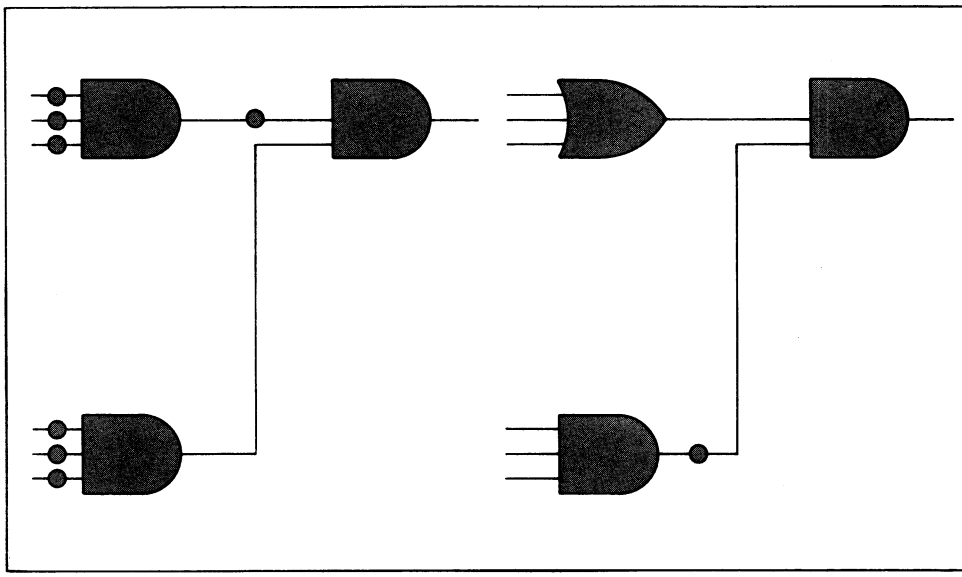


$\overline{A}+\overline{B}+\overline{C}=\overline{ABC}$

LOGMIN

LOGIC CIRCUIT MINIMIZATION

TYMSHARE®

Price: $1.25

TYMSHARE MANUALS

REFERENCE SERIES

LOGMIN

Logic Map Minimization Program

January 1970

Tymshare, Inc.
10340 Bubb Road
Cupertino, California 95014

ymshare, Inc., Litho in U.S.A.

TABLE OF CONTENTS

I. INTRODUCTION

The Tymshare Library program LOGMIN was developed to aid the logic circuit design engineer in obtaining the minimized Boolean equation from a logic map.[1] (references in section VI) The program can compute the equation for cases with don't care product terms.

The input to the program is the number of variables in the Boolean equation, the map location and the desired resultant for a given product term. Only product terms of logical one (true) and don't care cases need be specified. All cases not specified are assumed logical zero (false). Don't care case is two (2).

The minimized equation is of the sigma pi ( ) form.

f = (product term) + (product term) + . . .

A final check on the validity of the equation may be produced in the form of a truth table with the User Program Library program LOGIC. An example of a check of the minimized equation is shown in Section IV, Example 4.

II. PROGRAM PERFORMANCE CHARACTERISTICS

The present version (version 4) can handle up to a maximum of eleven variables. The time required to calculate the minimized equation is dependent on the number of variables and the number of product terms specified by logical true and don't care cases. However, a feeling for the execution time can be gained by observing the sample executions in Section IV.

A map is defined as the relationship between the vertical position within the truth table of the input variables and the logical value (0, 1, or 2) of the output. For example,

Input 2	Input 1	Output Value	Map Position
0	0	0	0
0	1	1	1
1	0	0	2
1	1	2	3

III. OPERATING INSTRUCTIONS

Input Capabilities

The program has the following input capabilities.

- Enter data directly into the program from the terminal.
- Enter data from a file. The file may be created in EDITOR or it could be created by reading a paper tape (which could be punched "off line" before logging in) using the Tymshare paper tape program TAPE.
- Modify existing data in the program and recompute.
- List the existing data in the program to the user's terminal.
- Write the input data in the program to a file for use at a later date.

Conversational Mode

The following conversational mode capabilities are available.

