DCNAP

THE DC NETWORK

Let your computer solve for DC-network node voltages and avoid brain strain

By James E. Tarchinski

IN THE OCTOBER 1987 ISSUE OF **Hands-on Electronics**, we published *ACNAP*, a program written in BASIC that enables a personal computer to calculate the node voltages of alternating-current (AC) networks. Due to the number of requests we've received for a direct-current version of that program, we would now like to introduce *DCNAP*: The *D*irect-*C*urrent *N*etwork *A*nalysis *P*rogram.

A Little Background

DCNAP (also written in BASIC) solves for the node voltages in DC circuits using a technique called *nodal analysis*. The circuits entered can have anywhere from 1 to 25 nodes and contain resistors, conductors, independent current sources, voltage-controlled current sources, and independent voltage sources.

The DCNAP program listing is given in Table 1, and a flow chart of the program is shown in Fig. 1. Note that the numbers in brackets on the flow chart correspond to the line numbers in the program listing shown in Table 1.

Those of you with sharp eyes and good memories may notice the resemblance between DCNAP and its AC cousin, ACNAP. The reason for that is, as you may already suspect, that DCNAP was created by optimizing the original ACNAP program to handle DC circuits. As mentioned in the ACNAP article, that program is capable of analyzing DC circuits by setting the frequency of operation to zero hertz and being certain not to enter any inductive elements in the circuit description when analyzing DC networks. However, ACNAP would be somewhat inconvenient for a large number of DC circuits; hence the need for DCNAP.

While creating DCNAP, care was taken not to modify any line numbers from the original (ACNAP) program listing. To use DCNAP, enter the pro-

