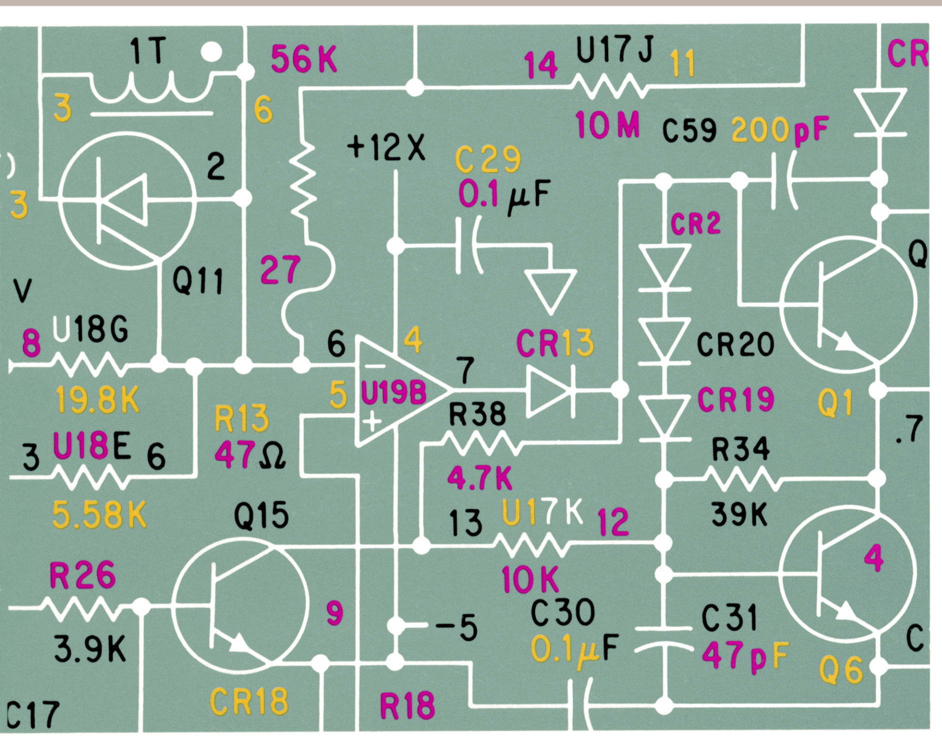
HEWLETT-PACKARD

Step-by-Step Solutions
For Your HP Calculator

Electrical Engineering



HP-42S

HEWLETT
PACKARD

Electrical Engineering

Step-by-Step Solutions for Your HP-42S Calculator

HEWLETT PACKARD

Edition 1 November 1988
Reorder Number 00042-90021

Notice

Corvallis Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.

Printing History

Edition 1 November 1988 Mfg. No. 00042-90023

Contents

How To Use This Book

Please take a moment to familiarize yourself with the formats used in this book.

Organization. Each chapter in this book covers a different area of electrical engineering. Sections within each chapter highlight the use of each program. The sections are organized like this:

- Description of the program, including equations and variables used.
- Special remarks and limitations.
- General instructions.
- Keystroke examples.
- Program listings.

About the Examples. Unless otherwise stated, the keystrokes and displays shown in each section assume the following conditions:

- The required programs have been keyed into the calculator.
- The stack is clear and you're using the specified display format. Generally, this does not affect the results of the example, but your displays may not exactly match the ones in this book.
- The SIZE is set to 25 registers (the default). The number of registers needed (if any) is listed under "Remarks."

As you work the examples, remember that lowercase letters are displayed as uppercase letters when they appear in menu labels.

If You Have a Printer. Many of the programs in this book will produce printed output if printing is enabled. Press ■[PRINT] [▲] PON to enable printing.

If you are not using a printer, be sure to disable printing (■[PRINT] [▲] POFF) to avoid losing results.

About Program Listings. It is assumed that you understand how to key programs into your calculator. If you're not sure, review part 2, "Programming," in the owner's manual.

If you print your programs, remember that the printer may print some characters differently than they are displayed. (For example, the → character is printed as ⊬.) Also note that some printers cannot print the angle character (∡).

About the Subject Matter. Discussions on the various topics included are beyond the scope of this book. Refer to basic texts on the subjects of interest. Many references are available in university libraries and in technical and college bookstores. The examples in this book demonstrate approaches to solving problems, but they do not cover the many ways to approach general problems in electrical engineering.

Our thanks to Dex Smith of TwentyEighth Street Publishing for developing this book.

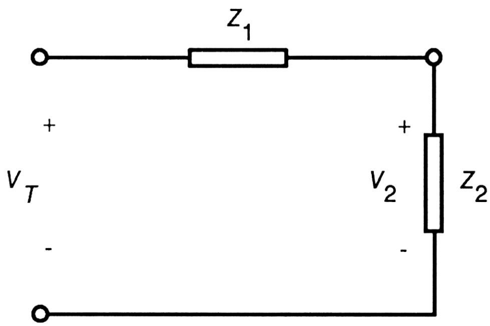
1

Circuit Calculations

This chapter contains programs that solve for an unknown circuit parameter (when the other parameters are known), define a power triangle from voltage and current, and convert impedances between delta and wye circuit configurations.

Voltage Division ("V÷")

For a circuit in the following general form, the "V÷" program solves for any of the four complex values provided the other three are known.



$$V_2 = \frac{Z_2 V_T}{Z_1 + Z_2}$$

Variables Used.

In Equation	Description	In Program
V_T	Terminal voltage (volts).	VT
V_2	Voltage across impedance Z_2.	V2
Z_1	Impedance (ohms).	Z1
Z_2	Impedance (ohms).	Z2

Since any of these values can be a complex number, the Solver cannot be used. The following program ("V÷") emulates the Solver by displaying a menu containing the four variables in the above equation.

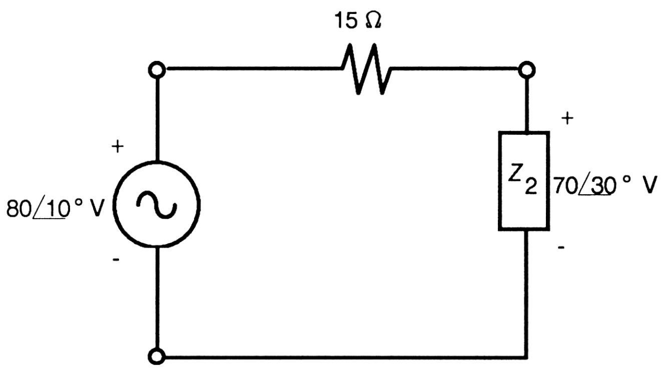
Remarks.

- "V÷" does not alter the angular and coordinate modes; you may use them as you wish when keying in complex values.
- Flag 21 (*printer enable*) is set or cleared to match flag 55 (*printer existence*). This automatically produces printer output if flag 55 is set.

Program Instructions.

1. Key the "V÷" program (listed on page 10) into your calculator.
2. Press [XEQ] V÷ (to run the "V÷" program).
3. Use the variable menu displayed by the program to store the known values.
4. Press the key for the value you want to calculate.
5. To work another problem, go to step 3; to quit, press [EXIT].

Example. Given the following circuit and voltage measurements, what must the impedance, Z_2, be?



Select Degrees and Polar modes, select the FIX 2 display format, and run the "V÷" program.

■[MODES] DEG ■[MODES] POLAR
■[DISP] FIX 02 [XEQ] V÷

```
x: 0.00
 VT  V2  Z1  Z2
```

Store the known values.

80 [ENTER] 10 ■[COMPLEX] VT

```
VT=80.00 ∡10.00
 VT  V2  Z1  Z2
```

15 [Z1]

```
Z1=15.00
 VT  V2  Z1  Z2
```

70 [ENTER] 30 ■[COMPLEX] [V2]

```
V2=70.00 ∡30.00
 VT  V2  Z1  Z2
```

Now, solve for the unknown impedance.

[Z2]

```
Z2=37.71 ∡79.29
 VT  V2  Z1  Z2
```

The unknown impedance (Z_2) is 37.71 ∡79.29° ohms.

"V÷" Program Listing.

Program:	**Comments:**
`00 { 148-Byte Prgm }`	

`01 LBL "V÷"`
`02 MVAR "VT"`
`03 MVAR "V2"`
`04 MVAR "Z1"`
`05 MVAR "Z2"`
	Defines menu variables.

`06 CF 21`
`07 FS? 55`
`08 SF 21`
	Sets or clears flag 21 to match flag 55.

`09 LBL 00`
`10 CLA`
`11 VARMENU "V÷"`
`12 STOP`
	Displays the variable menu and stops.

```
13 ATOX
14 ATOX
15 -
16 XEQ IND ST X
17 GTO 00
```

Determines which variable was selected by subtracting the ASCII codes of the first two characters in the variable name. For example, when you press Z1 to calculate *Z1*, the program branches to LBL 41 because the ASCII code of "Z" is 90, the ASCII code of "1" is 49, and 90 − 49 = 41.

```
18 LBL 36
19 RCL "Z2"
20 RCL× "VT"
21 RCL "Z1"
22 LASTX
23 +
24 ÷
25 STO "V2"
26 VIEW "V2"
27 RTN
```

Calculates *V2*.

```
28 LBL 02
29 RCL "Z1"
30 RCL+ "Z2"
31 RCL× "V2"
32 RCL÷ "Z2"
33 STO "VT"
34 VIEW "VT"
35 RTN
```

Calculates *VT*.

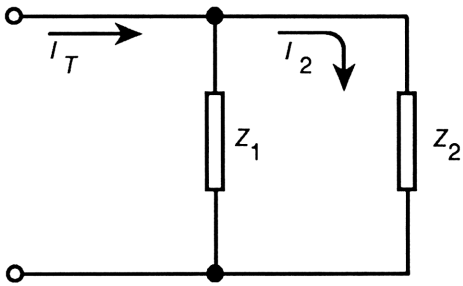
```
36 LBL 40
37 RCL "V2"
38 RCL× "Z1"
39 RCL "VT"
40 LASTX
41 -
42 ÷
43 STO "Z2"
44 VIEW "Z2"
45 RTN
```

Calculates *Z2*.

```
46 LBL 41                Calculates Z1.
47 RCL "Z2"
48 RCL× "VT"
49 RCL÷ "V2"
50 RCL- "Z2"
51 STO "Z1"
52 VIEW "Z1"
53 END
```

Current Division ("I÷")

For a circuit in the following general form, the "I÷" program solves for any of the four complex values provided the other three are known.



$$I_2 = \frac{Z_1 I_T}{Z_1 + Z_2}$$

Variables Used.

In Equation	Description	In Program
I_T	Terminal current (amps).	`I.T`
I_2	Current in impedance Z_2.	`I2`
Z_1	Impedance$_1$ (ohms).	`Z1`
Z_2	Impedance$_2$ (ohms).	`Z2`

Since any of these values can be a complex number, the Solver cannot be used. The following program ("I÷") emulates the Solver by displaying a menu containing the four variables in the above equation.

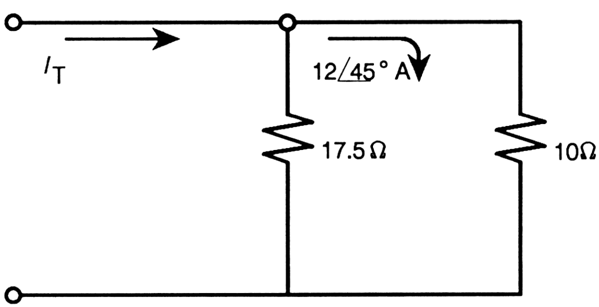
Remarks.

- "I÷" does not alter the angular and coordinate modes. You may use them as you wish when keying in complex values.
- Flag 21 (*printer enable*) is set or cleared to match flag 55 (*printer existence*). This automatically produces printer output if flag 55 is set.

Program Instructions.

1. Key the "I÷" program (listed on page 15) into your calculator.
2. Press [XEQ] I÷ (to run the "I÷" program).
3. Use the variable menu displayed by the program to store the known values.
4. Press the key corresponding to the value you want to calculate.
5. To work another problem, go to step 3; to quit, press [EXIT].

Example. Given the following circuit, what is the input current (I_T) if the current (I_2) through the 10Ω resistor is 12 ∡45° amperes?



Select Degrees and Polar modes, select the FIX 2 display format, and run the I÷ program.

■[MODES] DEG ■[MODES] POLAR
■[DISP] FIX 02 [XEQ] I÷

x: 0.00
I.T I2 Z1 Z2

Key in the three known values.

17.5 Z1

Z1=17.50
I.T I2 Z1 Z2

10 Z2

```
Z2=10.00
 I.T | I2 | Z1 | Z2 |  |
```

12 [ENTER] 45 ■[COMPLEX] I2

```
I2=12.00 ∡45.00
 I.T | I2 | Z1 | Z2 |  |
```

Now calculate the unknown current.

I.T

```
I.T=18.86 ∡45.00
 I.T | I2 | Z1 | Z2 |  |
```

The unknown current (I_T) is 18.86 ∡45° amperes.

"I÷" Program Listing.

Program:	Comments:
00 { 155-Byte Prgm }	
01 LBL "I÷" 02 MVAR "I.T" 03 MVAR "I2" 04 MVAR "Z1" 05 MVAR "Z2"	Defines menu variables.
06 CF 21 07 FS? 55 08 SF 21	Sets or clears flag 21 to match flag 55.
09 LBL 00 10 CLA 11 VARMENU "I÷" 12 STOP	Displays the variable menu and stops.
13 ATOX 14 ATOX 15 - 16 XEQ IND ST X 17 GTO 00	Determines the selected variable by subtracting the ASCII codes of the first two characters in the variable name. For example, when you press I.T , the program branches to LBL 27 because the ASCII code of "I" is 73, the ASCII code of "." is 46, and 73 − 46 = 27.

Calculates *I2*.

```
18 LBL 23
19 RCL "Z1"
20 RCL× "I.T"
21 RCL "Z2"
22 LASTX
23 +
24 ÷
25 STO "I2"
26 VIEW "I2"
27 RTN
```

Calculates *I.T*.

```
28 LBL 27
29 RCL "Z2"
30 RCL+ "Z1"
31 RCL× "I2"
32 RCL÷ "Z1"
33 STO "I.T"
34 VIEW "I.T"
35 RTN
```

Calculates *Z2*.

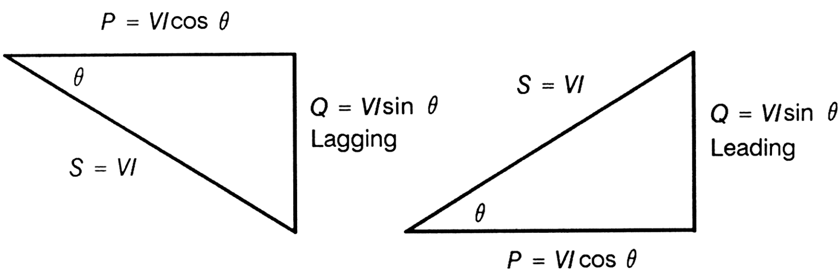
```
36 LBL 40
37 RCL "Z1"
38 RCL× "I.T"
39 RCL÷ "I2"
40 RCL- "Z1"
41 STO "Z2"
42 VIEW "Z2"
43 RTN
```

Calculates *Z1*.

```
44 LBL 41
45 RCL "I2"
46 RCL× "Z2"
47 RCL "I.T"
48 LASTX
49 -
50 ÷
51 STO "Z1"
52 VIEW "Z1"
53 END
```

Power Triangle ("PWR3")

The "PWR3" program calculates any value for the power triangle, provided that certain other values are known.



Variables Used.

In Figure	Description	In Program
V	Voltage (volts).	V
I	Current (amperes).	I
P	Average power (watts).	P
Q	Reactive power (vars).	Q
S	Apparent power (watts).	S
	Power factor (cos θ).	Pf

Remarks.

- Be sure to enter *V* and *I* as complex numbers. If the values in *V* and *I* are not complex numbers, the program will stop and display `Invalid Type`. If you generate this error, *restart* the program by pressing [EXIT] [XEQ] PWR3 .
- The "PWR3" program sets Degrees and Polar modes.

- The "PWR3" program uses flag 10 to control the calculations of *V* and *I*.
- A minus sign preceding a (result or input) value for power factor indicates a *lagging* power factor.

Program Instructions.

1. Key the "PWR3" program (listed on page 20) into your calculator.
2. Press [XEQ] PWR3 (to run the "PWR3" program).
3. Use the variable menu displayed by the program to store the known values and to calculate the unknowns:

   To calculate *P*, *Q*, *S*, or *pf*:

   a. Key in the voltage (in *polar* form) and press V.
   b. Key in the current (in *polar* form) and press I.
   c. Calculate any of the four values by pressing P, Q, S, or PF.

   To calculate the voltage, *V*:

   a. Key in the current (in *polar* form) and press I.
   b. Key in the apparent power and press S.
   c. Key in the power factor and press PF.
   d. Calculate the voltage by pressing V.

   To calculate the current, *I*:

   a. Key in the voltage (in *polar* form) and press V.
   b. Key in the apparent power and press S.
   c. Key in the power factor and press PF.
   d. Calculate the current by pressing I.
4. To work another problem, go to step 3; to quit, press [EXIT].

Example. For a circuit with an applied voltage of 100 ∡10° volts and a resulting current of 2.85 ∡ −40° amperes, determine the power triangle and the power factor.

Select the FIX 2 display format, and run the "PWR3" program.

■[DISP] FIX 02 [XEQ] PWR3	x: 0.00 (V I P Q S PF)
100 [ENTER] 10 ■[COMPLEX] V	V=100.00 ∡10.00 (V I P Q S PF)
2.85 [ENTER] 40 [+/−] ■[COMPLEX] I	I=2.85 ∡-40.00 (V I P Q S PF)
P	P=183.19 (V I P Q S PF)

The average power is 183.19 watts.

Q	Q=218.32 (V I P Q S PF)

The reactive power is 218.32 vars. (The *sign* of the power factor (pf) indicates if Q is a leading or a lagging value.)

S	S=285.00 (V I P Q S PF)

The apparent power is 285 watts.

PF	pf=-0.64 (V I P Q S PF)

The power factor is 0.64. Because a minus sign indicates a lagging power factor, the reactive power, Q, is also lagging. If all other variables remain unchanged, what voltage would be required to increase the apparent power to 300 watts?

300 S	S=300.00 (V I P Q S PF)

V

```
V=105.26 ∡10.00
 V    I    P    Q    S    PF
```

The required voltage is 105.26 ∡10.00° volts.

"PWR3" Program Listing.

Program:	Comments:
00 { 193-Byte Prgm } 01 LBL "PWR3" 02 MVAR "V" 03 MVAR "I" 04 MVAR "P" 05 MVAR "Q" 06 MVAR "S" 07 MVAR "pf"	Declares the menu variables.
08 DEG 09 POLAR 10 CF 21 11 FS? 55 12 SF 21	Sets Degrees and Polar modes. Sets or clears flag 21 to match flag 55.
13 LBL A 14 VARMENU "PWR3" 15 CLA 16 STOP 17 ATOX 18 X=0? 19 GTO A	Displays variable menu and stops. Pressing R/S redisplays the menu.
20 XTOA 21 1 22 AROT 23 R↓ 24 73 25 - 26 XEQ IND ST X	The ASCII character code of the first letter in the variable name is subtracted from 73 to determine which routine to execute.

```
27 ASTO ST L
28 STO IND ST L
29 VIEW IND ST L
30 GTO A
```

Stores the variable name into the Last X-register and displays the result.

```
31 LBL 13
32 RCL "I"
33 SF 00
34 XEQ B
35 RTN
```

Calculates *V*.

```
36 LBL 00
37 RCL "V"
38 CF 00
39 XEQ B
40 RTN
```

Calculates *I*.

```
41 LBL 07
42 XEQ 39
43 RCL "V"
44 RCL× "I"
45 ABS
46 RCL× "pf"
47 ABS
48 RTN
```

Calculates *P*.

```
49 LBL 08
50 XEQ 39
51 XEQ C
52 RCL "pf"
53 ACOS
54 SIN
55 ×
56 RTN
```

Calculates *Q*.

```
57 LBL 10
58 XEQ C
59 ABS
60 RTN
```

Calculates *S*.

```
61 LBL B
62 COMPLEX
63 RCL "pf"
64 FS? 00
65 +/-
66 ACOS
67 +
68 ENTER
69 ENTER
70 RCL "S"
71 R↑
72 ÷
73 X<>Y
74 COMPLEX
75 FC?C 00
76 +/-
77 RTN
```

Calculates *V* if flag 00 is set, or *I* if flag 00 is clear.

```
78 LBL C
79 RCL "I"
80 RCL× "V"
81 ABS
82 RTN
```

Calculates the magnitude of *VI*.

```
83 LBL 39
84 RCL "V"
85 COMPLEX
86 RCL "I"
87 COMPLEX
88 X<>Y
89 R↓
90 -
91 COS
92 LASTX
93 SIGN
94 +/-
95 ×
96 STO "pf"
97 END
```

Calculates *pf*.

