

User Commands

ESPRESSO(1)

NAME

espresso - Boolean Minimization

SYNOPSIS

espresso [type] [file] [options]

DESCRIPTION

Espresso takes as input a two-level representation of a two-valued (or a 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.

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.

Type specifies the logical format for the function. The allowed types are -f, -r, -fr, -fd, -dr, and -fdr which have the same meanings assigned in espresso(5).

The command line options described below can be specified anywhere on the command line and must be separated by spaces:

- d        Verbose detail describing the progress of the minimization is written to standard output. Useful only for those familiar with the algorithms used.
- do [s] This option executes subprogram [s]. Some of the more useful ones are:
  - check - 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)
  - echo - implies "-out fdr" and echoes the function to standard output. This can be used to compute the complement of a function.
  - map - draw the Karnaugh maps for a function.
  - opo - choose a good assignment of output function phases, and then minimize the function. The chosen phase is reported in the output file.
  - qm - generate all prime implicants of a function, compute the reduced prime implicant table, and perform a heuristic covering of this table. Provides a

bound on the size of the minimum solution. Will perform an exact minimization if option `-exact` is used.

`single_output` - Minimize each function one at a time as a single-output function. Terms will not be shared among the functions.

`stats` - provide simple statistics on the size of the function

The remaining subprograms (`absmin`, `allphase`, `badopo`, `compact`, `contain`, `dlmerge`, `dcset`, `essen`, `expand`, `intersect`, `irred`, `lexsort`, `make_sparse`, `mincov`, `minterms`, `npc`, `opotol`, `pair`, `pairsasao`, `pop`, `primes`, `reduce`, `sharp`, `taut`, `union`, `unravel`, `verify`) are intended for those heavily into manipulating Boolean functions.

- exact Use an exact algorithm to solve the covering problems encountered during the execution of the algorithms. Not recommended for casual users.
- fast Stop after the first EXPAND and IRREDUNDANT operations (i.e., do not iterate over the solution).
- kiss Sets up a kiss-style minimization problem.
- ness Essential primes will not be detected and removed from the minimization.
- nirr The final result will not necessarily be forced irredundant.
- help Provides a quick summary of the available command line options.
- onset 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).
- out [s] Selects the output format. By default, only the ON-set (i.e., type f) is output after the minimization. [s] 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). [s] may also be `eqntott` to output algebraic equations acceptable to `eqntott(1)`, or `pleasure` to output an unmerged PLA (with the `.label` and `.group` keywords) acceptable to `pleasure(1)`.
- pos 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` in the input file can also specify an arbitrary choice of output phases.
- 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.

#### 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(1), pleasure(1), pla(5), espresso(5)

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

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

#### AUTHOR

Richard Rudell

#### BUGS

Always passes comments from the input file, and passes unrecognized options straight from the input file to standard output (sometimes this isn't what you want).

There are a lot of options, but the most typical use is the following:

```
eqntott -r file.eqn | espresso >file.pla
```

The -R option of eqntott should not be used (it is much too expensive).