TABLE 1

```
1000 'DC NETWORK ANALYSIS PROGRAM  (DCNAP)
1010 '
1020 CLEAR : DIM A(25,26)
1030 SCREEN 0,0,0,0 : COLOR 10,0,0 : WIDTH 80 : CLS
1060 '
1070 PRINT "*******************************************************************"
1080 PRINT "*                                                                 *"
1090 PRINT "*                DCNAP - D.C. Network Analysis Program            *"
1100 PRINT "*                                                                 *"
1110 PRINT "*                   (c) 1988 by James E. Tarchinski               *"
1120 PRINT "*                                                                 *"
1130 PRINT "*******************************************************************"
1140 COLOR 11
1150 PRINT
1160 PRINT "  This program uses Nodal Analysis to determine the node voltages"
1170 PRINT "of linear D.C. networks.  These networks may range in size from"
1180 PRINT "1 to 25 nodes, not counting the ground, or datum, node."
1190 PRINT
1200 PRINT "  Models for five types of devices are incorporated in this"
1210 PRINT "program:"
1220 PRINT "      1. Resistors"
1230 PRINT "      2. Conductors"
1250 PRINT "      3. Independent current sources (ICS)"
1260 PRINT "      4. Voltage-controlled current sources (VCCS)"
1270 PRINT "      5. Independent voltage sources (IVS)"
1280 PRINT "
1290 LOCATE 23,1 : COLOR 7 : PRINT "Press any key ...";
1300 '
1310 INS=INKEYS:IF INS<>"" THEN GOTO 1310
1320 INS=INKEYS:IF INS= "" THEN GOTO 1320
1330 CLS : COLOR 10
1340 PRINT "For all circuits, these conventions must be followed:"
1350 COLOR 11
1360 PRINT
1370 PRINT "1. All nodes of the circuit must be labeled with consecutive"
1380 PRINT "   integers   (1, 2, 3, etc.)."
1390 PRINT
1400 PRINT "2. The ground node must be Node 0."
1410 PRINT
1420 PRINT "3. The ground node is not counted when determining the number"
1430 PRINT "   of nodes in a circuit."
1440 PRINT
1530 PRINT "4. No two voltage sources (IVS) may be connected to the same node,"
1540 PRINT "   not counting the ground node (Node 0)."
1550 PRINT
1560 LOCATE 23,1 : COLOR 7 : PRINT "Press any key ..."; : COLOR 10
1570 '
1580 INS=INKEYS:IF INS<>"" THEN GOTO 1580
1590 INS=INKEYS:IF INS= "" THEN GOTO 1590
1600 CLS
1610 PRINT "Enter the number of nodes in the circuit,"
1620 INPUT "not counting the ground node (1-25): ",N
1630 IF N > 25 OR N < 1 THEN  PRINT "PLEASE ENTER A VALID NUMBER!": GOTO 1610
1640 N1 = N + 1
1650 '
1660 PRINT
1750 '
1760 '
1770 '********************** RESISTOR SECTION ********************************
1780 '
1790 CLS : INPUT "Enter number of resistors: ",NC
1800 IF NC < 1 THEN GOTO 1950
1810 POS = "RESISTOR"
1820 P1S = "INITIAL NODE:  "
1830 P2S = "FINAL NODE:    "
1840 P3S = "VALUE (Ohms):  "
1850 FOR I = 1 TO NC
1860   GOSUB 3740
1870   VL = 1 / VL
1880      A(IN,IN) = A(IN,IN) + VL : A(IN,EN) = A(IN,EN) - VL
1890      A(EN,EN) = A(EN,EN) + VL : A(EN,IN) = A(EN,IN) - VL
1900 NEXT I
1910 '
1920 '
1930 '********************** CONDUCTOR SECTION *******************************
1940 '
1950 CLS : INPUT "Enter number of conductors: ",NC
1960 IF NC < 1 THEN GOTO 2400
1970 POS = "CONDUCTOR"
1980 P1S = "INITIAL NODE:  "
1990 P2S = "FINAL NODE:    "
2000 P3S = "VALUE (Mhos):  "
2010 FOR I = 1 TO NC
2020   GOSUB 3740
2030      A(IN,IN) = A(IN,IN) + VL : A(IN,EN) = A(IN,EN) - VL
2040      A(EN,EN) = A(EN,EN) + VL : A(EN,IN) = A(EN,IN) - VL
2050 NEXT I
2060 '
2070 '
2400 '************* INDEPENDENT CURRENT SOURCE SECTION **********************
2410 '
2420 CLS : INPUT "Enter number of ICSs: ",NC
2430 IF NC < 1 THEN GOTO 2560
```

