range displays the entire stored text (equivalent to LIST 1,$). After a LIST command, the current line (the line addressed by a period) is the last line listed.

Example

* **LIST $-2,$** ↲

TI=1
FI=5
PR,NV,CA

* **LI .** ↲

PR,NV,CA

EDIT

The EDIT command allows the user to alter lines of text in the current block. The form of the command is

EDIT range

ECAP will display the range and await editing commands. The control characters listed below tell ECAP what editing is to be done. For example, a Control Z (Z^C) copies the text from the old line to the new line up to and including the character typed after the Z^C.

* **EDIT B5** ↲
B5 N(3,0),R=150
Z^C=B5 N(3,0),R=2D^C50 *The user typed Z^C= to copy to the equal sign, then 2 D^C to make the change and end the edit.*

Control Z does not print on the terminal. Please refer to the Tymshare *EDITOR Manual, Reference Series,* for a detailed discussion of editing on the Tymshare system.

A forced edit mode will result if a line containing a syntax error is entered.

EDITING CONTROL CHARACTERS

Control Character	Symbol Printed	Action
For Deleting		
S^C	%	Deletes next character in old line.
K^C		Deletes next character in old line and prints it.
P^C *and a character*	%	Deletes up to but not including character typed after it.
X^C *and a character*	%	Deletes up to and including the character typed after it.
For Copying		
C^C		Copies next character in old line.
O^C *and a character*		Copies up to but not including the character typed after it.
Z^C *and a character*		Copies up to and including the character typed after it.
H^C		Copies and prints rest of old line and continues the edit at the end of the line.
Y^C		Copies but does not print rest of old line and continues edit at the beginning of the new line.
R^C		Copies and prints rest of old line plus the new line; continues edit from where R^C was typed.

EDITING CONTROL CHARACTERS (Continued)		
Control Character	**Symbol Printed**	**Action**
T^C		Same as R^C except that it aligns the rest of the old line with the new line.
U^C		Copies from old line up to next tab stop in new line.
For Inserting		
E^C *text* E^C	$<$ $>$	Inserts text into old line; first E^C prints $<$, second E^C prints $>$.
Other		
N^C	←	Backspaces in the old and new lines.
The following are used for termination.		
Carriage Return		Deletes rest of old line and ends the edit.
D^C		Copies the rest of the old line to the new line, prints the line, and ends the edit.
F^C		Copies but does not print the rest of the old line; ends the edit.

INSERT

The INSERT command allows the user to insert a new line or lines before the line given in the INSERT command. The INSERT mode is terminated by a Control D. The form of the command is

INSERT line ↵
text to be inserted
D^C

NOTE: Unlike all other Analysis Commands, at least three letters of INSERT must be given.

DELETE

The DELETE command removes the specified lines from the text area. The form of the command is

DELETE range ↵

EXECUTE

The EXECUTE command causes ECAP to analyze the current circuit. If the circuit cannot be analyzed, an error message will result. If the analysis is completed, the block will no longer be available for editing.

DC, AC, TR

DC, AC, and TR transfers to the DC, AC or transient Analysis Mode. All stored text is erased. The command stands as the first line of the new text.

GO

If the user has set the PAUSE option in the Command Mode, GO will continue processing after a pause. Please see the discussion of PAUSE on Page 19.

QUIT and END

The QUIT command returns the user to the Command Mode. All stored text is lost. The END command terminates the ECAP run, returning the user to the EXECUTIVE.

HELP

HELP prints a partial list of the direct commands available in the Analysis Mode and a brief description of each command.

SECTION 3

THE DC ANALYSIS MODE

The DC Analysis Mode accepts ECAP DC programs. DC Analysis provides steady-state solutions for circuits with time-invariant sources. The Introduction discusses how to code a circuit for ECAP. This section describes the inputs and outputs acceptable in DC Analysis and also describes techniques for modifying circuits. Refer to Section 6 for extensions to ECAP.

DC Analysis has the following capacity:

B Statements: 60
T Statements: 10
Nodes: 20
Simultaneous Modifications: 20

DC CIRCUITRY

A DC circuit is constructed from the following elements and parameters:[1]

R = resistance (ohms)

G = conductance (mhos)

E = independent time-invariant voltage source (volts)

I = independent time-invariant current source (amps)

BETA = current gain from control branch to dependent branch

GM = transconductance from control branch to dependent branch

node numbers

branch numbers

A DC branch contains one and only one passive element (resistor). Each branch is assigned a number, and each node—the junction between branches—is assigned a number. The circuit is then described to ECAP branch-by-branch using B statements. The form of the B statement is

Bbranch number N(node$_1$,node$_2$), R=resistance,
E=voltage source
or
I=current source

1 - See list of ECAP units in the Appendix.

