Department of Applied Analysis
and Computer Science
Research Report ~~CRSS-2017~~ CSTR 1002
March, 1970

THE DIALATOR SYSTEM
TRAIZE USER'S GUIDE

by
Doron J.Cohen, Paul M.Fawcett
Eric G.Manning & Larry Smith

We wish to thank the Defence Research Board of Canada, the Northern Electric Research & Development Laboratories, Ottawa, and the Faculty of Mathematics of the University of Waterloo for financial, technical and moral support which made this system of programs possible.

USER'S GUIDE-DIALATOR SYSTEM-TRAIZE

This is a user's guide to program TRAIZE of the DIALATOR system. (A system programmer's guide,explaining internal workings of the program, appears later). This guide is laid out as follows:

Section 1 - Tells how to describe a logic circuit to the program.

Section 2 - Tells how to punch the circuit description on cards.

Section 3 - Describes the Job Control Language cards needed to make a TRAIZE run on a System/360 computer under OS/360.

Section 4 - Describes the output and what it means.

Section 5 - Contains TRAIZE run examples.

Section 6 - Error Diagnostics.

Section 1

How To Describe A Circuit

Here is a circuit. Let us refer to it from now on as CIRCUIT1.

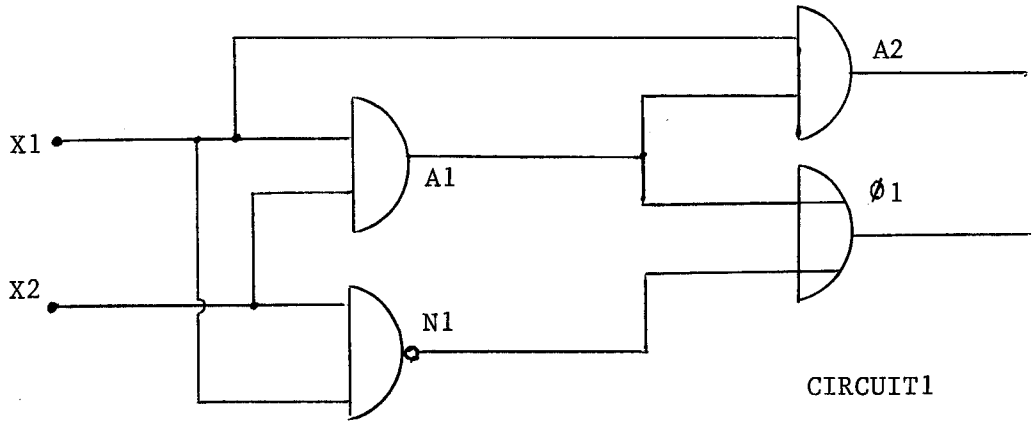

DIAGRAM 1

Here is how we describe CIRCUIT1 to TRAIZE:

```
(column 1 of card)
↓
% RUN:    TRAIZE   CIRCUIT=CIRCUIT1, DISP=NØ;
          INPUTS   (2)  X1,X2;
          OUTPUTS  (2)  Ø1,A2;
  X1:     INP;
  X2:     INP;
  A1:     AND  X1,X2;
  N1:     NAND X1,X2;
  A2:     AND  X1,A1;
  Ø1:     ØR   A1,N1;
          END  CIRCUIT1;
```

DIAGRAM 2

In CIRCUIT1 (see diagram 1) there are six points with labels. These labelled points are called the leads of CIRCUIT1.

Each lead performs a function in CIRCUIT1. X1 and X2 are primary inputs. A1 and A2 are AND gates. N1 is a NAND gate. Ø1 is an ØR gate. Also A2 and O1 are outputs for CIRCUIT1.

To describe CIRCUIT1 in words, for each lead we could give (see diagram 2 - lines 4 to 9):

1) its label.
2) the function (NAND, NOR, INPUT, etc.) it performs.
3) the labels of the leads which feed it (its inputs)
   or
   we could give the labels of its outputs.

In TRAIZE, we must give the input labels. We will see later how output labels can be optionally supplied to provide a powerful check for user coding errors.

Section 2

Circuit Description On Computer Cards

I Order And Content Of Card Statements

To make it easy for the system to read the description of CIRCUIT1 a certain order is followed (see diagram 2)

1. Control card which contains
    a) a run name ----- RUN
    b) the keyword TRAIZE
    c) a circuit name ----- CIRCUIT1
    d) a disposition code ----- NO
2. Statement which contains
    a) keyword ----- INPUTS
    b) number of inputs ----- 2
    c) list of inputs ----- X1, X2
3. Statement which contains
    a) keyword ----- OUTPUTS
    b) number of outputs ----- 2
    c) list of outputs ----- A2, Ø1
4. Statement which contains
    a) label of lead ----- X1
    b) function of lead ----- INP
    c) input references of lead ----- none

5. Additional statements as in 4.

6. Statement which contains

a) keyword --- END

b) a circuit name --- CIRCUIT1

II Card Format Rules

A) Control Card (see diagram 2 - line 1)

1. "%" must appear in column 1
2. followed by run name or program name
3. followed by ":"
4. followed by "TRAIZE"
5. followed by at least one blank
6. followed by "CIRCUIT"
7. followed by "="
8. followed by the circuit name
9. followed by ","
10. followed by "DISP"
11. followed by "="
12. followed by "NO" *
13. followed by ";"

The control card must be the first card in the deck (except for JCL cards). The "%" sign must be used only in the control card.