ANALYSIS PROGRAM

TABLE 1 *(continued)*

```
2440 P0$ = "ICS"
2450 P1$ = "INITIAL NODE  (The tail):   "
2460 P2$ = "FINAL NODE  (The point):    "
2470 P3$ = "MAGNITUDE (Amps):           "
2490 FOR I = 1 TO NC
2500    GOSUB 3740
2510      A(IN,N1) = A(IN,N1) - VL
2520      A(EN,N1) = A(EN,N1) + VL
2530 NEXT I
2540 '
2550 '
2560 '************ VOLTAGE CONTROLLED CURRENT SOURCE SECTION *************
2570 '
2580 CLS : INPUT "Enter number of VCCSs: ",NC
2590 IF NC < 1 THEN GOTO 2740
2600 P0$ = "VCCS"
2610 P1$ = "INITIAL NODE  (The tail):  "
2620 P2$ = "FINAL NODE  (The point):   "
2630 P3$ = "MAGNITUDE (Amps/volt):     "
2640 FOR I = 1 TO NC
2650    GOSUB 3740
2660    PRINT
2670    INPUT "CONTROLLING NODE (Positive): ",CP: IF CP > N OR CP < 0 THEN BEEP:
 GOTO 2660
2680    INPUT "CONTROLLING NODE (Negative): ",CN: IF CN > N OR CN < 0 OR CN = CP
 THEN BEEP: GOTO 2680
2690       A(IN,CP) = A(IN,CP) + VL : A(IN,CN) = A(IN,CN) - VL
2700       A(EN,CP) = A(EN,CP) - VL : A(EN,CN) = A(EN,CN) + VL
2710 NEXT I
2720 '
2730 '
2740 '************ INDEPENDENT VOLTAGE SOURCE SECTION *********************
2750 '
2760 CLS : INPUT "Enter number of IVSs: ",NC
2770 IF NC < 1 THEN GOTO 2960
2780 P0$ = "IVS"
2790 P1$ = "NEGATIVE NODE:          "
2800 P2$ = "POSITIVE NODE:          "
2810 P3$ = "MAGNITUDE (Volts):      "
2830 FOR I = 1 TO NC
2840    GOSUB 3740
2850    IF IN > EN THEN VL=-VL : T0=IN : IN=EN : EN=T0
2860      FOR J = 1 TO N1
2870        A(IN,J) = A(IN,J) + A(EN,J)
2890        A(EN,J) = 0
2900      NEXT J
2910      A(EN,EN) = 1 : A(EN,IN) = -1 : A(EN,N1) = VL
2930 NEXT I
2940  '
2950  '
2960  '************** GAUSSIAN ELIMINATION SECTION ************************
2970 CLS
2980 PRINT "CALCULATING, PLEASE WAIT..."
2990 FOR I = 1 TO N
3000    HF = I
3010    B =  ABS(A(I,I))
3020    FOR K = I+1 TO N
3030     T = ABS(A(K,I)): IF T > B THEN B = T : HF = K
3040    NEXT K
3050    IF I = HF THEN GOTO 3110
3060      FOR K = 1 TO N1
3070        T = A(I,K)
3080        A(I,K) = A(HF,K)
3090        A(HF,K) = T
3100      NEXT K
3110    IF B < 9.999999E-21 THEN CLS : PRINT "ERROR! - The circuit entered is no
t valid.": GOTO 3590
3120 '
3130    T = A(I,I)
3140    FOR K = I TO N1
3150        A(I,K) = A(I,K) / T
3180    NEXT K
3190 '
3200    FOR K = 1 TO N
3210     IF K = I THEN 3280
3220     T =  - A(K,I)
3230     A(K,I) = 0
3240      FOR L = I+1 TO N1
3250        A(K,L) = A(K,L) + T*A(I,L)
3270      NEXT L
3280    NEXT K
3290 NEXT I
3300 '
3310 CLS
3320 PRINT "THE NODE VOLTAGES ARE:": PRINT
3330 FOR I = 1 TO N
3340     IF I=20 THEN IN$="":LOCATE 23,1:PRINT "Press any key ...";:WHILE IN$=""
:IN$=INKEY$:WEND:CLS
3410     PRINT "V(";I;") = ";A(I,N1);" volts"
3570 NEXT I
3580 '
3590 LOCATE 23,1
3600 PRINT "Press any key to continue...";
```

gram given in Table 1 into your computer, taking special care to enter the program line numbers exactly as shown. Because of the program's origin, DCNAP does not always have line numbers that are perfect increments of ten, as is customary in magazine listings. Once the program is entered, you must run BASIC in order to use DCNAP.

Program Description

DCNAP is initialized by the first three lines of code (1000–1030). The variables are cleared, the "A" array is given dimension, and the screen is cleared, and set up in the 80-column text mode.

The next section of the program, lines 1070–1590, displays two screens of introductory information. The first screen contains general comments about the program (as shown in Fig. 2), while the second screen notes the four major rules that must be followed when using DCNAP. Basically, those rules state that each node in the circuit to be analyzed must be given a *unique* integer node number, and that the ground (or datum) node *must* be called Node 0.

No integer value may be skipped and the ground node is not counted when determining the number of nodes in a network. Also, no two independent voltage sources may be connected to the same node, with the single exception of the ground node (Node 0). After the instruction screens are displayed, the computer prompts you to enter the number of nodes in the circuit. If the number of nodes is less than one or greater than 25, an error message is displayed and the user is given another opportunity to input a valid number.

