Electrical Engineering

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

Edition 1  November 1988
Reorder Number  00042-90021
Notice

This book and any keystroke programs contained herein are provided “as is” and are subject to change without notice. Hewlett-Packard Company makes no warranty of any kind with regard to this book or the keystroke programs contained herein, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard Company shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance, or use of this book or the keystroke programs contained herein.

Hewlett-Packard Company 1988. All rights reserved. Reproduction, adaptation, or translation of this book, including any programs, is prohibited without prior written permission by Hewlett-Packard Company; except as allowed under the copyright laws. Hewlett-Packard Company grants you the right to use any program contained in this book in a Hewlett-Packard calculator.

The programs that control your calculator are copyrighted and all rights are reserved. Reproduction, adaptation, or translation of those programs without prior written permission of Hewlett-Packard Company is also prohibited.

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

Printing History

Edition 1 November 1988 Mfg. No. 00042-90023
### Contents

5  How To Use This Book

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>Circuit Calculations</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Voltage Division (&quot;V÷&quot;)</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>Current Division (&quot;I÷&quot;)</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>Power Triangle (&quot;PWR3&quot;)</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>Frequency Response of Transfer Function (&quot;FQRS&quot;)</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>RC Timing (&quot;RC&quot;)</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>Delta-Wye Conversions (&quot;DY&quot;)</td>
</tr>
<tr>
<td>2</td>
<td>38</td>
<td>Network Analysis</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>Using the Circuit Editor (&quot;CIRCT&quot;)</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>Starting the &quot;CIRCT&quot; Program</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>Initializing the Circuit</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>Displaying Circuit Elements</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>Adding Circuit Elements</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>Deleting a Circuit Element</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>Printing the Circuit</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>Quitting &quot;CIRCT&quot;</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>Saving Circuits</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>Storing Elements</td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>Mesh Analysis (&quot;MESH&quot;)</td>
</tr>
<tr>
<td></td>
<td>63</td>
<td>Nodal Analysis (&quot;NODAL&quot;)</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>Impedance of a Ladder Network (&quot;LADDR&quot;)</td>
</tr>
<tr>
<td>Contents</td>
<td>72</td>
<td>Filter Design</td>
</tr>
<tr>
<td>----------</td>
<td>-----</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>Active Filter Design (&quot;AF&quot;)</td>
</tr>
<tr>
<td></td>
<td>82</td>
<td>Butterworth Filter Design (&quot;BF&quot;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>95</th>
<th>Transmission Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>96</td>
<td>Transmission Line Calculations (&quot;LINE&quot;)</td>
</tr>
<tr>
<td></td>
<td>108</td>
<td>Transmission Line Impedance (&quot;TLI&quot;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
<th>115</th>
<th>Amplifier Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>116</td>
<td>Transistor Amplifier Performance (&quot;TAP&quot;)</td>
</tr>
<tr>
<td></td>
<td>123</td>
<td>Transistor Amplifier Bias Optimization (&quot;BIAS&quot;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>136</th>
<th>Truth Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>137</td>
<td>Writing a Logical Expression as a Program</td>
</tr>
<tr>
<td></td>
<td>139</td>
<td>Printing a Truth Table (&quot;PTTBL&quot;)</td>
</tr>
<tr>
<td></td>
<td>143</td>
<td>An Interactive Truth Table (&quot;ITTBL&quot;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7</th>
<th>147</th>
<th>Utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>148</td>
<td>Circuit Calculation Utilities</td>
</tr>
<tr>
<td></td>
<td>148</td>
<td>Impedance of an Element (&quot;C→Z&quot; and &quot;L→Z&quot;)</td>
</tr>
<tr>
<td></td>
<td>149</td>
<td>Combining Parallel Impedances (&quot;ZP&quot;)</td>
</tr>
<tr>
<td></td>
<td>149</td>
<td>Entering Radian Frequency (&quot;FQ?&quot;)</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>Circuit Element Input Utility (&quot;EL?&quot;)</td>
</tr>
<tr>
<td></td>
<td>156</td>
<td>Function Name Utility (&quot;FCN?&quot;)</td>
</tr>
<tr>
<td></td>
<td>157</td>
<td>Yes/No Utility (&quot;Y?N&quot;)</td>
</tr>
<tr>
<td></td>
<td>158</td>
<td>Product Over Sum Utility (&quot;P/S&quot;)</td>
</tr>
<tr>
<td></td>
<td>159</td>
<td>Size Utility (&quot;SIZE?&quot;)</td>
</tr>
</tbody>
</table>
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 \[ PRINT \ A \ P O N \] to enable printing.

If you are not using a printer, be sure to disable printing (\[ PRINT \ A \ P O F F \]) 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 (\( \angle \)).

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.
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.

<table>
<thead>
<tr>
<th>In Equation</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_T )</td>
<td>Terminal voltage (volts).</td>
<td>( VT )</td>
</tr>
<tr>
<td>( V_2 )</td>
<td>Voltage across impedance ( Z_2 ).</td>
<td>( V2 )</td>
</tr>
<tr>
<td>( Z_1 )</td>
<td>Impedance (ohms).</td>
<td>( Z1 )</td>
</tr>
<tr>
<td>( Z_2 )</td>
<td>Impedance (ohms).</td>
<td>( Z2 )</td>
</tr>
</tbody>
</table>

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.

Store the known values.

```
80 ENTER 10 [COMPLEX] 
```

```
VT=80.00 <10.00
VT: V2 21 22
```
Now, solve for the unknown impedance.

The unknown impedance \((Z_2)\) is 37.71 \(\angle 79.29^\circ\) ohms.

**"V\(\div\)" Program Listing.**

**Program:**

```plaintext
00 { 148-Byte Prgm }
01 LBL "V\(\div\"
02 MVAR "VT"
03 MVAR "V2"
04 MVAR "Z1"
05 MVAR "Z2"
06 CF 21
07 FS? 55
08 SF 21
09 LBL 00
10 CLA
11 YARMENU "V\(\div\"
12 STOP
```

**Comments:**

- Defines menu variables.
- Sets or clears flag 21 to match flag 55.
- Displays the variable menu and stops.
Determines which variable was selected by subtracting the ASCII codes of the first two characters in the variable name. For example, when you press \[\text{Z1}\] 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 \[90 - 49 = 41\].

Calculates \(V_2\).

Calculates \(V_T\).

Calculates \(Z_2\).
46 LBL 41
47 RCL "Z2"
48 RCL× "VT"
49 RCL÷ "V2"
50 RCL- "Z2"
51 STO "Z1"
52 VIEW "Z1"
53 END

Calculates $Z_1$. 

12 1: Voltage Division
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.

<table>
<thead>
<tr>
<th>In Equation</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_T )</td>
<td>Terminal current (amps).</td>
<td>( \text{I.T} )</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>Current in impedance ( Z_2 ).</td>
<td>( \text{I2} )</td>
</tr>
<tr>
<td>( Z_1 )</td>
<td>Impedance(_1) (ohms).</td>
<td>( \text{Z1} )</td>
</tr>
<tr>
<td>( Z_2 )</td>
<td>Impedance(_2) (ohms).</td>
<td>( \text{Z2} )</td>
</tr>
</tbody>
</table>

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 10\( \Omega \) resistor is 12 \( \angle 45^\circ \) amperes?

![Circuit Diagram]

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+]

Key in the three known values.

17.5 [Z1]
Now calculate the unknown current.

The unknown current ($I_T$) is $18.86 \angle 45^\circ$ amperes.

"I÷" Program Listing.

**Program:**

```
Program Listing.

00 ( 155-Byte Prgm )
01 LBL "I÷"
02 MVAR "I.T"
03 MVAR "I2"
04 MVAR "Z1"
05 MVAR "Z2"
06 CF 21
07 FS? 55
08 SF 21
09 LBL 00
10 CLA
11 VARMENU "I÷"
12 STOP
13 ATOX
14 ATOX
15 -
16 XEQ IND ST X
17 GTO 00
```

**Comments:**

- Defines menu variables.
- Sets or clears flag 21 to match flag 55.
- Displays the variable menu and stops.
- 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 $73 - 46 = 27$. 

1: Current Division
Calculates $I_2$.

18 LBL 23
19 RCL "Z1"
20 RCLx "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 RCLx "I2"
32 RCL ÷ "Z1"
33 STO "I.T"
34 VIEW "I.T"
35 RTN

Calculates $Z2$.

36 LBL 40
37 RCL "Z1"
38 RCLx "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 RCLx "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.

\[ P = VI \cos \theta \]

\[ S = VI \]

\[ Q = VI \sin \theta \]

Lagging

Leading

<table>
<thead>
<tr>
<th>Variables Used.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In Figure</strong></td>
</tr>
<tr>
<td>( V )</td>
</tr>
<tr>
<td>( I )</td>
</tr>
<tr>
<td>( P )</td>
</tr>
<tr>
<td>( Q )</td>
</tr>
<tr>
<td>( S )</td>
</tr>
<tr>
<td>Power factor (( \cos \theta )).</td>
</tr>
</tbody>
</table>

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 \( \angle 10^\circ \) volts and a resulting current of 2.85 \( \angle -40^\circ \) amperes, determine the power triangle and the power factor.

