Table of Contents

Name

espresso - Boolean Minimization

Synopsis

espresso [*options*] [*file*]

Description

Espresso takes as input a two-level representation of a two-valued (or multiple-valued) Boolean function, and produces a minimal equivalent representation. The algorithms used are new and represent an advance in both speed and optimality of solution in heuristic Boolean minimization.

Espresso reads the *file* provided (or standard input if no files are specified), performs the minimization, and writes the minimized result to standard output. *Espresso* automatically verifies that the minimized function is equivalent to the original function. Options allow for using an exact minimization algorithm, for choosing an optimal phase assignment for the output functions, and for choosing an optimal assignment of the inputs to input decoders.

The default input and output file formats are compatible with the Berkeley standard format for the physical description of a PLA . The input format is described in detail in espresso(5) . Note that the input file is a *logical* representation of a set of Boolean equations, and hence the input format differs slightly from that described in pla(5) (which provides for the *physical* representation of a PLA). The input and output formats have been expanded to allow for multiple-valued logic functions, and to allow for the specification of the don’t-care set which will be used in the minimization.

A complete list of the command line options is given below. Be warned that many of the command line options are not intended for general use.

-d
: Enables debugging. Useful only for those familiar with the algorithms used.

-Dcheck
: Checks that the function is a partition of the entire space (i.e., that the ON -set, OFF -set and DC -set are pairwise disjoint, and that their union is the Universe).

-Dd1merge
: Performs a quick distance-1 merge on the input file. This is useful when the input file is very large (e.g., a truth table with more than 1000 terms) because distance-1 merge is O(n log n) rather than the EXPAND step of Espresso which is O(n * n). The output should then be run through Espresso to complete the minimization. A range of variables to be merged can also be specified using **-rn-m** (the default is to merge over all variables).

-Decho
: Echoes the function to standard output. This can be used to get the complement of a function when combined with **-o** .

-Dequiv
: Identify output variables which are equivalent. Takes into account the don’t-care set and checks for equivalence of both the ON -set and OFF -set.

-Dexact
: Exact minimization algorithm (guarantees minimum number of product terms, and heuristically minimizes number of literals). Potentially expensive.

-Dmany
: Reads and minimizes PLA’s until end-of-file is detected. PLA’s in the same file are separated by *.e* .

-Dmap
: Draw the Karnaugh maps for a binary-valued function.

-Dmapdc
: Derive from the binary-valued variable *DONT_CARE* a don’t-care set, and then delete this variable. All input conditions for which an output changes when *DONT_CARE* changes define the don’t-care conditions for that output. This is a hack to support don’t-cares from high-level languages without a notion of don’t-cares.

-Dopo

Perform output phase optimization (i.e., determine which functions to complement to reduce the number of terms needed to implement the function). After choosing an assignment of phases for the outputs, the function is minimized. A simple algorithm is used which may become very expensive for a large number of outputs (e.g., more than 40).

-Dopoall
Minimize the function with all possible phase assignments. A range of outputs to cycle through can be given with **-rn-m** (the default is to use all outputs). The option **-S1** will perform an exact minimization for each phase assignment. Be warned that opoall requires an exponential number of minimizations !

-Dpair
Choose an assignment of the inputs to two-bit decoders, and minimize the function. The function MUST be minimized first to achieve good results. There are actually 4 different algorithms, of increasing cost, which may be selected with **-S1**, **-S2**, or **-S3**. The default is **-S0** which seems to give the best results for the cost.

-Dpairall
Minimize the function with all possible assignments of inputs to two-bit decoders. The option **-S1** will perform an exact minimization for each assignment of inputs to decoders, and the option **-S2** will perform an output-phase assignment for each assignment of inputs to decoders. Be warned that pairall requires an exponential number of minimizations !

-Dseparate
Remove the don't-care set from the ON -set of the function.

-Dso
Minimize each function one at a time as a single-output function. Terms will not be shared among the functions. The option **-S1** will perform an exact minimization for each single-output function.

-Dso_both
Minimize each function one at a time as a single-output function, but choose the function or its complement based on which has fewer terms. The option **-S1** will perform an exact minimization for each single-output function and its complement to determine which has fewer terms.