When the number of nodes is correctly entered, the program then prompts you to enter the circuit's description, which is done one element at a time. Resistors are entered first, followed by conductors, Independent Current Sources (ICS), Voltage-Controlled Current Sources (VCCS), and finally Independent Voltage Sources (IVS). Because each type of element is entered using the same format, a common subroutine (which begins on line 3740) is used to enter component data.

For resistors and conductors, it doesn't matter which node you consider

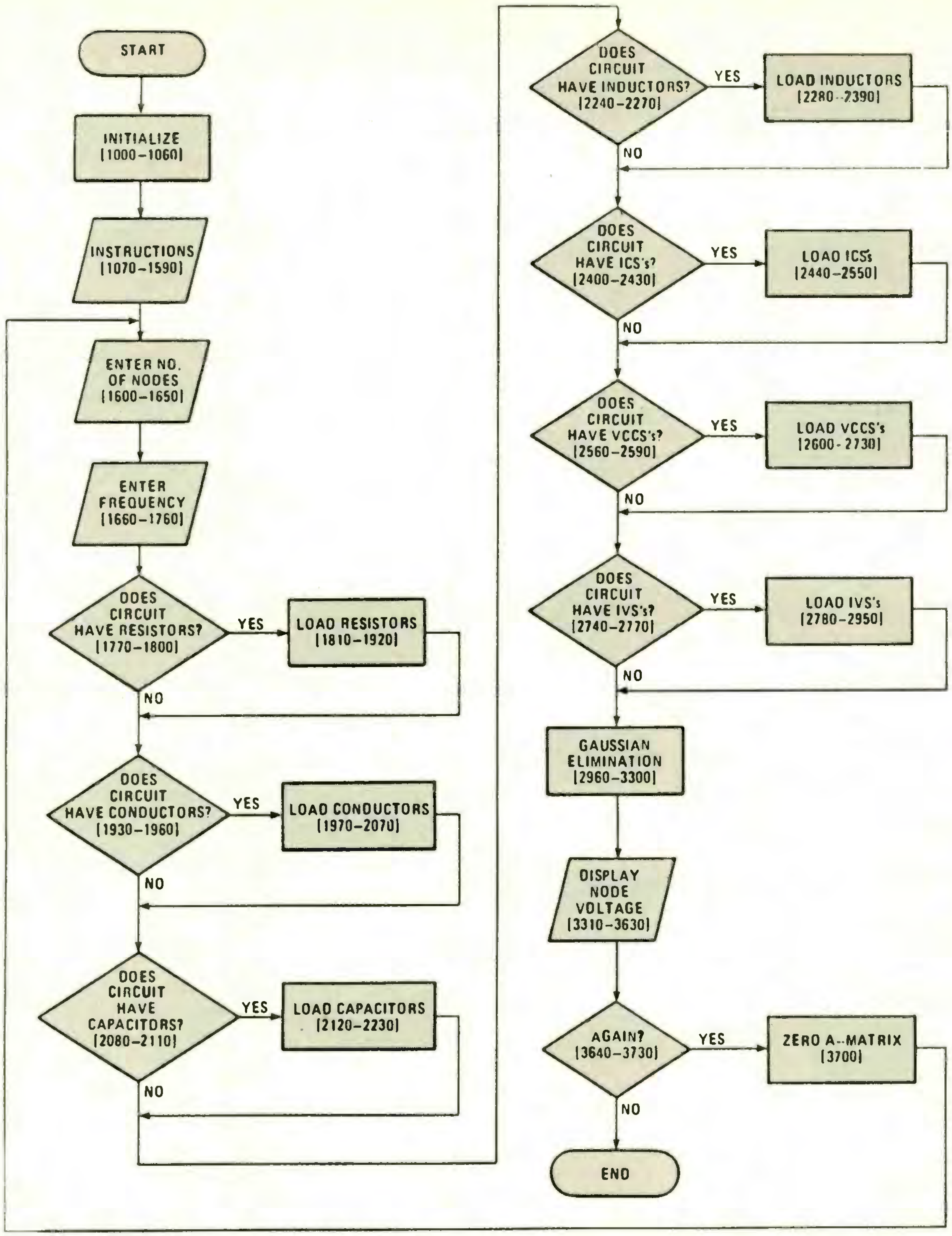

Fig. 1—Because DCNAP was derived from the original ACNAP program, the line numbers are not always in perfect increments of ten. Take care when entering the program not to miss a line number transition.

to be the "Initial Node," and which you consider to be the "Final Node," as long as there is one of each. For the other three element types however, the node names must not be confused or DCNAP will yield incorrect results. For those elements, the input prompts specify what names are associated with which terminal of the circuit. For example, the "Initial Node" of an independent current source (ICS) is the tail of the arrow, while the "Final Node" is the arrow's point.

When all the component values are entered, lines 2960–3290 of the program solve for the circuit's node voltages, using a Gaussian elimination algorithm. If the numeric values in the equations become too small for the computer to deal with, or if the circuit description entered is not valid, the error message of line 3110 is displayed.

The circuit's node voltages, with respect to the ground node (Node-0), are printed to the screen by lines 3310–3620. To make certain that the

TABLE 1 *(continued)*

```
3610 IN$ = INKEY$ : IF IN$ <> "" THEN GOTO 3610
3620 IN$ = INKEY$ : IF IN$ =  "" THEN GOTO 3620
3630 '
3640 CLS : PRINT "Would you like to analyize another circuit?"
3650 PRINT "  1. Yes,  or"
3660 PRINT "  2. No"
3670 INPUT "Your choice: ",A
3680 IF A =  2 THEN END                    'exit program