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

```
100 ENTER 10 [COMPLEX] V
2.85 ENTER 40 +/- [COMPLEX] I
```

The average power is 183.19 watts.

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

The apparent power is 285 watts.

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
```
The required voltage is 105.26 °10.00° volts.

“PWR3” Program Listing.

**Program:**

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"
08 DEG
09 POLAR
10 CF 21
11 FS? 55
12 SF 21
13 LBL A
14 VARMENU "PWR3"
15 CLA
16 STOP
17 ATOX
18 X=0?
19 GTO A
20 XTOA
21 1
22 AROT
23 R→
24 73
25 -
26 XEQ IND ST X

**Comments:**

Declares the menu variables.

Sets Degrees and Polar modes. Sets or clears flag 21 to match flag 55.

Displays variable menu and stops. Pressing [R/S] redisplaysthe menu.

The ASCII character code of the first letter in the variable name is subtracted from 73 to determinewhich routine to execute.
27 ASTO ST L
28 STO IND ST L
29 VIEW IND ST L
30 GTO A

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

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

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

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

57 LBL 10
58 XEQ C
59 ABS
60 RTN

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

Calculates \( V \).

Calculates \( I \).

Calculates \( P \).

Calculates \( Q \).

Calculates \( S \).

1: Power Triangle
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 $pf$. 

22 1: Power Triangle
Frequency Response of Transfer Function ("FQRS")

For a transfer function of the form

\[ G(S) = \frac{K(Z_{1}S + 1)}{S^{N}(Z_{2}S + 1)(Z_{3}S + 1)\left[\frac{S^{2}}{\omega_{0}^{2}} + \frac{2Z_{4}S}{\omega_{0}} + 1\right]} \]

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

Variables Used.

<table>
<thead>
<tr>
<th>In Equation</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K )</td>
<td>Transfer function parameter.</td>
<td>( K )</td>
</tr>
<tr>
<td>( N )</td>
<td>Transfer function parameter.</td>
<td>( N )</td>
</tr>
<tr>
<td>( Z_{1} )</td>
<td>Transfer function parameter.</td>
<td>( Z_{1} )</td>
</tr>
<tr>
<td>( Z_{2} )</td>
<td>Transfer function parameter.</td>
<td>( Z_{2} )</td>
</tr>
<tr>
<td>( Z_{3} )</td>
<td>Transfer function parameter.</td>
<td>( Z_{3} )</td>
</tr>
<tr>
<td>( Z_{4} )</td>
<td>Transfer function parameter.</td>
<td>( Z_{4} )</td>
</tr>
<tr>
<td>( \omega_{0} )</td>
<td>Transfer function parameter.</td>
<td>( \omega_{0} )</td>
</tr>
<tr>
<td>( \omega )</td>
<td>Input frequency, ( 2\pi f ) (radians/sec).</td>
<td>( \omega )</td>
</tr>
</tbody>
</table>

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 \( \omega_{0} \) 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 \( \text{[XEQ] FQRS} \) (to run the "FQRS" program).

3. The program prompts you for values of \( K, N, Z_1, Z_2, Z_3, Z_4, \) and \( \omega_0 \). At each prompt, key in the value and press \( \text{[R/S]} \).

4. The program displays \( G(S) \) and \( \log |G(S)| \) and stops. Press \( \text{[R/S]} \) 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, \( \omega \), is 0.01 rad/sec.

First put \( G(S) \) into proper form:

\[
G (S) = \frac{0.2(1.67S + 1)}{S(S + 1)\left[\left(\frac{S}{6}\right)^2 + \left(\frac{S}{6}\right) + 1\right]}
\]

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

\[
\text{[DISP] [FIX] 02 [XEQ] FQRS}
\]

\[
\text{Y: 0.00 K:0.00}
\]

\[
\text{0.2 [R/S]}
\]

\[
\text{Y: 0.20 N:0.00}
\]

\[
\text{1 [R/S]}
\]

\[
\text{Y: 1.00 Z1:0.00}
\]

\[
\text{1.67 [R/S]}
\]

\[
\text{Y: 1.67 Z2:0.00}
\]
G(S) and log |G(S)| for the given conditions are 20.00 ∠ -89.71° and 1.30.
"FQRS" Program Listing.

Program:

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"

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

Comments:

Prompts for each input value.

Calculates G(S).
36 RCLx "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 RCLx "Z1"
51 RCLx "ω"
52 COMPLEX
53 ×

54 POLAR
55 "G(S)="
56 ARCL ST X
57 ABS
58 LOG
59 -"|LOG |G(S)|="
60 ARCL ST X
61 AVIEW
62 END

Displays results.
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:

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

where \( e \) is the base of natural logarithms.
Variables Used.

<table>
<thead>
<tr>
<th>In Equations</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>Voltage before step (volts).</td>
<td>$V1$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>Voltage after step (volts).</td>
<td>$V2$</td>
</tr>
<tr>
<td>$V_i$</td>
<td>Instantaneous voltage (volts).</td>
<td>$Vi$</td>
</tr>
<tr>
<td>$R$</td>
<td>Resistance (ohms).</td>
<td>$R$</td>
</tr>
<tr>
<td>$C$</td>
<td>Capacitance (farads).</td>
<td>$C$</td>
</tr>
<tr>
<td>$t$</td>
<td>Time (seconds).</td>
<td>$t$</td>
</tr>
</tbody>
</table>

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 $\text{SOLVER} \rightarrow \text{RC}$.

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 $\text{EXIT} \rightarrow \text{EXIT}$ 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 $VI = 0$. Given a supply voltage of 12V and a 47 $\mu$F capacitor, what size resistor should you use to generate a one-second pulse?
Select the ENG 3 display format and the "RC" Solver program.

1. **Disp ENG 03**
2. **Solver RC**

- **0 V1**
- **12 V2**
- **2 ENTER 3 + 12 × V1**
- **47 E 6 +/- C**
- **1 T**
- **R**

V1=0.000E0
V1=12.000E0
V1=8.000E0
C=47.000E-6
V1=1.000E0
R=19.37E3

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 RCLx "C"
20 RCL÷ "t"
21 1/X
22 RCL− "R"
23 END

1: RC Timing
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.

<table>
<thead>
<tr>
<th>In Figure</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_{Y1}$</td>
<td>Impedance (ohms)</td>
<td>$Z_{Y1}$</td>
</tr>
<tr>
<td>$Z_{Y2}$</td>
<td>Impedance (ohms)</td>
<td>$Z_{Y2}$</td>
</tr>
<tr>
<td>$Z_{Y3}$</td>
<td>Impedance (ohms)</td>
<td>$Z_{Y3}$</td>
</tr>
<tr>
<td>$Z_{12}$</td>
<td>Impedance (ohms)</td>
<td>$Z_{12}$</td>
</tr>
<tr>
<td>$Z_{13}$</td>
<td>Impedance (ohms)</td>
<td>$Z_{13}$</td>
</tr>
<tr>
<td>$Z_{23}$</td>
<td>Impedance (ohms)</td>
<td>$Z_{23}$</td>
</tr>
</tbody>
</table>
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.

![Diagram of a delta circuit with labels: 1, 2, 3, 0 + i22Ω, 45Ω, 5 - i17Ω.]

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

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

Enter the delta values.

45 [ ]

0 [ENTER] 22 [COMPLEX] [ ]

5 [ENTER] 17 [ ]

Now calculate each of the wye values.

[ ]
"DY" Program Listing.

Program:

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"
08 CF 21
09 FS? 55
10 SF 21
11 LBL A
12 VARMENU "DY"
13 STOP
14 ATOX
15 XTOA
16 ATOX
17 XTOA
18 ATOX
19 XTOA
20 +
21 95
22 -
23 XEQ IND ST X

Comments:

Declarations menu variables.

Sets or clears flag 21 to match flag 55.

Returns the ASCII character codes of the selected variable name and restores the name in the Alpha register.

Adds the ASCII codes of the last two characters of the selected variable and subtracts 95 to determine which subroutine to execute.
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.

Calculates $Z_{Y1}$.

Calculates $Z_{Y2}$.

Calculates $Z_{Y3}$.

Shared subroutines.

Calculates $Z_{12}$.

Calculates $Z_{13}$.

Calculates $Z_{23}$. 

1: Delta-Wye Conversions
57 LBL 02
58 RCL "ZY1"
59 RCLx "ZY2"
60 LASTX
61 RCLx "ZY3"
62 +
63 RCL "ZY2"
64 RCLx "ZY3"
65 +
66 X<>Y
67 RTN

Shared subroutine.

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

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

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 }

You can view the elements in the circuit by using the [A] and [V] 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 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 \textbf{INIT YES}.

\begin{center}
\textit{INIT}
\end{center}

\begin{center}
\textbf{INITIALIZE Circuit?}
\end{center}

\begin{center}
\textbf{YES} \quad \textbf{NO}
\end{center}

\begin{center}
\textit{YES}
\end{center}

\begin{center}
\textbf{0-Element Circuit}
\end{center}

When there are no circuit elements, note that the \textbf{DEL}, \textbf{PRINT}, and \textbf{INIT} menu keys are\textit{ not} displayed or active.

Displaying Circuit Elements

Use the \( \textbf{\textup{A}} \) and \( \textbf{\textup{V}} \) keys to move up and down through the list of elements.

Here's a typical display:

\begin{center}
Element Number \quad \rightarrow \quad Element Location
\end{center}

\begin{center}
Element Type \quad \rightarrow \quad Element Value
\end{center}

\begin{center}
(3): 2\( \rightarrow \) 4
\end{center}

\begin{center}
+120.00\( \triangle \) 60.00
\end{center}

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\( \triangle \) 60.00° volts." (For mesh analysis, the location numbers would represent meshes 2 and 4.)

When a circuit element is displayed, the menu keys \textbf{ADD}, \textbf{DEL}, \textbf{PRINT}, and \textbf{INIT} are active even though they are not visible. (If you press \( \textbf{\textup{C}} \) 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:

```plaintext
{ 7-Element Circuit }

(1): 1 => 1
150.00 ± 10.00

(2): 1 ~ 1
16.00

(3): 1 ← 2
0.00 ± 10.00

(4): 1 ← 3
0.00 -124.00

(5): 2 ~ 2
32.00

(6): 2 ~ 3
12.00

(7): 3 ← 3
0.00 -116.00
```
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.
<table>
<thead>
<tr>
<th>Element Type</th>
<th>Units</th>
<th>Type Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistor</td>
<td>Ohms</td>
<td>82</td>
</tr>
<tr>
<td>Capacitor</td>
<td>Farads</td>
<td>67</td>
</tr>
<tr>
<td>Inductor</td>
<td>Henrys</td>
<td>76</td>
</tr>
<tr>
<td>General impedance</td>
<td>Ohms</td>
<td>90*</td>
</tr>
<tr>
<td>Voltage source</td>
<td>Volts</td>
<td>86*</td>
</tr>
<tr>
<td>Current source</td>
<td>Amperes</td>
<td>73*</td>
</tr>
</tbody>
</table>

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

**Remarks.**

- *CIRCT* is an $n \times 3$ 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:

00 ( 626-Byte Pgm )
01 LBL "CIRCT"
02 PROFF
03 CLX
04 STO 00
05 STO 01
06 CF 10
07 SF 25
08 INDEX "CIRCT"
09 FC?C 25
10 SF 10
11 FS? 10
12 GTO A
13 RCL "CIRCT"
14 DIM?
15 X<>Y
16 STO 01
17 LBL A
18 CLMEMU
19 "ADD"
20 KEY 1 GTO 01
21 "DEL"
22 FC? 10
23 KEY 2 GTO 02
24 "PRINT"
25 FC? 10
26 KEY 5 XEQ 12
27 "INIT"
28 FC? 10
29 KEY 6 GTO 06
30 KEY 7 XEQ 07
31 KEY 8 XEQ 08
32 KEY 9 GTO 99

Comments:

Initializes the program.

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

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

 Inserts the new circuit element into the CIRCT matrix.

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

Stores the new circuit element into CIRCT.
69 X<>Y
70 →
71 X<>Y
72 REAL?
73 GTO 11
74 COMPLEX
75 X<>Y
76 →
77 X<>Y
78 LBL 11
79 STOEL
80 RCLIJ
81 X<>Y
82 STO 00
83 1
84 STO+ 01
85 GTO 01

Inserts a new row into the *CIRCT* matrix.

86 LBL 05
87 I+
88 INSR
89 R↓
90 R↓
91 RCLIJ
92 GTO 04

93 LBL 10
94 R↓
95 1
96 ENTER
97 STOIJ
98 INSR
99 GTO 04

Inserts a new row at the top of the *CIRCT* matrix.
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 \( \text{-Element} \)
170 \( \text{Circuit}\)
171 AVIEW
172 RTN

173 LBL 82
174 "\(\)\(\)\(\)\(\)
175 RTN

Displays top-of-list message.

Alpha string to produce graphic resistor.
176 LBL 67
177 "<---<--| /|<--<--"  
178 RTN

179 LBL 76
180 "<---E"  
181 4
182 XTOA
183 t"E"
184 XTOA
185 t"E<->"  
186 R↓
187 RTN

188 LBL 90
189 "<--8<<<<<<8<<<<"  
190 RTN

191 LBL 86
192 "ffff8DDD8f<f"  
193 RTN

194 LBL 73
195 "<--8DDDS<T8<"  
196 RTN

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

Alpha string to produce graphic capacitor.

Alpha strings to produce graphic inductor.

Alpha string to produce graphic impedance.

Alpha string to produce graphic voltage source.

Alpha string to produce graphic current source.

Deletes the current circuit element.
211 STO 00
212 1
213 STO- 01
214 GTO A

215 LBL 06
216 "INITIALIZE ":
217 \texttt{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

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

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

Initializes the pointers and deletes \textit{CIRCT} variable.

Decrements the element pointer.

Increments the element pointer.
245 LBL 12
246 RCL 01
247 1E-3
248 x
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

262 LBL 99
263 EXITALL
264 END

Prints the graphic display for circuit elements in CIRCT.

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:

\[
I = \frac{Z}{V} = \frac{MATA}{MATB} = MATX
\]

Variables Used.

<table>
<thead>
<tr>
<th>In Equation</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n \times 3 ) matrix containing circuit elements.</td>
<td>CIRCT</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>Impedance matrix, ( Z ).</td>
<td>MATA</td>
</tr>
<tr>
<td>V</td>
<td>Voltage matrix, ( V ).</td>
<td>MATB</td>
</tr>
<tr>
<td>I</td>
<td>Solutions matrix.</td>
<td>MATX</td>
</tr>
<tr>
<td></td>
<td>Radian frequency, ( 2\pi f ) (radians/sec).</td>
<td>( w )</td>
</tr>
</tbody>
</table>

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 \( \boxed{\text{MATRIX}} \ SIMO \boxed{nn} \) (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
Be sure to set the SIZE to at least four registers (M Modes \[ S I Z E \ 4 \ \text{ENTER}) before running “MESH”.


**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 A PON to enable printing. If you’re not using a printer, be sure to disable printing (press PRINT A 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, I₁, I₂, and I₃.
First, select FIX 2 display format and then enter the circuit elements using the "CIRCT" program.

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

Now, add the new elements to the circuit.

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

Enter the value for the voltage source.

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]).

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.

Enter the capacitor shared by mesh 1 and mesh 3 in the same way.
Now enter the elements in mesh 2 that have not already been entered.

2 [ENTER] R 32 OHM

Location: # [ENTER] #

R C L 2 Y 1

2 [ENTER] 3 R 12 OHM

Location: # [ENTER] #

R C L 2 Y 1

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 2 Y 1

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 A 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= 6.28 4.72

56 2: Mesh Analysis
If you’re not using a printer, press \( \text{R/S} \) to display the current in mesh 2.

\[
\begin{array}{c}
\text{R/S} \\
I_2 = x: 2.61 \angle 53.37
\end{array}
\]

If you’re not using a printer, press \( \text{R/S} \) again to display the current in mesh 3.

\[
\begin{array}{c}
\text{R/S} \\
I_3 = x: 3.09 \angle -2.73
\end{array}
\]

“MESH” and “NODAL” Program Listing.

**Program:**

00 \{ 491-Byte Prgm \}
01 LBL "MESH"
02 CF 08
03 "No. Mesh "
04 \( \{ \text{"Currents?"} \) 
05 GTO A

06 LBL "NODAL"
07 SF 08
08 "No. Nodes?"

09 LBL A
10 RCL 02
11 PROMPT
12 STO 02

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?"

**Comments:**

Clears flag 08 (to indicate mesh analysis) and prompts for the number of mesh currents.

Sets flag 08 (to indicate nodal analysis) and prompts for the number of nodes.

Stores the number of mesh currents or nodes.

Creates the complex matrices \( \text{MATA} \) and \( \text{MATB} \), and prompts for a frequency.
Displays a "working" message and sets or clears flag 21 to match flag 55.

Sets the index pointer to the current element (identified in R₀₀). If this generates an error, then the end of the CIRCT matrix has been reached.

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

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

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

79 LBL b
Stores a value into the matrix.
80 +
81 STOEL
82 RTN

83 LBL 82
Stores a resistor value.
84 XEQ F
85 XEQ e
86 GTO B

87 LBL 67
Stores a capacitor value.
88 XEQ F
89 XEQ e
90 XEQ "C⇒Z"
91 GTO B

92 LBL 76
Stores an inductor value.
93 XEQ F
94 XEQ e
95 XEQ "L⇒Z"
96 GTO B

97 LBL 90
Stores a general impedance value.
98 XEQ F
Stores a current or voltage source into \textit{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.

Adds a current source connected to the reference node.

Adds a voltage source (mesh analysis) or a current source \textit{not} connected to the reference node (nodal analysis).
137 ENTER
138 RCLEL
139 R+
140 +
141 STOEL
142 R↓
143 X=γ?  
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"

157 SF 21
158 INDEX "MATX"

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

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

Calculates the results using matrix division.

Prepares to display the results.

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.

Flag 77 is tested to determine when the end of $MATX$ has been reached.
175 LBL e
176 J+
177 RCLEL
178 RTN

179 LBL F
180 ABS
181 FP
182 100
183 ×
184 IP
185 LASTX
186 FP
187 100
188 ×

189 CF 10
190 X=Y?
191 SF 10
192 FC? 08
193 RTN

194 FS? 10
195 GTO J
196 XEQ G
197 XEQ G
198 X<>Y
199 FC? 18
200 RTN
201 +
202 ENTER
203 RTN

204 LBL G
205 X=0?
206 SF 10
207 X<>Y
208 RTN

209 LBL J
210 "Invalid Data"
211 PROMPT
212 END

Increments the column pointer and recalls a matrix element.

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

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

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.)

Sets flag 10 if the reference node is used.

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:

\[
V = \frac{Z}{I} = \frac{MATA}{MATB} = MATX
\]

Variables Used.

<table>
<thead>
<tr>
<th>In Equation</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>n x 3 matrix containing circuit elements.</td>
<td>CIRCT</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>Impedance matrix.</td>
<td>MATA</td>
</tr>
<tr>
<td>V</td>
<td>Voltage matrix.</td>
<td>MATX</td>
</tr>
<tr>
<td>I</td>
<td>Current matrix.</td>
<td>MATB</td>
</tr>
<tr>
<td>Radian frequency, (2\pi f) (radians/sec).</td>
<td>(\omega)</td>
<td></td>
</tr>
</tbody>
</table>

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] SIMO 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 ON to enable printing. If you’re not using a printer, be sure to press PRINT OFF 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 $V_3$. 

![Circuit Diagram]
Begin by selecting FIX 2 display format and running “CIRCT”.

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

Now, add the new elements to the circuit.

Enter the current source from node 0 to node 1.

Enter the resistor from node 1 to node 2.

Enter the impedance from node 0 to node 2.

Enter the resistor from node 0 to node 3.

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: $-10 \angle 5^\circ \text{A}$.)

Enter the impedance from node 2 to node 3.
Press **EXIT** to return to the program's main menu.

At this point you can use **A** and **V** to view the elements in the circuit. Exit the "CIRCT" program and calculate the node voltages. If you want the results printed, press **PRINT A PON** to enable printing.

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