Example

B1 N(0,1),R=100
B2 N(1,2),R=12,E=5.2

Positive current flows from node$_1$ to node$_2$. Node zero is always the reference node. Every circuit must have a reference node.

Worst case analysis and standard deviation require that maximum and minimum values be supplied for some or all of the parameters. These values may be specified in two ways. The maximum and minimum values can be enclosed in parentheses after the nominal value. For example,

B7 N(3,0),R=100(80,110)

Alternatively, a fractional tolerance can be enclosed in parentheses after the nominal value. For example,

B7 N(3,0),R=100(.15)

implies that branch 7 can have a minimum resistance of 85 ohms and a maximum resistance of 115 ohms.

Another statement acceptable in DC Analysis is the T statement which describes dependent current sources. The T statement is discussed on Page 26.

DC OUTPUT

The quantities calculated in DC Analysis are illustrated in Figure 6 below.

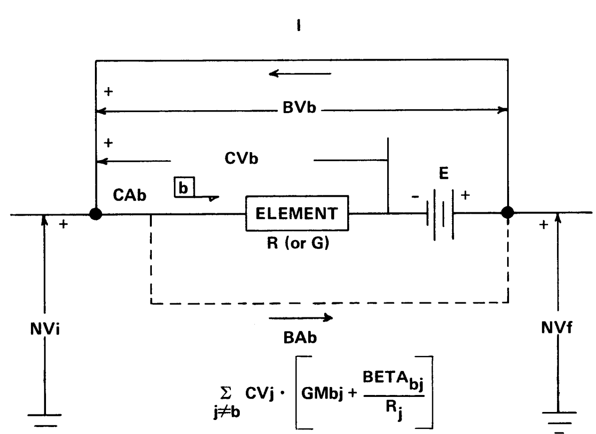

Figure 6 - A Typical DC Analysis Circuit

The output quantities are:

NV, node voltage

BV, branch voltage drop ($NV_i - NV_f$)

CV, element voltage (BV + E)

CA, element current

$$(CA_b = CV_b * Y_b + \sum_{j \neq b} CV_j \cdot (GM_{bj} + \frac{BETA_{bj}}{R_j}))$$

BA, branch current (CA - I)

BP, element power dissipation (CV · CA)

SE, sensitivity $SE_{ij} = \frac{\partial NV_i}{\partial \, parameter_j}$

WO, worst case values, minimum and maximum voltage for each node.

ST, standard deviation for node voltages.

MISC, miscellaneous data including the nodal conductance matrix, the nodal impedance matrix, and the equivalent current vector.

Any combination of these output quantities can be requested in a PRINT or PUNCH statement. PRINT displays the output on the terminal, whereas PUNCH writes the output on a file. The form of the commands is

PR, quantity, quantity, etc.

or

PU, quantity, quantity, etc.

Example

PR,NV,CA
PU,WO,SE,MISC

The output for specific branches or nodes can be requested by including the branch or node numbers in parentheses. For example,

PR,NV(3,4,5,12),CA

prints the node voltages for the specified nodes and the currents for all elements.

The sensitivity, worst case, and standard deviation are special calculations. The worst case values are calculated by taking the minimum and maximum values for all the parameters. The choice of which extreme to use is determined by the sign of the partial derivative of the node voltage with respect to the individual parameters. The user can select the nodes for which the worst case sensitivities will be calculated. The statements

WO
SE

instruct ECAP to calculate the worst case and sensitivities for all nodes, but the statements

WO,4
SE,4
PR,SE,WO

calculate and print the values for node 4 only.

DEPENDENT CURRENT SOURCES

All three analysis modes allow dependent current sources. A BETA (current gain) or GM (transconductance) defines a dependent current source in the dependent branch. There may be more than one dependent current source in a branch. These dependent current sources are used to model transistors and other sophisticated devices.

Dependent current sources are specified with a T statement. The form of the T statement is

Tnumber B(control branch, dependent branch), [BETA=current gain | GM=transconductance]

For example,

T2 B(3,4),BETA=2

specifies that the second dependent current source is in branch 4; the element current in B4 is twice the element current in B3. That is, positive current in the source branch creates positive current in the dependent branch. (Positive current flows from the first node listed in the branch statement to the second node listed.) A negative BETA would mean that a positive current in the source branch would create a negative current in the dependent branch. If the control branch resistance has a non-zero tolerance, BETA rather than GM should be specified. ($BETA = GM * R_{from}$)
NOTE: The T card must not precede the control branch.