3690 IF A <> 1 THEN BEEP: GOTO 3640        'inproper entry
3700 FOR I=1 TO N : FOR J=1 TO N1 : A(I,J)=0 : NEXT : NEXT
3710 GOTO 1600
3720  '
3730  '
3740  '****************** INPUT SUBROUTINE ******************************
3750 PRINT : PRINT
3760 PRINT "========== ";POS;" #";I;" =========="
3770 PRINT
3780 PRINT P1$;
3790 INPUT "",IN   : IF IN > N OR IN < 0 THEN BEEP : GOTO 3780
3800 PRINT P2$;
3810 INPUT "",EN   : IF EN > N OR EN < 0 OR IN = EN THEN BEEP : GOTO 3800
3820 PRINT P3$;
3830 INPUT "",VL
3840 RETURN
```

```
**********************************************************************"
*                                                                    *"
*          DCNAP - D.C. Network Analysis Program                     *"
*                                                                    *"
*             (c) 1988 by James E. Tarchinski                        *"
*                                                                    *"
**********************************************************************"


  This program uses Nodal Analysis to determine the node voltages"
of linear D.C. networks.  These networks may range in size from"
1 to 25 nodes, not counting the ground, or datum, node."

  Models for five types of devices are incorporated in this"
program:"
     1. Resistors"
     2. Conductors"
     3. Independent current sources (ICS)"
     4. Voltage-controlled current sources (VCCS)"
     5. Independent voltage sources (IVS)"

Press any key ...
```

Fig. 2—The first screen contains general comments about the program.

top lines do not scroll off the screen before you have an opportunity to write the node voltages down, line 3340 pauses for a key press when the variable "I" is equal to 20 (assuming that there are at least 20 nodes in the circuit).

Example Circuits

After you enter DCNAP and save it to disk, you'll probably want to run some test data to verify that the program is functioning properly. Let's go through a few sample solutions. As previously mentioned, you must first call up BASIC; load and run DCNAP.