```
3 [R/S]  
```

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

```
V1= x: 68.66 Λ 47.03
```

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

```
V2= x: 45.46 Λ 53.34
```

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

```
V3= x: 58.07 Λ 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:

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

Adding a series \( R, L, \) or \( C \), we have:

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

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

Variable Used. \( Z_{in} \) 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 \( \text{RP} \) and \( \text{RS} \).

Flag 00 is set when \( Z_{\text{in}} = 0 \).

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 \( \text{XEQ LADDR} \) (to run the “LADDR” program).

3. When you see Radian Frequency \( (2\pi f) \) ?, key in the radian frequency and press \( \text{R/S} \). (If the correct frequency is already in the X-register, just press \( \text{R/S} \).)

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 \( Z_{\text{in}} \) is measured), key in the value of the element and then press the corresponding key.

\( \text{RP} \) (parallel resistor, ohms)

\( \text{LP} \) (parallel inductor, henrys)

\( \text{CP} \) (parallel capacitor, farads)

\( \text{RS} \) (series resistor, ohms)

\( \text{LS} \) (series inductor, henrys)

\( \text{CS} \) (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 \( \text{EXIT} \). After quitting, you can restart “LADDR” by pressing \( \text{R/S} \).
**Example.** Find the input impedance of the following circuit at a frequency of 1 MHz:

\[ Z_{in} \]

\[ \begin{array}{c}
1000 \Omega \\
120 \mu \text{H} \\
100 \Omega \\
\end{array} \]

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

\[ \text{Radian Frequency}(2\pi f) = 0.01 \]

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

\[ 6 \times \pi \times 2 \times \text{R/S} \]

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

100 \[ \text{RP} \]

650 \[ 12 \times \text{CS} \]

120 \[ 6 \times \text{CP} \]

1000 \[ \text{RP} \]

2: Impedance of a Ladder Network 69
The input impedance is 306.73 $\Delta - 41.82^\circ$ ohms.

"LADDR" Program Listing.

Program:

```
00 { 154-Byte Prgm }
01 LBL "LADDR"
02 XEQ "FQ?"
03 SF 00
04 CLX
05 ENTER
06 COMPLEX
07 STO "Zin"
08 CF 21
09 FS? 55
10 SF 21
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
19 FS? 00
20 GTO B
21 "RS"
22 KEY 4 XEQ 04
23 "LS"
24 KEY 5 XEQ 05
25 "CS"
26 KEY 6 XEQ 06
```

Comments:

- Inputs the frequency using the "FQ?" utility on page 149.
- Sets flag 00 (until the first element is entered).
- Initializes $Z_{in}$ and sets or clears flag 21 to match flag 55.
- Defines the menu keys for entering elements in parallel.
- If flag 00 is clear, declares the menu keys for entering elements in series.
- Decodes the element location into two mesh or node numbers in the X- and Y-registers.
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.)

Converts a resistance to admittance.

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

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

Adds a parallel element.

Converts a capacitance to impedance.

Converts an inductance to impedance.

Adds a series element.
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 $\alpha$ (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.

![Low Pass Filter Diagram]

$C_5 = C$

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

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

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

$R_4 = GR_1$
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.

\[ C_3 = C_4 = C \]

\[ R_1 = \frac{1}{G \cdot 2\pi f_0 C PF} \]

\[ R_2 = \frac{1}{\left( \frac{2}{PF^2} - G \right) \cdot 2\pi f_0 C PF} \]

\[ R_5 = \frac{2}{PF \cdot 2\pi f_0 C} \]
Variables Used.

<table>
<thead>
<tr>
<th>In Equations</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_0$</td>
<td>Center frequency (Hz).</td>
<td>$F$</td>
</tr>
<tr>
<td>$G$</td>
<td>Midband gain (db).</td>
<td>$G$</td>
</tr>
<tr>
<td>$PF$</td>
<td>Peaking factor.</td>
<td>$PF$</td>
</tr>
<tr>
<td>$C$</td>
<td>Capacitor (farads).</td>
<td>$C$</td>
</tr>
</tbody>
</table>

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 \texttt{XEQ} 	exttt{AF} (to run the “AF” program).
   a. Key in a frequency, $F$, in Hertz; press \texttt{R/S}.
   b. Key in the midband gain, $G$; press \texttt{R/S}.
   c. Key in a peaking factor, $PF$; press \texttt{R/S}.
   d. Key in a capacitance, $C$, in Farads; press \texttt{R/S}.

After entering these values, the program returns to the main menu.

4. Press \texttt{LOWP} (low pass), \texttt{HIGHP} (high pass), or \texttt{BAND} (band pass) to calculate the elements needed to build the particular filter.

5. When all of the elements have been displayed, press \texttt{R/S} to return to the main menu. Then go to step 3 to work another problem or press \texttt{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.

\texttt{DISP ENG 03 XEQ AF}
INPUT

10 R/S

10 R/S

1 R/S

E 6 +/- R/S

HIGHP

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

[R/S]

[R/S]

[R/S]

[R/S]
“AF” Program Listing.

Program:

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

Comments:

- Defines the menu for selecting a filter type.
- Clears any message that may be displayed, displays the menu and stops. Pressing [R/S] redisplay the menu.
- Calculates the elements for a low pass filter.
Calculates the elements for a high pass filter.

35 "R4"
36 R+ 
37 XEQ 08 
38 "C2"
39 1 
40 RCL "G"
41 RCL× "C"
42 4 
43 × 
44 RCL "PF"
45 ×+2 
46 ÷ 
47 XEQ 08 
48 "C5=C"
49 LBL 10 
50 RCL "C"
51 GTO 08 
52 LBL B 
53 "R2"
54 2 
55 XEQ 04 
56 RCL "G"
57 1/X 
58 2 
59 + 
60 × 
61 RCL÷ "PF"
62 1/X 
63 XEQ 08 
64 "R5"
65 2 
66 RCL× "G"
67 1 
68 + 
69 2 
70 XEQ 04 
71 RCL× "PF"
72 ÷ 
73 XEQ 08 
74 "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

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

Calculates the elements for a band pass filter.

Calculates $\pi FC$. 
113 LBL I
114 EXITALL
115 INPUT "F"
116 INPUT "G"
117 INPUT "PF"
118 INPUT "C"
119 RTN

120 LBL 08
121 EXITALL
122 R="="
123 FS? 55
124 ARCL ST X
125 AVIEW

126 LBL 09
127 END

Exits all menus and inputs $F$, $G$, $PF$, and $C$.