Standard Front End Commands

```
:HELP

LEGAL COMMANDS:

     HELP  (OR) ?    REPRINTS THIS LIST
     CAPABILITIES    DESCRIBES PROGRAM CAPABILITIES
     INSTRUCTIONS    HOW TO EXECUTE THE PROGRAM
     CREDITS         THOSE RESPONSIBLE FOR THIS PROGRAM
     CHARGES         ADDITIONAL COST (IF ANY)
     VERSION         LATEST UPDATE
     EXPERT          LESS CONVERSATIONAL FOR EXPERIENCED USERS
     RUN             BEGINS EXECUTION
     INPUT           SPECIFY DATA FILE NAME
     LIST            LISTS ALL INPUT DATA ON USERS TERMINAL
     REVISIONS       CHANGE DATA WHICH HAS BEEN INPUT TO PROGRAM
     COMPUTE         COMPUTE & PRINT PRODUCT TERMS
     WRITE           WRITES INPUT DATA TO A FILE
     QUIT  (OR) Q    QUITS TO EXEC

     ANY OF THESE COMMANDS MAY BE SHORTENED TO THE
     FIRST THREE LETTERS
```

```
:CAPABILITIES

1 MINUTE OF PRINTOUT

THIS PROGRAM COMPUTES THE MINIMIZED OR SIMPLIFIED BOOLEAN EQUATION
FROM A MAP.  THE RESULTANT EQUATION IS THE SUM OF PRODUCT TERMS.
A SIGMA PI EQUATION.
THE PROGRAM ONLY REQUIRES SPECIFYING THE MAP LOCATION FOR ALL
LOGICAL TRUE AND DON'T CARE CASES.   ALL LOCATIONS NOT SPECIFIED
ARE ASSUMED LOGICAL FALSE.

THE COMPUTING METHOD IS SIMILIAR TO THE TABULATION METHOD OF MCCLUSKEY.

REFERENCES:
TORNG, H.C.; INTRODUCTION TO THE LOGICAL DESIGN OF SWITCHING SYSTEMS;
ADDISON-WESLEY PUB., INC.; 1964 (TABULATION METHOD)

INPUT SPECIFICATIONS:
THE PROGRAM CAN HANDLE UP TO 11 VARIABLES.


:INSTRUCTIONS

2 MINUTES OF PRINTOUT

TYPE 'RUN' TO START PROGRAM.

THIS PROGRAM MAKES AN INITIAL REQUEST OF:

INPUT FROM:

TO SPECIFY INPUT FROM THE TERMINAL, TYPE  TEL, TE, OR T
TO SPECIFY INPUT FROM A FILE, TYPE THE FILE NAME.

IN EITHER CASE, FOLLOW TEL OR THE FILE NAME BY A CARRIAGE RETURN.

*** WHEN INPUT FROM THE TEL IS REQUESTED:
THE INPUT OF NEW DATA OR REVISIONS IS OF THE FOLLOWING
FORM:

  ENTER NO. OF VARIABLES: N    (NOT ASKED, FOR REVISIONS)
  MAP POSITION: MP        WHERE  MP = MAP POSITION (PRODUCT TERM
  VALUE: V                       V = 1 OR 2                FROM A MAP)
  MAP POSITION: MP               V = 1 FOR TRUE
  VALUE: V                       V = 2 FOR DON'T CARE CASE
     .                           ALL OTHERS ARE ASSUMED FALSE
     .
     .
  MAP POSITION: END        (END OF INPUT OR REVISION)
```

cont'ed next page

Standard Front End Commands (cont'ed)

ALTERNATE INPUT FORM

MAP POSTION: MP,V,MP,V,MP,V,...,END (ALL VALUES SEPERATED BY A COMMA)

*** WHEN INPUT FROM A FILE IS REQUESTED:
THE FOLLOWING FILE FORMAT IS REQUIRED.

```
N           WHERE N=NO. OF BOOLEAN VARIABLES
MP,V              MP= MAP POSITION (PRODUCT TERM)
MP,V              V= 1 OR 2
  .
  .
  .
MP,V
```

*** PROGRAM OUTPUT:
THE OUTPUT OF THE PROGRAM MAY CONSIST OF ONE OR MORE PRODUCT TERMS OF THE FOLLOWING FORM.

X5 -X4 -X3 X2 X1 X0

X5 -X4 X3 -X2 X0

ETC.

THE RESULTANT BOOLEAN EQUATION WOULD BE OF THE FORM:

F = [X5 * -X4 * -X3 * X2 * X1 * X0] + [X5 * -X4 * X3 * -X2 * X0
+ ETC.

