

ESPRESSO TOOL GUIDE

Input format is PLA format. An example is shown below:

```
.i 4
.o 1
.ilb a b c d
.olb y
.p 9
0000 1
0001 1
0100 1
0101 1
0111 1
1000 1
1001 1
1100 1
1111 1
.dc
.e
```

To run Espresso, just type the command espresso and the file name. For example, suppose that the above PLA is stored in a file named example.pla. Then, we can perform two-level minimization by running the following command:

```
Espresso example.pla
```

The output will appear in the screen as shown below:

```
.olb y
.dc
.i 4
.o 1
.ilb a b c d
.p 4
0-0- 1
-111 1
--00 1
-00- 1
.e
```

You can direct the output to a file using the command:

```
espresso example.pla > example_min.pla
```

To print all the prime implicants of a function use the following command:

```
espresso -Dprimes example.pla
```

To run expand only, use the following command:

```
espresso -Dexpand example.pla
```

To run reduce only, use the following command:

```
espresso -Dreduce example.pla
```

To run irredundant only, use the following command:

```
espresso -Dirred example.pla
```

To run exact two-level minimization, use the following command:

```
espresso -Dexact example.pla
```

To run single-output two-level minimization, use the following command:

```
espresso -Dsingle_output example.pla
```

To check if a function is tautology, use the following command:

```
espresso -Dtaut example.pla
```

A summary of ESPRESSO commands is shown below:

UC Berkeley, Espresso Version #2.3, Release date 01/31/88

SYNOPSIS: espresso [options] [file]

```
-d          Enable debugging
-e[opt]    Select espresso option:
           fast, ness, nirr, nunwrap, onset, pos,
strong,
           eat, eatdots, kiss, random
-o[type]   Select output format:
           f, fd, fr, fdr, pleasure, eqntott, kiss,
cons
-rn-m      Select range for subcommands:
           dlmerge: first and last variables (0 ... m-
1)
           minterms: first and last variables (0 ...
m-1)
           opoall: first and last outputs (0 ... m-1)
-s         Provide short execution summary
-t         Provide longer execution trace
-x         Suppress printing of solution
-v[type]   Verbose debugging detail (-v '' for all)
-D[cmd]    Execute subcommand 'cmd':
           ESPRESSO, many, exact, qm, single_output,
so, so_both,
           simplify, echo, opo, opoall, pair, pairall,
check, stats,
           verify, PLVerify, equiv, map, mapdc, fsm,
contain, dlmerge,
           dlmerge_in, disjoint, dsharp, intersect,
minterms, primes,
           separate, sharp, union, xor, essen, expand,
gasp, irred,
           make_sparse, reduce, taut, super_gasp,
lexsort, test
-Sn        Select strategy for subcommands:
           opo: bit2=exact bit1=repeated bit0=skip
sparse
           opoall: 0=minimize, 1=exact
```

3=exact

pair: 0=algebraic, 1=strongd, 2=espresso,

pairall: 0=minimize, 1=exact, 2=opo

so_espresso: 0=minimize, 1=exact

so_both: 0=minimize, 1=exact