Exits all menus. If results are being printed (flag 55 set), puts label and value on the same line.

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:

\[
\begin{align*}
\text{Low Pass} & \quad \omega_n = \frac{\omega}{\omega_0} \\
\text{High Pass} & \quad \omega_n = \frac{\omega_0}{\omega} \\
\text{Band Pass} & \quad \omega_n = \frac{\omega^2 - \omega_0^2}{BW\omega} \\
\text{Band Elimination} & \quad \omega_n = \frac{BW\omega}{\omega_0^2 - \omega^2}
\end{align*}
\]

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

\[\text{whose elements are given by the following set of formulas:}\]
\[ C_i = \frac{1}{\pi f_c R} \sin \left( \frac{(2i - 1)\pi}{2n} \right), \quad i = 1, 3, 5, \ldots, n - 1 \]

\[ L_i = \frac{R}{\pi f_c} \sin \left( \frac{(2i - 1)\pi}{2n} \right), \quad i = 2, 4, 6, \ldots, n \]

where:

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

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 | ![Low pass circuit](image)
| \( C_n \) | \( L_n \)
High pass | \( L_h = \frac{1}{\omega_0 C_n} \) | \( C_h = \frac{1}{\omega_0 L_n} \)
Band pass | \( L_{bp} = \frac{BW}{\omega_0^2 C_n} \) | \( C_{bp} = \frac{C_n}{BW} \)
| \( \frac{L_n}{BW} \) | \( \frac{BW}{\omega_0^2 L_n} \)
Band elimination | \( L_{be} = \frac{L_n BW}{\omega_0^2} \) | \( C_{be} = \frac{1}{L_n BW} \)

84 3: Butterworth Filter Design
Variables and Storage Registers Used.

<table>
<thead>
<tr>
<th>In Equations</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Resistor (ohms).</td>
<td>R</td>
</tr>
<tr>
<td>$F_0$</td>
<td>Center frequency (Hz).</td>
<td>$F_0$</td>
</tr>
<tr>
<td>$\omega_0$</td>
<td>Center frequency (radians/sec.).</td>
<td>$R_{01}$</td>
</tr>
<tr>
<td>$F_1$</td>
<td>Attenuation frequency (Hz).</td>
<td>$F_1$</td>
</tr>
<tr>
<td></td>
<td>Amount of attenuation (dB).</td>
<td>A</td>
</tr>
<tr>
<td>$BW$</td>
<td>Band width (Hz).</td>
<td>$BW$</td>
</tr>
<tr>
<td></td>
<td>Band width (radians/sec.).</td>
<td>$R_{02}$</td>
</tr>
<tr>
<td></td>
<td>Filter type (1-4).</td>
<td>$R_{07}$</td>
</tr>
<tr>
<td>$n$</td>
<td>Filter order.</td>
<td>$R_{09}$</td>
</tr>
<tr>
<td>$i$</td>
<td>Element counter.</td>
<td>$R_{11}$</td>
</tr>
</tbody>
</table>

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` `V` `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 $\Delta dB$ is close to Loss ($\omega_0$)).
Program Instructions.

1. Key the "BF" program (listed on page 88) into your calculator.

2. Press \[ XEQ \] (to run the "BF" program).

3. The program displays a variable menu containing \( R, F_0, F_1, A, \) 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 \textit{Type}? and a menu containing the four types of filters. Press one of these keys to select a filter type:

\begin{itemize}
  \item \underline{LOWP} (low pass)
  \item \underline{HIGHP} (high pass)
  \item \underline{BPASS} (band pass)
  \item \underline{BELIM} (band elimination)
\end{itemize}

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.

\begin{verbatim}
[ disp ] ENG 03 XEQ BF
\end{verbatim}

Store the five inputs.

\begin{itemize}
  \item \underline{R}=50.00E0
  \item \underline{F}_0=800.0E0
  \item \underline{F}_1=900.0E0
\end{itemize}

86  3: Butterworth Filter Design
Continue with the program.

Select a band-pass filter. If you are not using a printer, press \textbf{R/S} after each result is displayed.
Therefore, the filter you’ve calculated looks like this:

```
\[ \begin{array}{c}
50 \Omega & 113 mH & 154 mH & 41 mH \\
& .35 \mu F & .26 \mu F & .96 \mu F \\
& 16.5 \mu F & 61 \mu F & 45 \mu F \\
2.4 mH & 644 \mu H & 880 \mu H & \\
\end{array} \]
```

"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:**

```
00 ( 394-Byte Prgm )
01 LBL "BF"
02 MYAR "R"
03 MYAR "F0"
04 MYAR "F1"
05 MYAR "A"
06 MYAR "BW"
07 RAD
08 LBL A
09 VARMENU "BF"
10 CLA
11 STOP
12 ALENG
13 X≠0?
14 GTO A
```

**Comments:**

- Declares the menu variables and sets Radians mode.
- Displays the variable menu and stops. Pressing \[\text{R/S}\] is the only way to continue the program.
 Converts the center frequency to radians/second.

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

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

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

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

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

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

 Exits all menus, stores the filter type, and sets flag 21 for proper output.
48 RCL "BW"  
49 XEQ 10  
50 STO 02

Calculates the filter order.

51 LBL 00  
52 RCL "F1"  
53 RCL "A"  
54 10  
55 ÷  
56 10+x  
57 2  
58 x  
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

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

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

83 LBL 08  
84 RCL 09  
85 RCL- 10

Evaluates the Butterworth equations.

Converts the band width to radians/second.
Calculate the frequency transformation for the particular filter type.

3: Butterworth Filter Design
126 1/X
127 RCL 01
128 X+2
129 /
130 XEQ 06
131 +/-
132 GTO 00
133 LBL 04
134 SF 01
135 RCL X 02
136 RCL 01
137 X+2
138 /
139 XEQ 06
140 XEQ 00
141 ABS
142 RCL 01
143 X+2
144 X
145 1/X
146 XEQ 06
147 +/-
148 LBL 00
149 FS? 01
150 CLA
151 X≥0?
152 IF "L"
153 X<0?
154 IF "C"
155 RCL 11
156 AIP
157 IF "="
158 R+R
159 ABS
160 ARCL ST X
161 IF "r"
162 FC? 01
163 AVIEW
164 FS?C 01
165 RTN

Displays a pair of filter elements.
Repeats the Butterworth calculations for each element.

Calculates the normalized frequency for the particular filter type.

Multiplies by $-1^i$. 

3: Butterworth Filter Design 93
203 LBL 10
204 2
205 ×
206 PI
207 ×
208 END

Converts the frequency in the X-register (in Hz) to radians/second.
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 $\sqrt{L/C}$), $R$, 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:

\[ Z_{in} \]
From this model the following equations can be derived:

\[ R_0 = \sqrt{\frac{L}{C}} \]

\[ r = \frac{R}{L} = \frac{vR}{R_0} \]

\[ g = \frac{G}{C} = vR_0G \]

where:

- \( L \) = inductance/unit length.
- \( C \) = capacitance/unit length.
- \( G \) = conductance/unit length.
- \( R \) = resistance/unit length.
- \( v = 3 \times 10^8 \nu_r \).
- \( \nu_r \) = relative phase velocity.
- \( f \) = frequency, Hz.
- \( \omega = 2\pi f \) radians/second.

and

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

\[ \beta = \frac{1}{\sqrt{2v}} \left[ \omega^2 - rg + \sqrt{(r^2 + \omega^2)(g^2 + \omega^2)} \right]^{1/2} \]
The approximate solution is:

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

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

\[
\alpha = \alpha_C + \alpha_D
\]

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

where:

\(\alpha_C\) = conductor loss, nepers/unit length = 0.5\((R/R_0)\)

\(\alpha_D\) = dielectric loss, nepers/unit length = 0.5\(\text{GR}\)

\(\beta_0 = \omega/v\)

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 = \(\text{Re}\{Z_0\} + j \text{Im}\{Z_0\}\) (ohms)

\(\gamma = \) propagation constant of line = \(\alpha + j \beta\)

98 4: Transmission Line Calculations
$Z_0$ and $\gamma$ are computed differently depending on which solution is selected.

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

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

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

**Variables Used.**

<table>
<thead>
<tr>
<th>In Equations</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>Frequency.</td>
<td>$f$</td>
</tr>
<tr>
<td>$\nu_r$</td>
<td>Relative phase velocity.</td>
<td>$\nu_r$</td>
</tr>
<tr>
<td>$R_0$</td>
<td>Characteristic impedance.</td>
<td>$R_0$</td>
</tr>
<tr>
<td>$l$</td>
<td>Line length.</td>
<td>1</td>
</tr>
<tr>
<td>$Z_L$</td>
<td>Impedance of termination.</td>
<td>$Z_L$</td>
</tr>
</tbody>
</table>

**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 [V 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 [EXEC] LINE (to run the “LINE” program). The program displays a variable menu containing $f, \nu_r, R_0, l,$ 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 \textbf{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 \textbf{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 \textbf{EXIT} and go to step 4.