WHERE X5, X4, X3, X2, X1, X0 ARE BOOLEAN VARIABLES
AND -X5 = NOT X5 (THE COMPLIMENT)

:VERSION

VERSION 4 DECEMBER, 1969

:CREDITS

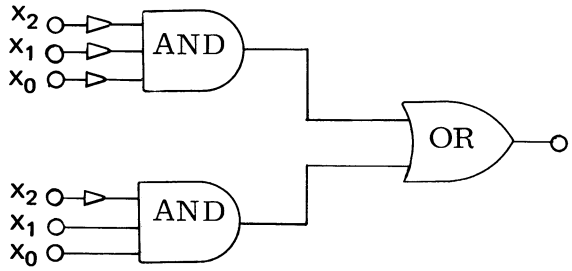
WRITTEN BY TYMSHARE

:

Conversational Terminal Input

```
PLEASE LOG IN: @7;RJT;

TYMSHARE  11/16  10:43


-LOGMIN


:RUN
INPUT FROM: T                    Note: Input from terminal can
                                       be T, TE, or TEL
ENTER NO. OF VARIABLES: 3
MAP POSITION: 0
VALUE: 1
MAP POSITION: 3
VALUE: 1
MAP POSITION: 5
VALUE: 2
MAP POSITION: E          Note: Ending map input can be
                               E, EN, or END

REVISONS? N
LIST INPUT DATA? Y       Note: Answers to questions can be
                               Y, YE, or YES
                               N, or NO

  3 VARIABLES

   MAP
POSITION   VALUE

    0        1
    3        1
    5        2

COMPUTE & PRINT PRODUCT TERMS? Y


-X2  -X1  -X0

-X2   X1   X0


WRITE INPUT DATA TO A FILE? Y
OUTPUT TO: JOB1          ------>  Note: Output to a file can be
OK                                      aborted by entering the
:QUIT                                   word NOTHING.
-LOG
```

Resultant Minimized Circuit



EXPERT Terminal Input

```
-LOGMIN

:EXPERT

OK

:RUN

INPUT FROM: TEL

N: 3
MP: 1
V: 1
MP: 2
V: 1
MP: 6
V: 2
MP: END

REVISONS? N
LIST INPUT DATA? Y

  3 VARIABLES

   MAP
POSITION  VALUE

    1       1
    2       1
    6       2
```

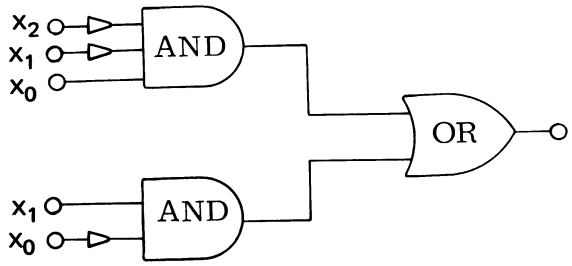
NOTE: EXPERT mode has reduced printout for map position and value. Once the EXPERT mode is set, it can only be reset to the normal mode by starting the program with LOGMIN from EXEC.

cont'ed next page

EXPERT Terminal Input (cont'ed)

```
COMPUTE & PRINT PRODUCT TERMS? Y

-X2  -X1   X0

 X1  -X0

WRITE INPUT DATA TO A FILE? Y

OUTPUT TO: /JOB2/

OK


:QUIT

-LOG
```

Resultant Minimized Circuit



Conversational File Input

```
-LOGMIN

:RUN

INPUT FROM: JOB1

REVISONS? NO
LIST INPUT DATA? YES

  3 VARIABLES

   MAP
POSITION  VALUE

    0      1
    3      1
    5      2

COMPUTE & PRINT PRODUCT TERMS? YES

-X2  -X1  -X0

-X2   X1   X0

WRITE INPUT DATA TO A FILE? NO

:QUIT

-LOG
```

This was the file created during the Conversational Termal Input.

Direct Command Mode (for advanced LOGMIN users)

INPUT & LIST Commands

```
*LOGMIN

:INPUT TEL

ENTER NO. OF VARIABLES: 4
MAP POSITION: 2
VALUE: 2
MAP POSITION: 5
VALUE: 1
MAP POSITION: END


:LIST


  4 VARIABLES

   MAP
POSITION  VALUE

    2       2
    5       1
```

If the file name is not entered directly behind the command, the response will be;

INPUT FROM:

cont'ed next page

```
:INPUT /JOB2/
```

```
:LIST
```

Note loading a second set of input data deletes the original input data.

```
  3 VARIABLES

   MAP
POSITION   VALUE

    1        1
    2        1
    6        2
```

```
:QUIT
```

```
-LOG
```

```
*LOGMIN

:INPUT /JOB1/



:LIST

  3 VARIABLES

   MAP
POSITION   VALUE

    0        1
    3        1
    5        2


:REVISIONS

MAP POSITION: 3
VALUE: 0
MAP POSITION: END
```

Setting the value of a map position to zero eliminates it from the listing as seen on the next page.

cont'ed next page

```
:LIST

  3 VARIABLES

   MAP
POSITION   VALUE

    0        1
    5        2


:WRITE /JOB3/

OK

:WRITE

OUTPUT TO: /JOB3/

OK

:QUIT

-LOG
```

NOTE: When the WRITE command is given followed by a carriage return, a file name is required. If the user writes to the same file twice, the first contents are destroyed; only the second contents remain in the file.

