

Efficient and Effective Redundancy Removal for Million-Gate Circuits

Michel Berkelaar
Magma Design Automation
michel@magma-da.com

Koen van Eijk
koen@magma-da.com

1 Extended Abstract

Redundancy removal of combinational circuits has been the subject of many papers over the last decades. Most of these papers work with the relatively small circuits available as benchmarks in the logic synthesis community. In Magma's BlastFusion and BlastChip software, very large blocks of logic (millions of gates) are handled flat (BlastFusion and BlastChip are registered trademarks of Magma Design Automation). We implemented redundancy removal in a way that will allow it to run efficiently (fast, low memory usage) and robustly (no run time or memory explosion on *any* netlist) on industrial designs of up to several million gates. We achieve this without resorting to partitioning.

Other than most published approaches we do not try to indentify all redundancies in a circuit, as an exact solution to this NP-hard problem is infeasible for the large circuits we face. Instead we try to identify as many as possible in a reasonable run time.

We use a carefully engineered combination of Fault Collapsing, Random Test Generation (RTG) and the good old

D-algorithm. As the goal is finding redundancies, and not sets of test vectors, these algorithms need changes and adaptations for optimal efficiency and robustness. Fault Collapsing can be more aggressive than for test generation. RTG was implemented with a novel dynamic control of the bit-parallelism employed. The D-algorithm's effort control was not implemented with a traditional backtrack limit, but on a more fine-grain level, to increase robustness. For details, please refer to [1].

Results on 11 industrial netlists are shown in table 1. All tests were run on a Sun Ultra-80 workstation. A comparison is shown to a state-of-the-art SAT-based approach. Our approach is clearly faster while identifying more redundancies.

References

- [1] Michel Berkelaar and Koen van Eijk "Efficient and Effective Redundancy Removal for Million-Gate Circuits", *Proc. International Workshop on Logic Synthesis 2001*, pp. 249ff.

Table 1. redundancy removal results

circuit	nodes	after fault collapsing faults	after RTG		redundancies found / CPU time			
			faults	CPU	SAT approach	low effort	med. effort	high effort
ind1	15087	18969	5595	6.0s	172 / 25.9m	208 / 34.8s	266 / 1.3m	268 / 4.5m
ind2	67538	96006	15199	37.0s	432 / 43.7m	553 / 1.5m	774 / 3.5m	780 / 11.0m
ind3	89357	100656	12082	1.2m	188 / 30.0m	167 / 1.8m	366 / 3.7m	432 / 5.5m
ind4	126607	178234	24542	1.7m	2794 / 53.0m	2625 / 4.8m	4625 / 9.2m	4931 / 47.5m
ind5	134321	187829	10834	56.0s	899 / 30.3m	990 / 2.0m	1183 / 2.7m	1226 / 5.5m
ind6	147437	136115	18102	43.0s	783 / 42.5m	2581 / 5.7m	2811 / 15.1m	2970 / 28.0m
ind7	186185	204705	10430	1.5m	987 / 13.8m	914 / 2.3m	1039 / 2.9m	1068 / 5.2m
ind8	219529	312975	33932	1.8m	1078 / 1.0h	1015 / 5.0m	1417 / 10.3m	1563 / 33.6m
ind9	221018	221799	48577	43.0s	1205 / 2.9h	2440 / 2.7m	3672 / 13.3m	3825 / 54.2m
ind10	325262	479569	69311	5.6m	2577 / 1.8h	13005 / 15.4m	14966 / 29.1m	17047 / 1.4h
ind11	956851	1036689	83798	7.3m	3887 / 10.4h	1783 / 12.4m	4572 / 32.5m	5422 / 3.4h