Frequency Response of Transfer Function ("FQRS")

For a transfer function of the form

$$G(S) = \frac{K(Z_1S + 1)}{S^N (Z_2S + 1)(Z_3S + 1)\left(\frac{S^2}{\omega_0^2} + \frac{2Z_4S}{\omega_0} + 1\right)}$$

the "FQRS" program calculates G(S) and log |G(S)| for any input frequency ω (where $S = j\omega$).

Variables Used.

In Equation	Description	In Program
K	Transfer function parameter.	K
N	Transfer function parameter.	N
Z_1	Transfer function parameter.	Z1
Z_2	Transfer function parameter.	Z2
Z_3	Transfer function parameter.	Z3
Z_4	Transfer function parameter.	Z4
ω_0	Transfer function parameter.	ω0
ω	Input frequency, 2πf (radians/sec).	ω

Remarks.

- For type 0 systems, enter $N = 0$.
- Z_1, Z_2, and Z_3 can be entered as 0. If there is no quadratic term, enter Z_4 as 0 and ω_0 very large compared to $1/Z_3$, where Z_3 is the smallest first-order term used (other than zero).
- The "FQRS" program sets Degrees and Polar modes.

Program Instructions.

1. Key the "FQRS" program (listed on page 26) into your calculator.
2. Press [XEQ] FQRS (to run the "FQRS" program).
3. The program prompts you for values of K, N, Z_1, Z_2, Z_3, Z_4, and ω_0. At each prompt, key in the value and press [R/S].
4. The program displays G(S) and log |G(S)| and stops. Press [R/S] to go to step 3 for another problem.

Example. Find G(S) and log |G(S)| for

$$G(S) = \frac{12(S + 0.6)}{S(S + 1)(S^2 + 6S + 36)}$$

The frequency, ω, is 0.01 rad/sec.

First put G (S) into proper form:

$$G(S) = \frac{.2(1.67S + 1)}{S(S + 1)\left[\left(\frac{S}{6}\right)^2 + \left(\frac{S}{6}\right) + 1\right]}$$

Select the FIX 2 display format and run the "FQRS" program.

Keystrokes	Display
■ [DISP] FIX 02 [XEQ] FQRS	Y: 0.00 K?0.00
.2 [R/S]	Y: 0.20 N?0.00
1 [R/S]	Y: 1.00 Z1?0.00
1.67 [R/S]	Y: 1.67 Z2?0.00

1 [R/S]

```
Y: 1.00
Z3?0.00
```

0 [R/S]

```
Y: 0.00
Z4?0.00
```

.5 [R/S]

```
Y: 0.50
w0?0.00
```

6 [R/S]

```
Y: 6.00
w?0.00
```

.01 [R/S]

```
G(S)=20.00 ∡-89.71
LOG |G(S)|=1.30
```

G(S) and log |G(S)| for the given conditions are 20.00 ∡ −89.71° and 1.30.

“FQRS” Program Listing.

Program: | **Comments:**

```
00 { 159-Byte Prgm }
01 LBL "FQRS"
02 INPUT "K"
03 INPUT "N"
04 INPUT "Z1"
05 INPUT "Z2"
06 INPUT "Z3"
07 INPUT "Z4"
08 INPUT "w0"
```

Prompts for each input value.

```
09 LBL 00
10 INPUT "w"
11 DEG
12 RECT
13 1
14 X<>Y
15 RCL÷ "w0"
16 X↑2
17 2
18 RCL× "Z4"
19 RCL× "w"
20 RCL÷ "w0"
21 +
22 COMPLEX
23 1
24 RCL "Z3"
25 RCL× "w"
26 COMPLEX
27 ×
28 1
29 RCL "Z2"
30 RCL× "w"
31 COMPLEX
32 ×
33 POLAR
34 COMPLEX
35 90
```

Calculates G(*S*).

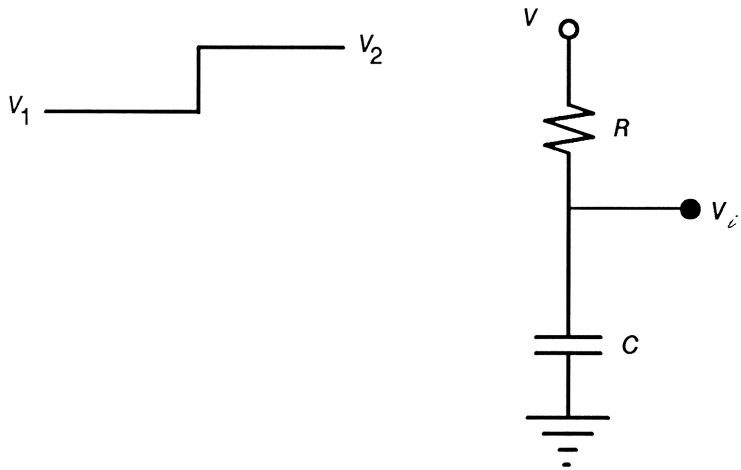
```
36 RCL× "N"
37 +
38 +/-
39 X<>Y
40 RCL "ω"
41 RCL "N"
42 Y↑X
43 ×
44 1/X
45 X<>Y
46 COMPLEX
47 RECT
48 RCL "K"
49 ENTER
50 RCL× "Z1"
51 RCL× "ω"
52 COMPLEX
53 ×
```

`54 POLAR`	Displays results.

```
55 "G(S)="
56 ARCL ST X
57 ABS
58 LOG
59 ⊢"␊LOG |G(S)|="
60 ARCL ST X
61 AVIEW
62 END
```

RC Timing ("RC")

The "RC" Solver program computes any of the six variables in the following figure and equation, provided the other five are known.



$$V_i = V_1 e^{(-t/RC)} + V_2\left(1 - e^{(-t/RC)}\right)$$

By rearranging terms, the equation can be written as:

$$\left(\frac{tC}{\left[-e\left(1 - \frac{V_i - V_1}{V_2 - V_1}\right)\right]}\right) - R = 0.$$

where *e* is the base of natural logarithms.

Variables Used.

In Equations	Description	In Program
V_1	Voltage before step (volts).	V1
V_2	Voltage after step (volts).	V2
V_i	Instantaneous voltage (volts).	Vi
R	Resistance (ohms).	R
C	Capacitance (farads).	C
t	Time (seconds).	t

Remarks.

- The "RC" Solver program uses only *real*-number inputs.
- For voltages across the resistor and capacitor, $V_R + V_C = V$ applies *at all times*.

Program Instructions.

1. Key the "RC" Solver program (listed on page 31) into your calculator.
2. Select the "RC" Solver program: press ■[SOLVER] RC.
3. Use the variable menu to store the five known variables: key in the value and press the corresponding key.
4. Press the key corresponding to the unknown variable. The Solver searches for the unknown and displays the solution (if one can be found).
5. To work another problem, go to step 3, or press [EXIT] [EXIT] to quit.

Example. A 555 type of integrated circuit timer uses an external RC configuration for time determination. When used as a one-shot, its output pulse terminates when the capacitor charges to two-thirds of the supply voltage. Until the pulse starts, the capacitor is shorted so $V1 = 0$. Given a supply voltage of 12V and a 47 μF capacitor, what size resistor should you use to generate a one-second pulse?

Select the ENG 3 display format and the "RC" Solver program.

■ [DISP] ENG 03 ■ [SOLVER] RC	x: 0.000E0 V1 V2 VI R C T
0 V1	V1=0.000E0 V1 V2 VI R C T
12 V2	V2=12.00E0 V1 V2 VI R C T
2 [ENTER] 3 [÷] 12 [×] VI	Vi=8.000E0 V1 V2 VI R C T
47 [E] 6 [+/-] C	C=47.00E-6 V1 V2 VI R C T
1 T	t=1.000E0 V1 V2 VI R C T
R	R=19.37E3 V1 V2 VI R C T

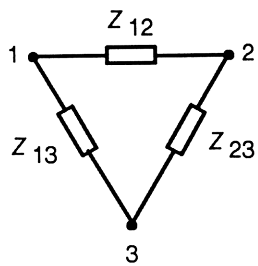
Use a 19.37 Kohm resistor.

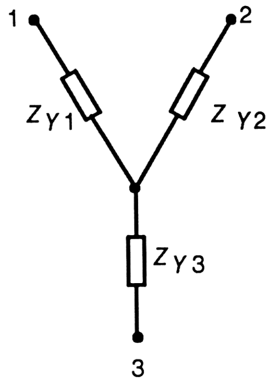
"RC" Program Listing.

```
00 { 58-Byte Prgm }
01 LBL "RC"
02 MVAR "V1"
03 MVAR "V2"
04 MVAR "Vi"
05 MVAR "R"
06 MVAR "C"
07 MVAR "t"
08 1
09 RCL "Vi"
10 RCL "V1"
11 -
12 RCL "V2"
13 LASTX
14 -
15 ÷
16 -
17 LN
18 +/-
19 RCL× "C"
20 RCL÷ "t"
21 1/X
22 RCL- "R"
23 END
```

Delta-Wye Conversions ("DY")

This program allows you to convert impedance values between delta and wye configurations. That is, given the three wye values (Z_{Y1}, Z_{Y2}, and Z_{Y3}), you can calculate any of the three delta values (Z_{12}, Z_{23}, and Z_{13}). Likewise, given the delta values, you can calculate any of the wye values.





Variables Used.

In Figure	Description	In Program
Z_{Y1}	Impedance (ohms).	ZY1
Z_{Y2}	Impedance (ohms).	ZY2
Z_{Y3}	Impedance (ohms).	ZY3
Z_{12}	Impedance (ohms).	Z12
Z_{13}	Impedance (ohms).	Z13
Z_{23}	Impedance (ohms).	Z23

Remarks.

- Flag 21 (*printer enable*) is set or cleared to match flag 55 (*printer existence*). This automatically produces printer output if flag 55 is set.
- The program will give erroneous results (original inputs) if the *calculated* outputs are used directly as inputs.

Program Instructions.

1. Key the "DY" program (listed on page 35) into your calculator.
2. Select the coordinate mode you want to use: press ■[MODES] RECT for Rectangular mode, or ■[MODES] POLAR for Polar mode.
3. Press [XEQ] DY to run the "DY" program.
4. The program displays a variable menu containing the six variables in the above illustration.

   To calculate wye values,

   a. Store each of the three delta values by keying in the value and pressing the corresponding menu key.

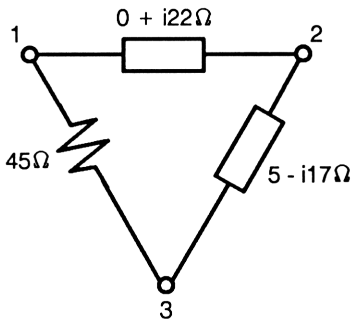
   b. Press the key for the wye value you want to calculate. (Repeat for each of the other two unknown wye values.)

   To calculate delta values,

   a. Store each of the three wye values by keying in the value and pressing the corresponding menu key.

   b. Press the key for the delta value you want to calculate. (Repeat for each of the other two unknown delta values.)
5. You can work as many problems as you want. The menu stays active until you press [EXIT] or select an application menu.

Example. Given the following delta circuit, determine the equivalent wye circuit.



Select the Rectangular mode and the FIX 4 display format, and run the "DY" program.

■[MODES] RECT ■[DISP] FIX 04 [XEQ] DY	x: 0.0000 ZY1 ZY2 ZY3 Z12 Z23 Z13

Enter the delta values.

45 Z13	Z13=45.0000 ZY1 ZY2 ZY3 Z12 Z23 Z13
0 [ENTER] 22 ■[COMPLEX] Z12	Z12=0.0000 i22.0000 ZY1 ZY2 ZY3 Z12 Z23 Z13
5 [ENTER] 17 [+/-] ■[COMPLEX] Z23	Z23=5.0000 -i17.0000 ZY1 ZY2 ZY3 Z12 Z23 Z13

Now calculate each of the wye values.

ZY1	ZY1=1.9604 i19.6040 ZY1 ZY2 ZY3 Z12 Z23 Z13

ZY2 — ZY2=7.6238 i1.4376

ZY1 | ZY2 | ZY3 | Z12 | Z23 | Z13

ZY3 — ZY3=2.9406 -i15.5941

ZY1 | ZY2 | ZY3 | Z12 | Z23 | Z13

"DY" Program Listing.

Program:	Comments:
00 { 219-Byte Prgm }	
01 LBL "DY" 02 MVAR "ZY1" 03 MVAR "ZY2" 04 MVAR "ZY3" 05 MVAR "Z12" 06 MVAR "Z23" 07 MVAR "Z13"	Declares menu variables.
08 CF 21 09 FS? 55 10 SF 21	Sets or clears flag 21 to match flag 55.
11 LBL A 12 VARMENU "DY" 13 STOP 14 ATOX 15 XTOA 16 ATOX 17 XTOA 18 ATOX 19 XTOA	Returns the ASCII character codes of the selected variable name and restores the name in the Alpha register.
20 + 21 95 22 - 23 XEQ IND ST X	Adds the ASCII codes of the last two characters of the selected variable and subtracts 95 to determine which subroutine to execute.

```
24 SF 25
25 ÷
26 ASTO ST L
27 FC?C 25
28 XEQ 03
29 STO IND ST L
30 FC? 50
31 VIEW IND ST L
32 GTO A
```

Stores the variable name in the Last X register and displays the result. Division by zero is detected with flag 25 (*error ignore*) and an appropriate message is displayed. If this happens, line 31 is skipped because flag 50 (*message*) is set.

```
33 LBL 43
34 RCL "Z12"
35 RCL× "Z13"
36 GTO 01
```

Calculates Z_{Y1}.

```
37 LBL 44
38 RCL "Z12"
39 GTO 00
```

Calculates Z_{Y2}.

```
40 LBL 45
41 RCL "Z13"
```

Calculates Z_{Y3}.

```
42 LBL 00
43 RCL× "Z23"
44 LBL 01
45 RCL "Z12"
46 RCL+ "Z13"
47 RCL+ "Z23"
48 RTN
```

Shared subroutines.

```
49 LBL 04
50 RCL "ZY3"
51 GTO 02
```

Calculates Z_{12}.

```
52 LBL 05
53 RCL "ZY2"
54 GTO 02
```

Calculates Z_{13}.

```
55 LBL 06
56 RCL "ZY1"
```

Calculates Z_{23}.

```
57 LBL 02
58 RCL "ZY1"
59 RCL× "ZY2"
60 LASTX
61 RCL× "ZY3"
62 +
63 RCL "ZY2"
64 RCL× "ZY3"
65 +
66 X<>Y
67 RTN
```

Shared subroutine.

```
68 LBL 03
69 ⊢"=Open Circuit"
70 9.99E499
71 AVIEW
72 END
```

Division by zero indicates an open circuit (infinite resistance). Infinity is approximated with 9.99×10^{499}.

2

Network Analysis

This chapter contains programs for performing mesh and nodal network analyses for circuits containing any combination of resistors, capacitors, inductors, and general impedances. Circuits for mesh analysis may also contain voltage sources and circuits for nodal analysis may contain current sources.

Circuits are entered into the calculator using the "CIRCT" program and then processed by running the "MESH" or "NODAL" program.

Using the Circuit Editor ("CIRCT")

"CIRCT" is a menu-driven program that allows you to add, delete, and print elements in a circuit. The circuit elements are stored in a matrix named *CIRCT*, where each row in the matrix contains an element.

Required Programs. "CIRCT" (page 45), "EL?" (page 151), and "Y?N" (page 157).

Starting the "CIRCT" Program

If there are no elements stored in *CIRCT* (if no circuit exists), you'll see this display when you run the "CIRCT" program:

[XEQ] CIRCT

```
{ 0-Element Circuit }
 ADD
```

Or, if the matrix *CIRCT* exists, the program assumes that it contains *n* circuit elements (where *n* is the number of rows in the matrix), and you'll see a display like this:

[XEQ] CIRCT

```
{ 7-Element Circuit }
 ADD  DEL            PRINT INIT
```

You can view the elements in the circuit by using the [▲] and [▼] keys.

Note If the *CIRCT* matrix exists but does not contain circuit information that the "CIRCT" program recognizes, the program can be thrown far off course. To prevent this from happening, either clear the *CIRCT* variable (■[CLEAR] CLV CIRCT) before starting the program, or initialize the circuit as shown in the next section.

Initializing the Circuit

To initialize *CIRCT* (that is, to delete the *CIRCT* matrix), press INIT YES.

INIT

```
INITIALIZE Circuit?
YES                    NO
```

YES

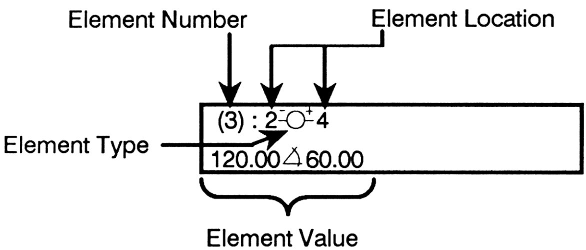
```
{ 0-Element Circuit }
ADD
```

When there are no circuit elements, note that the DEL, PRINT, and INIT menu keys are *not* displayed or active.

Displaying Circuit Elements

Use the [▲] and [▼] keys to move up and down through the list of elements.

Here's a typical display:



For nodal analysis this display reads: "The third circuit element is a voltage source connected between node 2 and node 4 with a value of 120.00 ∠60.00° volts." (For mesh analysis, the location numbers would represent *meshes* 2 and 4.)

When a circuit element is displayed, the menu keys ADD, DEL, PRINT, and INIT are active even though they are not visible. (If you press [◄] to clear the message, you'll see that the menu *is* active.)

Adding Circuit Elements

1. Use the [▲] and [▼] keys to move to the position in the list of circuit elements where you want to add an element. The new element will be inserted *after* the displayed element.
2. Press ADD, the *first* menu key. (The menu keys are active, even though the message temporarily covers them.) The program displays `Location: # [ENTER] #` and a menu of the six types of circuit elements.
3. Key in the location of the element.

   For mesh analysis: mesh# [ENTER] mesh#, where the two mesh numbers indicate the meshes that share the element. For an element in a single mesh, press mesh# [ENTER].

   For nodal analysis: from-node# [ENTER] to-node#, where the circuit element is connected from the first node to the second.

   The order of the location numbers is important only when entering a voltage or current source. For example, the illustration on page 40 shows a voltage *rise* from node 2 to node 4. If the node numbers had been entered in the opposite order, the value should be negative to represent a voltage *drop* from node 4 to node 2.
4. Press a menu key to specify the *type* of element you want to add:

   R (resistor, ohms)

   C (capacitor, farads)

   L (inductor, henrys)

   Z (general impedance, ohms)

   V (voltage source, volts)

   I (current source, amperes)
5. The program displays a menu of the common units for the element you're entering. Key in the value of the element and then press the appropriate key. (For example, to enter a 1,000-ohm resistor you could press 1000 OHM or 1 KOHM.)