COMPUTE Command

```
*LOGMIN

:INPUT /JOB1/


:LIST

  3 VARIABLES

   MAP
POSITION   VALUE

    0        1
    3        1
    5        2


:COMPUTE


-X2   -X1   -X0

-X2    X1    X0
```

cont'ed next page

COMPUTE Command (cont'ed)

```
:INPUT /JOB3/
```

The user wants to compute answers for the data in a second file after the first is completed.

```
:LIST

  3 VARIABLES

   MAP
POSITION  VALUE

    0       1
    5       2


:COMPUTE


-X2  -X1  -X0


:QUIT

-LOG
```

IV. EXAMPLES

The following examples are presented to show the capabilities of the program as well as show some representive problems.

1. Input data from a file with RUN command.
2. Computing a problem with 11 variables.
3. Computing a logical chain case.
4. Check of logical chain equation with the LOGIC program.

Example 1: Input data from a file with RUN command.

```
*LOGMIN

:RUN

INPUT FROM: LOGICD


REVISONS? NO
LIST INPUT DATA? YES


  6 VARIABLES

   MAP
POSITION  VALUE

    0      1
    3      2
    5      2
    6      1
    7      2
    8      1
    9      2
   10      1
   13      1
   14      1
   15      1
   16      2
   17      1
   19      1
   20      1
   21      1
   22      2
   23      2
   24      1
   25      1
   26      1
   28      2
   31      1
   34      1
   39      1
   40      2
```

cont'ed next page

Example 1: (cont'ed)

```
41      2
43      1
44      2
45      2
46      1
48      2
49      2
52      1
53      1
55      2
57      2
58      1
61      1
63      2
```

COMPUTE & PRINT PRODUCT TERMS? YES

```
 X5  -X4  -X3  -X2   X1  -X0
 X5  -X4   X3  -X2   X0
 X4   X3  -X2   X1  -X0
-X5  -X2  -X1  -X0
-X3   X2   X1   X0
-X4   X3   X2   X1  -X0
-X5   X3  -X2  -X0
-X5   X4  -X3   X0
-X5   X2   X1   X0
-X5  -X3   X2   X1
 X4  -X3  -X1
-X4   X3  -X1   X0
 X3  -X2  -X1   X0
 X5   X3  -X1   X0
```

NOTE: For 23 specifications of required logical true product terms, only 14 product terms are required in the minimized Boolean equation.

cont'ed next page

Example 1: (cont'ed)

```
WRITE INPUT DATA TO A FILE? NO

:QUIT

-LOG
```

Example 2: Computing a problem with 11 variables

```
-LOGMIN

:INPUT TEL

ENTER NO. OF VARIABLES: 11
MAP POSITION: 0
VALUE: 1
MAP POSITION: 100
VALUE: 1
MAP POSITION: 300
VALUE: 1
MAP POSITION: 400
VALUE: 2
MAP POSITION: 600
VALUE: 1
MAP POSITION: 902
VALUE: 2
MAP POSITION: 1500
VALUE: 1
MAP POSITION: 1501
VALUE: 2
MAP POSITION: 2046
VALUE: 1
MAP POSITION: 2047
VALUE: 1
MAP POSITION: END
```

cont'ed next page

Example 2: (cont'ed)

```
:LIST

  11 VARIABLES

   MAP
POSITION  VALUE

    0       1
  100       1
  300       1
  400       2
  600       1
  902       2
 1500       1
 1501       2
 2046       1
 2047       1

:COMPUTE

-X10  -X9  -X8  -X7  -X6  -X5  -X4  -X3  -X2  -X1  -X0

-X10  -X9  -X8  -X7   X6   X5  -X4  -X3   X2  -X1  -X0

-X10  -X9   X8  -X7  -X6   X5  -X4   X3   X2  -X1  -X0

-X10   X9  -X8  -X7   X6  -X5   X4   X3  -X2  -X1  -X0

 X10  -X9   X8   X7   X6  -X5   X4   X3   X2  -X1

 X10   X9   X8   X7   X6   X5   X4   X3   X2   X1

:QUIT

-LOG
```

Example 3: Computing a logical chain case.

```
*LOGMIN

:EXPERT

OK

:INPUT TEL

N: 3
MP: 1
V: 1
MP: 2
V: 1
MP: 3
V: 1
MP: 4
V: 1
MP: 5
V: 1
MP: 6
V: 1
MP: E


:LIST


  3 VARIABLES

   MAP
POSITION  VALUE

    1       1
    2       1
    3       1
    4       1
    5       1
    6       1
```

cont'ed next page

Example 3: (cont'ed)

:**COMPUTE**

-X1 X0

-X2 X1

X2 -X0

NOTE: If the Boolean equation was written directly from the map, it would have six product terms of three variables in each term. Here the equation is just three product terms with only two variables each.