-Dstats
Provide simple statistics on the size of the function.

-Dverify
Checks for Boolean equivalence of two PLA's. Reads two filenames from the command line, each containing a single PLA.

-DPLAverify
Checks for Boolean equivalence of two PLA's by first permuting the columns based on the user supplied variable names. Reads two filenames from the command line.

-eeat
Normally comments are echoed from the input file to the output file. This options discards any comments in the input file.

-efast
Stop after the first EXPAND and IRREDUNDANT operations (i.e., do not iterate over the solution).

-ekiss
Sets up a *kiss* -style minimization problem. This is a hack.

-eness
Essential primes will not be detected.

-enirr
The result will not necessarily be made irredundant in the final step which removes redundant literals.

-enunwrap
The ON -set will not be unwrapped before beginning the minimization.

-eonset
Recompute the ON -set before the minimization. Useful when the PLA has a large number of product terms (e.g., an exhaustive list of minterms).

-epos
Swaps the ON -set and OFF -set of the function after reading the function. This can be used to minimize the OFF -set of a function. *.phase* (see espresso(5)) in the input file can also specify an arbitrary choice of output phases.

-estrong
Uses the alternate strategy SUPER_GASP (as a replacement for LAST_ GASP) which is more expensive, but occasionally provides better results.

-o[type]
Selects the output format. By default, only the ON -set (i.e., type f) is output after the minimization. [type] can be one of **f** , **d** , **r** , **fd** , **dr** , **fr** , or **fdr** to select any combination of the ON -set (f), the OFF -set (r) or the DC -set (d). [type] may also be **eqntott** to output algebraic equations acceptable to *eqntott*(1OCTTOOLS) , or **pleasure** to output an unmerged PLA (with the *.label* and *.group* keywords) acceptable to

pleasure(1OCTTOOLS) .

-s
: Will provide a short summary of the execution of the program including the initial cost of the function, the final cost, and the computer resources used.

-t
: Will produce a trace showing the execution of the program. After each main step of the algorithm, a single line is printed which reports the processor time used, and the current cost of the function.

-x
: Suppress printing of the solution.

-v [type]
: Specifies verbose debugging detail. Not generally useful.

Diagnostics

Espresso will issue a warning message if a product term spans more than one line. Usually this is an indication that the number of inputs or outputs of the function is specified incorrectly.

See Also

kiss(1OCTTOOLS) , pleasure(1OCTTOOLS) , pla(5OCTTOOLS) , espresso(5OCTTOOLS) , nova(5OCTTOOLS)

R. Brayton, G. Hachtel, C. McMullen, and A. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis* , Kluwer Academic Publishers, 1984.

R. Rudell, A. Sangiovanni-Vincentelli, "Espresso-MV: Algorithms for Multiple-Valued Logic Minimization," *Proc. Cust. Int. Circ. Conf.* , Portland, May 1985.

R. Rudell, "Multiple-Valued Minimization for PLA Synthesis," Master’s Report, University of California, Berkeley, June 1986.

R. Rudell, A. Sangiovanni-Vincentelli, "Exact Minimization of Multiple-Valued Functions for PLA Optimization", *Int. Conf. Comp. Aid. Des.* , Santa Clara, November 1986.

Author

Please direct any questions or comments to:

```
Richard Rudell
 205 Cory Hall
 Dept. of EECS
 University of California
 Berkeley, California  94720
```

Arpanet mail address is rudell@ic.Berkeley.EDU.

Comments

Default is to pass comments and unrecognized options from the input file to standard output (sometimes this isn’t what you want).

It is no longer possible to specify the type on the command line.

There are a lot of options, but typical use doesn’t need them.

This manual page refers to Version 2.3 of Espresso. The major change from Version 2.2 to Version 2.3 is the addition of a fast sparse-matrix covering algorithm for the **-Dexact** mode.

The -Dopo option becomes very slow for many outputs (> 20).

Table of Contents

- Name
- Synopsis
- Description
- Diagnostics
- See Also
- Author
- Comments