Since all ECAP nodes must be connected to ground by at least one path through the circuit, dummy branches are sometimes used for the dependent branches. These dummy branches connect the reference node with ground and have no effect on the circuit.

Dependent voltage sources can be approximated by the dependent current sources.

MODIFY

The parameter values in a circuit can be executed, altered, and re-executed using the MODIFY command. The MODIFY command is followed by the statement label and the new values for the parameters to be changed only. Each new statement begins on a new line. For example,

MODIFY
B5 R=100
B7 R=10
EX

will change the resistances in branches 5 and 7 and re-execute.

In any modify routine, one parameter can be iterated in equal steps through a given range of values. For example,

MO
B5 R=100(5)200

will run six analyses with the resistance in branch 5 equal to 100, 120, 140, 160, 180, and 200. Parameters can be iterated only in a modification. If a control branch is modified, any corresponding T statement must be respecified so that the transconductance will be recalculated. Maximum and minimum tolerances in modified branches are automatically set to zero if they are not respecified. Output cards and special calculation statements can be included after a MODIFY. *NOTE: The circuit topology cannot be changed by a MODIFY.*

NODAL CURRENT UNBALANCES

As a check on the accuracy of the calculation, ECAP adds currents into each node. The sum of all the currents should be zero. Unless the user specifies otherwise, ECAP will print an error message if the absolute value of the sum of the current into any node is greater than .001 ampere. The user can change the minimum acceptable current unbalance with a 1ERROR statement. The form of the statement is

1E = maximum permissible current unbalance (amps)

SECTION 4

THE AC ANALYSIS MODE

The ECAP AC Analysis Mode performs steady state analyses of circuits excited by a sinusoidal source. This section discusses the input and output requests acceptable in AC analysis and introduces a new statement, the M statement.

The capacity of Tymshare's ECAP AC analysis is

B Statements: 60
T Statements: 10
Nodes: 20
M Statements: 10
Simultaneous Modifications: 20

AC CIRCUITRY

An AC circuit is constructed from the following elements and parameters:

R = resistance (ohms)
G = conductance (mhos)
C = capacitance (farads)
L = inductance (henries)
} *Only one per branch.*

E = sinusoidal voltage source (volts/degrees[1])

I = sinusoidal current source (amps/degrees[1])

BETA = current gain (real[2])

GM = transconductance (real[2])

branch numbers

node numbers

FR = frequency (hertz)

An AC branch contains one and only one passive element. This element may be a resistor, capacitor, or inductor. Both the magnitude and phase must be given for all sources in the form magnitude/phase.

Example

B1 N(0,1),R=12,E=52/0,I=3.5/180
B2 N(1,2),L=5.4
B3 N(2,0),C=2E-3

NOTE: If the controlling branch has an inductor or capacitor, only voltage dependent current sources are permitted. That is, GM rather than BETA should be specified in the T statement.

Every AC circuit must have a frequency statement to specify the frequency of all of the sources. The form of the statement is

FR = frequency (Hz)

The parameter iteration capability of the MODIFY command permits the user to sweep the frequency. The convention for specifying frequency iteration is different from other parameters.[3] The user can specify either a multiplicative factor or the number of linear steps to be taken.

A multiplicative factor is simply enclosed in parentheses:

MO
FR = lower limit (factor) upper limit

The frequency is multiplied by the factor to determine the frequency for the next analysis. The frequency continues to be increased until it is greater than the upper limit. The statements

MO
FR = .03(1.2).1

would perform analyses at frequencies .03, .036, .0432, .05184, etc.

Alternatively, the user can specify the number of frequency steps. These steps are linear rather than multiplicative; that is, a fixed increase is added to each frequency to obtain the next frequency for analysis.

The number of steps is indicated by a plus sign:

FR = lower limits (+ number of steps) upper limit

There is always one more analysis performed than the number of steps. For example,

FR = .04(+8).2

would perform 9 analyses at .04, .06, .08, ..., .2.

The phase and magnitude of the sources are iterated by the conventions discussed in Section 3 with the restriction that phase and magnitude must always be iterated in the same number of steps.