* Later we will see other values for DISP.

B) Header Statement Cards (see diagram 2, lines 2 & 3)

1. First word on the card must be one of the following keywords: "INPUTS"
   "OUTPUTS"
   "FEEDBACKS"
2. followed by "("
3. followed by number of inputs, outputs, or feedbacks
4. followed by ")"
5. followed by at least one blank
6. followed by an input, output or feedback label
7. followed by (a) "," if more labels
   (b) ";" if last label

Inputs and outputs statements must appear; feedbacks statement only if there are feedback loops in the circuit. The header statement cards must follow the control card; their relative order does not matter, however.

i.e. outputs statement can appear before inputs statement.

C) Lead Statement Cards (see diagram 2, lines 4 to 9)

1. First word is lead label
2. followed by ":"
3. followed by lead function
4. followed by at least one blank
5. followed by an input reference label
6. followed by a) "," if more input references
   b) ";" if last input reference

The lead statements must appear after the header statements; their relative order does not matter, however.

D) End Statement

1. First word is keyword "END"
2. followed by at least one blank
3. followed by circuit name
4. followed by ";"

III Other Rules

1. The card description is totally free format and blank is a delimiter. Thus any number of blanks may appear between fields and a field may start in any column. Moreover, a statement can overflow into one or more cards with no special action.
2. No word or number may have intervening blanks.

   eg. in CIRCUIT1 Card 1 Ø Ø TRØAIZE is invalid.
3. Comments may be inserted between any fields and must be enclosed in quotes.

   eg. X1: INP; "X1 IS AN INPUT"
4. Columns 73 through 80 are ignored.
5. Lead labels may not exceed 6 characters in length.
6. Run name and Circuit name may not exceed 8 characters in length.

Option

The programmer may list the output references, for any lead, in the lead statement. For example, in CIRCUIT1 the output reference labels for lead A1 are Ø1 and A2. The lead statement for A1 with output references would read:

```
A1:          AND          X1,X2/Ø1,A2;
```

RULES

1) programmer must list all or none of the output references for a particular lead but need not do output references for all leads.
2) input references and output references must be separated by a "/".

NOTE: If you give the optional output references for every lead of a circuit, program TRAIZE will make a very thorough check of your coding work for you. This is possible because either the input references or the output references alone provide enough information to completely define your circuit. Hence if you give both, you have really described your circuit twice. TRAIZE takes advantage of this by comparing the two descriptions and complaining loudly if any discrepancies are found.

The practice of giving both input and output references is therefore highly recommended, whenever the simulation results must be trustworthy.

Section 3) Control Cards for TRAIZE

The following deck is an example of the necessary control cards used to make a TRAIZE run on a system 360 under OS/360.

```
//TRAIZE  JOB  'U0664NORELEC,TIME=(,20),PAGES=99',MSGLEVEL=1,
//             REGION=250K
//JOBLIB  DD  DSN=P0785.LMAN2,DISP=SHR,UNIT=2314
//    EXEC  PGM=TRAIZE,REGION=250K
//SYSPRINT  DD  SYSOUT=A
//RECFILE  DD  DSN=U0664.RECFILE,DISP=OLD,UNIT=2314
//RECDIR   DD  DSN=U0664.RECDIR,DISP=OLD,UNIT=2314
//SCSYMTA  DD  DSN=&SCSYMTA,DISP=(NEW,PASS),SPACE=(TRK,(10)),
//             UNIT=2314,DCB=(RECFM=FB,BLKSIZE=200,LRECL=20)
//SCLDSC  DD  DSN=&SCLDSC,DISP=(NEW,PASS),SPACE=(TRK,(10)),
//             UNIT=2314,DCB=(RECFM=FB,BLKSIZE=200,LRECL=40)
//SCSS    DD   DSN=&SCSS,DISP=(NEW,PASS),SPACE=(TRK,(10)),
//             UNIT=2314,DCB=(RECFM=FB,BLKSIZE=200,LRECL=20)
//STRMIN  DD  DSN=&STRMIN,DISP=(NEW,PASS),SPACE=(TRK,(10)),
//             UNIT=2314,DCB=(RECFM=FB,BLKSIZE=500,LRECL=100)
//SYSIN  DD  *
```

Your TRAIZE deck is inserted here.

```
/*
```

Section 4

Output Description

The first output is the control card of your description. Next will be the Operators Table Listing.

It will look like

= = = = OPERATORS TABLE LISTING = = = =

NO.	SYMBOL	ADDRESS
1		0
2	AND	13
3	END	64
4	FBK	2
5	FBKS	36
6	FEEDBACKS	36
7	INP	1
8	INPS	34
9	INPUTS	34
10	NAND	8
11	NOR	12
12	NOT	7
13	OR	14
14	OUP	33
15	OUPS	35
16	OUTPUTS	35

The symbols are all the possible keywords that a programmer can use. They consist of Header Statement and Lead Statement keywords, plus "END".