   All elements are converted to their default units (see the table on page 44) before being added to *CIRCT*.

To add another element, go to step 3. To return to the main menu, press [EXIT].

Examples later in this chapter demonstrate how to use the "CIRCT" program for entering circuit elements for mesh and nodal analyses.

Deleting a Circuit Element

To delete a circuit element:

1. Use the [▲] and [▼] keys to display the element you want to delete.
2. Press `DEL`, the *second* menu key. (The menu keys are active, even though the message temporarily covers them.)
3. To prevent accidentally deleting an element, the program displays `DELETE Element?` Press `YES` to complete the operation. (Press `NO` or [EXIT] if you change your mind.)

Printing the Circuit

If you want a printed record of the circuit, press `PRINT`, the fifth menu key. Note that this printing is slower than most other printing operations because it involves printing graphics in the display.

The following sample output was printed after entering the circuit on page 54:

```
{ 7-Element Circuit }

(1):1 -o+ 1
150.00 ∡10.00

(2):1 -w- 1
16.00

(3):1 -▭- 2
0.00 i20.00

(4):1 -▭- 3
0.00 -i24.00

(5):2 -w- 2
32.00

(6):2 -w- 3
12.00

(7):3 -▭- 3
0.00 -i16.00
```

Quitting "CIRCT"

Press [EXIT] to exit the "CIRCT" program. After exiting, if you don't run any other programs (or move the program pointer in any other way), you can restart "CIRCT" by pressing [R/S].

Saving Circuits

To save a copy of your circuit:

1. Press [EXIT] to quit "CIRCT".
2. Press [RCL] CIRCT to recall a copy of *CIRCT*.
3. Press [STO] [ENTER] *name* [ENTER] to store a copy of *CIRCT* (where *name* is the name of a new variable).
4. Press [R/S] to restart the "CIRCT" program.

To restore a saved copy of a circuit:

1. Press [EXIT] to quit "CIRCT".
2. Press [RCL] name to recall a copy of a circuit that's been saved.
3. Press [STO] CIRCT to store a copy of the matrix into *CIRCT*.
4. Press [R/S] to restart the "CIRCT" program.

Storing Elements

The *CIRCT* variable contains a three-column matrix. Each row in the matrix represents a single circuit element:

- The first column contains a number in the form *tt.mmnn* where *tt* is the type code for the element (see the table below), *mm* is the first location number, and *nn* is the second location number. If *tt.mmnn* is negative, the value for the element is complex.
- The second column contains the *real* value of the element in the default units (see the table below). Voltage and current sources are stored in Polar form; all other elements are stored in Rectangular form.
- The third column contains the *imaginary* part of the element value.

Element Type	Units	Type Code
Resistor	Ohms	82
Capacitor	Farads	67
Inductor	Henrys	76
General impedance	Ohms	90*
Voltage source	Volts	86*
Current source	Amperes	73*
*If the element value is a complex number, the type code is negative.		

Remarks.

- *CIRCT* is an *n* x 3 matrix containing the circuit elements. An example of entering the elements of a circuit is contained in the next section, "Mesh Analysis."
- Values for all elements are displayed using the default units (see the table above).
- Flag 09 is set when the editor is extracting a complex value from *CIRCT*.
- Flag 10 is set if there is no variable *CIRCT*.
- Flag 25 is used to determine if *CIRCT* exists.
- Register R_{00} contains the current circuit-element number.
- Register R_{01} contains the total number of circuit elements.

"CIRCT" Program Listing.

Program: / **Comments:**

```
{ 626-Byte Prgm }
LBL "CIRCT"
PROFF
CLX
STO 00
STO 01
CF 10
SF 25
INDEX "CIRCT"
FC?C 25
SF 10
FS? 10
GTO A
RCL "CIRCT"
DIM?
X<>Y
STO 01
```

Initializes the program.

```
LBL A
CLMENU
"ADD"
KEY 1 GTO 01
"DEL"
FC? 10
KEY 2 GTO 02
"PRINT"
FC? 10
KEY 5 XEQ 12
"INIT"
FC? 10
KEY 6 GTO 06
KEY 7 XEQ 07
KEY 8 XEQ 08
KEY 9 GTO 99
```

Defines the menu for the program. If flag 10 is set (no circuit), defines only the "ADD" key.

```
33 LBL B
34 XEQ 00
35 MENU
36 STOP
37 GTO B
```

Displays the menu and stops.

```
38 LBL 01
39 XEQ "EL?"
40 FC?C 08
41 GTO A
```

Inputs a circuit element using the "EL?" utility on page 151.

```
42 FS? 10
43 1
44 FS?C 10
45 GTO 03
46 RCL 00
47 X=0?
48 GTO 10
49 RCL 01
50 X≠Y?
51 GTO 05
52 R↓
53 R↓
54 RCL "CIRCT"
55 DIM?
56 R↓
57 1
58 +
```

Inserts the new circuit element into the *CIRCT* matrix.

```
59 LBL 03
60 3
61 DIM "CIRCT"
62 INDEX "CIRCT"
```

Increases (or creates) the *CIRCT* matrix for the new element.

```
63 LBL 04
64 CLX
65 1
66 STOIJ
67 R↓
68 R↓
```

Stores the new circuit element into *CIRCT*.

```
X<>Y
→
X<>Y
REAL?
GTO 11
COMPLEX
X<>Y
→
X<>Y
LBL 11
STOEL
RCLIJ
X<>Y
STO 00
1
STO+ 01
GTO 01
```

```
LBL 05
I+
INSR
R↓
R↓
RCLIJ
GTO 04
```

Inserts a new row into the *CIRCT* matrix.

```
LBL 10
R↓
1
ENTER
STOIJ
INSR
GTO 04
```

Inserts a new row at the top of the *CIRCT* matrix.

```
100 LBL 00
101 RCL 00
102 IP
103 X=0?
104 GTO C
105 FS? 10
106 GTO C
107 "("
108 AIP
109 ⊢"):"
110 1
111 STOIJ
112 RCLEL
113 CF 09
114 X<0?
115 SF 09
116 ABS
117 IP
118 LASTX
119 FP
120 100
121 ×
122 IP
123 AIP
124 LASTX
125 FP
126 100
127 ×
128 ⊢"   "
129 AIP
130 ⊢"␊"
131 J+
132 R↓
133 R↓
134 RECT
135 86
136 X=Y?
137 POLAR
138 R↓
```

Displays the current element. If R_{00} = 0, then displays top-of-list message. (Note that line 128 contains three blank spaces between the double quotes.)

```
139 73
140 X=Y?
141 POLAR
142 R↓
143 RCLEL
144 ENTER
145 FS? 09
146 →
147 FS?C 09
148 COMPLEX
149 ARCL ST X
150 AVIEW
151 REAL?
152 R↓
153 R↓
154 1
155 ENTER
156 32
157 POSA
158 6
159 ×
160 4
161 +
162 XEQ IND ST T
163 AGRAPH
164 RTN
```

```
165 LBL C
166 RCL 01
167 "{ "
168 AIP
169 ⊢"-Element "
170 ⊢"Circuit }"
171 AVIEW
172 RTN
```

Displays top-of-list message.

```
173 LBL 82
174 "←←¿← ←¿← ←←"
175 RTN
```

Alpha string to produce graphic resistor.

```
176 LBL 67
177 "←←←←|÷|←←←←"
178 RTN
```

Alpha string to produce graphic capacitor.

```
179 LBL 76
180 "←←E"
181 4
182 XTOA
183 ├"E"
184 XTOA
185 ├"E←←"
186 R↓
187 RTN
```

Alpha strings to produce graphic inductor.

```
188 LBL 90
189 "←←8((((8←←"
190 RTN
```

Alpha string to produce graphic impedence.

```
191 LBL 86
192 "£££8DDD8£∡£"
193 RTN
```

Alpha string to produce graphic voltage source.

```
194 LBL 73
195 "←←8DDD8←T8←"
196 RTN
```

Alpha string to produce graphic current source.

```
197 LBL 02
198 RCL 00
199 X=0?
200 GTO A
201 "DELETE Element?"
202 XEQ "Y?N"
203 X=0?
204 GTO A
205 SF 25
206 DELR
207 FC?C 25
208 GTO 14
209 RCLIJ
210 X<>Y
```

Deletes the current circuit element.

```
211 STO 00
212 1
213 STO- 01
214 GTO A
```

```
215 LBL 06
216 "INITIALIZE "
217 ├"Circuit?"
218 XEQ "Y?N"
219 X=0?
220 GTO A
221 LBL 14
222 CLV "CIRCT"
223 CLX
224 STO 00
225 STO 01
226 SF 10
227 GTO A
```

Initializes the pointers and deletes *CIRCT* variable.

```
228 LBL 07
229 RCL 00
230 1
231 -
232 X<0?
233 RCL 01
234 STO 00
235 RTN
```

Decrements the element pointer.

```
236 LBL 08
237 1
238 STO+ 00
239 RCL 01
240 RCL 00
241 X>Y?
242 CLX
243 STO 00
244 RTN
```

Increments the element pointer.

```
245 LBL 12
246 RCL 01
247 1E-3
248 ×
249 STO 00
250 LBL 13
251 XEQ 00
252 PRON
253 PRLCD
254 CLA
255 PRA
256 PROFF
257 ISG 00
258 GTO 13
259 CLX
260 STO 00
261 RTN
```

Prints the graphic display for circuit elements in *CIRCT*.

```
262 LBL 99
263 EXITALL
264 END
```

Exits all menus.

Mesh Analysis ("MESH")

The "MESH" program (listed on page 57) calculates the mesh currents of a circuit containing any combination of resistors, capacitors, inductors, general impedances, and voltage sources. (If the circuit you want to analyze contains *current* sources, convert them to voltage sources.)

How "MESH" Works. "MESH" uses the elements in the *CIRCT* matrix to create the following matrices:

$$\mathbf{I} = \frac{\mathbf{Z}}{\mathbf{V}} = \frac{MATA}{MATB} = MATX$$

Variables Used.

In Equation	Description	In Program
	$n \times 3$ matrix containing circuit elements.	CIRCT
Z	Impedance matrix, **Z**.	MATA
V	Voltage matrix, **V**.	MATB
I	Solutions matrix.	MATX
	Radian frequency, $2\pi f$ (radians/sec).	ω

Since the variable names *MATA*, *MATB*, and *MATX* are used, you can use the Simultaneous Equations application to work with the data after using "MESH". Press ■[MATRIX] SIMQ *nn* (where *nn* is the number of mesh currents).

Remarks.

- Flag 08 is cleared for mesh analysis.
- Flag 10 is set when the location numbers are equal (for mesh analysis).
- "MESH" leaves the calculator in the Polar mode.
- Register R_{00} contains the element counter. Register R_{02} contains the number of mesh currents. Register R_{03} is used for intermediate

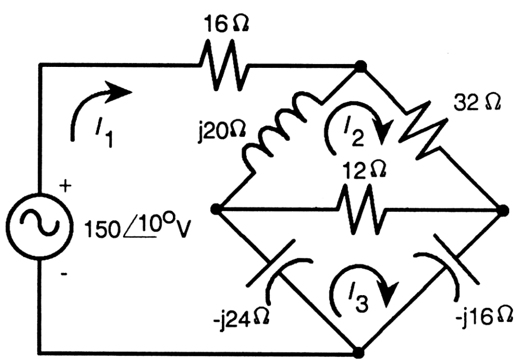
results. Be sure to set the SIZE to at least four registers (■[MODES] [▼] SIZE 4 [ENTER]) before running "MESH".

Programs Required. "MESH" (page 57), "CIRCT" (page 45), "C→Z" (page 148), "EL?" (page 151), "FQ?" (page 149), and "Y?N" (page 157).

Program Instructions.

1. Enter the circuit elements using the "CIRCT" program. Press [EXIT] when you are finished.
2. If you want the results to be printed, press ■[PRINT] [▲] PON to enable printing. If you're *not* using a printer, be sure to disable printing (press ■[PRINT] [▲] POFF).
3. Press [XEQ] MESH (to run the "MESH" program).
4. When you see `No. Mesh Currents?`, key in the number of mesh currents and press [R/S].
5. When you see `Radian Frequency(2πf)?`, key in the radian frequency and press [R/S]. (This value is used only if the circuit contains capacitance or inductance.)
6. The mesh currents are then calculated and displayed. If you're not using a printer, press [R/S] after each result is displayed.

Example: Calculating Mesh Currents. Use the "CIRCT" program to enter the following circuit. Then execute "MESH" to calculate the mesh currents, I_1, I_2, and I_3.



First, select FIX 2 display format and then enter the circuit elements using the "CIRCT" program.

■[DISP] FIX 02 [XEQ] CIRCT

```
{ 7-Element Circuit }
ADD DEL     PRINT INIT
```

If the circuit contains any elements, delete them by initializing it.

INIT YES

```
{ 0-Element Circuit }
ADD
```

Now, add the new elements to the circuit.

ADD

```
Location: # [ENTER] #
R  C  L  Z  V  I
```

Enter all of the elements in mesh 1. Since the voltage source isn't shared by another mesh, its location is entered like this:

1 [ENTER] V

```
Value?
MV  V  KV
```

Enter the value for the voltage source.

150 [ENTER] 10 ■[COMPLEX] V

```
Location: # [ENTER] #
R  C  L  Z  V  I
```

Enter the location and value for the 16-ohm resistor. Like the voltage source, the resistor is unique to mesh 1, so its location is entered the same way (1 [ENTER]).

1 [ENTER] R 16 OHM

```
Location: # [ENTER] #
R  C  L  Z  V  I
```

Since a complex impedance is provided for the inductor, enter it as a general impedance. This element is shared between mesh 1 and mesh 2, so enter both location numbers.

1 [ENTER] 2 Z 0 [ENTER] 20 ■[COMPLEX] OHM

```
Location: # [ENTER] #
R  C  L  Z  V  I
```

Enter the capacitor shared by mesh 1 and mesh 3 in the same way.

1 [ENTER] 3 Z 0 [ENTER] 24 [+/-] ■[COMPLEX] OHM

```
Location: # [ENTER] #
R  C  L  Z  V  I
```

Now enter the elements in mesh 2 that have not already been entered.

2 [ENTER] R 32 OHM

```
Location: # [ENTER] #
 R  C  L  Z  V  I
```

2 [ENTER] 3 R 12 OHM

```
Location: # [ENTER] #
 R  C  L  Z  V  I
```

Enter the last element, which is the capacitor unique to mesh 3. (Note that you could have entered these elements in any order.)

3 [ENTER] Z 0 [ENTER] 16 [+/-] ■[COMPLEX] OHM

```
Location: # [ENTER] #
 R  C  L  Z  V  I
```

Press [EXIT] to return to the main level. You'll see the last element that you entered.

[EXIT]

```
(7):3 -||- 3
0.00 -i16.00
```

At this point you can use [▲] and [▼] to view the elements in the circuit.

Exit from the "CIRCT" program and calculate the mesh currents. If you want the results printed, press ■[PRINT] [▲] PON to enable printing.

[EXIT]

```
Y: 1.00
X: 34.00
```

[XEQ] MESH

```
No. Mesh Currents?
X: 0.00
```

Key in the number of mesh currents.

3 [R/S]

```
Radian Frequency(2πf)?
X: 0.01
```

Since the inductor and capacitors were given as complex impedances, the frequency will *not* be used. Press [R/S] to display the current in mesh 1.

[R/S]

```
I1=
X: 6.28 ∡4.72
```

If you're not using a printer, press [R/S] to display the current in mesh 2.

[R/S]

```
I2=
x: 2.61 ∡53.37
```

If you're not using a printer, press [R/S] again to display the current in mesh 3.

[R/S]

```
I3=
x: 3.09 ∡-2.73
```

"MESH" and "NODAL" Program Listing.

Program:	Comments:
`00 { 491-Byte Prgm }`	

`01 LBL "MESH"`
`02 CF 08`
`03 "No. Mesh "`
`04 ├"Currents?"`
`05 GTO A`
	Clears flag 08 (to indicate mesh analysis) and prompts for the number of mesh currents.

`06 LBL "NODAL"`
`07 SF 08`
`08 "No. Nodes?"`
	Sets flag 08 (to indicate nodal analysis) and prompts for the number of nodes.

`09 LBL A`
`10 RCL 02`
`11 PROMPT`
`12 STO 02`
	Stores the number of mesh currents or nodes.

`13 ENTER`
`14 NEWMAT`
`15 ENTER`
`16 COMPLEX`
`17 STO "MATA"`
`18 DIM?`
`19 1`
`20 STO 00`
`21 NEWMAT`
`22 ENTER`
`23 COMPLEX`
`24 STO "MATB"`
`25 XEQ "FQ?"`
	Creates the complex matrices *MATA* and *MATB*, and prompts for a frequency.

```
26 CF 21
27 "Working..."
28 AVIEW
29 FS? 55
30 SF 21
```

Displays a "working" message and sets or clears flag 21 to match flag 55.

```
31 LBL a
32 INDEX "CIRCT"
33 RCL 00
34 1
35 SF 25
36 STOIJ
37 FC?C 25
38 GTO D
```

Sets the index pointer to the current element (identified in R_{00}). If this generates an error, then the end of the *CIRCT* matrix has been reached.

```
39 RCLEL
40 XEQ IND ST X
```

Recalls the element type code and executes the appropriate subroutine to add the element value to *MATA* and *MATB*.

```
41 1
42 STO+ 00
43 GTO a
```

Increments the element pointer and repeats the loop for the next element.

```
44 LBL B
45 FS? 08
46 1/X
47 ENTER
48 R↓
49 R↓
50 INDEX "MATA"
51 STOIJ
52 RCLEL
53 R↑
54 FS? 10
55 GTO b
56 -
57 STOEL
58 R↓
59 X<>Y
60 STOIJ
61 R↑
```

Accumulates an element value into *MATA*. If flag 10 is set (indicating an element shared by two meshes or connected to the reference node), adds the value only to the matrix element on the main diagonal. If flag 10 is clear, the value is accumulated to the appropriate matrix elements not on the diagonal. Note that values accumulated along the diagonal are positive; values accumulated in other matrix elements of *MATA* are negative.

```
62 STOEL
63 CLX
64 LASTX
65 R↑
66 STO 03
67 R↑
68 ENTER
69 STOIJ
70 R↓
71 RCLEL
72 R↑
73 XEQ b
74 RCL 03
75 ENTER
76 STOIJ
77 RCLEL
78 LASTX
```

Stores a value into the matrix.

```
79 LBL b
80 +
81 STOEL
82 RTN
```

Stores a resistor value.

```
83 LBL 82
84 XEQ F
85 XEQ e
86 GTO B
```

Stores a capacitor value.

```
87 LBL 67
88 XEQ F
89 XEQ e
90 XEQ "C→Z"
91 GTO B
```

Stores an inductor value.

```
92 LBL 76
93 XEQ F
94 XEQ e
95 XEQ "L→Z"
96 GTO B
```

Stores a general impedance value.

```
97 LBL 90
98 XEQ F
```

```
99 XEQ e
100 XEQ e
101 RECT
102 COMPLEX
103 GTO B
```

```
104 LBL 73
105 LBL 86
106 XEQ F
107 XEQ e
108 XEQ e
109 POLAR
110 COMPLEX
111 INDEX "MATB"
112 RCL ST Z
113 FC? 08
114 GTO c
115 FC? 10
116 GTO c
117 X≠0?
118 GTO C
119 X<> ST Z
120 X<>Y
121 -1
122 ×
123 X<>Y
```

Stores a current or voltage source into *MATB*. Since the two types are not distinguished, all sources are assumed to be voltage sources for mesh analysis or current sources for nodal analysis.

```
124 LBL C
125 1
126 STOIJ
127 RCLEL
128 R↑
129 +
130 STOEL
131 RTN
```

Adds a current source connected to the reference node.

```
132 LBL c
133 1
134 STOIJ
135 CLX
136 RCL ST T
```

Adds a voltage source (mesh analysis) or a current source *not* connected to the reference node (nodal analysis).

```
137 ENTER
138 RCLEL
139 R↑
140 +
141 STOEL
142 R↓
143 X=Y?
144 RTN
145 1
146 STOIJ
147 RCLEL
148 LASTX
149 -
150 STOEL
151 RTN
```

```
152 LBL D
153 POLAR
154 RCL "MATB"
155 RCL÷ "MATA"
156 STO "MATX"
```

Calculates the results using matrix division.

```
157 SF 21
158 INDEX "MATX"
```

Prepares to display the results.

```
159 LBL d
160 RCLIJ
161 R↓
162 "I"
163 FS? 08
164 "V"
165 AIP
166 ├"="
167 RCLEL
168 FS? 55
169 ARCL ST X
170 AVIEW
```

Recalls each result from *MATX* and displays it with the appropriate label. If a printer is being used (flag 55 set), the value is appended to the label.

```
171 I+
172 FC? 77
173 GTO d
174 RTN
```

Flag 77 is tested to determine when the end of *MATX* has been reached.

```
LBL e
J+
RCLEL
RTN
```

Increments the column pointer and recalls a matrix element.

```
LBL F
ABS
FP
100
×
IP
LASTX
FP
100
×
```

Decodes the element location into two mesh or node numbers in the X- and Y-registers.

```
CF 10
X=Y?
SF 10
FC? 08
RTN
```

Sets flag 10 to indicate that the two location numbers are the same.

```
FS? 10
GTO J
XEQ G
XEQ G
X<>Y
FC? 10
RTN
+
ENTER
RTN
```

For nodal analysis (flag 08 set), displays an error message if the location numbers are equal. (The numbers do not make sense if they're equal.)

```
LBL G
X=0?
SF 10
X<>Y
RTN
```

Sets flag 10 if the reference node is used.

```
LBL J
"Invalid Data"
PROMPT
END
```

Error message.