1 - Sinusoidal sources are described by separating the amplitude and phase by a slash.

2 - For modelling of complex BETA or GM, refer to R.W. Jensen and M.D. Lieberman, *IBM Electronic Circuit Analysis Program,* Prentice-Hall, 1968, Page 65.

3 - For comparison, refer to MODIFY on Page 26.

AC OUTPUT

The output quantities in AC Analysis are:

NV, node voltage

BV, branch voltage

CV, element voltage

CA, element current

BA, branch current

BP, power dissipation = Real $(CV \cdot \overline{CA})$
where $\overline{CA}$ is the complex conjugate of CA.

MISC, real and imaginary components of the complex nodal admittance matrix and equivalent current vector.

INDUCTIVELY COUPLED BRANCHES

Mutual inductances can be described to ECAP with an M statement. The form of the M statement is

Mnumber B(branch$_1$,branch$_2$), L = mutual inductance (henries)

Example

M1 B(2,5),L=-3.1

A positive mutual inductance means that positive current creates a voltage rise at the second node of the other branch. A negative mutual inductance means that positive current creates a voltage rise at the first node of the other branch. If one branch is inductively coupled to two other branches, an M statement must be included to define the coupling between those two branches.

If the coupling constant is close to 1, a model for an ideal transformer should be used.[1]

NODAL CURRENT UNBALANCES

The nodal current balance is checked by adding the real and imaginary parts of the currents separately. The total current is determined for each node and then the sum of the absolute value of the unbalances for all the nodes is compared to the 1ERROR.

1 - See Jensen & Lieberman, Op. Cit., Page 119.

SECTION 5
THE TRANSIENT ANALYSIS MODE

ECAP Transient Analysis provides time-dependent solutions for circuits with linear and non-linear elements. This section discusses the basic elements of an ECAP TR program. Refer to Jensen and Lieberman, Op. Cit., for more detailed discussions of TR modelling.

TR Analysis has the following capacity

B Statements: 60

T Statements: 10

Nodes: 20

M Statements: 10

E Statements: 5 (50 values/statement)

I Statements: 5 (50 values/statement)

Switches: As required for 60 switched branches

TR CIRCUITRY

A transient circuit is constructed from the following

R = resistance (ohms)

G = conductance (mhos)

C = capacitance (farads)

L = inductance (henries)

E = time independent voltage

EO = initial voltage

I = time independent current

IO = initial current

BETA = current gain

GM = transconductance

IN = initial time (zero if not specified)

TI = time interval between calculations (seconds)

OU = number of time steps between printout

FI = final time

SH = short circuit resistance for capacitors (ohms)

OP = open circuit resistance for inductors (ohms)

EQ requests equilibrium solution

1E = current unbalance error

2E = switching time error

A TR branch contains one and only one passive element, a resistor, capacitor, or inductor. For dependent current sources, if the controlling branch has a capacitor or inductor, GM rather than BETA should be specified in the T statement.

Every TR circuit must include TIME STEP, OUTPUT INTERVAL, and FINAL TIME specifications. Each is an independent statement. For example,

TI = .05
OU = 10
FI = 1.0

will perform calculations every .05 second, print results at every 10*.05 = .5 second, and stop after 1 second. The table below shows suggested values for time step selections as related to the node of interest. If more than one node is under study and the time steps indicated by the chart will be three or four orders of magnitude different for the two nodes, two different analyses should be run with two different time steps.

Time dependent sources are specified in an E statement or an I statement (see Page 32). Mutual inductances are specified by M statements as discussed in Section 4. Sense switch number 3 can be used to interrupt the calculations to make modifications (see Section 6).

Table Of Time Step Selections

Elements At Node	Time Step
R, C only	.01 RC
R, L only	.01 L/R
L, C only	$.001*2\pi\sqrt{LC}$
R, C & L	.01 RC for $\frac{2RC}{\sqrt{LC}} \leqslant 1$
	$.001*2\pi\sqrt{LC}$ for $\frac{2RC}{\sqrt{LC}} > 1$

The output quantities in TR Analysis are:

NV, node voltage

BV, branch voltage

CV, element voltage

CA, element current

BA, branch current

BP, element power dissipation

MI, nodal conductance matrix and equivalent current vector, error levels, and time step information

TIME-DEPENDENT SOURCES

TR ECAP allows both time-dependent current sources and time-dependent voltage sources. There are three ways to describe the time-dependent sources to ECAP: as non-periodic functions, as general periodic functions, and as sinusoidal functions. A time-dependent voltage source is described in an E statement. The forms of the E statement are:

Non-Periodic

E## (n), $V_1, V_2, \dots, V_k$

where ## = identifying number of branch containing source

n = number of time steps between values

$V_1, V_2, \dots, V_k$ = value of function at time $k \cdot n \cdot TI$

For example,

E17 (4),0,1.5,1.0,.5,0
I6 (10),0,.1,.1,.2,.4,1

Periodic

E## P(n), $V_1, V_2, \dots, V_k$

Sinusoidal (see Figure 7 below)

E## SIN(t),a,k_o,t_o

where t = period of sine wave (seconds)

a = amplitude (volts)

k_o = dc offset (volts)

t_o = time offset (seconds)

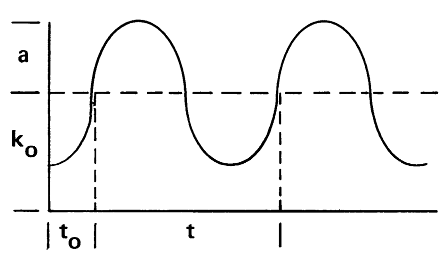

Figure 7 - Sinusoidal Time-Dependent Source

For the periodic and non-periodic functions, as many as 50 source values can be given on each card. A time dependent source will **replace** any fixed source already in the branch.

To assure accuracy and speed, the period of the sine wave must be greater than three times the time step and less than 50 times the time step. If the period is not between these limits, it will be adjusted by ECAP.

SWITCHES

Switches are used to model non-linear characteristics that can be approximated by piecewise linear functions. For a detailed discussion of modelling techniques refer to one of the standard ECAP references. A switch is described in an S statement giving the switch branch, the branches controlled by the switch, and the polarity of the switch. The current direction in the switch branch determines which values will be used for the characteristics of the controlled branches. The form of the statement is

S## B=a, ($b_1, b_2, \dots b_n$), **OFF** *or* **ON**

where ## = switch number

a = switch branch

$b_1, b_2, \dots, b_n$ = controlled branches

Some or all of the parameters in the controlled branches are given two values enclosed in parentheses. For example,

B7 N(3,0),R=(8,12),E=(0,1.5)

If the polarity of the switch is OFF, the first value is in effect when the current in the switch branch is negative, and the second value is in effect when the current in the switch branch is positive. The reverse is true if the polarity of the switch is ON. If only one value is given, that value is maintained regardless of the state of the switch. The polarity of the switch should agree with the initial current direction in the switch branch.

Any branch may be a switch or controlled branch. A branch may even control itself. A parameter may be controlled by several switches. The value of the parameter is determined by the number of switches with current reverse of their polarity (the sum of the number of ON switches with negative current and the number of OFF switches with positive current). If this sum is an even number, the first value is in effect; if this sum is an odd number, the second value is in effect.

The user can specify a 2ERROR which determines the difference between the actual switching time and the program-calculated switching time. The difference is less than 2ERROR*TI. If 2ERROR is not specified by the user, it is set to .001.

INITIAL CONDITIONS

The first solution in any transient analysis and the first solution after any switching is an initial condition solution. All capacitors are replaced by very small resistances and all inductors by large resistances. These resistances are .01 ohms and 10^7 ohms unless otherwise specified by the user. The user can set these resistances with the SHORT and OPEN statements.

SH = resistance substituted for all capacitors

OP = resistance substituted for all inductors

The initial voltages and currents across branches can be specified in the branch statements. EO is the initial voltage and IO is the initial current.

B14 N(12,2),C=1.5E-6,EO=5

Sometimes it is desirable to reduce the time steps immediately after an initial condition solution. This can be done with the 3ERROR statement.

3E=P

If 3ERROR $\geqslant$ 1, there is no reduction in time step.
If 3ERROR $<$ 1, four solutions will be done in the next specified time step. Only the first time step is affected. All subsequent solutions are performed as specified.

EQUILIBRIUM

A transient analysis can be run to calculate only one solution, the steady state solution. A steady state solution is requested simply by typing the statement

EQ

The OPEN resistance is substituted for all capacitors and the SHORT resistance for all inductors. Initial condition values are used for all other parameters. No TI or FI statements are required in an equilibrium solution.