Optional Header Statement keywords are also listed; They are:

INPS for INPUTS

OUPS for OUTPUTS

FBKS for FEEDBACKS

This is followed by a listing of your source deck; that is, the program describing the circuit.

Next is a printout of the variables:

	Value in CIRCUIT1
N# - number of leads in circuit	N# = 6
E# - number of labels in circuit	E# = 6
* K# - pointer to next available space in $SUCS.SS	K# = 1
I# - number of inputs	I# = 2
O# - number of outputs	O# = 2
F# - number of feedbacks	F# = 0
JOBPARM.CNAME - circuit name	JOBPARM.CNAME = 'CIRCUIT1'
JOBPARM.RUNAME - run name	JOBPARM.RUNAME = 'RUN'
JOBPARM.DISP - disposition parameter	JOBPARM.DISP = 'NO'

The OPERATORS TABLE LISTING is printed out again as above.

Next the number of levels of the circuit is printed.

* K# explained in Programmer's Guide

Here are the levels in CIRCUIT1

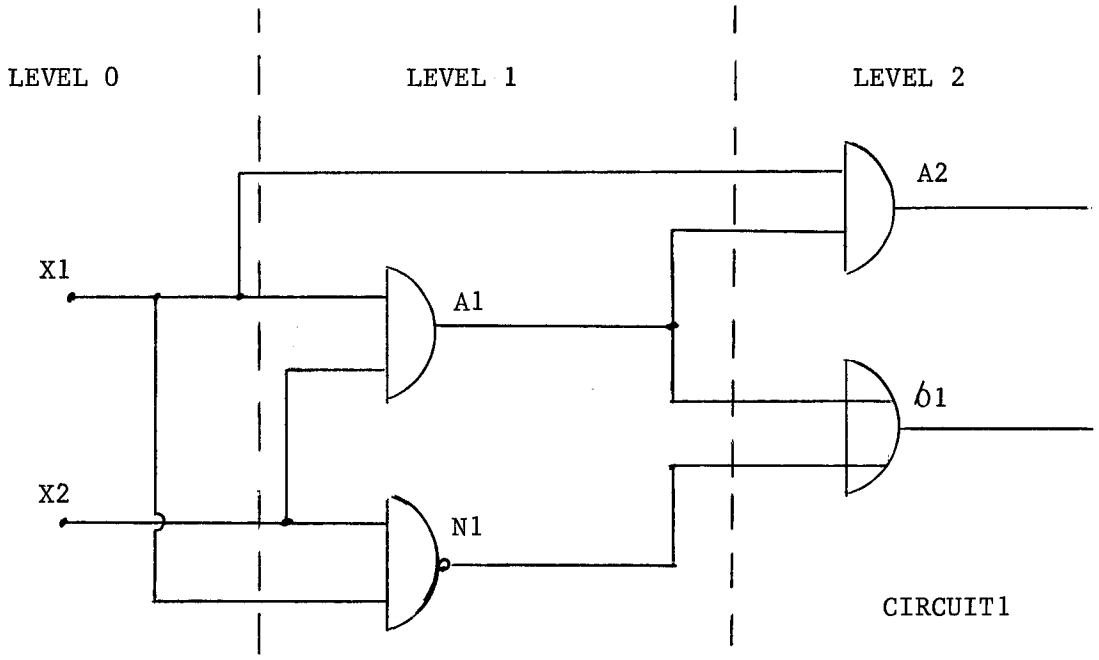

DIAGRAM 3

By dividing CIRCUIT1 into levels, we can picture CIRCUIT1 as the union of 3 circuits: CIRCUIT1. LEVEL 0

CIRCUIT1. LEVEL 1

CIRCUIT1. LEVEL 2

The outputs of LEVEL 0 are the inputs of LEVEL 1 and the outputs of LEVEL 1 are the inputs of LEVEL 2.

The last output is a mapping of the programmer's ordering of the leads to required ordering of the leads.

In CIRCUIT1 the leads of LEVEL 0 must be first, followed by the leads of LEVEL 1, followed by the leads of LEVEL 2.

For example (see diagram 3)

(1)

Lead	Programmer's Order	Required Order	Mapping Output
X1	1	1	1 ---- 1
X2	2	2	2 ---- 2
A1	3	3	3 ---- 3
N1	4	4	4 ---- 4
A2	5	5	5 ---- 5
Ø1	6	6	6 ---- 6

If we had written the leads in a different order, the picture would change:

(2)

X1	1	1	1 ---- 1
Ø1	2	5	2 ---- 5
N1	3	3	3 ---- 3
A2	4	6	4 ---- 6
A1	5	4	5 ---- 4
X2	6	2	6 ---- 2

In general, to keep output simple try to list leads of LEVEL 0, followed by leads of LEVEL 1, etc., (in program deck).

Section 5

The first example is CIRCUIT1. The DISP used is YES. (An explanation of DISP values is contained in APPENDIX A of this manual.)

In addition to the normal output, described earlier, a listing of the file directory tree is printed. The reason is, we have asked to have the description of CIRCUIT1 stored on the file. If it has been stored, CIRCUIT1 should appear as a node in the tree.

Following this the file description of CIRCUIT1 is fetched and printed. If you understand this description you should be able to draw from it, CIRCUIT1, as shown in DIAGRAM 1, Section 1.