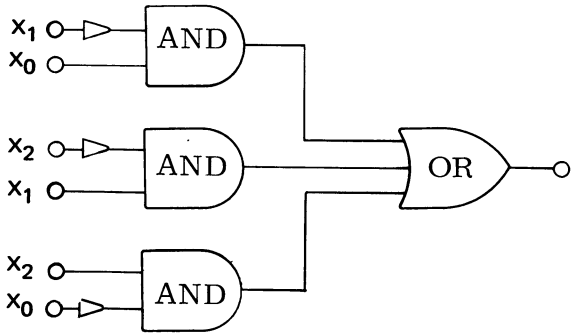
:**WRITE /LOGICAL CHAIN/**

OK

:**QUIT**

-**LOG**

Minimized Circuit



Example 4: Check of logical chain equation with the LOGIC program.

From example 3. (Page 21)

From LOGMIN		For LOGIC
X0	=	I1
X1	=	I2
X2	=	I3

NOTE:

-X0	=	(NOT I1)
-X1-	=	(NOT I2)
-X2	=	(NOT I3)

```
PLEASE LOG IN: A3;PO;

TYMSHARE  2/3  11:33
-#LOGIC

NEED INSTRUCTIONS (YES OR NO)? NO
AFTER ENTERING EQUATIONS TYPE 'RUN'

>100 A= (NOT I2) AND I1
>110 B= (NOT I3) AND I2
>120 C= I3 AND (NOT I1)
>130 A1= A OR B OR C
>RUN
HOW MANY INPUTS ? 3
HOW MANY OUTPUTS ? 1
COMPLETE TRUTH TABLE ? YES

  INPUT        OUTPUT

I1  I2  I3      A1        MP
--  --  --      --
 0   0   0       0         0
 1   0   0       1         1
 0   1   0       1         2
 1   1   0       1         3
 0   0   1       1         4
 1   0   1       1         5
 0   1   1       1         6
 1   1   1       0         7

-LOG
```

If you have not used this program before, type YES.

Note only map positions (MP) 1-6 are true. This proves that indeed this circuit creates the map as defined on the bottom of page 21.

V. ERROR MESSAGES

The following error messages may occur during input from a file or the teletype:

XXX NO. OF VARIABLES NOT NUMERIC

The first item in the input file is not a number. In this case it was the letters XXX. Reading of the input file will stop because of this error.

70 MAP POSITION TOO LARGE. PLEASE RE-ENTER

If n is the number of variables, then the map position may not exceed 2^n-1. In this case the number of variables, which is the first data item on the input file, must be 7 or larger for a map position of 70 to exist.

20, 4 LOGICAL VALUE CAN ONLY BE 0, 1 OR 2. PLEASE RE-ENTER

The line 20, 4 was encountered in the input file. The value 4 is an illegal logical value. It may only be a numeric 0, 1, or 2.

NUMBER OF VARIABLES EXCEEDS 11

The number of variables, the first item in the input file may not exceed 11.

A check is made of the map after input is complete for the following three cases and the error message is printed as indicated.

ALL MAP POSITIONS FALSE AND/OR DON'T CARE

All map locations logical false or some locations false with the remainder don't care cases.

ALL MAP POSITIONS TRUE AND/OR DON'T CARE

All map locations logical true or some locations true with the remainder don't care cases.

ALL MAP POSITIONS DON'T CARE CASES

All map locations are don't care cases.

VI. REFERENCES

1. H.C. Torng, Introduction to the Logical Design of Switching Systems, Addison-Weseley Publishing Co., Inc., Reading, Mass., 1964

2. W.S. Humphery, Jr., Switching Circuits with Computer Applications, McGraw-Hill Book Company, Inc., New York, 1968

3. F.E. Hohn & D.E. Muller, Applied Boolean Algebra, The MacMillian Co., New York, 1960

4. Binay Logic, Reprint from Product Engineering, R101, McGraw-Hill Book Company, Inc., New York, 1963

5. Logic Handbook, Digital Equipment Corp., Maynard, Mass. 1969

TYMSHARE, INC.

10340 BUBB ROAD
CUPERTINO, CALIFORNIA 95014