Nodal Analysis ("NODAL")

The "NODAL" program (listed on page 57) calculates the node voltages of a circuit containing any combination of resistors, capacitors, inductors, general impedances, and current sources. (If the circuit you want to analyze contains *voltage* sources, convert them to current sources.)

How "NODAL" Works. "NODAL" uses the circuit information in the *CIRCT* matrix to create the following matrices:

$$\mathbf{V} = \frac{\mathbf{Z}}{\mathbf{I}} = \frac{MATA}{MATB} = MATX$$

Variables Used.

In Equation	Description	In Program
	$n \times 3$ matrix containing circuit elements.	CIRCT
Z	Impedance matrix.	MATA
V	Voltage matrix.	MATX
I	Current matrix.	MATB
	Radian frequency, $2\pi f$ (radians/sec).	ω

Since the variable names *MATA*, *MATB*, and *MATX* are used, you can use the Simultaneous Equations application to work with the data after you've finished using "NODAL". Press ■[MATRIX] SIMQ *nn* (where *nn* is the number of nodes).

Remarks.

- Flag 08 is set for nodal analysis.
- Flag 10 is set when one location number is zero (for nodal analysis).
- "NODAL" leaves the calculator in Polar mode.
- Register R_{00} contains the element counter. Register R_{02} contains the number of nodes. Register R_{03} is used for intermediate results. Be

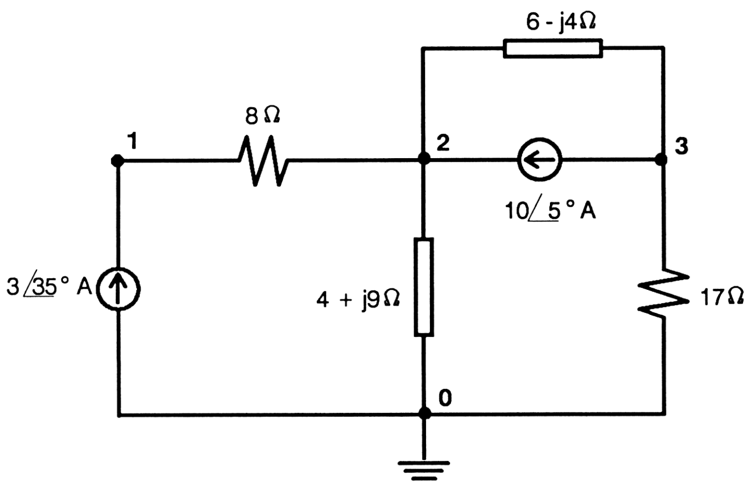
sure to set the SIZE to at least four registers (■[MODES] [▼] SIZE 4 [ENTER]) before running "NODAL".

Programs Required. "MESH" (page 57), "CIRCT" (page 45), "C→Z" (page 148), "EL?" (page 151), "FQ?" (page 149), "Y?N" (page 157).

Program Instructions.

1. Enter the circuit elements using the "CIRCT" program. Press [EXIT] when you're finished.
2. If you want the results to be printed, press ■[PRINT] [▲] PON to enable printing. If you're *not* using a printer, be sure to press ■[PRINT] [▲] POFF to disable printing.
3. Press [XEQ] NODAL (to run the "NODAL" program).
4. When you see `No. Nodes?`, key in the number of nodes and press [R/S].
5. When you see `Radian Frequency(2πf)?`, key in the radian frequency and press [R/S]. (This value is used only if the circuit contains capacitance or inductance.)
6. The node voltages are then calculated and displayed. If you're not using a printer, press [R/S] after each result is displayed.

Example: Calculating Node Voltages. Use the "CIRCT" program to enter the following circuit. Then execute "NODAL" to calculate the node voltages, V_1, V_2, and V_3.



Begin by selecting FIX 2 display format and running "CIRCT".

■[DISP] FIX 02 [XEQ] CIRCT

`{ 7-Element Circuit }`
`ADD | DEL | | | PRINT | INIT`

If "CIRCT" has any elements stored in it, delete them by initializing it.

INIT YES

`{ 0-Element Circuit }`
`ADD | | | | |`

Now, add the new elements to the circuit.

ADD

`Location: # [ENTER] #`
`R | C | L | Z | V | I`

Enter the current source from node 0 to node 1.

0 [ENTER] 1 I 3 [ENTER] 35
■[COMPLEX] A

`Location: # [ENTER] #`
`R | C | L | Z | V | I`

Enter the resistor from node 1 to node 2.

1 [ENTER] 2 R 8 OHM

`Location: # [ENTER] #`
`R | C | L | Z | V | I`

Enter the impedance from node 0 to node 2.

0 [ENTER] 2 Z 4 [ENTER] 9
■[COMPLEX] OHM

`Location: # [ENTER] #`
`R | C | L | Z | V | I`

Enter the resistor from node 0 to node 3.

0 [ENTER] 3 R 17 OHM

`Location: # [ENTER] #`
`R | C | L | Z | V | I`

Enter the current source from node 3 to node 2. (Note that if you entered this source from node 2 to node 3, the *sign* is reversed: $-10\ \measuredangle 5°$A.)

3 [ENTER] 2 I 10 [ENTER]
5 ■[COMPLEX] A

`Location: # [ENTER] #`
`R | C | L | Z | V | I`

Enter the impedance from node 2 to node 3.

2 [ENTER] 3 Z 6 [ENTER]
4 [+/-] ■[COMPLEX] OHM

`Location: # [ENTER] #`
`R | C | L | Z | V | I`

Press [EXIT] to return to the program's main menu.

[EXIT]

```
(6):2 ⇔ 3
6.00 -i4.00
```

At this point you can use [▲] and [▼] to view the elements in the circuit. Exit the "CIRCT" program and calculate the node voltages. If you want the results printed, press ■[PRINT] [▲] POH to enable printing.

[EXIT]

```
Y: 1.00
x: 34.00
```

[XEQ] NODA

```
No. Nodes?
x: 0.00
```

Key in the number of nodes. (Do not include the reference node).

3 [R/S]

```
Radian Frequency(2πf)?
x: 0.01
```

Since no capacitors or inductors were entered, the frequency is not needed; press [R/S] to calculate the results.

[R/S]

```
V1=
x: 68.66 ∡47.03
```

If you're not using a printer, press [R/S] to display the voltage at node 2.

[R/S]

```
V2=
x: 45.46 ∡53.34
```

If you're not using a printer, press [R/S] again to display the voltage at node 3.

[R/S]

```
V3=
x: 58.07 ∡126.80
```

Impedance of a Ladder Network ("LADDR")

This program calculates the input impedance, Z_{in}, of a ladder network. Elements are added one at a time starting with the element furthest from the terminals where Z_{in} is measured. The first element must be connected in parallel.

Given an input *admittance* of Y_{in}, adding a shunt (parallel) R, L, or C results in a new input impedance:

$$Y_{new} = \begin{cases} Y_{in} + \left(\dfrac{1}{R_p} + j0\right) \\ Y_{in} + \left(0 - j\dfrac{1}{\omega L_p}\right) \\ Y_{in} + (0 + j\omega C_p) \end{cases}$$

Adding a series R, L, or C, we have:

$$Y_{new} = \begin{cases} \left(\dfrac{1}{Y_{in}} + (R_s + j0)\right)^{-1} \\ \left(\dfrac{1}{Y_{in}} + (0 + j\omega L_s)\right)^{-1} \\ \left(\dfrac{1}{Y_{in}} + \left(0 - j\dfrac{1}{\omega C_s}\right)\right)^{-1} \end{cases}$$

where $Y = 1/Z$ and $\omega = 2\pi f$.

Variable Used. `Zin` is the impedance looking into the ladder network.

Remarks.

- Elements are entered in rectangular form; however, the input impedance is displayed in polar form. (If you want the impedance displayed in rectangular mode, you can change line 29 of the "LADDR" program listing to `29 RECT` and then delete line 31.)

- If a circuit element is given as a complex impedance, key in the complex value and then add it to the circuit using RP and RS.
- Flag 00 is set when $Z_{in} = 0$.
- Flag 21 (*printer enable*) is set or cleared to match flag 55 (*printer existence*). This automatically produces printer output if flag 55 is set.

Programs Required. "LADDR" (page 70), "C→Z" (page 148), and "FQ?" (page 149).

Program Instructions.

1. Key the required programs into your calculator.
2. Press [XEQ] LADDR (to run the "LADDR" program).
3. When you see Radian Frequency(2πf)?, key in the radian frequency and press [R/S]. (If the correct frequency is already in the X-register, just press [R/S].)
4. The program displays a menu of the elements that can be added. For each element in the network (starting with the element furthest from the terminals where Z_{in} is measured), key in the value of the element and then press the corresponding key.

   RP (parallel resistor, ohms)

   LP (parallel inductor, henrys)

   CP (parallel capacitor, farads)

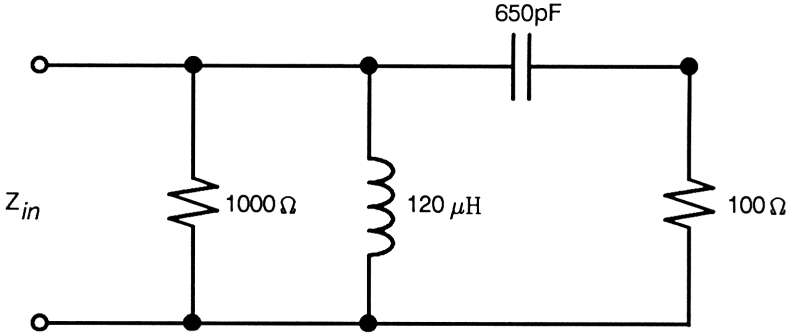
   RS (series resistor, ohms)

   LS (series inductor, henrys)

   CS (series capacitor, farads)

   The series elements are not displayed until the first parallel element has been entered. Each time you add an element, the new impedance is displayed.
5. To quit, press [EXIT]. After quitting, you can restart "LADDR" by pressing [R/S].

Example. Find the input impedance of the following circuit at a frequency of 1 MHz:



Select the FIX 2 display format and run the "LADDR" program.

■[DISP] FIX 02 [XEQ] LADDR

```
Radian Frequency(2πf)?
x: 0.01
```

The frequency is given in Hz. Convert it to radians/second.

[E] 6 ■[π] [×] 2 [×] [R/S]

```
Zin=0.00 ∡0.00
RP  LP  CP
```

Enter the four elements (working *right to left*). Notice that the current value of Z_{in} is displayed after each element is entered.

100 RP

```
Zin=100.00 ∡0.00
RP  LP  CP  RS  LS  CS
```

650 [E] 12 [+/-] CS

```
Zin=264.49 ∡-67.78
RP  LP  CP  RS  LS  CS
```

120 [E] 6 [+/-] LP

```
Zin=384.34 ∡-56.67
RP  LP  CP  RS  LS  CS
```

1000 RP

```
Zin=306.73 ∡-41.82
RP  LP  CP  RS  LS  CS
```

The input impedance is 306.73 ∡ −41.82° ohms.

"LADDR" Program Listing.

Program:	**Comments:**
`00 { 154-Byte Prgm }`	
`01 LBL "LADDR"` `02 XEQ "FQ?"`	Inputs the frequency using the "FQ?" utility on page 149.
`03 SF 00`	Sets flag 00 (until the first element is entered).

`04 CLX`
`05 ENTER`
`06 COMPLEX`
`07 STO "Zin"`
`08 CF 21`
`09 FS? 55`
`10 SF 21`
	Initializes Z_{in} and sets or clears flag 21 to match flag 55.

`11 CLMENU`
`12 LBL A`
`13 "RP"`
`14 KEY 1 XEQ 01`
`15 "LP"`
`16 KEY 2 XEQ 02`
`17 "CP"`
`18 KEY 3 XEQ 03`
	Defines the menu keys for entering elements in parallel.

`19 FS? 00`
`20 GTO B`
`21 "RS"`
`22 KEY 4 XEQ 04`
`23 "LS"`
`24 KEY 5 XEQ 05`
`25 "CS"`
	If flag 00 is clear, declares the menu keys for entering elements in series.

`26 KEY 6 XEQ 06`	Decodes the element location into two mesh or node numbers in the X- and Y-registers.

```
27 LBL B
28 MENU
29 POLAR
30 VIEW "Zin"
31 RECT
32 STOP
33 GTO A
```

Displays the menu and the input impedance. (If you want the impedance displayed in rectangular mode, you can change line 29 to `29 RECT` and delete line 31.)

```
34 LBL 01
35 1/X
36 GTO 07
```

Converts a resistance to admittance.

```
37 LBL 02
38 XEQ "C→Z"
39 GTO 07
```

Converts an inductance to admittance using the "C→Z" utility on page 148.

```
40 LBL 03
41 XEQ "L→Z"
```

Converts a capacitance to admittance using the "L→Z" utility on page 148.

```
42 LBL 07
43 RCL "Zin"
44 FC?C 00
45 1/X
46 +
47 1/X
48 STO "Zin"
49 RTN
```

Adds a parallel element.

```
50 LBL 06
51 XEQ "C→Z"
52 GTO 04
```

Converts a capacitance to impedance.

```
53 LBL 05
54 XEQ "L→Z"
```

Converts an inductance to impedance.

```
55 LBL 04
56 STO+ "Zin"
57 END
```

Adds a series element.

3

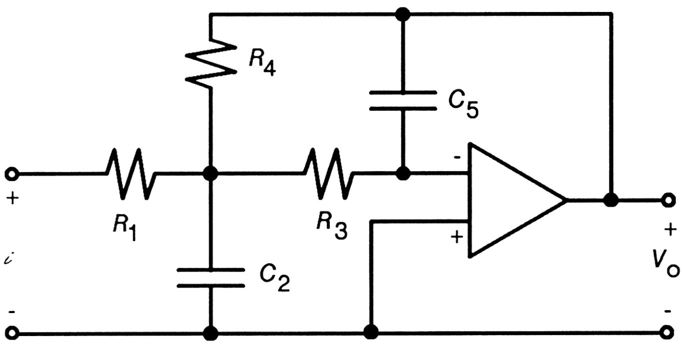
Filter Design

This chapter contains programs for calculating component values for standard active filters and for Butterworth filters between equal terminations.

Active Filter Design ("AF")

This program calculates element values for the standard active filter circuits shown below. You must provide *F* (the corner or center frequency), *G* (the midband gain), *PF* or α (the peaking factor), and *C* (a capacitor). The program then displays (and optionally prints) the list of elements that form the desired filter.

Low Pass Filter.



$$C_5 = C$$

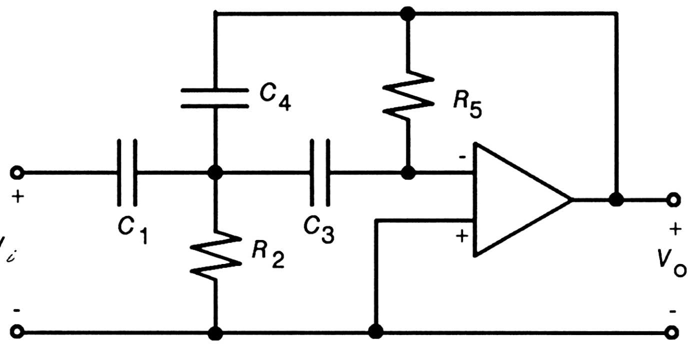
$$C_2 = \frac{4C(G+1)}{PF^2}$$

$$R_1 = \frac{PF}{4G\pi f_0 C}$$

$$R_3 = \frac{PF}{4\pi f_0 C(G+1)} = \frac{G}{G+1} R_1$$

$$R_4 = GR_1$$

High Pass Filter.



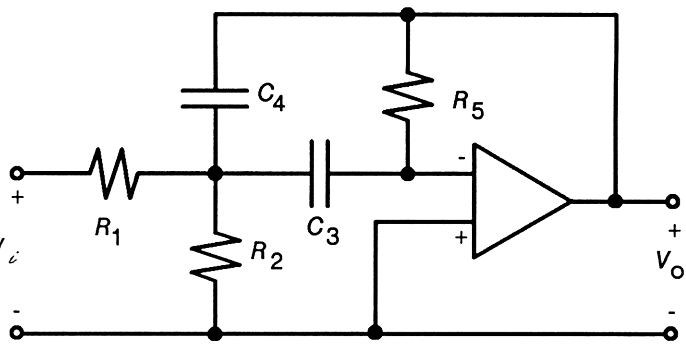
$$C_1 = C_3 = C$$

$$C_4 = \frac{C}{G}$$

$$R_2 = \frac{PF}{2\pi f_0 C\left(2 + \frac{1}{G}\right)}$$

$$R_5 = \frac{2G + 1}{PF\ 2\pi f_0 C}$$

Bandpass Filter.



$$C_3 = C_4 = C$$

$$R_1 = \frac{1}{G\, 2\pi f_0 C\, PF}$$

$$R_2 = \frac{1}{\left(\frac{2}{PF^2} - G\right) 2\pi f_0 C\, PF}$$

$$R_5 = \frac{2}{PF\, 2\pi f_0 C}$$

Variables Used.

In Equations	Description	In Program
f_0	Center frequency (Hz).	F
G	Midband gain (db).	G
PF	Peaking factor.	PF
C	Capacitor (farads).	C

Remarks. Flag 21 (*printer enable*) is set by the program.

Program Instructions.

1. Key the "AF" program (listed on page 78) into your calculator.
2. Press [XEQ] AF (to run the "AF" program).
3. Press INPUT. The calculator prompts for *F*, *G*, *PF*, and *C*.
   a. Key in a frequency, *F*, in Hertz; press [R/S].
   b. Key in the midband gain, *G*; press [R/S].
   c. Key in a peaking factor, *PF*; press [R/S].
   d. Key in a capacitance, *C*, in Farads; press [R/S].