Here is the run output.

EXAMPLE 1

```
%RUN:   TRAIZE  CIRCUIT=CIRCUIT1,DISP=YES;                    00010000SOURCE.

==FDIR==OLD DIRECTORY FETCHED.

==FETCH==TRAIZE.$OPTAB
STRUCTURE: $OPTAB   OF NODE NO.  51,FETCHED     228 BYTES,STARTING REGION NO. 11 WORD NO. 621


 ===== STRUCTURE NO.  14 =====

 SIZE:   228 BYTES, RECORD NAME:TRAIZE.$OPTAB

 $OPTAB.OP#=       16;


 ===== OPERATORS TABLE LISTING=====

   NO.   SYMBOL   ADDRESS
     1                 0
     2   AND          13
     3   END          64
     4   FBK           2
     5   FBKS         36
     6   FEEDBACK     36
     7   INP           1
     8   INPS         34
     9   INPUTS       34
    10   NAND          8
    11   NOR          12
    12   NOT           7
    13   OR           14
    14   OUP          33
    15   OUPS         35
    16   OUTPUTS      35
```

```
        INPUTS  (2)  X1,X2;                                          00020000SOURCE.
        OUTPUTS (2)  O1,A2;                                          00030000SOURCE.
  X1:   INP;                                                         00040000SOURCE.

***TRAIZE*** COMBINATIONAL CIRCUIT.
  X2:   INP;                                                         00050000SOURCE.
  A1:   AND  X1,X2;                                                  00060000SOURCE.
  N1:   NAND  X1,X2;                                                 00070000SOURCE.
  A2:   AND  X1,A1;                                                  00080000SOURCE.
  O1:   OR   A1,N1;                                                  00090000SOURCE.
        END  CIRCUIT1;                                               00100000SOURCE.
```

(a second listing of TRAIZE.$OPTAB has been omitted)


```
          PHASE  BUILDER
THE GIVEN CIRCUIT PARAMETERS ARE


N#=        6            E#=        6            K#=        1            I#=        2
F#=        0            CNAME='CIRCUIT1'        JOBPARM.RUNAME='RUN'    JOBPARM.DISP='YES'

==FDIR==OLD DIRECTORY FETCHED.

==FETCH==TRAIZE.$OPTAB
STRUCTURE: $OPTAB   OF NODE NO.  51,FETCHED     228 BYTES,STARTING REGION NO. 11 WORD NO. 621
```

```
CIRCUIT1.$CIDS.$SUCS                CREATED.

==LECR==THE CIRCUIT ORGANIZED AND LEVELED IN:    3 LEVELS.

THE MAPPING:
       1->    1        2->    2        3->    3        4->    4
       5->    5        6->    6

==STORE==CIRCUIT1.$CIDS
STRUCTURE: $CIDS     OF NODE NO.  71, STORED      86 BYTES,STARTING REGION NO. 12 WORD NO. 525

==STORE==CIRCUIT1.$CIDS.$LDSC
STRUCTURE: $LDSC     OF NODE NO.  78, STORED     168 BYTES,STARTING REGION NO. 12 WORD NO. 547

==STORE==CIRCUIT1.$CIDS.$SYMTA
STRUCTURE: $SYMTA    OF NODE NO.  81, STORED     124 BYTES,STARTING REGION NO. 12 WORD NO. 492

==STORE==CIRCUIT1.$CIDS.$REFS
STRUCTURE: $REFS     OF NODE NO.  82, STORED      56 BYTES,STARTING REGION NO. 12 WORD NO. 589

==STORE==CIRCUIT1.$CIDS.$SUCS
STRUCTURE: $SUCS     OF NODE NO.  83, STORED      75 BYTES,STARTING REGION NO.  0 WORD NO.  20
```


O#= 2

```
===== DIRECTORY LISTING =====

    98 AVAIL POINTER.
    40 REGIONS USED .
    24 STRUCTURES RECOGNIZED.
   136 NODES IN THE DIRECTORY.
   141 DESIRED NO. OF NODES.
    22 KEY-WORDS RECOGNIZED.
700308 DATE OF LAST CHANGE.


===== DIALATOR KEY-WORDS TABLE =====

NO.  KEY-WORD  INDEX
  1                0
  2                0
  3                0
  4                0
  5                0
  6                0
  7                0
  8                0
  9  $CIDS         2
 10  $COPT         1
 11  $CTRES       10
 12  $DIATR       11
 13  $FMCS         4
 14  $LDSC         3
 15  $LVBS        13
 16  $OPTAB       14
 17  $REFS         6
 18  $SFMC         9
 19  $SUCS         7
 20  $SYMTA        5
 21  $TEDS        12
 22  $TERM         8
```

```
===== THE DIRECTORY TREE =====

**********************
*NO. REC_NAME     SIZ*
* STB SRG TYP SIS DAU*
**********************
*  0 REC_FILE       0*
*  -1  -1   0   4   1*
**********************************************
                     *  1 HEADFREE       0*
                     *  -1  -1   0  88  -1*
                     **********************
                     * 88 ..FREE..     273*
                     * 718  12   0  87  -1*
                     **********************
                     * 87 ..FREE..      74*
                     * 612  12   0  89  -1*
                     **********************
                     * 89 ..FREE..     113*
                     * 218  12   0  77  -1*
                     **********************
                     * 77 ..FREE..       3*
                     * 699  12   0  85  -1*
                     **********************
                     * 85 ..FREE..      74*
                     * 882  11   0  80  -1*
                     **********************
                     * 80 ..FREE..      37*
                     * 603  12   0  70  -1*
                     **********************
                     * 70 ..FREE..      11*
                     *  ?0   0   0  7?  -1*
```

(Part of the tree print out has been omitted.