**Example.** A transmission line has the following properties:

\[ R = 1.2664 \text{ ohms/cm}. \]

\[ G = 0.00004187 \text{ siemens/cm}. \]

\[ R_0 = 55 \text{ ohms}. \]

\[ \nu_r = 0.85. \]

What is the input impedance of 3.5 cm of this line at 2 GHz if it is terminated in \( Z_L = 75 \angle -30^\circ \text{ ohms} \)?

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

\begin{verbatim}
2 E 9   \textbf{F}
\end{verbatim}

100 4: Transmission Line Calculations
The required input impedance is 48.01 \( \pm 28.48^\circ \) ohms.
"LINE" Program Listing.

Program:

00 { 404-Byte Prgm }
01 LBL "LINE"
02 MYAR "f"
03 MVAR "Vr"
04 MVAR "R0"
05 MVAR "1"
06 MVAR "2L"
07 CF 21
08 FS? 55
09 SF 21
10 LBL A
11 CLA
12 VARMENU "LINE"
13 STOP
14 ALENG
15 X≠0?
16 GTO A
17 RCL "f"
18 1E10
19 ÷
20 STO 04
21 2
22 PI
23 ×
24 ×
25 STO 03
26 RCL "1"
27 2
28 ×
29 3
30 RCL× "Vr"
31 STO 00
32 ÷
33 STO 02

Comments:

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

Displays the variable menu and stops. The program continues only when [R/S] is pressed.

Calculates intermediate results used by both solutions.
Displays a menu for selecting the type of solution. The [EXIT] key is defined to go back to the variable menu.

Calculates the "exact" solution.
Calculates the "approximate" solution.

```plaintext
73 STO 05
74 RCL 06
75 RCL- 03
76 STO "Yr"
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 CLM MENU
87 "C loss?"
88 PROMPT
89 STO 00
90 "D loss?"
91 PROMPT
92 STO 03
93 RCL "1"
94 RCL "Yr"
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 ÷
```
4: Transmission Line Calculations

```
112 RCL ÷ 06
113 STO× 00
114 STO× 03
115 RCL 03
116 RCL- 00
117 ENTER
118 STO 01
119 RCL 03
120 3
121 ×
122 RCL 00
123 +
124 ×
125 2
126 ÷
127 +/-
128 1
129 +
130 →POL
131 STO× "R0"
132 X<>Y
133 STO "Yr"
134 RCL 01
135 X+2
136 2
137 ÷
138 1
139 +
140 RCL 00
141 RCL+ 03
142 →POL
143 STO× 02
144 X<>Y
145 STO 05
```
Completes the calculations for both solutions.
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 x
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:

$$Z_0 = \frac{60}{\sqrt{\varepsilon_r}} \ln \left( \frac{D}{d} \right)$$

Variables Used.

<table>
<thead>
<tr>
<th>In Equations</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>Wire spacing.</td>
<td>$D$</td>
</tr>
<tr>
<td>$d$</td>
<td>Wire diameter.</td>
<td>$\text{dia}$</td>
</tr>
<tr>
<td>$h$</td>
<td>Height of wire (above ground).</td>
<td>$h$</td>
</tr>
<tr>
<td>$\varepsilon_r$</td>
<td>Relative permittivity.</td>
<td>$e$</td>
</tr>
</tbody>
</table>
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 $\varepsilon_r$ of the polyethylene insulation = 2.3.

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

```
[DISP] FIX 02 [XEQ] TLI
```

Store the inputs.

```
.68    D
```

4: Transmission Line Impedance 109
Calculate $Z_0$ for the **coaxial** configuration.

Now calculate $Z_0$ for an open two-wire (air) line with $D = 6$ in. and $\text{dia} = 0.0808$ in. ($\varepsilon_r$ of air = 1).
This time, calculate $Z_0$ of an air line ($\varepsilon_r$ of air = 1) consisting of a single 0.1285 inch diameter wire 6 inches from a ground plane.

\[
\text{dia}=0.13
\]

\[
\text{h}=6.00
\]

\[
\text{e}=1.00
\]

**"TLI" Program Listing.**

**Program:**

```
00 { 225-Byte Prgm }
01 LBL "TLI"
02 MVAR "D"
03 MVAR "dia"
04 MVAR "h"
05 MVAR "e"
06 CF 21
07 FS? 55
08 SF 21
```

**Comments:**

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

Calculates $Z_0$ for wires in parallel near the ground.

Calculates $Z_0$ for a coaxial line configuration.

Completes the calculation and displays the result.
79 LBL 07
80 2
81 RCL× "D"
82 RCL÷ "dia"
83 RTN

84 LBL 08
85 4
86 RCL× "h"
87 RCL÷ "dia"
88 RTN

89 LBL 09
90 2
91 RCL× "h"
92 RCL÷ "D"
93 X+2
94 END

Calculates an intermediate result.
Calculates an intermediate result.
Calculates an intermediate result.
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.

![Transistor Amplifier Diagram]

**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.**

<table>
<thead>
<tr>
<th>In Equations</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h_i )</td>
<td>Matrix h-parameter.</td>
<td>( h_i )</td>
</tr>
<tr>
<td>( h_r )</td>
<td>Matrix h-parameter.</td>
<td>( h_r )</td>
</tr>
<tr>
<td>( h_f )</td>
<td>Matrix h-parameter.</td>
<td>( h_f )</td>
</tr>
<tr>
<td>( h_o )</td>
<td>Matrix h-parameter.</td>
<td>( h_o )</td>
</tr>
<tr>
<td>( Z_S )</td>
<td>Source impedance.</td>
<td>( Z_S )</td>
</tr>
<tr>
<td>( Z_L )</td>
<td>Load impedance.</td>
<td>( Z_L )</td>
</tr>
</tbody>
</table>

**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 \( \text{XEQ TAP} \) (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 \( \text{R/S} \) to display the result menu.
5. Press the appropriate menu key for each result you want to calculate:
   - \( \text{R1} \) to display the current gain, \( A_i \).
   - \( \text{AV} \) to display the voltage gain, \( A_v \).


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

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?

$$h = \begin{bmatrix} 1100 & 250 \times 10^{-6} \\ 50 & 25 \times 10^{-6} \end{bmatrix}$$

Select the FIX 4 display format and run the “TAP” program.

![Display Settings]

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

- 1100 **HI**
- $250 \times 10^{-6}$ **HR**
- 50 **HF**
- $25 \times 10^{-6}$ **HO**
- 1000 **ZS**
For each of the outputs you want to calculate, press the corresponding menu key.

\[ R_v = -40.000 \]

\[ A_v = -400.000 \]

\[ A_{vs} = -200.000 \]

\[ Z_{in} = 1,000.000 \]

\[ Z_{out} = 52,500.000 \]

Press [EXIT] to return to the input menu.

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

Program:

00 { 227-Byte Prgm }
01 LBL "TAP"
02 MVAR "hi"
03 MVAR "hr"
04 MVAR "hf"
05 MVAR "ho"
06 MVAR "Zs"
07 MVAR "Z1"
08 CF 21
09 FS? 55
10 SF 21

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

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

30 LBL 00
31 MENU
32 STOP
33 GTO 00

Comments:

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

Displays the variable menu and stops.

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

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 \div
41 "Av"
42 GTO 06

43 LBL 03 Calculates $A_{av}$.
44 XEQ 08
45 RCL+ "Zs"
46 \div
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 \times
59 LASTX
60 RCL\times "Zs"
61 +
62 RCL "hf"
63 RCL\times "hr"
64 -
65 \div
66 "Zout"
67 LBL 06                    Displays a result.
68 \="
69 ARCL ST \times
70 AVIEW
71 RTN

72 LBL 07                    Calculates \( A_i \).
73 RCL "hf"
74 +/-
75 1
76 RCL "ho"
77 RCL× "Z1"
78 +
79 ÷
80 RTN

81 LBL 08                    Calculates \( Z_{in} \).
82 XEQ 07
83 RCL× "Z1"
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:

$$\theta_{JA} = \frac{(T_{\text{max}} - 25^\circ C)}{P_D}$$

and the minimum load resistance and emitter resistance are estimated:

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

$$R_{E1} = 0.1 R_{L1} = R_{En}$$
Next, the quiescent, maximum, and minimum collector currents are calculated:

\[ I_{CQ} = \frac{V_{CC}}{2(R_L + R_E)} \]

\[ I_{C_{\text{max}}} = I_{CQ} \left(1 + \Delta I_{CQ}\right) \]

\[ I_{C_{\text{min}}} = I_{CQ} \left(1 - \Delta I_{CQ}\right) \]

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

\[ T_{\text{max}} = \theta_{JA} I_{CQ} \left(\frac{V_{CC}}{2}\right) + T_{A_{\text{max}}} \]

\[ V_{\text{BEX}} = V_{\text{BE1}_{\text{min}}} + \Delta V_{BE} \log \left(\frac{I_{C_{\text{max}}}}{I_1}\right) - 0.0022 (T_{\text{max}} - 25^\circ\text{C}) \]

\[ T_{\text{min}} = \theta_{JA} I_{CQ} \left(\frac{V_{CC}}{2}\right) \left(1 - (\Delta I_{CQ})^2\right) + T_{A_{\text{min}}} \]

\[ V_{\text{BEN}} = V_{\text{BE1}_{\text{max}}} + \Delta V_{BE} \log \left(\frac{I_{C_{\text{min}}}}{I_1}\right) - 0.0022 (T_{\text{min}} - 25^\circ\text{C}) \]

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

\[ R_{E(n+1)} = \frac{-2(V_{\text{BEX}} - V_{\text{BEN}})}{I_{C_{\text{max}}} - I_{C_{\text{min}}}} \]

From this point, if \( V_{\text{BEX}} > V_{\text{BEN}} \), 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 \( T_{\text{max}} > T_{J_{\text{max}}} \) occurs, \( R_L \) is increased by 10%.