   After entering these values, the program returns to the main menu.
4. Press LOWP (low pass), HIGHP (high pass), or BAND (band pass) to calculate the elements needed to build the particular filter.
5. When all of the elements have been displayed, press [R/S] to return to the main menu. Then go to step 3 to work another problem or press [EXIT] to quit.

Example. Design a high-pass active filter with the following characteristics: $F = 10$ Hz, $G = 10$, $PF = 1$, and $C = 1\ \mu$F.

Select the ENG 3 display format and run the "AF" program.

■[DISP] ENG 03 [XEQ] AF

```
x: 0.000E0
LOWP HIGHP BAND          INPUT
```

INPUT	Y: 0.000E0 F?0.000E0
10 [R/S]	Y: 10.00E0 G?0.000E0
10 [R/S]	Y: 10.00E0 PF?0.000E0
1 [R/S]	Y: 1.000E0 C?0.000E0
[E] 6 [+/-] [R/S]	x: 1.000E-6 LOWP HIGHP BAND INPUT
HIGHP	R2= x: 7.579E3

If you're not using a printer, press [R/S] after each result.

[R/S]	R5= x: 334.2E3
[R/S]	C1=C3=C= x: 1.000E-6
[R/S]	C4= x: 100.0E-9
[R/S]	x: 100.0E-9 LOWP HIGHP BAND INPUT

"AF" Program Listing.

Program: / **Comments:**

```
00 { 309-Byte Prgm }
01 LBL "AF"
02 SF 21
03 CLMENU
04 "LOWP"
05 KEY 1 XEQ A
06 "HIGHP"
07 KEY 2 XEQ B
08 "BAND"
09 KEY 3 XEQ C
10 "INPUT"
11 KEY 6 XEQ I
12 KEY 9 GTO 09
```

Defines the menu for selecting a filter type.

```
13 LBL 00
14 CLD
15 MENU
16 STOP
17 GTO 00
```

Clears any message that may be displayed, displays the menu and stops. Pressing [R/S] redisplays the menu.

```
18 LBL A
19 "R1"
20 RCL "PF"
21 4
22 RCL× "G"
23 XEQ 04
24 ÷
25 XEQ 08
26 "R3"
27 RCL× "G"
28 ENTER
29 ENTER
30 LASTX
31 1
32 +
33 ÷
34 XEQ 08
```

Calculates the elements for a low pass filter.

```
"R4"
R↓
XEQ 08
"C2"
1
RCL "G"
RCLx "C"
4
x
RCL "PF"
X↑2
÷
XEQ 08
"C5=C"
LBL 10
RCL "C"
GTO 08
```

Calculates the elements for a high pass filter.

```
LBL B
"R2"
2
XEQ 04
RCL "G"
1/X
2
+
x
RCL÷ "PF"
1/X
XEQ 08
"R5"
2
RCLx "G"
1
+
2
XEQ 04
RCLx "PF"
÷
XEQ 08
"C1=C3=C"
```

```
75 XEQ 10
76 "C4"
77 RCL÷ "G"
78 GTO 08
```

```
79 LBL C
80 "R1"
81 2
82 XEQ 04
83 RCL× "G"
84 ENTER
85 ENTER
86 RCL× "G"
87 1/X
88 XEQ 08
89 "R2"
90 CLX
91 2
92 RCL "PF"
93 X↑2
94 ÷
95 RCL- "G"
96 ×
97 1/X
98 XEQ 08
99 "R5"
100 CLX
101 2
102 X<>Y
103 ÷
104 XEQ 08
105 "C3=C4=C"
106 GTO 10
```

Calculates the elements for a band pass filter.

```
107 LBL 04
108 PI
109 RCL× "F"
110 RCL× "C"
111 ×
112 RTN
```

Calculates πFC.

113 LBL I 114 EXITALL 115 INPUT "F" 116 INPUT "G" 117 INPUT "PF" 118 INPUT "C" 119 RTN	Exits all menus and inputs *F*, *G*, *PF*, and *C*.
120 LBL 08 121 EXITALL 122 ⊢"=" 123 FS? 55 124 ARCL ST X 125 AVIEW	Exits all menus. If results are being printed (flag 55 set), puts label and value on the same line.
126 LBL 09 127 END	Ends program.

Butterworth Filter Design ("BF")

This program calculates component values for Butterworth filters between *equal* terminations. Inputs are termination resistance, passband characteristics, and attenuation at some out-of-band frequency.

Before the filter elements can be calculated, a normalized frequency must be computed from the desired cutoff or center frequency and passband characteristics. The normalized frequency is computed by one of these formulas:

Low Pass

$$\omega_n = \frac{\omega}{\omega_0}$$

High Pass

$$\omega_n = \frac{\omega_0}{\omega}$$

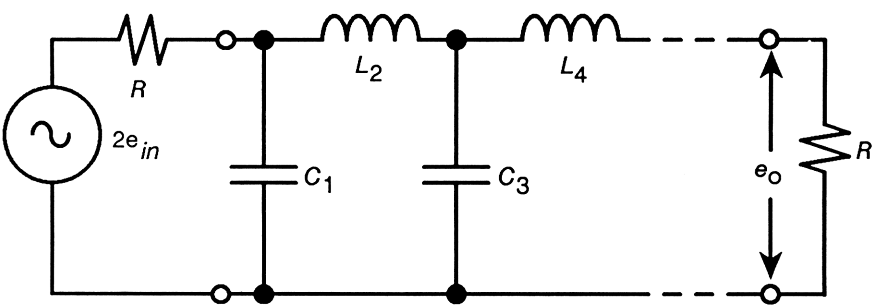
Band Pass

$$\omega_n = \frac{\omega^2 - \omega_0^2}{BW\omega}$$

Band Elimination

$$\omega_n = \frac{BW\omega}{\omega_0^2 - \omega^2}$$

The basic form of the filter is this *low*-pass prototype:



whose elements are given by the following set of formulas:

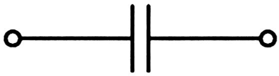
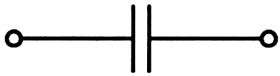
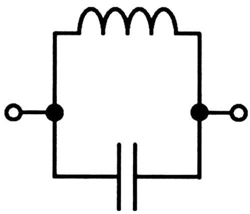
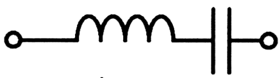
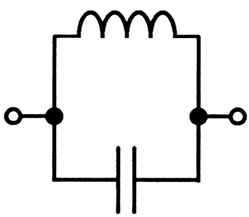
$$C_i = \frac{1}{\pi f_c R} \sin \frac{(2i - 1)\pi}{2n}, \; i = 1, 3, 5, \ldots, n - 1$$

$$L_i = \frac{R}{\pi f_c} \sin \frac{(2i - 1)\pi}{2n}, \; i = 2, 4, 6, \ldots, n$$

where:

$$n = INT \left[\frac{1 + \ln(2 \times 10^{-\Delta dB/10} - 1)}{2 \ln(\omega / \omega_0)} \right]$$

Once the low-pass values have been calculated, if another passband characteristic is desired, the filter components are changed by one of the frequency transformations shown on the next page.

Passband Characteristic	Circuit Elements	
Low pass	 C_n	 L_n
High pass	 $L_h = \frac{1}{\omega_0 C_n}$	 $C_h = \frac{1}{\omega_0 L_n}$
Band pass	$L_{bp} = \frac{BW}{\omega_0^2 C_n}$  $C_{bp} = \frac{C_n}{BW}$	L_{bp} C_{bp}  $\frac{L_n}{BW}$ $\frac{BW}{\omega_0^2 L_n}$
Band elimination	L_{be} C_{be}  $\frac{1}{C_n BW}$ $\frac{C_n BW}{\omega_0^2}$	$L_{be} = \frac{L_n BW}{\omega_0^2}$  $C_{be} = \frac{1}{L_n BW}$

Variables and Storage Registers Used.

In Equations	Description	In Program
R	Resistor (ohms).	R
F_0	Center frequency (Hz).	F0
ω_0	Center frequency (radians/sec.).	R_{01}
F_1	Attenuation frequency (Hz).	F1
	Amount of attenuation (dB).	A
BW	Band width (Hz).	BW
	Band width (radians/sec.).	R_{02}
	Filter type (1-4).	R_{07}
n	Filter order.	R_{09}
i	Element counter.	R_{11}

Remarks.

- Flag 01 (set and cleared by the program) is used for branch control.
- Flag 21 (*printer enable*) is set by the program.
- Registers R_{08}, R_{10}, R_{13}, and R_{14} are used to store intermediate results. Be sure to set the SIZE to at least 15 registers (■[MODES] [▼] SIZE 15 [ENTER]).
- The "BF" program sets Radians mode.

Note The program will give erroneous results if asked to calculate a filter order when A is small (when ΔdB is close to Loss (ω_0)).

Program Instructions.

1. Key the "BF" program (listed on page 88) into your calculator.
2. Press [XEQ] BF (to run the "BF" the program).
3. The program displays a variable menu containing *R*, *F0*, *F1*, *A*, and *BW*. Store a value into each variable by keying in the value and then pressing the corresponding menu key.
4. After each of the five values has been stored, press [R/S].
5. The program displays `Type?` and a menu containing the four types of filters. Press one of these keys to select a filter type:

   LOWP (low pass)

   HIGHP (high pass)

   BPASS (band pass)

   BELIM (band elimination)
6. The program then calculates and displays *N* and the filter elements. If you're not using a printer, press [R/S] after each result is displayed.

Example. Design a 100-Hz wide Butterworth filter centered at 800 Hz with a 30-dB attenuation at 900 Hz. R_0 is 50 ohms. The termination resistance, *R*, is also 50 ohms.

Select ENG 3 display format and run the "BF" program.

■[DISP] ENG 03 [XEQ] BF

```
x: 0.000E0
 R | F0 | F1 | A | BW |
```

Store the five inputs.

50 R

```
R=50.00E0
 R | F0 | F1 | A | BW |
```

800 F0

```
F0=800.0E0
 R | F0 | F1 | A | BW |
```

900 F1

```
F1=900.0E0
 R | F0 | F1 | A | BW |
```

30 A

A=30.00E0
R | F0 | F1 | A | BW

100 BW

BW=100.0E0
R | F0 | F1 | A | BW

Continue with the program.

[R/S]

Type?
LOWP | HIGHP | BPASS | BELIM

Select a band-pass filter. If you are not using a printer, press [R/S] after each result is displayed.

BPASS

N=
x: 6.000E0

[R/S]

C1=16.48E-6
L1=2.402E-3

[R/S]

L2=112.5E-3
C2=351.7E-9

[R/S]

C3=61.49E-6
L3=643.6E-6

[R/S]

L4=153.7E-3
C4=257.5E-9

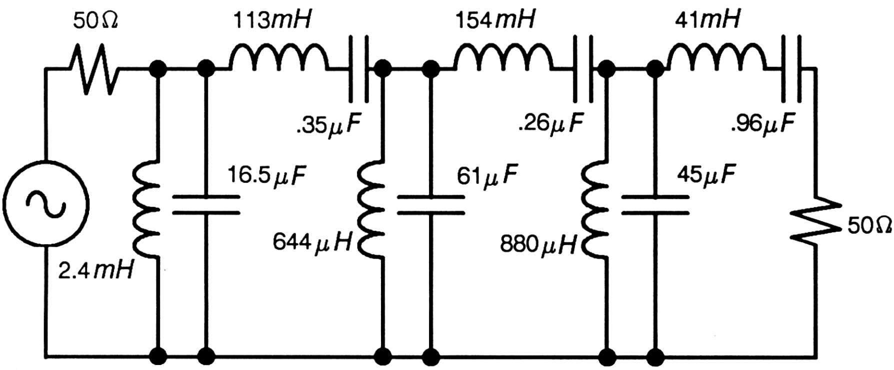
[R/S]

C5=45.02E-6
L5=879.2E-6

[R/S]

```
L6=41.19E-3
C6=960.8E-9
```

Therefore, the filter you've calculated looks like this:



"BF" Program Listing. This program demonstrates that local labels do not have to be unique within a program as long as careful consideration is given to the local label search order (described in the owner's manual).

Program:	**Comments:**
`00 { 394-Byte Prgm }`	

`01 LBL "BF"`
`02 MVAR "R"`
`03 MVAR "F0"`
`04 MVAR "F1"`
`05 MVAR "A"`
`06 MVAR "BW"`
`07 RAD`
	Declares the menu variables and sets Radians mode.

`08 LBL A`
`09 VARMENU "BF"`
`10 CLA`
`11 STOP`
`12 ALENG`
`13 X≠0?`
`14 GTO A`
	Displays the variable menu and stops. Pressing [R/S] is the only way to continue the program.

```
15 RCL "F0"
16 XEQ 10
17 STO 01
```

Converts the center frequency to radians/second.

```
18 CLMENU
19 "LOWP"
20 KEY 1 GTO C
21 "HIGHP"
22 KEY 2 GTO D
23 "BPASS"
24 KEY 3 GTO E
25 "BELIM"
26 KEY 4 GTO F
27 KEY 9 GTO A
```

Defines the menu for selecting a filter type. The [EXIT] key is defined to return to the variable menu.

```
28 LBL B
29 MENU
30 "Type?"
31 PROMPT
32 GTO B
```

Displays the menu and prompts for a filter type. Pressing [R/S] redisplays the menu.

```
33 LBL C
34 1
35 GTO 01
```

Calculates the elements of a low pass filter (type 1).

```
36 LBL D
37 2
38 GTO 01
```

Calculates the elements of a high pass filter (type 2).

```
39 LBL E
40 3
41 GTO 01
```

Calculates the elements of a band pass filter (type 3).

```
42 LBL F
43 4
```

Calculates the elements of a band elimination filter (type 4).

```
44 LBL 01
45 EXITALL
46 STO 07
47 SF 21
```

Exits all menus, stores the filter type, and sets flag 21 for proper output.

```
48 RCL "BW"
49 XEQ 10
50 STO 02
```

Converts the band width to radians/second.

```
51 LBL 00
52 RCL "F1"
53 RCL "A"
54 10
55 ÷
56 10↑X
57 2
58 ×
59 1
60 -
61 LN
62 STO 08
63 X<>Y
64 XEQ 10
65 XEQ 07
66 RCL 08
67 X<>Y
68 LN
69 ABS
70 ÷
71 1
72 +
73 2
74 ÷
75 IP
76 STO 09
77 STO 10
```

Calculates the filter order.

```
78 "N="
79 FS? 55
80 AIP
81 AVIEW
82 ADV
```

Displays the filter order. If results are being printed (flag 55 set), the label and result are displayed on the same line.

```
83 LBL 08
84 RCL 09
85 RCL- 10
```

Evaluates the Butterworth equations.

```
86 1
87 +
88 STO 11
89 2
90 ×
91 1
92 -
93 PI
94 ×
95 2
96 ÷
97 RCL÷ 09
98 SIN
99 2
100 ×
101 LBL 09
102 STO 14
103 RCL "R"
104 -1
105 RCL 11
106 Y↑X
107 Y↑X
108 ×
```

```
109 GTO IND 07
110 LBL 01
111 RCL÷ 01
112 XEQ 06
113 GTO 00
114 LBL 02
115 RCL× 01
116 1/X
117 XEQ 06
118 +/-
119 GTO 00
120 LBL 03
121 SF 01
122 RCL÷ 02
123 XEQ 06
124 XEQ 00
125 ABS
```

Calculates the frequency transformation for the particular filter type.

```
1/X
RCL 01
X↑2
÷
XEQ 06
+/-
GTO 00
LBL 04
SF 01
RCL× 02
RCL 01
X↑2
÷
XEQ 06
XEQ 00
ABS
RCL 01
X↑2
×
1/X
XEQ 06
+/-

LBL 00                Displays a pair of filter elements.
FS? 01
CLA
X≥0?
⊢"L"
X<0?
⊢"C"
RCL 11
AIP
⊢"="
R↓
ABS
ARCL ST X
⊢"␊"
FC? 01
AVIEW
FS?C 01
RTN
```

```
166 DSE 10
167 GTO 08
168 RTN
```

Repeats the Butterworth calculations for each element.

```
169 LBL 07
170 STO 14
171 GTO IND 07
172 LBL 04
173 XEQ 03
174 GTO 00
175 LBL 02
176 XEQ 01
177 LBL 00
178 1/X
179 +/-
180 GTO 05
181 LBL 01
182 RCL 14
183 RCL÷ 01
184 GTO 05
185 LBL 03
186 RCL 14
187 X↑2
188 RCL 01
189 X↑2
190 -
191 RCL÷ 14
192 RCL÷ 02
193 LBL 05
194 ABS
195 STO 13
196 RTN
```

Calculates the normalized frequency for the particular filter type.

```
197 LBL 06
198 -1
199 RCL 11
200 Y↑X
201 ×
202 RTN
```

Multiplies by -1^i.

```
203 LBL 10
204 2
205 ×
206 PI
207 ×
208 END
```

Converts the frequency in the X-register (in Hz) to radians/second.

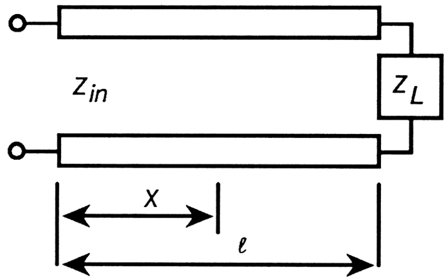
4

Transmission Lines

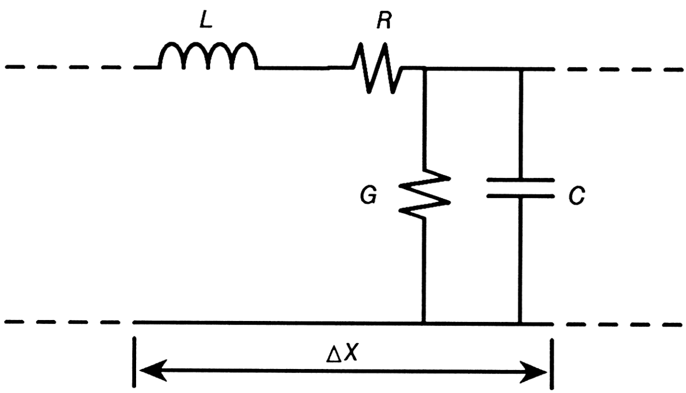
This chapter contains programs that calculate the impedance of a lossy high-frequency transmission line and the high-frequency characteristic impedances for five types of transmission line configurations.

Transmission Line Calculations ("LINE")

This program calculates the input impedance of a lossy transmission line terminated in Z_L. The program provides an exact solution when the distributed line parameters R_0 (defined as $\sqrt{L \div C}$), R, and G are given. It provides an approximate solution when R_0 and the conductor and dielectric losses are given.



The transmission line shown has a lumped model composed of elements L, C, R, and G as follows:



From this model the following equations can be derived:

$$R_0 = \sqrt{\frac{L}{C}}$$

$$r = \frac{R}{L} = \frac{v\,R}{R_0}$$

$$g = \frac{G}{C} = v\,R_0 G$$

where:

L = inductance/unit length.

C = capacitance/unit length.

G = conductance/unit length.

R = resistance/unit length.

$v = 3 \times 10^8\, v_r$.

v_r = relative phase velocity.

f = frequency, Hz.

