DECUS
PROGRAM LIBRARY

DECUS NO.	8-648
TITLE	LOGMIN - LOGIC MINIMIZATION PROGRAM
AUTHOR	David Wu
COMPANY	Princeton University Princeton, New Jersey
DATE	June 30, 1973
SOURCE LANGUAGE	PAL-8

LOGMIN - LOGIC MINIMIZATION PROGRAM

DECUS Program Library Write-up

DECUS NO. 8-648

Abstract

LOGMIN is useful to the logic designer for determining or checking the two-level minimized representation of a logic function, given that function in its sum-of-products or product-of-sums form. The function need not be in its canonical representation.

Requirements

LOGMIN operates within an environment that contains a 4K PDP-8 and an ASR33 teletype as a subset. LOGMIN resides in locations 20 through 2000 and uses the rest of core as working area. If available core is exhausted LOGMIN will issue

INSUFFICIENT MEMORY. EXECUTION TERMINATED.

and halt. Press CONTINUE to restart program execution.

Restriction

The way LOGMIN handles data imposes no restriction on the number of variables a logic function to be minimized may have. However, because the ASR33 teletype has a maximum width of 72 columns, the author has limited the number of variables a logic function may have to a maximum of 72, which is quite adequate for most applications.

Please direct any suggestions and comments to
David Wu '74
Electrical Engineering Department
Princeton University
Princeton New Jersey 08540

Application Procedure

1. LOGMIN starts at location 200 and begins by typing the following message,

ENTER NUMBER OF VARIABLES :

The proper response to this should be a 1 to 2 digit decimal number followed by a carriage return. Any input character other than carriage return, rubout and numerals are ignored. LOGMIN will restart if any of the following conditions prevails :

a) a carriage return is issued before any numeral was entered,

b) the value of the decimal number is 0 or > 72,

c) a 3 or more digit number was entered,

d) a rubout is issued.

LOGMIN must have a valid response to this question before it will continue.

2. After a legal response has been made to the above message LOGMIN enters the first stage of the input phase and issues the following,

ENTER ESSENTIAL 'MIN/MAXTERMS'

The legal input characters during the input phase are rubout, $, -, 1 and 0; all other characters are ignored.

The user response to the above message for a 4-variable logic function

$$f(a,b,c,d) = ab\bar{c}d + bc + \bar{a}d$$

should be

```
1 1 0 1      [ abc̄d ]
- 1 1 -      [  bc  ]
0 - - 1      [  ād  ]
```

Each legal character is echoed and if the number of variables is less than 37 then a space will separate them so that the printout will be esthetically more pleasing.

If an erroneous entry is made, it can be deleted with a rubout and a back-arrow is typed pointing to that entry to indicate that it has been deleted. A rubout can be issued in the middle of an entry to delete the un-finished entry. When an entry is full, that is, when the total number of 1, 0 and - entered equals the number of variables, any keyboard

input will cause LOGMIN to issue a carriage return and a line feed, and ready itself to accept the next entry. If the keyboard input is a legal character, then it is accepted by LOGMIN and echoed after a carriage return and a line feed.

The dollar sign is used to indicate that all entries representing the function to be minimized have been made and LOGMIN will proceed to the second stage of the input phase. If the dollar sign is issued in the middle of an entry, that entry is deleted and LOGMIN goes on to input the don't care conditions. If the dollar sign is issued before any legal entry has been made LOGMIN restarts itself, since there is nothing to be minimized.

Since LOGMIN does not buffer I/O, fast keyboard input will often cause a rubout to be read in by mistake.

The reason MIN/MAXTERMS is in quotes is that LOGMIN does not require the function to be minimized to be in the canonical representation, and thus provides the user with an extra degree of freedom by saving him the trouble of manually reducing the function to the canonical representation.

3. LOGMIN types

ENTER DCN'T CARE 'MIN/MAXTERMS'

and the proper response is the same as described in 2.

4. LOGMIN now enters the output phase and types

ESSENTIAL PRIME IMPLICANTS

and outputs the two-level absolutely minimized logic function. The output has the same format of representation as is described in 2. LOGMIN then restarts program execution and is ready to minimize another logic function.

Follows are a truth table of a 4-variable logic function and the corresponding input/output of LOGMIN. The worked example should be self-explanatory.

p	q	r	s	f(p,q,r,s)	
0	0	0	0	d	[don't care condition]
0	0	0	1	0	
0	0	1	0	d	[don't care condition]
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	1	
1	0	0	1	0	
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	1	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	

$$f(p,q,r,s) = qs + p\bar{r}\bar{s} + \bar{q}r$$

```
ENTER NUMBER OF VARIABLES : 4

ENTER ESSENTIAL 'MIN/MAXTERMS'
0 0 1 1
0 1 0 1
0 1 1 1
1 0 0 0
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 1
$

ENTER DON'T CARE 'MIN/MAXTERMS'
0 0 0 0
0 0 1 0
$

ESSENTIAL PRIME IMPLICANTS
- 1 - 1
1 - 0 0
- 0 1 -
```

This manuscript was printed with the PRINCETON PRINT CHAIN and prepared with the automatic text editing program ROFF version 3.0 under OS/360.