Section containing CIRCUIT1 & DECODER are shown on next page.)


79 CIRCUIT1 0
-1 -1 0 91 71
71 $CIDS 86
525 12 2 96 78
78 $LDSC 168
547 12 3 81 -1
81 $SYMTA 124
492 12 5 82 -1
82 $REFS 56
589 12 6 83 -1
83 $SUCS 75
20 0 7 84 -1
91 DECODER 0
-1 -1 0 -1 90
90 $CIDS 86
70 12 2 -1 92
92 $LDSC 718
786 12 3 93 -1
93 $SYMTA 424
92 12 5 94 -1
94 $REFS 77
198 12 6 95 -1
95 $SUCS 270
631 12 7 -1 -1

```
**********************

    98 NODES IN THE DIR TREE     29 NODES IN THE FREE LIST.
 89056 BYTES USED IN RECORDS, 198944 BYTES FREE.
    39 UNUSED NODES.

==SDIR==DIRECTORY REWRITTEN  700312   003025680


==LCID==CIRCUIT NAME: CIRCUIT1, OPTIONS: 0100 FILE ONLY.
FAULTS OPTION:


===== CIRCUIT1 CIRCUIT RECORDS =====


CIRCUIT1 NODE NO.  79
$CIDS    NODE NO.  71      86 BYTES,STARTING REGION NO.    12 WORD NO. 525
$LDSC    NODE NO.  78     168 BYTES,STARTING REGION NO.    12 WORD NO. 547
$FMCS    NODE NO.  84     132 BYTES,STARTING REGION NO.    12 WORD NO. 450
$SYMTA   NODE NO.  81     124 BYTES,STARTING REGION NO.    12 WORD NO. 492
$REFS    NODE NO.  82      56 BYTES,STARTING REGION NO.    12 WORD NO. 589
$SUCS    NODE NO.  83      75 BYTES,STARTING REGION NO.     0 WORD NO.  20
$TERM    NOT FOUND.
```

```
===== STRUCTURES LOCATIONS =====
STRUCTURE: $CIDS    OF NODE NO.  71,FETCHED     86 BYTES,STARTING REGION NO. 12 WORD NO. 525
STRUCTURE: $LDSC    OF NODE NO.  78,FETCHED    168 BYTES,STARTING REGION NO. 12 WORD NO. 547
STRUCTURE: $SYMTA   OF NODE NO.  81,FETCHED    124 BYTES,STARTING REGION NO. 12 WORD NO. 492
STRUCTURE: $REFS    OF NODE NO.  82,FETCHED     56 BYTES,STARTING REGION NO. 12 WORD NO. 589
STRUCTURE: $SUCS    OF NODE NO.  83,FETCHED     75 BYTES,STARTING REGION NO.  0 WORD NO.  20

EV='01101110'B          R=      110;


      6 LEADS.

      2 OUTPUTS  :  O1      ,A2

      2 INPUTS   :  X1      ,X2

      0 FEEDBACKS:

      0 SINGLE FAULTS.

     12 SUCCESSORS IN SS.

      0 TERMINALS.

      6 SYMBOLIC NAMES.

      0 FAUST RUNS.

      3 LOGIC LEVELS.
```

NO.	NAME	ATOM	LVL.	INPUT REFERENCES			FANOUT	OUTPUT REFERENCES			ASSOCIATED FAULTS NO.S
1	X1	1	0	0	0	0	3..	3	4	5	INPUT# 1
				0	0						
2	X2	1	0	0	0	0	2..	3	4		INPUT# 2
				0	0						
3	A1	13	1	1	2	0	2..	5	6		
				0	0						
4	N1	8	1	1	2	0	1..	6			
				0	0						
5	A2	13	2	1	3	0	0				
				0	0						OUTPUT# 2
6	O1	14	2	3	4	0	0				
				0	0						OUTPUT# 1

The second example is a description of a circuit named DECODER. Here DISP=NEW is used. The description of the circuit is not stored, because there is a circuit in the file with the name DECODER. This is evident if we look at the previous example. If we had put DISP=OLD or DISP=YES, then we would have overwritten the description presently in the file. To understand the implications of this see APPENDIX A.

Here is the run output.