$\omega = 2\pi f$ radians/second.

and

$$\alpha = \frac{1}{\sqrt{2}\,v}\left[rg - \omega^2 + \sqrt{(r^2 + \omega^2)(g^2 + \omega^2)}\right]^{1/2}$$

$$\beta = \frac{1}{\sqrt{2}\,v}\left[\omega^2 - rg + \sqrt{(r^2 + \omega^2)(g^2 + \omega^2)}\right]^{1/2}$$

The approximate solution is:

$$\mathrm{Re}\{Z_0\} = R_0\left[1 + \frac{1}{2}\left(\frac{\alpha_C - \alpha_D}{\beta_0}\right)\left(\frac{3\alpha_D + \alpha_C}{\beta_0}\right)\right]$$

$$\mathrm{Im}\{Z_0\} = R_0\left[\frac{\alpha_D - \alpha_C}{\beta_0}\right]$$

$$\alpha = \alpha_C + \alpha_D$$

$$\beta = \beta_0\left[1 + \frac{1}{2}\left(\frac{\alpha_C - \alpha_D}{\beta_0}\right)^2\right]$$

where:

α_C = conductor loss, nepers/unit length = $0.5(R/R_0)$

α_D = dielectric loss, nepers/unit length = $0.5(GR)$

$\beta_0 = \omega/\mathrm{v}$

Then

$$Z_{in} = Z_0\left(\frac{1 + \Gamma_L e^{-2\gamma l}}{1 - \Gamma_L e^{-2\gamma l}}\right)\text{ohms.}$$

where:

$$\Gamma_L = \frac{Z_L - Z_0}{Z_L + Z_0}$$

l = line length

Z_L = impedance of termination (ohms)

Z_0 = characteristic impedance of line = $\mathrm{Re}\{Z_0\} + j\,\mathrm{Im}\{Z_0\}$ (ohms)

γ = propagation constant of line = $\alpha + j\,\beta$

Z_0 and γ are computed differently depending on which solution is selected.

$$\mathrm{Re}\{Z_0\} = \frac{R_0}{\sqrt{2(g^2 + \omega^2)}}\left[rg + \omega^2 + \sqrt{(r^2 + \omega^2)(g^2 + \omega^2)}\right]^{1/2}$$

$$\mathrm{Im}\{Z_0\} = \frac{\pm R_0}{\sqrt{2(g^2 + \omega^2)}}\left[-(rg + \omega^2) + \sqrt{(r^2 + \omega^2)(g^2 + \omega^2)}\right]^{1/2}$$

The + sign is chosen when $g \geq r$ and the − sign is chosen when $g < r$.

Variables Used.

In Equations	Description	In Program
f	Frequency.	f
v_r	Relative phase velocity.	Vr
R_0	Characteristic impedance.	R0
l	Line length.	l
Z_L	Impedance of termination.	ZL

Remarks.

- Flag 21 (*printer enable*) is set or cleared to match flag 55 (*printer existence*). This automatically produces printer output if flag 55 is set.
- Registers R_{00} thru R_{06} are used for storing intermediate results. Be sure to set the SIZE to at least seven registers (■ [MODES] [▼] SIZE 7 [ENTER]) before running "LINE".

Program Instructions.

1. Key the "LINE" program (listed on page 102) into your calculator.
2. Select the desired angular and coordinate modes.
3. Press [XEQ] LINE (to run the "LINE" program). The program displays a variable menu containing f, ν_r, R_0, l, and Z_L (displayed as F, VR, R0, L, and ZL).
4. Store a value into each variable by keying in the value and pressing the corresponding menu key.

5. After all five variables have been stored, press [R/S].
6. Select an exact or approximate solution.

   For an exact solution:

   a. Press EXACT.
   b. The program prompts for G. Key in the conductance value and press [R/S].
   c. The program then prompts for R. Key in the resistance value and press [R/S].

   For an approximate solution:

   a. Press APROX.
   b. The program prompts for the conductor loss. Key in the value and press [R/S].
   c. The program then prompts for the dielectric loss. Key in the value and press [R/S].
7. The value for Z_{in} is calculated and displayed. To calculate another solution, go to step 6. To work a new problem, press [EXIT] and go to step 4.

Example. A transmission line has the following properties:

R = 1.2664 ohms/cm.

G = 0.00004187 siemens/cm.

R_0 = 55 ohms.

v_r = 0.85.

What is the input impedance of 3.5 cm of this line at 2 GHz if it is terminated in Z_L = 75 $\measuredangle -30°$ ohms?

Select Degrees and Polar modes, select FIX 2 display format, and run the "LINE" program.

Keys	Display
■[MODES] DEG ■[MODES] POLAR ■[DISP] FIX 02 [XEQ] LINE	x: 0.00 F VR R0 L ZL
2 [E] 9 F	f=2,000,000,000.00 F VR R0 L ZL

.85 VR	Vr=0.85 F VR R0 L ZL
55 R0	R0=55.00 F VR R0 L ZL
3.5 L	l=3.50 F VR R0 L ZL
75 [ENTER] 30 [+/-] ■ [COMPLEX] ZL	ZL=75.00 ∡-30.00 F VR R0 L ZL
[R/S]	x: 2.75 EXACT APROX
EXACT	Y: 2.75 G?0.00
4.187 [E] 5 [+/-] [R/S]	Y: 4.19E-5 R?0.00
1.2664 [R/S]	Zin=48.01 ∡28.48 EXACT APROX

The required input impedance is 48.01 ∡28.48° ohms.

"LINE" Program Listing.

Program:	Comments:
00 { 404-Byte Prgm } 01 LBL "LINE" 02 MVAR "f" 03 MVAR "Vr" 04 MVAR "R0" 05 MVAR "l" 06 MVAR "ZL" 07 CF 21 08 FS? 55 09 SF 21	Defines the menu variables and sets or clears flag 21 to match flag 55.
10 LBL A 11 CLA 12 VARMENU "LINE" 13 STOP 14 ALENG 15 X≠0? 16 GTO A	Displays the variable menu and stops. The program continues only when [R/S] is pressed.
17 RCL "f" 18 1E10 19 ÷ 20 STO 04 21 2 22 PI 23 × 24 × 25 STO 03 26 RCL "l" 27 2 28 × 29 3 30 RCL× "Vr" 31 STO 00 32 ÷ 33 STO 02	Calculates intermediate results used by both solutions.

```
34 LBL B
35 CLMENU
36 "EXACT"
37 KEY 1 XEQ C
38 "APROX"
39 KEY 6 XEQ D
40 KEY 9 GTO A
41 MENU
42 STOP
43 GTO B
```

Displays a menu for selecting the type of solution. The EXIT key is defined to go back to the variable menu.

```
44 LBL C
45 EXITALL
46 INPUT "G"
47 INPUT "R"
48 RCL× 00
49 RCL÷ "R0"
50 STO 01
51 RCL "G"
52 RCL× "R0"
53 STO× 00
54 RCL 03
55 RCL 01
56 →POL
57 SQRT
58 STO 01
59 X<>Y
60 2
61 ÷
62 STO 06
63 RCL 03
64 RCL 00
65 →POL
66 SQRT
67 STO 00
68 X<>Y
69 2
70 ÷
71 STO 03
72 RCL+ 06
```

Calculates the "exact" solution.

```
73 STO 05
74 RCL 06
75 RCL- 03
76 STO "Vr"
77 RCL 01
78 RCL÷ 00
79 STO× "R0"
80 RCL 01
81 RCL× 00
82 STO× 02
83 GTO E
```

```
84 LBL D
85 EXITALL
86 CLMENU
87 "C loss?"
88 PROMPT
89 STO 00
90 "D loss?"
91 PROMPT
92 STO 03
93 RCL "l"
94 RCL "Vr"
95 RCL 04
96 PI
97 ×
98 1.5
99 ÷
100 X<>Y
101 ÷
102 STO 06
103 ×
104 2
105 ×
106 STO 02
107 RCL 03
108 10
109 LN
110 20
111 ÷
```

Calculates the “approximate” solution.

```
RCL÷ 06
STO× 00
STO× 03
RCL 03
RCL- 00
ENTER
STO 01
RCL 03
3
×
RCL 00
+
×
2
÷
+/-
1
+
→POL
STO× "R0"
X<>Y
STO "Vr"
RCL 01
X↑2
2
÷
1
+
RCL 00
RCL+ 03
→POL
STO× 02
X<>Y
STO 05
```

Completes the calculations for both solutions.

```
LBL E
POLAR
RCL "ZL"
COMPLEX
STO 06
X<>Y
STO 01
RCL 05
RCL 02
→REC
+/-
E↑X
STO 02
X<>Y
180
×
PI
÷
STO 03
RCL 06
RCL- "Vr"
RCL 01
RCL÷ "R0"
→REC
1
+
→POL
1/X
-2
×
X<>Y
+/-
X<>Y
→REC
1
+
→POL
RCL× 02
```

```
184 1/X
185 X<>Y
186 RCL- 03
187 +/-
188 X<>Y
189 →REC
190 1
191 -
192 →POL
193 1/X
194 2
195 ×
196 X<>Y
197 +/-
198 X<>Y
199 →REC
200 1
201 +
202 →POL
203 RCL× "R0"
204 X<>Y
205 RCL "Vr"
206 +
207 COMPLEX
```

```
208 "Zin="
209 ARCL ST X
210 AVIEW
211 END
```

Displays the result and returns to the menu at LBL B.

Transmission Line Impedance ("TLI")

This program calculates the high frequency characteristic impedance (Z_0) for five types of transmission lines:

Open two-wire line:

$$Z_0 = \frac{120}{\sqrt{\varepsilon_r}} \ln\left(\frac{2D}{d}\right)$$

Single wire near ground:

$$Z_0 = \frac{138}{\sqrt{\varepsilon_r}} \log\left(\frac{4h}{d}\right)$$

Balanced wires near ground:

$$Z_0 = \frac{276}{\sqrt{\varepsilon_r}} \log\left\{\frac{2D}{d}\left[1 + \left(\frac{D}{2h}\right)^2\right]^{-1/2}\right\}$$

Wires in parallel near ground:

$$Z_0 = \frac{69}{\sqrt{\varepsilon_r}} \log\left\{\frac{4h}{d}\left[1 + \left(\frac{2h}{D}\right)^2\right]^{+1/2}\right\}$$

Coaxial line:

$$Z_0 = \frac{60}{\sqrt{\varepsilon_r}} \ln \frac{D}{d}$$

Variables Used.

In Equations	Description	In Program
D	Wire spacing.	D
d	Wire diameter.	dia
h	Height of wire (above ground).	h
ε_r	Relative permittivity.	e

Remarks. Flag 21 (*printer enable*) is set or cleared to match flag 55 (*printer existence*). This automatically produces printer output if flag 55 is set.

Program Instructions.

1. Key the "TLI" program (listed on page 111) into your calculator.
2. Press [XEQ] TLI (to run the "TLI" program).
3. Use the variable menu displayed by the program to store the required inputs for the particular line configuration you're working with.
4. Press [R/S].
5. The program displays `Line Config?` and a menu containing the five types of line configurations. Select a configuration by pressing the corresponding menu key:

   OP (open two-wire line)

   SW (single wire near ground)

   B (balanced wires near ground)

   P (wires in parallel near ground)

   C (coaxial line)
6. The value of Z_0 is displayed and the program returns to the input menu (step 3).

Example. Calculate Z_0 of RG-218/U coaxial cable with wire spacing, D = 0.68 in., wire *dia* = 0.195 in., and ε_r of the polyethylene insulation = 2.3.

Select FIX 2 display format and run the "TLI" program.

■[DISP] FIX 02 [XEQ] TLI	x: 0.00 D DIA H E

Store the inputs.

.68 D	D=0.68 D DIA H E

.195 DIA — dia=0.20
D DIA H E

2.3 E — e=2.30
D DIA H E

Calculate Z_0 for the *coaxial* configuration.

[R/S] — Line Config?
OP SW B P C

C — Z0=49.42
D DIA H E

Now calculate Z_0 for an open two-wire (air) line with $D = 6$ in. and *dia* $= 0.0808$ in. (ε_r of air $= 1$).

6 D — D=6.00
D DIA H E

.0808 DIA — dia=0.08
D DIA H E

1 E — e=1.00
D DIA H E

[R/S] — Line Config?
OP SW B P C

OP — Z0=600.08
D DIA H E

This time, calculate Z_0 of an air line (ε_r of air = 1) consisting of a single 0.1285 inch diameter wire 6 inches from a ground plane.

.1285 DIA	dia=0.13 D DIA H E
6 H	h=6.00 D DIA H E
1 E	e=1.00 D DIA H E
R/S	Line Config? OP SW B P C
SW	Z0=313.44 D DIA H E

"TLI" Program Listing.

Program:	**Comments:**
`00 { 225-Byte Prgm }`	
`01 LBL "TLI"`	Declares menu variables and sets or clears flag 21 to match flag 55.
`02 MVAR "D"`	
`03 MVAR "dia"`	
`04 MVAR "h"`	
`05 MVAR "e"`	
`06 CF 21`	
`07 FS? 55`	
`08 SF 21`	

```
09 LBL A
10 VARMENU "TLI"
11 CLA
12 STOP
13 ALENG
14 X≠0?
15 GTO A
```

Displays the variable menu and stops.

```
16 CLMENU
17 "OP"
18 KEY 1 XEQ 01
19 "SW"
20 KEY 2 XEQ 02
21 "B"
22 KEY 3 XEQ 03
23 "P"
24 KEY 4 XEQ 04
25 "C"
26 KEY 5 XEQ 05
27 KEY 9 GTO A
```

Defines the programmable menu for the five configurations. The [EXIT] key is defined to return to the variable menu.

```
28 MENU
29 "Line Config?"
30 PROMPT
31 GTO A
```

Displays the menu and prompts for a selection. After the appropriate routine is executed, returns to the variable menu.

```
32 LBL 01
33 XEQ 07
34 LN
35 120
36 GTO 06
```

Calculates Z_0 for an open two-wire configuration.

```
37 LBL 02
38 XEQ 08
39 LOG
40 138
41 GTO 06
```

Calculates Z_0 for a single wire near the ground.

Calculates Z_0 for balanced wires near the ground.

```
LBL 03
XEQ 09
1/X
1
+
SQRT
1/X
XEQ 07
×
LOG
276
GTO 06
```

Calculates Z_0 for wires in parallel near the ground.

```
LBL 04
XEQ 09
1
+
XEQ 08
SQRT
XEQ 08
×
LOG
69
GTO 06
```

Calculates Z_0 for a coaxial line configuration.

```
LBL 05
RCL "D"
RCL÷ "dia"
LN
60
```

Completes the calculation and displays the result.

```
LBL 06
RCL "e"
SQRT
÷
×
"Z0="
ARCL ST X
AVIEW
RTN
```

79 LBL 07 80 2 81 RCL× "D" 82 RCL÷ "dia" 83 RTN	Calculates an intermediate result.
84 LBL 08 85 4 86 RCL× "h" 87 RCL÷ "dia" 88 RTN	Calculates an intermediate result.
89 LBL 09 90 2 91 RCL× "h" 92 RCL÷ "D" 93 X↑2 94 END	Calculates an intermediate result.

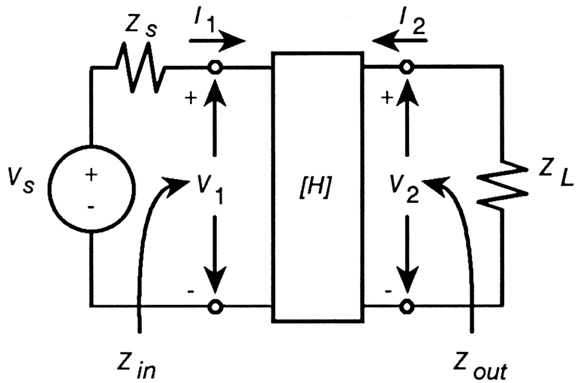
5

Amplifier Analysis

This chapter contains programs that calculate small-signal properties of a transistor amplifier and automate a method of transistor bias optimization.

Transistor Amplifier Performance ("TAP")

This program calculates certain small-signal properties of a transistor amplifier given the h-parameter matrix and the source and load impedances. The program calculates current and voltage gains, and input and output impedances.



Equations. The definition of the h-parameter matrix is:

$$\begin{bmatrix} V_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} h_i & h_r \\ h_f & h_o \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \end{bmatrix}$$

The current gain is:

$$A_i = \frac{i_2}{i_1} = \frac{-h_f}{1 + h_o Z_L}$$

The voltage gain is:

$$A_v = \frac{v_2}{v_1} = \frac{A_i Z_L}{Z_{in}}$$

The voltage gain with a source resistor is:

$$A_{vs} = \frac{v_2}{v_S} = \frac{A_i Z_L}{Z_{in} + Z_S}$$

The input impedance is:

$$Z_{in} = h_i + h_r Z_L A_i$$

The output impedance is:

$$Z_{out} = \frac{h_i + Z_S}{h_o h_i + h_o Z_S - h_f h_r}$$

Variables Used.

In Equations	Description	In Program
h_i	Matrix h-parameter.	hi
h_r	Matrix h-parameter.	hr
h_f	Matrix h-parameter.	hf
h_o	Matrix h-parameter.	ho
Z_S	Source impedance.	Zs
Z_L	Load impedance.	Zl

Remarks. Flag 21 (*printer enable*) is set or cleared to match flag 55 (*printer existence*). This automatically produces printer output if flag 55 is set.

Program Instructions.

1. Key the "TAP" program (listed on page 120) into your calculator.
2. Press [XEQ] TAP (to run the "TAP" program).
3. Use the variable menu displayed by the program to store the four h-parameter values (h_i, h_r, h_f, and h_o) and the source and load impedances (Z_S and Z_L).
4. Press [R/S] to display the result menu.
5. Press the appropriate menu key for each result you want to calculate:
   - AI to display the current gain, A_i.
   - AV to display the voltage gain, A_v.

- AVS to display the voltage gain with a source resistor, A_{vs}.
- ZIN to display the input impedance, Z_{in}.
- ZOUT to display the output impedance, Z_{out}.

6. Press [EXIT] to return to the variable menu (step 3).

Example. What are the small-signal properties of a transistor that has the following h-parameter matrix and has source and load impedances of 1,000 and 10,000 ohms, respectively?

$$h = \begin{bmatrix} 1100 & 250E-6 \\ 50 & 25E-6 \end{bmatrix}$$

Select the FIX 4 display format and run the "TAP" program.

■[DISP] FIX 04 [XEQ] TAP	x: 0.0000 HI HR HF HO ZS ZL

Using the variable menu, enter each of the six input values and then press [R/S].

1100 HI	hi=1,100.0000 HI HR HF HO ZS ZL
250 [E] 6 [+/-] HR	hr=0.0003 HI HR HF HO ZS ZL
50 HF	hf=50.0000 HI HR HF HO ZS ZL
25 [E] 6 [+/-] HO	ho=2.5000E-5 HI HR HF HO ZS ZL
1000 ZS	Zs=1,000.0000 HI HR HF HO ZS ZL

10000 ZL

```
Zl=10,000.0000
 HI  HR  HF  HO  ZS  ZL
```

[R/S]

```
x: 0.0000
 AI  AV  AVS  ZIN  ZOUT
```

For each of the outputs you want to calculate, press the corresponding menu key.

AI

```
Ai=-40.0000
 AI  AV  AVS  ZIN  ZOUT
```

AV

```
Av=-400.0000
 AI  AV  AVS  ZIN  ZOUT
```

AVS

```
Avs=-200.0000
 AI  AV  AVS  ZIN  ZOUT
```

ZIN

```
Zin=1,000.0000
 AI  AV  AVS  ZIN  ZOUT
```

ZOUT

```
Zout=52,500.0000
 AI  AV  AVS  ZIN  ZOUT
```

Press [EXIT] to return to the input menu.

[EXIT]

```
x: 52,500.0000
 HI  HR  HF  HO  ZS  ZL
```

From here you can work another problem or press [EXIT] again to quit.

"TAP" Program Listing.