In addition to verifying that you've entered the program correctly, the solutions also demonstrate exactly how to

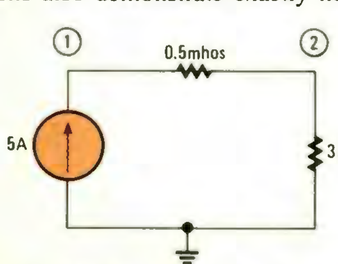

Fig. 3—This simple two-node circuit is used for the first example.

use the DC Network Analysis Program. The circuit for our first example is the two-node network shown in Fig. 3. The DCNAP solution for that circuit would be as follows (assuming that you've gotten past the *press any key* prompt:

```
Enter the number of nodes in the circuit,
not counting the ground node (1-25): 2

Enter the number of resistors: 1

============= RESISTORS # 1 =============

INITIAL NODE: 2
FINAL NODE:  0
VALUE (Ohms):  3

Enter the number of conductors:  1

=========== CONDUCTOR # 1 ============

INITIAL NODE:  1
FINAL NODE:  2
VALUE (Mhos):  0.5

Enter the number of ICSs:  1

============== ICS # 1 ===============

INITIAL NODE (The Tail):  0
FINAL NODE  (The Point):  1
VALUE (Amps):  5

Enter the number of DCSs:  0
Enter the number of IVSs:  0

THE NODE VOLTAGES ARE:
V(1) = 25 volts
V(2) = 15 volts

Would you like to analyze another circuit?
1. Yes, or
2. No
Your choice:  1
```

The next example that we'll analyze is the three-node circuit shown in Fig. 4. That circuit contains a voltage-controlled current source whose output current is dependent on V_x, the voltage from Node-1 to Node-2. A solution for

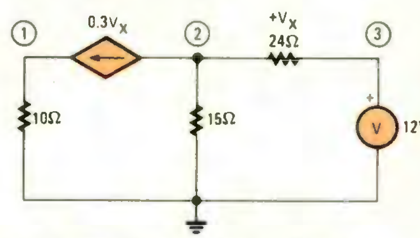

Fig. 4—The three-node circuit is a bit more complicated than the first, but is still child's play for DCNAP.

that circuit can readily be obtained as follows:

```
Enter the number of nodes in the circuit,
not counting the ground node (1-25):  3

Enter the number of resistors:  3

============= RESISTOR # 1 =============

INITIAL NODE:  1
FINAL NODE:  0
VALUE (Ohms):  10

============= RESISTOR # 2 =============

INITIAL NODE:  2
FINAL NODE:  0
VALUE (Ohms):  15

============ RESISTOR # 3 =============

INITIAL NODE:  2
FINAL NODE:  3
VALUE (Ohms):  24

Enter the number of conductors:  0
Enter the number of ICSs:  0
Enter the number of VCCSs:  1

========== VCCS # 1 ==================

INITIAL NODE (The Tail):  2
FINAL NODE  (The Point):  1
VALUE (Amps):  0.3
CONTROLLING NODE (Positive):  2
CONTROLLING NODE (Negative):  3

Enter the number of IVSs:  1

=========== IVS # 1 ===================

NEGATIVE NODE:  0
POSITIVE NODE:  3
MAGNITUDE (Volts):  12

CALCULATING, PLEASE WAIT...
THE NODE VOLTAGES ARE:

V(1) = -5.877554 volts
V(2) =  10.04082 volts
V(3) =  12 volts
```

The two test circuits that have been analyzed contain all of the circuit elements that DCNAP can handle. So if the version of DCNAP that you enter into your computer correctly solves both networks, you can be somewhat confident that you have entered the program correctly. If your solutions do not agree with the ones above, double check your work.

It may take a bit of work to enter DCNAP and then eliminate any typographical errors. But when you are done you'll have a powerful program capable of solving even the most complex DC circuits in just seconds. ■