EXAMPLE 2

```
%RUN1: TRAIZE  CIRCUIT=DECODER,  DISP=NEW;                    00010000SOURCE.

==FDIR==OLD DIRECTORY FETCHED.

==FETCH==TRAIZE.$OPTAB
STRUCTURE: $OPTAB    OF NODE NO.  51,FETCHED     228 BYTES,STARTING REGION NO. 11 WORD NO. 621

===== STRUCTURE NO.  14 =====

SIZE:   228 BYTES, RECORD NAME:TRAIZE.$OPTAB

$OPTAB.OP#=       16;


===== OPERATORS TABLE LISTING=====

  NO.   SYMBOL   ADDRESS
    1                 0
    2   AND          13
    3   END          64
    4   FBK           2
    5   FBKS         36
    6   FEEDBACK     36
    7   INP           1
    8   INPS         34
    9   INPUTS       34
   10   NAND          8
   11   NOR          12
   12   NOT           7
   13   OR           14
   14   OUP          33
   15   OUPS         35
   16   OUTPUTS      35
```

```
INPUTS(7)                    A,B,C,D,E,F,G;                    00020000SOURCE.
OUTPUTS(4)                    ND1,ND2,ND3,ND4;                 00030000SOURCE.
A    :            INP          /NOR1;                         00040000SOURCE.

***TRAIZE*** COMBINATIONAL CIRCUIT.
B    :            INP          /NOR1,NOR3,NOR6,NOR9;          00050000SOURCE.
C    :            INP          /NOR2,NOR3,NOR4;               00060000SOURCE.
D    :            INP          /NOR2,NOR5,NOR8;               00070000SOURCE.
E    :            INP          /NOR4,NOR7,NOR10;              00080000SOURCE.
F    :            INP          /NOR5,NOR6,NOR7;               00090000SOURCE.
G    :            INP          /NOR8,NOR9,NOR10;              00100000SOURCE.
NOR1:             NOR        A,B/NOR15,NOR19;                 00110000SOURCE.
NOR2:             NOR        C,D/NOR12;                       00120000SOURCE.
NOR3:             NOR        B,C/NOR14;                       00130000SOURCE.
NOR4:             NOR        C,E/NOR13,NOR13,NOR14;           00140000SOURCE.
NOR5:             NOR        D,F/NOR17;                       00150000SOURCE.
NOR6:             NOR        B,F/NOR12,NOR17;                 00160000SOURCE.
NOR7:             NOR        E,F/NOR16,NOR16,NOR18;           00170000SOURCE.
NOR8:             NOR        D,G/NOR11,NOR15,NOR18;           00180000SOURCE.
NOR9:             NOR        B,G/NOR19;                       00190000SOURCE.
NOR10:            NOR        E,G/NOR11,NOR20,NOR20;           00200000SOURCE.
NOR11:            NOR        NOR10,NOR8/ND1;                  00210000SOURCE.
NOR12:            NOR        NOR2,NOR6/ND1;                   00220000SOURCE.
NOR13:            NOR        NOR4,NOR4/ND1;                   00230000SOURCE.
NOR14:            NOR        NOR4,NOR3/ND2;                   00240000SOURCE.
NOR15:            NOR        NOR1,NOR8/ND2;                   00250000SOURCE.
NOR16:            NOR        NOR7,NOR7/ND2;                   00260000SOURCE.
NOR17:            NOR        NOR5,NOR6/ND3;                   00270000SOURCE.
NOR18:            NOR        NOR8,NOR7/ND3;                   00280000SOURCE.
NOR19:            NOR        NOR9,NOR1/ND4;                   00290000SOURCE.
NOR20:            NOR        NOR10,NOR10/ND4;                 00300000SOURCE.
ND1  :            NAND       NOR11,NOR12,NOR13;               00310000SOURCE.
ND2  :            NAND       NOR14,NOR15,NOR16;               00320000SOURCE.
ND3  :            NAND       NOR17,NOR18;                     00330000SOURCE.
ND4  :            NAND       NOR19,NOR20;                     00340000SOURCE.
               END DECODER;                                    00350000SOURCE.
```

```
            PHASE  BUILDER
THE GIVEN CIRCUIT PARAMETERS ARE

                                                                                    O#=          4

N#=         31              E#=         31              K#=         78              I#=          7
F#=          0              CNAME='DECODER'             JOBPARM.RUNAME='RUN1'    JOBPARM.DISP='NEW';

==FDIR==OLD DIRECTORY FETCHED.

==FETCH==TRAIZE.$OPTAB
STRUCTURE: $OPTAB   OF NODE NO.  51,FETCHED     228 BYTES,STARTING REGION NO. 11 WORD NO. 621

              (  a second listing of TRAIZE.$OPTAB has been omitted   )

DECODER.$CIDS.$SUCS                  CREATED.

==LEOR==THE CIRCUIT ORGANIZED AND LEVELED IN:     4 LEVELS.

THE MAPPING:
       1->    1        2->    2        3->    3        4->    4
       5->    5        6->    6        7->    7        8->    8
       9->    9       10->   10       11->   11       12->   12
      13->   13       14->   14       15->   15       16->   16
      17->   17       18->   18       19->   19       20->   20
      21->   21       22->   22       23->   23       24->   24
      25->   25       26->   26       27->   27       28->   28
      29->   29       30->   30       31->   31
**STORE**FAILED,"NEW" DECODER.$CIDS                          EXISTS.
***TRAIZE***  FAILED TO STORE TABLES IN FILE SYSTEM.

***TRAIZE***  ERRORS FOUND BY BUILDER OR DISPOSITION ='NO'.
NO TABLES WRITTEN INTO  FILE SYSTEM.
```

Section 6

Error Diagnostics

Here are 2 examples of incorrect descriptions.