Program: | **Comments:**

```
00 { 227-Byte Prgm }
01 LBL "TAP"
02 MVAR "hi"
03 MVAR "hr"
04 MVAR "hf"
05 MVAR "ho"
06 MVAR "Zs"
07 MVAR "Zl"
08 CF 21
09 FS? 55
10 SF 21
```

Declares the menu variables and sets or clears flag 21 to match flag 55.

```
11 LBL A
12 VARMENU "TAP"
13 CLA
14 STOP
15 ALENG
16 X≠0?
17 GTO A
```

Displays the variable menu and stops.

```
18 CLMENU
19 "Ai"
20 KEY 1 XEQ 01
21 "Av"
22 KEY 2 XEQ 02
23 "Avs"
24 KEY 3 XEQ 03
25 "Zin"
26 KEY 4 XEQ 04
27 "Zout"
28 KEY 5 XEQ 05
29 KEY 9 GTO A
```

Defines the programmable menu for displaying the results. The [EXIT] key is defined to return to the variable menu.

```
30 LBL 00
31 MENU
32 STOP
33 GTO 00
```

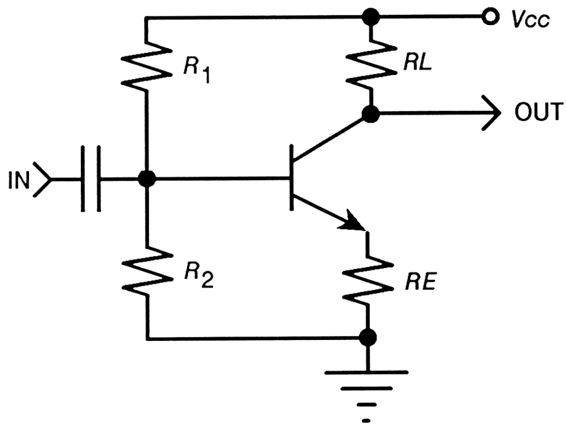
Displays the programmable menu and stops. The menu is redisplayed after each result.

```
34 LBL 01              Calculates A_i.
35 XEQ 07
36 "Ai"
37 GTO 06

38 LBL 02              Calculates A_v.
39 XEQ 08
40 ÷
41 "Av"
42 GTO 06

43 LBL 03              Calculates A_vs.
44 XEQ 08
45 RCL+ "Zs"
46 ÷
47 "Avs"
48 GTO 06

49 LBL 04              Calculates Z_in.
50 XEQ 08
51 "Zin"
52 GTO 06

53 LBL 05              Calculates Z_out.
54 RCL "hi"
55 RCL+ "Zs"
56 LASTX
57 RCL "ho"
58 ×
59 LASTX
60 RCL× "Zs"
61 +
62 RCL "hf"
63 RCL× "hr"
64 -
65 ÷
66 "Zout"
```

67 LBL 06	Displays a result.
68 ⊢"="	
69 ARCL ST X	
70 AVIEW	
71 RTN	
72 LBL 07	Calculates A_i.
73 RCL "hf"	
74 +/-	
75 1	
76 RCL "ho"	
77 RCL× "Zl"	
78 +	
79 ÷	
80 RTN	
81 LBL 08	Calculates Z_{in}.
82 XEQ 07	
83 RCL× "Zl"	
84 ENTER	
85 RCL× "hr"	
86 RCL+ "hi"	
87 END	

Transistor Amplifier Bias Optimization ("BIAS")

This program automates the method of bias optimization described in "Designing Class 'A' Amplifiers to Meet Specified Tolerances," by Ward J. Helms (*Electronics*, August 8, 1974). The program requires you to specify a set of parameters from which it determines, by an iterative technique, the optimum values for R_1, R_2, R_E, and R_L. The minimum power gain is also computed.



How "BIAS" Works. First, values are input for the variables listed in the table on page 126. Then, the transistor's thermal resistance is calculated:

$$\theta_{JA} = (T_{\max} - 25°C) / P_D$$

and the minimum load resistance and emitter resistance are estimated:

$$R_{L1} = \frac{\theta_{JA} V_{CC}^2}{4.4(T_{J\max} - T_{A\max})} = R_{Ln}$$

$$R_{E1} = 0.1 R_{L1} = R_{En}$$

Next, the quiescent, maximum, and minimum collector currents are calculated:

$$I_{CQ} = \frac{V_{CC}}{2(R_{Ln} + R_{En})}$$

$$I_{C\max} = I_{CQ}(1 + \Delta I_{CQ})$$

$$I_{C\min} = I_{CQ}(1 - \Delta I_{CQ})$$

From these, we can calculate the base-emitter voltage under hot, high-current conditions (V_{BEX}) and under cold, low-current conditions (V_{BEN}).

$$T_{\max} = \theta_{JA} I_{CQ}(V_{CC}/2) + T_{A\max}$$

$$V_{BEX} = V_{BE1\min} + \Delta V_{BE} \log(I_{C\max}/I_1) - 0.0022(T_{\max} - 25°C)$$

$$T_{\min} = \theta_{JA} I_{CQ}(V_{CC}/2)(1 - (\Delta I_{CQ})^2) + T_{A\min}$$

$$V_{BEN} = V_{BE1\max} + \Delta V_{BE} \log(I_{C\min}/I_1) - 0.0022(T_{\min} - 25°C)$$

Now, a better estimate of the emitter resistance can be made:

$$R_{E(n+1)} = \frac{-2(V_{BEX} - V_{BEN})}{I_{C\max} - I_{C\min}}$$

From this point, if $V_{BEX} > V_{BEN}$, then R_E is set to zero, R_L is increased by 10% and the design procedure is repeated. Iterations continue until

$$\frac{R_{E(n+1)} - R_{En}}{R_{En}} < 0.5\%$$

If at any time the condition $T_{\max} > T_{J\max}$ occurs, R_L is increased by 10%.

When the iterative procedure is complete, $T_{\max}$, $I_{C\max}$, $T_{\min}$, and $I_{C\min}$ are displayed.

Values for

$h_{FE\,max}$ = maximum worst-case current gain at T_{max} or T_{min} and $I_{C\,max}$ or $I_{C\,min}$

and

$h_{FE\,min}$ = minimum worst-case current gain at T_{max} or T_{min} and $I_{C\,max}$ or $I_{C\,min}$

are determined from the transistor's data sheet. The Thevenin-equivalent resistance (R_B) and voltage (V_{BB}) of the amplifier's bias network are calculated:

$$R_B = \frac{h_{FE\,max}\, h_{FE\,min} \left[R_{E(n+1)} \left(I_{C\,max} - I_{C\,min} \right) + V_{BEX} - V_{BEN} \right]}{h_{FE\,max} I_{C\,min} - h_{FE\,min} I_{C\,max}}$$

$$V_{BB} = V_{BEN} + I_{C\,min}\left(\left(R_B / h_{FE\,min} \right) + R_{E(n+1)} \right)$$

Now the bias resistors are calculated:

$$R_1 = \frac{R_B V_{CC}}{V_{BB}}$$

$$R_2 = \frac{R_B V_{CC}}{V_{CC} - V_{BB}}$$

Finally, the minimum power gain and minimum signal power are calculated:

$$A_P = \frac{R_B R_L h_{FE\,min}}{R_E \left(R_B + h_{FE\,min} R_E \right)}$$

$$P_S = \left(1 - \Delta I_{CQ} \right)^2 \left(\frac{V_{CC}^{\,2} R_L}{8 \left(R_L + R_E \right)^2} \right)$$

Variables Used.

In Equations	Description	In Program
V_{CC}	Source voltage (volts).	VCC
ΔI_{CQ}	Maximum desired percentage variation of quiescent current.	dICQ
$T_{A\ \max}$	Maximum ambient temperature (use the maximum case temperature for a transistor mounted on a heat sink).	TAmax
$T_{A\ \min}$	Minimum ambient temperature.	TAmin
$T_{J\ \max}$	Maximum junction temperature.	TJmax
P_D	Maximum rated power dissipation at 25°C.	PD
I_1	Collector current, usually selected for convenience so that I_1 and 10 I_1 at 25°C bracket the expected operating point.	I1
ΔV_{BE}	Typical base-emitter voltage change over the range of I_1 to 10 I_1 at 25°C.	dVBE
$V_{BE1\ \min}$	Minimum base-emitter voltage at I_1 at 25°C.	VBE1min
$V_{BE1\ \max}$	Maximum base-emitter voltage at I_1 at 25°C.	VBE1max
$h_{FE\ \max}$	Maximum worst-case current gain at $T_{\max}$ or $T_{\min}$ and $I_{C\ \max}$ or $I_{C\ \min}$.	R_{01}
$h_{FE\ \min}$	Minimum worst-case current gain at $T_{\max}$ or $T_{\min}$ and $I_{C\ \max}$ or $I_{C\ \min}$.	R_{02}

Remarks.

- Flag 01 is used for branching control.
- Flag 02 is used to control the labels given to results.
- Flag 21 (*printer enable*) and flag 55 (*printer existence*) control printer output.
- Registers R_{00} through R_{09} are used for storing intermediate results. Be sure to set the SIZE to at least 10 registers (■[MODES] [▼] SIZE 10 [ENTER]) before running "BIAS".

Program Instructions.

1. Key the "BIAS" program (listed on page 130) into your calculator.
2. Press [XEQ] BIAS (to run the "BIAS" program).
3. Input the variables as prompted; press [R/S] after each entry.
4. After the last input, the program calculates and displays T_{max} and I_{max}. The program then calculates and displays T_{min} and I_{min}.
5. When you see Hmax?, key in a value for the maximum worst-case current gain at T_{max} or T_{min} and I_{Cmax} or I_{Cmin}. Press [R/S].
6. When you see Hmin?, key in a value for the minimum worst-case current gain at T_{max} or T_{min} and I_{Cmax} or I_{Cmin}. Press [R/S].
7. The program then calculates values for R_E, R_L, R_1, R_2, and A_P. If you are not using a printer, press [R/S] after each result is displayed.

Example. A single-stage class "A" amplifier is connected to a 30-volt power supply. Calculate the maximum power output and maximum power gain obtained from a transistor over an ambient temperature range of 0°C to 70°C, with a maximum quiescent-current variation of ±20% (or .2).

From the transistor's data sheet,

TJ_{max} = 150°C

PD = 0.36 W

ΔV_{BE} = 0.10 v from 3 to 30 mA

V_{BE1min} = 0.52 v at 3 mA at 25°C

V_{BE1max} = 0.72 v at 3 mA at 25°C

$I_1 = 0.001$ A

$h_{FEmax} = 600$

$h_{FEmin} = 100$

Select ENG 2 display format and run the "BIAS" program.

■ [DISP] ENG 02 [XEQ] BIAS	Y: 0.00E0 VCC?0.00E0
30 [R/S]	Y: 30.0E0 dICQ?0.00E0
.2 [R/S]	Y: 200.E-3 TAmax?0.00E0
70 [R/S]	Y: 70.0E0 TAmin?0.00E0
0 [R/S]	Y: 0.00E0 TJmax?0.00E0
150 [R/S]	Y: 150.E0 PD?0.00E0
.36 [R/S]	Y: 360.E-3 I1?0.00E0
.001 [R/S]	Y: 1.00E-3 dVBE?0.00E0

.1 [R/S]

```
Y: 100.E-3
VBE1min?0.00E0
```

.52 [R/S]

```
Y: 520.E-3
VBE1max?0.00E0
```

.72 [R/S]

```
Tmax=148.E0
Imax=18.0E-3
```

[R/S]

```
Tmin=74.8E0
Imin=12.0E-3
```

[R/S]

```
Hmax?
X: 500.E-3
```

Refer again to the transistor's data sheet and input h_{FEmax}.

600 [R/S]

```
Hmin?
X: 600.E0
```

Now, input h_{FEmin} .

100 [R/S]

```
RE=
X: 115.E0
```

[R/S]

```
RL=
X: 888.E0
```

[R/S]

```
R2=
X: 4.18E3
```

[R/S]

```
R1=
X: 45.0E3
```

[R/S]

```
AP=
x: 22.9E0
```

"BIAS" Program Listing.

Program:	Comments:
`00 { 553-Byte Prgm }`	
`01 LBL "BIAS"`	Inputs values.
`02 INPUT "VCC"`	
`03 INPUT "dICQ"`	
`04 INPUT "TAmax"`	
`05 STO 07`	
`06 INPUT "TAmin"`	
`07 STO 08`	
`08 INPUT "TJmax"`	
`09 INPUT "PD"`	
`10 STO 09`	
`11 INPUT "I1"`	
`12 INPUT "dVBE"`	
`13 INPUT "VBE1min"`	
`14 INPUT "VBE1max"`	
`15 CF 01`	Initializes values for iterative process.
`16 SF 02`	
`17 SF 21`	
`18 RCL "TJmax"`	
`19 25`	
`20 -`	
`21 RCL÷ 09`	
`22 STO 09`	
`23 RCL "VCC"`	
`24 X↑2`	
`25 ×`	
`26 RCL "TJmax"`	
`27 RCL- "TAmax"`	
`28 4.4`	
`29 ×`	
`30 ÷`	

```
STO 03
0.1
×
STO 04
LBL 00                          Begins iterative loop.
RCL "VCC"
2
÷
ENTER
ENTER
RCL 03
RCL+ 04
÷
STO 00
RCL× 09
×
RCL+ 07
RCL "TJmax"
X<>Y
X>Y?
GTO 05
XEQ 03
+/-
RCL "dICQ"
1
+
XEQ 04
RCL+ "VBE1min"
STO 05
1
RCL "dICQ"
X↑2
-
2
÷
RCL× 00
RCL× 09
RCL× "VCC"
RCL+ 08
XEQ 03
```

```
71 +/-
72 1
73 RCL- "dICQ"
74 XEQ 04
75 RCL+ "VBE1max"
76 STO 06
77 RCL 05
78 X>Y?
79 GTO 02
80 -
81 RCL÷ 00
82 RCL÷ "dICQ"
83 RCL 04
84 X<>Y
85 STO 04
```

Repeats iterative loop as needed.

```
86 %CH
87 0.5
88 X≤Y?
89 GTO 00
90 FS? 01
91 GTO 01
92 SF 01
93 GTO 00
```

Prompts for h_{FEmax} and h_{FEmin}.

```
94 LBL 01
95 CF 01
96 "Hmax?"
97 PROMPT
98 STO 01
99 "Hmin?"
100 PROMPT
101 STO 02
```

Calculates R_B.

```
102 X<>Y
103 RCL "dICQ"
104 2
105 ×
106 RCL× 00
107 RCL× 04
108 RCL+ 05
```

```
109 RCL- 06
110 RCL× 02
111 RCL× 01
112 1
113 RCL- "dICQ"
114 RCL× 01
115 1
116 RCL+ "dICQ"
117 RCL× 02
118 -
119 ÷
120 RCL÷ 00
121 STO 07
```

```
122 RCL÷ 02
123 RCL+ 04
124 RCL× 00
125 1
126 RCL- "dICQ"
127 ×
128 RCL+ 06
129 STO 08
```

Calculates V_{BB}.

```
130 RCL "VCC"
131 X<>Y
132 ÷
133 RCL× 07
```

Calculates R_1.

```
134 RCL "VCC"
135 RCL× 07
136 LASTX
137 RCL- 08
138 ÷
```

Calculates R_2.

```
139 RCL 03
140 RCL 04
141 "RE"
142 XEQ 02
143 "RL"
144 XEQ 02
145 "R2"
146 XEQ 02
```

Recalls R_L and R_E, and then displays each of the four values in the stack.

```
147 "R1"
148 XEQ 02
```

```
149 ÷
150 RCL× 02
151 RCL 07
152 ×
153 LASTX
154 RCL 04
155 RCL× 02
156 +
157 ÷
158 LOG
159 10
160 ×
161 "AP"
```

Calculates and then displays the power gain. (When the RTN at line 168 is reached, the program ends because there are no pending subroutine calls.)

```
162 LBL 02
163 ⊢"="
164 FS? 55
165 ARCL ST X
166 AVIEW
167 R↓
168 RTN
```

Displays a result and rolls the stack down one register (for the next result).

```
169 LBL 03
170 FS? 01
171 XEQ A
172 25
173 -
174 2.2E-3
175 ×
176 RTN
```

Calculates temperature.

```
177 LBL A
178 "Tmax="
179 FC? 02
180 "Tmin="
181 ARCL ST X
182 ⊢"°F"
183 RTN
```

Displays proper label and temperature.

Calculates current.

```
184 LBL 04
185 RCL× 00
186 FS? 01
187 XEQ I
188 RCL÷ "I1"
189 LOG
190 RCL× "dVBE"
191 +
192 RTN
```

Displays proper label and current.

```
193 LBL I
194 FS? 02
195 ⊢"Imax="
196 FC?C 02
197 ⊢"Imin="
198 ARCL ST X
199 AVIEW
200 RTN
201 STO 07
202 CLX
203 STO 04
```

Increases R_L by 10%.

```
204 LBL 05
205 1.1
206 STO× 03
207 GTO 00
208 END
```

6

Truth Tables

This chapter contains two programs for testing logical expressions. The first program, "PTTBL" (*print truth table*), allows you to print a complete truth table. The second program, "ITTBL" (*interactive truth table*), allows you to display any row of a truth table.

Writing a Logical Expression as a Program

Before using "PTTBL" or "ITTBL" you must enter a logical expression. This is done by writing a program that represents the expression.

Each of the storage registers R_{00} through R_{05} holds a 1 or 0. These are the inputs to the function that you write. In your program recall each register as it's needed and use the Boolean logic functions (AND, OR, and XOR) to create the expression.

Here's a simple example. This program represents the expression *A* OR *B*, where *A* is stored in R_{01} and *B* is stored in R_{00}.

```
00 { 10-Byte Prgm }
01 LBL "OR"
02 RCL 01
03 RCL 00
04 OR
05 END
```

The calculator's built-in NOT function returns the *36-bit* logical NOT of the number in the X-register. To perform a *single-bit* logical NOT, execute these three functions:

```
SIGN
LASTX
-
```

The logical expression

$$(A \text{ AND } B \text{ AND } C) \text{ OR } (A \text{ AND } \overline{B} \text{ AND } \overline{C})$$

can be represented with the following program (assuming *A* is in R_{02}, *B* is in R_{01}, and *C* is in R_{00}).

"EXMPL" Program Listing.

Program:	Comments:
00 { 31-Byte Prgm }	
01 LBL "EXMPL"	Calculates A AND B AND C.
02 RCL 02	
03 RCL 01	
04 AND	
05 RCL 00	
06 AND	
07 RCL 00	Calculates $\overline{C}$.
08 SIGN	
09 LASTX	
10 -	
11 RCL 01	Calculates $\overline{B}$.
12 SIGN	
13 LASTX	
14 -	
15 AND	Calculates $\overline{B}$ AND $\overline{C}$.
16 RCL 02	Calculates A AND $\overline{B}$ AND $\overline{C}$.
17 AND	
18 OR	Calculates (A AND B AND C) OR (A AND $\overline{B}$ AND $\overline{C}$)
19 END	

Note

Since the name of your program is stored in a variable (*FCN*), do not use a global label longer than six characters.

Printing a Truth Table ("PTTBL")

This program prints a truth table for a logical expression written as previously described. You must provide the name of the function (global label) and the number of the most significant input bit. (Bits are numbered right to left; the right-most bit is number 0 and is stored in R_{00}.)

Required Programs. "PTTBL" (page 141) and "FCN?" (page 156).

Variables Used.

Description	In Program
Function name.	FCN
Loop counter.	count
Most significant bit.	msb

Remarks.

- Registers R_{00} through R_{05} are used as input registers (bits) for the Boolean expression. Be sure to set the SIZE to at least six registers (■[MODES] [▼] SIZE 6 [ENTER]) before running "PTTBL".
- This program clears all of the storage registers.
- Flag 12 is used to produce double-wide output from the printer.

Program Instructions.