SECTION 6
EXTENDED ECAP

RETRIEVABLE MODELS

Tymshare ECAP can retrieve models from a file. The models stored on the file can be any ECAP circuit descriptions. Branch numbers, node numbers, etc. to be relocated must begin at 1. Node 0 can be included, but it will relocate to node 0 in the ECAP program; that is, node 0 is always ground. Many models may be stored on the same file. Each model begins with

MODEL: XXXXX

where XXXXX is the name of the model. The model name can be of any length, but must not contain any spaces.

The model is retrieved with an N statement. The N statement relocates each node, branch, etc., into the current circuit. The general form of the N statement is

**N-- relocation specifications FI = model file,
MO = model name**

The relocation specifications assign each node and statement number in the model to nodes and statements in the circuit. The specifications have the form:

ri(x_1, x_2, ... , x_n)

where ri is the index type from the relocation index table, and x_1 to x_n are the numbers to be assigned to the normalized model values (1 to n) as they are relocated.

**N1 BR(7,13,8), NO(27,28,29) FI=RETMODS
MO=TIFANY**

relocates B1 as B7, B2 as B13, etc., and node 1 as node 27, etc.

Relocation Index Table

BR	=	B statement
TR	=	T statement
NO	=	nodes
MU	=	M statement
SW	=	S statement
VO	=	E statement
CU	=	I statement

The use of relocatable models does not allow a larger ECAP program; B, T, S, M, E, and I statements must still be, after relocation, numbered sequentially, and be within the size constraints.

Example

Assume that the following model is on the file MODELS:

```
MODEL:  TRANS
B1      N(1,2),R=350,E=-.3
B2      N(3,2),R=10E3
T1      B(1,2),BETA=50
```

N statement in program:

N1 BR(6,13),	NO(7,8,9),	TR(1),	FI=MODELS,	MO=TRANS
Branch Relocation	*Node Relocation*	*T Statement Relocation*	*Model File*	*Model ID*

Relocated model in current analysis:

```
B6   N(7,8),R=350,E=-.3
B13  N(9,8),R=10E3
T1   B(6,13),BETA=50
```

MODIFICATIONS IN TRANSIENT ANALYSIS

Modifications can be made during a transient analysis through the use of sense switch 3. After each calculation, ECAP checks the status of sense switch 3. If it is set, control returns to the terminal and any of the statements listed below can be entered.

Sense switch 3 is set at any time during analysis by typing

3;S ⮐

and, when requested by ECAP, is reset by typing

3;R ⮐

The state of sense switch 3 is checked after each time step, but may appear to be delayed for two reasons:

1. Switching calculations (S statement action) are being performed,
2. Solution output before setting the sense switch is still being transmitted by the computer.

Input to the transient modify routine features A^C and Q^C editing. Allowable transient modify statements which may be shortened to the first two letters are:

INITIAL TIME = XX.XX	Restarts the solution with the time printout as specified.
TIME STEP = XX.XX	Initiates a new time step value.
OUTPUT INTERVAL = XX	Changes the number of solution cycles before output.
FINISH TIME = XX.XX	Redefines the final solution time.
SHORT = XX.XX	Redefines the short circuit value for capacitors for initial condition solutions.
OPEN = XX.XX	Redefines the open circuit value for inductors for initial condition solutions.
1ERROR = XX.XX	Redefines maximum allowable current unbalances.
2ERROR = XX.XX	Redefines switching resolution.
3ERROR = XX.XX	Redefines time step reduction for initial condition solutions.
EXECUTE	Continues the solution.
END	Returns to the EXECUTIVE.

APPENDIX

POPULAR ECAP UNITS

Listed below are the two most popular sets of units for ECAP parameters.

Parameter	Set 1 Ohms-Amps	Set 2 Kilohms-Milliamps
R	Ohms	Kilohms
G	Mhos	Millimhos
C	Farads	Picofarads
L	Henries	Microhenries
E	Volts	Volts
I	Amperes	Milliamperes
BETA	(Unitless)	(Unitless)
GM	Mhos	Millimhos
Time	Seconds	Nanoseconds
Frequency	Hertz	Gigahertz
Phase angle	Degrees	Degrees

BIBLIOGRAPHY

1. IBM Corporation, *1620 Electronic Circuit Analysis Program, H20-0170-1*, White Plains, New York, IBM Corporation, 1965.
2. Jensen, Randal W. and Mark D. Lieberman, *IBM Electronic Circuit Analysis Program*, Englewood Cliffs, New Jersey, Prentice-Hall, 1963.

INDEX

NOTE: Page numbers which appear in **bold** *face type refer to those pages where the listed item receives the most detailed discussion.*