Example (1)

```
          CIRCUIT1       Description And Error Messages

   % RUN:      TRAIZE   CIRCUIT = CIRCUIT1 , DISP = NO;
               INPUTS   (3)  X1,X2,X3;
               OUTPUTS  (2)  01,A2;
      X1:      INP;
   *** TRAIZE ***    COMBINATIONAL CIRCUIT.
      X2:      INP;
      A1:      AND           X1,X2;
      N1:      NAN           X1,X2;
   *** TRAIZE ***   ILLEGAL FUNCTION CODE ... NAN INSTATEMENT NR. 4
      A2:      AND           X1,A1,03;
      01       OR            A1,N1;
   *** TRAIZE *** NO LABEL ON THIS ATOM, STATEMENT NO. 6
   *** TRAIZE *** ILLEGAL FUNCTION CODE ... 01 IN STATEMENT NR. 6
CONDITION ENDF OCCURRED IN STATEMENT 00010  AT OFFSET  . +00130 FROM
ENTRY POINT NECAR.
                    .
                    .
                    .
                    .
                    .
                    .
```

TRAIZE has not gone through the second stage of processing. The reason is the absence of the END statement.

Omitting the END statement not only is an error, but it prevents other errors, those picked up in the second stage, from being found.

In CIRCUIT1 two errors that would be picked up in the second stage are checked.

Therefore we can not debug CIRCUIT1 in a single run.

Example (2)

In this circuit description all of the errors are found in the second stage. That is, none of the errors are syntax errors.

Therefore the error messages appear following the circuit description listing.

```
CIRCUIT2              Description
%RUN:          TRAIZE   CIRCUIT=CIRCUIT2, DISP=NO;
               INPUTS  (3)   X1,X2,X3;
               OUTPUTS (1)   O1;
               FEEDBACKS  (1)  F1;
  X1:          INP             /A1;
  X2:          INP             /A1,O1;
  A1:          AND      X1,X2  /O2;
  O1: OUT:     OR       A1,X2;
               END  CIRCUIT2;
```

Try to pick out the errors yourself before looking at the copy of error messages from TRAIZE on following page.

*** TRAIZE *** THE SUCCESSOR 02 AT POSITION 6 OF THE SUCCESSORS LIST HAS NOT BEEN DECLARED AS A LEAD OF THE CIRCUIT.

CIRCUIT2.$CIDS.$SUCS CREATED.

TRAIZE THE USER-SUPPLIED SUCCESSOR FOR LEAD A1 HAS NOT BEEN DECLARED AS A LEAD.

TRAIZE INPUT NAME X3 HAS BEEN GIVEN IN THE INPUT LIST BUT HAS NOT BEEN DECLARED AS AN INPUT LEAD.

TRAIZE FEEDBACK NAME F1 HAS BEEN GIVEN IN THE FEEDBACKS LIST BUT HAS NOT BEEN DECLARED AS A FBK LEAD.

==LEOR==THE CIRCUIT ORGANIZED AND LEVELED IN: 3 LEVELS.

THE MAPPING:

1- > 1 2- > 2 3- > 3 4- > 4

TRAIZE ERRORS FOUND BY BUILDER OR DISPOSITION = "NO".
NO TABLES WRITTEN INTO FILE SYSTEM.

GLOSSARY OF ERROR MESSAGES

Messages concerning the control card:

*** CCINT FAILED ** is printed out whenever there is any error in the control card or control card is missing.

** CCINT ** INVALID OPERAND FOR_____.
the operand appearing after CIRCUIT or DISP is not there.

* CONTROL CARD FOUND IN DATA
card beginning with "%" in column one was found.

. FLUSHED printed after each card appearing before the control card. These cards are ignored.

* INVALID DELIMITER_____.
an equal sign does not appear after CIRCUIT or DISP.

* INVALID KEYWORD_____.
either CIRCUIT or DISP do not appear in control card and some other keyword appears in their place.

* INVALID KEYWORD IN CONTROL CARD
TRAIZE does not appear after RUNAME.

* INVALID OR MISSING OPERATION _____.

TRAIZE does not appear or is followed by some delimiter other than a blank.

* RUNAME INVALID OR MISSING

first name after "%" is not followed by a colon or is blank.

Messages concerning TRAIZE in general. All of these messages are preceded by *** TRAIZE ***, which will be left out here.

* ATTEMPT TO READ TRAIZE.$OPTAB FROM DIAL FILE FAILED RUN ABORTED.

This is not your error but a failure of the system.

* COMBINATIONAL CIRCUIT

This is not an error, but indicates the type of circuit you have described.

* CONTROL CARD INVALID

There was an error encountered in control card.

* COULDN'T READ CONTENTS OF COPTA FROM LIBRARY

This is not your error but a failure of the system.

* COULDN'T RECOVER $SYMTA FROM REC_DIR

* CRESSA FAILED, RUN ABORTED

The input reference leads were not written in file.

Error in input reference.

* ERRORS FOUND BY BUILDER OR DISPOSITION = "NO". NO TABLES WRITTEN INTO FILE

Could also be errors found by syntax analizer.