When the iterative procedure is complete, \( T_{\text{max}}, I_{C_{\text{max}}}, T_{\text{min}}, \) and \( I_{C_{\text{min}}} \) are displayed.
Values for

\[ h_{FE_{\text{max}}} = \text{maximum worst-case current gain at } T_{\text{max}} \text{ or } T_{\text{min}} \text{ and } I_{C_{\text{max}}} \text{ or } I_{C_{\text{min}}} \]

and

\[ h_{FE_{\text{min}}} = \text{minimum worst-case current gain at } T_{\text{max}} \text{ or } T_{\text{min}} \text{ and } I_{C_{\text{max}}} \text{ or } I_{C_{\text{min}}} \]

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:

\[
R_B = \frac{h_{FE_{\text{max}}} h_{FE_{\text{min}}}}{h_{FE_{\text{max}}} I_{C_{\text{min}}} - h_{FE_{\text{min}}} I_{C_{\text{max}}}} \left[ R_E (n+1) (I_{C_{\text{max}}} - I_{C_{\text{min}}}) + V_{BEX} - V_{BEN} \right]
\]

\[
V_{BB} = V_{BEN} + I_{C_{\text{min}}} \left( R_B / h_{FE_{\text{min}}} + R_E (n+1) \right)
\]

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:

\[
A_p = \frac{R_B R_L h_{FE_{\text{min}}}}{R_E (R_B + h_{FE_{\text{min}}} R_E)}
\]

\[
P_S = (1 - \Delta I_{CQ})^2 \left[ \frac{V_{CC}^2 R_L}{8 (R_L + R_E)^2} \right]
\]
Variables Used.

<table>
<thead>
<tr>
<th>In Equations</th>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CC}$</td>
<td>Source voltage (volts).</td>
<td>$V_{CC}$</td>
</tr>
<tr>
<td>$\Delta I_{CQ}$</td>
<td>Maximum desired percentage variation of quiescent current.</td>
<td>$dI_{CQ}$</td>
</tr>
<tr>
<td>$T_{A_{max}}$</td>
<td>Maximum ambient temperature (use the maximum case temperature for a transistor mounted on a heat sink).</td>
<td>$T_{A_{\text{max}}}$</td>
</tr>
<tr>
<td>$T_{A_{min}}$</td>
<td>Minimum ambient temperature.</td>
<td>$T_{A_{\text{min}}}$</td>
</tr>
<tr>
<td>$T_{J_{max}}$</td>
<td>Maximum junction temperature.</td>
<td>$T_{J_{\text{max}}}$</td>
</tr>
<tr>
<td>$P_{D}$</td>
<td>Maximum rated power dissipation at 25°C.</td>
<td>$P_{D}$</td>
</tr>
<tr>
<td>$I_{1}$</td>
<td>Collector current, usually selected for convenience so that $I_{1}$ and 10 $I_{1}$ at 25°C bracket the expected operating point.</td>
<td>$I_{1}$</td>
</tr>
<tr>
<td>$\Delta V_{BE}$</td>
<td>Typical base-emitter voltage change over the range of $I_{1}$ to 10 $I_{1}$ at 25°C.</td>
<td>$dV_{BE}$</td>
</tr>
<tr>
<td>$V_{BE1_{\text{min}}}$</td>
<td>Minimum base-emitter voltage at $I_{1}$ at 25°C.</td>
<td>$V_{BE1_{\text{min}}}$</td>
</tr>
<tr>
<td>$V_{BE1_{\text{max}}}$</td>
<td>Maximum base-emitter voltage at $I_{1}$ at 25°C.</td>
<td>$V_{BE1_{\text{max}}}$</td>
</tr>
<tr>
<td>$h_{FE_{\text{max}}}$</td>
<td>Maximum worst-case current gain at $T_{\text{max}}$ or $T_{\text{min}}$ and $I_{C_{\text{max}}}$ or $I_{C_{\text{min}}}$</td>
<td>$R_{01}$</td>
</tr>
<tr>
<td>$h_{FE_{\text{min}}}$</td>
<td>Minimum worst-case current gain at $T_{\text{max}}$ or $T_{\text{min}}$ and $I_{C_{\text{max}}}$ or $I_{C_{\text{min}}}$</td>
<td>$R_{02}$</td>
</tr>
</tbody>
</table>
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 R00 through R09 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 $T_{\text{max}}$ and $I_{\text{max}}$. The program then calculates and displays $T_{\text{min}}$ and $I_{\text{min}}$.
5. When you see $H_{\text{max}}?$, key in a value for the maximum worst-case current gain at $T_{\text{max}}$ or $T_{\text{min}}$ and $I_{\text{Gmax}}$ or $I_{\text{Gmin}}$. Press [R/S].
6. When you see $H_{\text{min}}?$, key in a value for the minimum worst-case current gain at $T_{\text{max}}$ or $T_{\text{min}}$ and $I_{\text{Gmax}}$ or $I_{\text{Gmin}}$. 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,

$T_{\text{Jmax}} = 150°C$

$PD = 0.36 \text{ W}$

$\Delta V_{BE} = 0.10 \text{ v from 3 to 30 mA}$

$V_{BE1\text{min}} = 0.52 \text{ v at 3 mA at 25°C}$

$V_{BE1\text{max}} = 0.72 \text{ v at 3 mA at 25°C}$
\[ I_1 = 0.001 \text{ A} \]
\[ h_{FEmax} = 600 \]
\[ h_{FEmin} = 100 \]

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

<table>
<thead>
<tr>
<th>Action</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 [R/S]</td>
<td>Y: 30.00E0, dICQ?0.00E0</td>
</tr>
<tr>
<td>.2 [R/S]</td>
<td>Y: 200.00E-3, TAmx?0.00E0</td>
</tr>
<tr>
<td>70 [R/S]</td>
<td>Y: 70.00E0, TAmx?0.00E0</td>
</tr>
<tr>
<td>0 [R/S]</td>
<td>Y: 0.00E0, TJmax?0.00E0</td>
</tr>
<tr>
<td>150 [R/S]</td>
<td>Y: 150.00E0, PD?0.00E0</td>
</tr>
<tr>
<td>.36 [R/S]</td>
<td>Y: 360.00E-3, I?0.00E0</td>
</tr>
<tr>
<td>.001 [R/S]</td>
<td>Y: 1.00E-3, dYBE?0.00E0</td>
</tr>
</tbody>
</table>

128 5: Transistor Amplifier Bias Optimization
Refer again to the transistor's data sheet and input $h_{FEmax}$.

Now, input $h_{FEmin}$.

5: Transistor Amplifier Bias Optimization 129
Program Listing.

Program:

00 ( 553-Byte Prgm )
01 LBL "BIAS"
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
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 ÷

Comments:

Inputs values.

Initializes values for iterative process.
31 STO 03
32 0.1
33 ×
34 STO 04

35 LBL 00
36 RCL "VCC"
37 2
38 ÷
39 ENTER
40 ENTER
41 RCL 03
42 RCL+ 04
43 ÷
44 STO 00
45 RCL× 09
46 ×
47 RCL+ 07
48 RCL "TJmax"
49 X<>Y
50 X<>Y?
51 GTO 05
52 XEQ 03
53 +/-
54 RCL "dICQ"
55 1
56 +
57 XEQ 04
58 RCL+ "VBE1min"
59 STO 05
60 1
61 RCL "dICQ"
62 X+2
63 -
64 2
65 ÷
66 RCL× 00
67 RCL× 09
68 RCL× "VCC"
69 RCL+ 08
70 XEQ 03

Begins iterative loop.
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 $h_{FE_{max}}$ and $h_{FE_{min}}$.

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

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

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

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

Calculates $V_{BB}$.

Calculates $R_1$.

Calculates $R_2$. Recalls $R_L$ and $R_E$, and then displays each of the four values in the stack.
IIRI "R1"

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 F="="
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).