1. Key the "PTTBL" and "FCN?" programs into your calculator.
2. Key in the program that represents the logical expression (described in the "Writing a Logical Expression as a Program" section on page 137).
3. Press [XEQ] PTTBL (to run the "PTTBL" program).
4. When you see Function Name?, use the ALPHA menu to type the name of the function (global label). Press [R/S].

5. The program then prompts you for the most significant bit (msb?). Key in the number of the highest register you want to use as an input to your function ($1 \leq msb \leq 5$).
6. Press [R/S] to print the truth table.

Example. Print a truth table for the expression on page 137. (If you haven't done it already, key in the "EXMPL" program on page 138.)

Run the "PTTBL" program.

[XEQ] PTTBL

```
Function Name?
ABCDE FGHI JKLM NOPQ RSTUV WXYZ
```

Key in the name of the logical expression you want to print.

EXMPL

```
EXMPL_
ABCDE FGHI JKLM NOPQ RSTUV WXYZ
```

[R/S]

```
Y: 0
msb?0
```

Since the logical expression uses three inputs (A, B, and C), the most significant bit is stored in R_{02}.

2 [R/S]

```
Y: 1
X: 0
```

Printer Output.

```
2 1 0         EXMPL

0 0 0           0
0 0 1           0
0 1 0           0
0 1 1           0
1 0 0           1
1 0 1           0
1 1 0           0
1 1 1           1
```

"PTTBL" Program Listing.

Program:	Comments:
00 { 168-Byte Prgm } 01 LBL "PTTBL" 02 XEQ "FCN?" 03 PRON 04 CF 12 05 CLST 06 ALL	Prompts for a function name using the "FCN?" utility. Printing is enabled.
07 INPUT "msb" 08 1 09 + 10 2 11 X<>Y 12 Y↑X 13 1 14 - 15 1E3 16 ÷ 17 STO "count"	Prompts for the number of the most significant bit and sets up the loop counter.
18 CLA 19 PRA	Prints a single blank line.
20 RCL "msb" 21 LBL 00 22 ARCL ST X 23 ⊢" " 24 DSE ST X 25 GTO 00 26 ARCL ST X 27 ⊢" " 28 ARCL "FCN" 29 ⊢"␊" 30 PRA 31 SF 12	Prints a table header. (Note that line 27 has seven blank spaces between the double quotes.)

```
32 LBL 01
33 CLRG
34 RCL "count"
35 IP
36 RCL "msb"
```

Initializes the inputs (storage registers) and prepares to print a row in the truth table.

```
37 LBL 02
38 BIT?
39 XEQ 04
40 DSE ST X
41 GTO 02
42 BIT?
43 XEQ 04
```

Stores the input bits into the appropriate storage registers.

```
44 XEQ IND "FCN"
```

Evaluates the expression for the given inputs.

```
45 CLA
46 RCL "msb"
47 LBL 03
48 ARCL IND ST X
49 DSE ST X
50 GTO 03
```

Accumulates the input bits into the Alpha register.

```
51 ARCL IND ST X
52 ⊢"    "
53 ARCL ST Y
```

Accumulates the output bit into the Alpha register. (Note that line 52 has four blank spaces between the double quotes.)

```
54 PRA
55 ISG "count"
56 GTO 01
```

Prints a line in the truth table and completes the loop.

```
57 CF 12
58 RTN
```

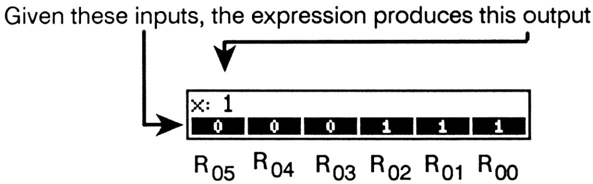
Resets the double-wide flag and ends.

```
59 LBL 04
60 1
61 STO IND ST Y
62 R↓
63 END
```

Stores an input bit into the storage register identified in the Y-register.

An Interactive Truth Table ("ITTBL")

This program allows you to change any of the inputs and display the corresponding output. Here's a typical display:



Required Programs. "ITTBL" (page 145) and "FCN?" (page 156).

Remarks.

- *FCN* (the function name) is the only variable used by this program.
- Registers R_{00} through R_{05} are used as input registers (bits) for the Boolean expression. Be sure to set the SIZE to at least six registers (■ [MODES] [▼] SIZE 6 [ENTER]) before running "ITTBL".
- This program clears all of the storage registers.
- This program does not produce printed output.

Program Instructions.

1. Key the "ITTBL" and "FCN?" programs into your calculator.
2. Key in the program that represents the logical expression (described in the "Writing a Logical Expression as a Program" section on page 137).
3. Press [XEQ] ITTBL (to run the "ITTBL" program).
4. When you see `Function Name?`, use the ALPHA menu to type the name of the function (global label). Press [R/S].
5. The program enters an interactive mode. The menu labels represent the inputs (R_{05} through R_{00}) as shown above.

To change one of the inputs, press the corresponding menu key. The program returns the result to the X-register. By toggling the inputs you can view the value of the expression for any combination of inputs.

6. To quit, press [EXIT].

Example. Change the inputs and view the value of the expression. (If you haven't done it already, key in the "EXMPL" program on page 138.)

Run the "ITTBL" program.

[XEQ] ITTBL

```
Function Name?
ABCDE FGHI JKLM NOPQ RSTUV WXYZ
```

Key in the name of the logical expression you want to print. (If you just worked the example in the previous section, *FCN* probably still contains "EXMPL". Press [←]; if you see EXMPL, then simply press [R/S] to continue.)

EXMPL

```
EXMPL_
ABCDE FGHI JKLM NOPQ RSTUV WXYZ
```

[R/S]

```
x: 0
0 0 0 0 0 0
```

Toggle the most significant bit (bit 2 for this example) by pressing the fourth menu key.

0 *(the [LOG] key)*

```
x: 1
0 0 0 1 0 0
```

Notice that the output (X-register) is now 1. Now toggle bits 1 and 0 (the fifth and sixth menu keys).

0 *(the [LN] key)*

```
x: 0
0 0 0 1 1 0
```

0 *(the [XEQ] key)*

```
x: 1
0 0 0 1 1 1
```

When you're finished, press [EXIT] to quit.

[EXIT]

```
Y: 1
X: 1
```

"ITTBL" Program Listing.

Program: / **Comments:**

```
00 { 125-Byte Prgm }
01 LBL "ITTBL"
02 XEQ "FCN?"
03 ALENG
04 X=0?
05 GTO 09
```

Prompts for a function name using the "FCN?" utility.

```
06 BINM
07 ALL
08 CLRG
09 CLMENU
```

Initializes by selecting Binary mode, selecting ALL display format, and clearing the storage registers and programmable menu definitions.

```
10 KEY 9 GTO 09
11 LBL A
12 CLA
13 ARCL 05
14 KEY 1 XEQ 01
15 CLA
16 ARCL 04
17 KEY 2 XEQ 02
18 CLA
19 ARCL 03
20 KEY 3 XEQ 03
21 CLA
22 ARCL 02
23 KEY 4 XEQ 04
24 CLA
25 ARCL 01
26 KEY 5 XEQ 05
27 CLA
28 ARCL 00
29 KEY 6 XEQ 06
```

Defines the [EXIT] key to branch to LBL 09. Defines the six top-row menu keys using the numbers in the corresponding storage registers as menu labels.

30 XEQ IND "FCN"	Evaluates the logical expression and leaves the value in the X-register.
31 MENU 32 STOP 33 GTO A	Displays the menu and stops. Pressing [R/S] redisplays the menu.
34 LBL 01 35 5 36 GTO 07 37 LBL 02 38 4 39 GTO 07 40 LBL 03 41 3 42 GTO 07 43 LBL 04 44 2 45 GTO 07 46 LBL 05 47 1 48 GTO 07 49 LBL 06 50 CLX	Produces the appropriate register number, depending on which menu key is pressed.
51 LBL 07 52 RCL IND ST X 53 SIGN 54 LASTX 55 - 56 STO IND ST Y 57 RTN	Toggles the bit in a particular storage register.
58 LBL 09 59 EXITALL 60 END	Exits all menus and ends.

7

Utilities

The programs in this chapter are general-purpose utilities and subroutines used by other programs in this book. You may also find them useful when writing your own programs.

Circuit Calculation Utilities

Impedance of an Element ("C→Z" and "L→Z")

This program converts the value for a capacitor or inductor in the X-register to a complex impedance. Before executing "C→Z" (*capacitance to impedance*) or "L→Z" (*inductance to impedance*), store the radian frequency in the variable *w*.

Remarks.

- *w* is the radian frequency, $2\pi f$ radians/second. (A lowercase "W" is used because the HP-42S does not have a lowercase omega character.)
- This program sets Rectangular mode.

"C→Z" and "L→Z" Program Listing.

Program:	Comments:
`00 { 30-Byte Prgm }`	

`01 LBL "C→Z"`
`02 XEQ 00`
`03 1/X`
`04 RTN`
	Calculates the impedance for the given capacitance (which is the reciprocal of impedance for inductance).

`05 LBL "L→Z"`
`06 LBL 00`
`07 RECT`
`08 RCL× "w"`
`09 0`
`10 X<>Y`
`11 COMPLEX`
`12 END`
	Calculates the impedance for the given inductance.

Combining Parallel Impedances (“ZP”)

The “ZP” program takes two complex impedances (in the X- and Y-registers) and returns the combined impedance for the two elements connected in parallel.

“ZP” Program Listing.

```
00 { 11-Byte Prgm }
01 LBL "ZP"
02 1/X
03 X<>Y
04 1/X
05 +
06 1/X
07 END
```

Entering Radian Frequency (“FQ?”)

The “FQ?” program prompts for a value of *w*, the radian frequency value used by several programs in this book. Whenever you see `Radian Frequency(2πf)?`, key in the frequency and press [R/S].

Remarks.

- *w* is the radian frequency, 2πf radians/second. (A lowercase “W” is used because the HP-42S does not have a lowercase omega character.)
- The program sets flag 25 (*error ignore*) to prevent an error from stopping the program if *w* doesn’t exist.

“FQ?” Program Listing.

```
00 { 44-Byte Prgm }
01 LBL "FQ?"
02 "Radian Frequenc"
03 ⊢"y(2πf)?"
04 SF 25
05 CLX
06 RCL "w"
07 CF 25
08 PROMPT
09 STO "w"
10 END
```

Circuit Element Input Utility ("EL?")

This program displays a menu for entering six types of circuit elements. It is designed to be used by other circuit analysis programs, such as the "CIRCT" program on page 45.

Each routine in this program displays a menu for entering an element using common units. For example, when you enter a resistor, the calculator displays:

```
Value?
 OHM  KOHM  MOHM                  S
```

You can enter a 2,000-ohm resistor by pressing 2000 OHM , 2 KOHM , or .002 MOHM .

The program returns two numbers to the stack:

Y: *ee.yyxx*
X: *element value*

where the X-register contains the *element value* you keyed in adjusted to the default units shown in the following table. *ee* is the element type in the table, and *yy* and *xx* are the location numbers entered into the stack. These numbers are used by programs such as "CIRCT" to indicate where a particular element occurs in a circuit.

Element Type	Units	Type Code
Resistor	Ohms	82
Capacitor	Farads	67
Inductor	Henrys	76
General impedance	Ohms	90*
Voltage source	Volts	86*
Current source	Amperes	73*

*If the element value is a complex number, the type code is negative.

Remarks.

- The type code is temporarily stored in a variable named *TYPE*.
- Flag 08 is set to indicate when an element has been entered successfully.

"EL?" Program Listing.

Program:	Comments:
`00 { 374-Byte Prgm }`	

`01 LBL "EL?"`
`02 LBL A`
`03 RECT`
`04 CF 08`
`05 "R"`
`06 KEY 1 GTO 09`
`07 "C"`
`08 KEY 2 GTO 12`
`09 "L"`
`10 KEY 3 GTO 13`
`11 "Z"`
`12 KEY 4 GTO 10`
`13 "V"`
`14 KEY 5 GTO 14`
`15 "I"`
`16 KEY 6 GTO 15`
`17 KEY 9 GTO 99`
	Defines menu for entering location and type of element. The [EXIT] key is defined to cause a branch to the "END" (which causes a return to the calling program).

```
18 "Location: "
19 ├"# [ENTER] #"
20 MENU
21 PROMPT
22 GTO A
```

Displays the input message and the menu. Pressing [R/S] redisplays the menu.

```
23 LBL 09
24 82
25 GTO 11
```

Enters the code for a resistor.

```
26 LBL 10
27 90
```

Enters the code for a general impedance.

```
28 LBL 11
29 XEQ 00
30 "OHM"
31 KEY 1 XEQ 01
32 "KOHM"
33 KEY 2 XEQ 02
34 "MOHM"
35 KEY 3 XEQ 03
36 "S"
37 KEY 6 XEQ 08
38 GTO B
```

Defines a menu for entering units for a resistor of general impedance. (The S in line 36 refers to the SI unit "siemens.")

```
39 LBL 12
40 67
41 XEQ 00
42 "pF"
43 KEY 1 XEQ 07
44 "nF"
45 KEY 2 XEQ 06
46 "μF"
47 KEY 3 XEQ 05
48 "mF"
49 KEY 4 XEQ 04
50 "F"
51 KEY 5 XEQ 01
52 GTO B
```

Enters the code for a capacitor and defines a menu for entering units of capacitance.

```
53 LBL 13
54 76
55 XEQ 00
56 "mH"
57 KEY 1 XEQ 04
58 "H"
59 KEY 2 XEQ 01
60 "KH"
61 KEY 3 XEQ 02
62 "MH"
63 KEY 4 XEQ 03
64 GTO B
```

Enters the code for an inductor and defines a menu for entering units of inductance.

```
65 LBL 14
66 POLAR
67 86
68 XEQ 00
69 "mV"
70 KEY 1 XEQ 04
71 "V"
72 KEY 2 XEQ 01
73 "KV"
74 KEY 3 XEQ 02
75 GTO B
```

Enters the code for a voltage source and defines a menu for entering voltage units.

```
76 LBL 15
77 POLAR
78 73
79 XEQ 00
80 "mA"
81 KEY 1 XEQ 04
82 "A"
83 KEY 2 XEQ 01
84 "KA"
85 KEY 3 XEQ 02
86 GTO B
```

Enters the code for a current source and defines a menu for entering current units.

```
87 LBL 08
88 1/X
89 RTN
90 LBL 07
91 XEQ 04
92 LBL 06
93 XEQ 04
94 LBL 05
95 XEQ 04
96 LBL 04
97 1E3
98 ÷
99 RTN
100 LBL 03
101 XEQ 02
102 LBL 02
103 1E3
104 ×
105 LBL 01
106 RTN
```

Adjusts the element value to be expressed in the *default* units.

```
107 LBL 00
108 X<>Y
109 1
110 %
111 R↑
112 +
113 1
114 %
115 R↑
116 +
117 STO "TYPE"
```

Combines the location number and element type code into a single number and stores it in the variable *TYPE*.

```
118 CLMENU
119 KEY 9 GTO A
120 RTN
```

Clears the programmable menu and defines the [EXIT] key to return to the first menu.

```
121 LBL B
122 MENU
123 1
124 "Value?"
125 PROMPT
```

Displays the units menu and prompts for a value. Pressing [R/S] causes the default value of 1 to be used.

```
126 SF 08
127 RCL "TYPE"
128 CLV "TYPE"
129 X<>Y
130 REAL?
131 RTN
132 X<>Y
133 +/-
134 X<>Y
```

Sets flag 08 to indicate an element has been entered. Returns the type code in the Y-register and the element value in the X-register. If the element value is complex, the type code is made negative.

```
135 LBL 99
136 END
```

Ends the program.

Function Name Utility ("FCN?")

This program prompts for a function name and then stores the name into a variable named *FCN*. If *FCN* contains a string, it is recalled into the Alpha register (press [←] to clear the `Function Name?` message). Use the ALPHA menu to type the name of the function (global program label) and then press [R/S] to continue.

Remarks.

- *FCN* contains the variable name (up to six characters).
- Flag 21 (*printer enable*) is cleared by the program.
- Flag 25 (*error ignore*) is used to prevent the program from stopping if *FCN* doesn't exist.
- The plotting programs in the owner's manual use a similar technique to prompt for a function name. If you have the "FCN?" program in your calculator, you can shorten one or both of the plotting programs by calling "FCN?" as a subroutine (`XEQ "FCN?"`).

"FCN?" Program Listing.

Program:	Comments:
`00 { 48-Byte Prgm }`	

`01 LBL "FCN?"`
`02 "Function Name?"`
`03 CF 21`
`04 AVIEW`
	Displays the prompt.

`05 SF 25`
`06 RCL "FCN"`
`07 CF 25`
`08 CLA`
`09 STR?`
`10 ARCL ST X`
	If *FCN* exists and contains an Alpha string, recalls that string to the Alpha register.

`11 AON` `12 STOP`	Turns on the ALPHA menu and stops.

`13 AOFF`
`14 ASTO "FCN"`
`15 END`
	Turns off the ALPHA menu and stores the first six characters in the Alpha menu into *FCN*.

Yes/No Utility ("Y?N")

This program displays a menu for a Yes/No decision. It returns a zero if you press NO, [R/S], or [EXIT]; it returns a 1 if you press YES.

If you want to use this utility in your own programs, simply place a message in the Alpha register and then execute the utility (XEQ "Y?N"). Your program can then test the X-register to detect a "yes" (1) or a "no" (0).

Remarks.

- Flag 21 (*printer enable*) is cleared by this program.
- This program redefines the programmable menu.

"Y?N" Program Listing.

```
00 { 41-Byte Prgm }
01 LBL "Y?N"
02 CF 21
03 AVIEW
04 CLMENU
05 "YES"
06 KEY 1 GTO 01
07 "NO"
08 KEY 6 GTO 00
09 KEY 9 GTO 00
10 MENU
11 STOP
12 LBL 00
13 0
14 RTN
15 LBL 01
16 1
17 END
```

Product Over Sum Utility ("P/S")

This routine is quite useful for many electrical engineering applications. It simply takes two values (in the X- and Y-registers) and returns the product of the two values divided by the sum of the two values:

$$\frac{xy}{x + y}$$

Note

If $x + y = 0$, the program will error at line 05 (`Divide by 0`).

"P/S" Program Listing.

```
00 { 15-Byte Prgm }
01 LBL "P/S"
02 RCL× ST Y
03 X<>Y
04 RCL+ ST L
05 ÷
06 END
```

Size Utility ("SIZE?")

This program returns the number of storage registers available.

"SIZE?" Program Listing.

```
00 { 27-Byte Prgm }
01 LBL "SIZE?"
02 SF 25
03 RCL "REGS"
04 FC? 25
05 0
06 FC?C 25
07 RTN
08 DIM?
09 x
10 END
```

Step-by-Step Solutions for Your HP-42S Calculator

Electrical Engineering contains a variety of programs and examples to provide solutions for electrical engineers and engineering students.

- **Circuit Calculations**
  Voltage Division • Current Division • Power Triangle • Frequency Response of Transfer Function • RC Timing • Delta-Wye Conversions
- **Network Analysis**
  Using the Circuit Editor • Mesh Analysis • Nodal Analysis • Impedance of a Ladder Network
- **Filter Design**
  Active Filter Design • Butterworth Filter Design
- **Transmission Lines**
  Transmission Line Calculations • Transmission Line Impedance
- **Amplifier Analysis**
  Transistor Amplifier Performance • Transistor Amplifier Bias Optimization
- **Truth Tables**
  Writing a Logical Expression as a Program • Printing a Truth Table • An Interactive Truth Table
- **Utilities**
  Circuit Calculation Utilities • Circuit Element Input Utility • Function Name Utility • Yes/No Utility • Product Over Sum Utility • Size Utility

HEWLETT PACKARD

Reorder Number
00042-90021

00042-90023 English
Printed in Canada 11/88

0 88698 00037 3