* FAILED TO STORE TABLES IN FILE SYSTEM

Description of circuit not printed correctly. Check syntax errors.

* FEEDBACK NAME _____ HAS BEEN DECLARED IN THE LIST OF FEEDBACKS BUT ITS FUNCTION IS NOT FBK.

* FEEDBACK NAME_____ HAS BEEN GIVEN IN THE FEEDBACKS LIST BUT HAS NOT BEEN DECLARED AS A FBK LEAD.

* FEEDBACK STATEMENT HAS NO TERMINAL SEMICOLON, OR NR OF FEEDBKS GIVEN, =_____ IS LESS THAN NR COUNTED.

* FUNCTION NOT YET IMPLEMENTED

The function name is in the table but system is not equipped to handle it.

* ILLEGAL FUNCTION CODE _____ IN STATEMENT NR _____.

* INPUT NAME _____ HAS BEEN GIVEN IN THE INPUT LIST BUT HAS NOT BEEN DECLARED AS AN INPUT LEAD.

* INPUT NAME _____ HAS BEEN GIVEN IN THE LIST OF INPUTS BUT ITS FUNCTION IS NOT INP.

* INPUT REFERENCE LIST FOR STATEMENT NR_____ DOES NOT END WITH SLASH OR SEMICOLON.

* INPUTS, OUTPUTS AND FEEDBACK STATEMENTS MUST PRECEDE ALL LEAD DECLARATIONS.

* INPUTS, OUTPUTS AND FEEDBACKS STATEMENTS CANNOT APPEAR IN THE MIDDLE OF LEAD DECLARATIONS.

* INPUTS STATEMENT HAS NO TERMINAL SEMICOLON, OR NR OF INPUTS GIVEN, =_____ IS LESS THAN NR COUNTED =_____.

* INVALID KEYWORD IN INPUTS, OUTPUTS OR FEEDBACKS STATEMENT_____.
The keyword was not found in table-check spelling.

* NO INPUT REFERENCES WERE FOUND FOR STATEMENT NR_____.
Input references for each lead are necessary.

* NO INPUTS STATEMENT FOUND.
Inputs statement missing or in wrong place.

* NO LABEL ON THIS LEAD STATEMENT NR.=_____

* NO OUTPUTS STATEMENT FOUND.
Outputs statement missing or in wrong place.

* NO PARENTHESES FOUND AROUND NUMBER OF FEEDBACKS IN FEEDBACKS STATEMENT.
Following "FEEDBACKS" must have number of feedbacks in parenthesis.

* NO PARENTHESES FOUND AROUND NUMBER OF INPUTS IN INPUTS STATEMENT.
  Following "INPUTS" must have number of inputs in parenthesis.

* NO PARENTHESES FOUND AROUND NUMBER OF OUTPUTS IN OUTPUTS STATEMENT.
  Following "OUTPUTS" must have number of outputs in parenthesis.

* NO TERMINAL SEMICOLON.

* NUMBER OF FEEDBACKS GIVEN =_____ IS MORE THAN NUMBER ACTUALLY COUNTED _____.

* NUMBER OF INPUTS GIVEN =_____IS MORE THAN NUMBER ACTUALLY COUNTED_____.

* NUMBER OF OUTPUTS GIVEN =_____IS MORE THAN NUMBER ACTUALLY COUNTED_____.

* OUTPUT NAME_____HAS BEEN GIVEN IN THE OUTPUT LIST BUT HAS NOT BEEN DECLARED AS AN OUTPUT LEAD.

* OUTPUTS STATEMENT HAS NO TERMINAL SEMICOLON OR NR OF OUTPUTS GIVEN,= _____IS LESS THAN NR COUNTED.

* THE FEEDBACK LEAD OF STATEMENT_____HAS MORE THAN 1 INPUT REFERENCE.
  A feedback lead can have and must have one input reference.

* THE INPUT_____OF THE LEAD_____HAS NOT BEEN DECLARED AS A LEAD OF THE CIRCUIT.

* THE SINGLE INPUT REFERENCE FOR THE FEEDBACK LEAD OF STATEMENT NR_____
IS FOLLOWED BY / OR ;

* THE SUCCESSOR_____AT POSITION_____OF THE SUCCESSORS LIST HAS
NOT BEEN DECLARED AS A LEAD OF THE CIRCUIT.
Check out put reference names for spelling and input
reference names.

* THE USER SUPPLIED SUCCESSOR_____FOR LEAD_____IS IN DISAGREEMENT
WITH THE COMPUTED ONES. THEY ARE ... _____

* THIS STATEMENT HAS A SLASH BUT NO OUTPUT REFERENCES CAN BE FOUND.

* TOO MANY INPUT REFERENCES FOR THE LEAD OF STATEMENT NR. _____

APPENDIX A

Other values for DISP

The following are alternative values for DISP. The user should become familiar with the system's file operation before he commands anything other than DISP=NO.

DISP=NO - do not save the run results in the file system.

DISP=YES - save the run results in the file system.

DISP=NEW - create new entries in the file for the run results.

DISP=OLD - overwrite already existing records of run results with the same name.

Notice that YES is a combination of OLD and NEW, that is, if there are records overwrite, or if there are no records create new entries.