Calculates temperature.

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

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

Displays proper label and temperature.

5: Transistor Amplifier Bias Optimization
184 LBL 04 Calculates current.
185 RCLx 00
186 FS? 01
187 XEQ I
188 RCL+ "I1"
189 LOG
190 RCLx "dVBE"
191 +
192 RTN

193 LBL I Displays proper label and current.
194 FS? 02
195 "I_{\text{max}}=
196 FC?C 02
197 "I_{\text{min}}=
198 ARCL ST X
199 ARVIEW
200 RTN
201 STO 07
202 CLX
203 STO 04

204 LBL 05 Increases \( R_L \) by 10%.
205 1.1
206 STOrx 03
207 GTO 00
208 END
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 Pgm )
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

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

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.**

<table>
<thead>
<tr>
<th>Program:</th>
<th>Comments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 ( 31-Byte Prgm )</td>
<td>Calculates ( A ) AND ( B ) AND ( C ).</td>
</tr>
<tr>
<td>01 LBL &quot;EXMPL&quot;</td>
<td></td>
</tr>
<tr>
<td>02 RCL 02</td>
<td></td>
</tr>
<tr>
<td>03 RCL 01</td>
<td></td>
</tr>
<tr>
<td>04 AND</td>
<td></td>
</tr>
<tr>
<td>05 RCL 00</td>
<td></td>
</tr>
<tr>
<td>06 AND</td>
<td></td>
</tr>
<tr>
<td>07 RCL 00</td>
<td>Calculates ( \overline{C} ).</td>
</tr>
<tr>
<td>08 SIGN</td>
<td></td>
</tr>
<tr>
<td>09 LASTX</td>
<td></td>
</tr>
<tr>
<td>10 -</td>
<td></td>
</tr>
<tr>
<td>11 RCL 01</td>
<td>Calculates ( \overline{B} ).</td>
</tr>
<tr>
<td>12 SIGN</td>
<td></td>
</tr>
<tr>
<td>13 LASTX</td>
<td></td>
</tr>
<tr>
<td>14 -</td>
<td></td>
</tr>
<tr>
<td>15 AND</td>
<td>Calculates ( \overline{B} ) AND ( \overline{C} ).</td>
</tr>
<tr>
<td>16 RCL 02</td>
<td>Calculates ( A ) AND ( \overline{B} ) AND ( \overline{C} ).</td>
</tr>
<tr>
<td>17 AND</td>
<td></td>
</tr>
<tr>
<td>18 OR</td>
<td>Calculates ( (A ) AND ( B ) AND ( C )) \ OR ( (A ) AND ( \overline{B} ) AND ( \overline{C} ) )</td>
</tr>
<tr>
<td>19 END</td>
<td></td>
</tr>
</tbody>
</table>

**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.**

<table>
<thead>
<tr>
<th>Description</th>
<th>In Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function name.</td>
<td>FCN</td>
</tr>
<tr>
<td>Loop counter.</td>
<td>count</td>
</tr>
<tr>
<td>Most significant bit.</td>
<td>msb</td>
</tr>
</tbody>
</table>

**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 ($\text{MODES} \text{ V SIZE} \ 6 \ \text{ENTER}$) 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 $\text{XEQ PTTBL}$ (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 $\text{R/S}$.
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 ≤ msb ≤ 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
```

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

```
EXMPL
```

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

Since the logical expression uses three inputs (A, B, and C), the most significant bit is stored in R₆².

```
2 [R/S]
Y: 1
x: 0
```

**Printer Output.**

```
2 1 0
EXMPL
```

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

140 6: Printing a Truth Table
“PTTBL” Program Listing.

Program:

00 { 168-Byte Prgm }
01 LBL "PTTBL"
02 XEQ "FCN?"
03 PRON
04 CF 12
05 CLST
06 ALL
07 INPUT "msb"
08 1
09 +
10 2
11 X<>Y
12 Y^X
13 1
14 -
15 1E3
16 ÷
17 STO "count"
18 CLA
19 PRA
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 +""Y"
30 PRA
31 SF 12

Comments:

Prompts for a function name using the “FCN?” utility. Printing is enabled.

Prompts for the number of the most significant bit and sets up the loop counter.

Prints a single blank line.

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"  

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

44 XEQ IND "FCN"  

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

51 ARCL IND ST X  
52 +/-  
53 ARCL ST Y  

54 PRA  
55 ISG "count"  
56 GTO 01  

57 CF 12  
58 RTN  

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

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

Stores the input bits into the appropriate storage registers.

Evaluates the expression for the given inputs.

Accumulates the input bits into the Alpha register.

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

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

 Resets the double-wide flag and ends.

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

---

6: Printing a Truth Table
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:

Given these inputs, the expression produces this output

\[
\begin{array}{cccccc}
0 & 0 & 0 & 1 & 1 & 1 \\
\end{array}
\]

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 \[ SIZE \] \[ ENTER \]) 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
```

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 [R/S]; if you see EXMPL, then simply press [R/S] to continue.)

```
EXMPL
```

```
R/S
```

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

```
0 (the LOG key)
```

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)
```

```
0 (the XEQ key)
```

---

144 6: An Interactive Truth Table
"ITTBL" Program Listing.

Program:

```
00 ( 125-Byte Prgm )
01 LBL "ITTBL"
02 XEQ "FCN?"
03 ALENG
04 X=0?
05 GTO 09
06 BINM
07 ALL
08 CLRG
09 CLMENU
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
```

Comments:

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

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

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.
HEQ IWD "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] redisplaysthe 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.
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, \( 2\pi \) 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:

```
00 { 30-Byte Prgm }
01 LBL "C→Z"
02 XEQ 00
03 1/X
04 RTN

05 LBL "L→Z"
06 LBL 00
07 RECT
08 RCLX "w"
09 0
10 X<>Y
11 COMPLEX
12 END
```

Comments:
- Calculates the impedance for the given capacitance (which is the reciprocal of impedance for inductance).
- 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 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\((2\pi f)\)?, key in the frequency and press \([R/S]\).

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.)
- 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 Frequency"
03 l"y(2\pi f)?"
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?

ΩM │ KΩM │ MΩM

5
```

You can enter a 2,000-ohm resistor by pressing 2000 \( \Omega \), 2 \( \text{KΩ} \), or .002 \( \text{MΩ} \).

The program returns two numbers to the stack:

\[ y: \text{ee}, yx \]
\[ x: \text{element value} \]

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.
<table>
<thead>
<tr>
<th>Element Type</th>
<th>Units</th>
<th>Type Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistor</td>
<td>Ohms</td>
<td>82</td>
</tr>
<tr>
<td>Capacitor</td>
<td>Farads</td>
<td>67</td>
</tr>
<tr>
<td>Inductor</td>
<td>Henrys</td>
<td>76</td>
</tr>
<tr>
<td>General impedance</td>
<td>Ohms</td>
<td>90*</td>
</tr>
<tr>
<td>Voltage source</td>
<td>Volts</td>
<td>86*</td>
</tr>
<tr>
<td>Current source</td>
<td>Amperes</td>
<td>73*</td>
</tr>
</tbody>
</table>

*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:

```
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 "Y"
14 KEY 5 GTO 14
15 "I"
16 KEY 6 GTO 15
17 KEY 9 GTO 99
```

Comments:

- 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).
Displays the input message and the menu. Pressing \textbf{R/S} redisplays the menu.

Enters the code for a resistor.

Enters the code for a general impedance.

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

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

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

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

118 CLMENU
119 KEY 9 GTO A
120 RTN

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

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

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

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

135 LBL 99
136 END

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

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.

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:

<table>
<thead>
<tr>
<th>Line</th>
<th>Instructions</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>( 48-Byte Pgm )</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>LBL &quot;FCN?&quot;</td>
<td>Displays the prompt.</td>
</tr>
<tr>
<td>02</td>
<td>&quot;Function Name?&quot;</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>CF 21</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>AVIEW</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>SF 25</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>RCL &quot;FCN&quot;</td>
<td>If FCN exists and contains an Alpha string, recalls that string to the Alpha register.</td>
</tr>
<tr>
<td>07</td>
<td>CF 25</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>CLA</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>STR?</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ARCL ST X</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>AON</td>
<td>Turns on the ALPHA menu and stops.</td>
</tr>
<tr>
<td>12</td>
<td>STOP</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>AOFF</td>
<td>Turns off the ALPHA menu and stores the first six characters in the Alpha menu into FCN.</td>
</tr>
<tr>
<td>14</td>
<td>ASTO &quot;FCN&quot;</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

156 7: Function Name Utility
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) 09 KEY 9 GTO 00
01 LBL "Y?N" 10 MENU
02 CF 21 11 STOP
03 AVIEW 12 LBL 00
04 CLMENU 13 0
05 "YES" 14 RTN
06 KEY 1 GTO 01 15 LBL 01
07 "NO" 16 1
08 KEY 6 GTO 00 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:

\[ \frac{xy}{x + y} \]

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 RCLx 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