

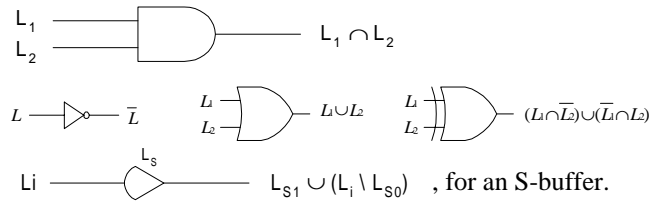
Solving. To solve the diagnosis problem and differentiate diagnoses $D=\{d_1, d_2\}$, either $\{d_1\}-N$ or $\{d_2\}-N$ must be present in a circuit output bit (or, equivalently, $L-N$, where $\#L=1$). For problem number 2 (maximisation), the goal is $maximise(\#\cup_b L_b)$ where b ranges over each circuit output bit b with signal L_b-N_b . The third problem (minimisation) is a typical set covering problem.

2.1 Improved Representation with Sets

With the previous representation, both the set and the Boolean value can be variables, which makes the representation hard to handle. Instead, we can represent $L-0$ simply as L , and $L-1$ as \bar{L} (the complement of L , with respect to D), by means of transformation *transf*:

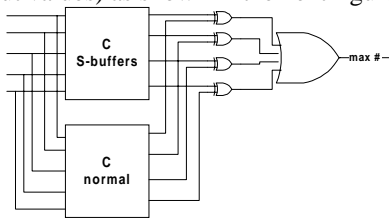
$$transf(S) = \begin{cases} L, & S = L-0 \\ \bar{L}, & S = L-1 \end{cases}$$

As such, gates are modelled with the usual set operations:



Solving. To solve the diagnosis problem using just sets, we can still simply ensure that a set L with cardinality 1 is present at a circuit output bit. With the set $D=\{d_1, d_2\}$ of diagnoses to differentiate, it is equivalent to have an L ($\#L=1$) as an output bit in the sets representation, or to have an $L-N$ ($\#L=1$) in the previous representation.

For the maximisation problem, we must know exactly if an output signal depends on its set or on its complement. This is done by recovering the lost information (the normal output values) as shown in the next figure.



Circuit c with S-buffers is kept, and we add the same circuit but with all lines normal (i.e. with no S-buffers). Circuits share the inputs. The xor-gates in the output bits receive a set L from the faulty circuit and either \emptyset or D (the universe) from the normal one. Therefore, L is kept as L if the normal value was 0, and recovered to \bar{L} if the normal value was 1. A maximisation on the union of these real dependencies can now be performed to reach our goal.

Since we now only have set constraints, these can be actively used by a set constraint solver and choice-points

are avoided. Also, labelling (the exponential component of search) is only performed at the circuit with S-buffers, in contrast with Boolean SAT approaches, which consider one extra circuit for each diagnosis, which is unacceptable, in practice, for a large set of diagnoses [2].

The minimisation problem is a meta-problem: it involves sets of solutions to set problems and is still an open problem.

3. Conclusions and Further Research

We have shown how a constraint programming approach is able to model a number of ATPG related problems. Clearly, the practical interest of this work depends on the ability to develop adequate constraint solvers to deal with the domains that have been used (8-valued logic and sets [6]). In both cases we have been developing constraint solvers over existing CLP languages (SICStus and ECLiPse) and checking our systems with the ISCAS standard circuits. We expect to be able to prove that the results are competitive with SAT based approaches (as shown in [7] in the pure ATPG problem), at least in some types of problems. Meanwhile, we reckon that the expressive power and flexibility of the constraint programming approach makes it very attractive and deserves further research work.

Acknowledgement. The first author was financially supported by "Sub-Programa Ciência e Tecnologia do 2º Quadro Comunitário de Apoio".

References

- 1 H. Simonis. *Constraint Logic Programming Language as a Digital Circuit Design Tool*, Thesis, 1992.
- 2 L. G. Silva, L. M. Silveira and J. P. Marques-Silva, *Algorithms for Solving Boolean Satisfiability in Combinational Circuits*, IEEE/ACM Design and Test in Europe Conf. (DATE), 1999.
- 3 T. Gruning, U. Mahlstedt, H. Koopmeiners, *DIATEST: A Fast Diagnostic Test Pattern Generator for Combinational Circuits*, Procs of ICCAD91, 194-197, 1991
- 4 I. Pomeranz, S.M. Reddy, *A Diagnostic Test Generation Procedure for Synchronous Sequential Circuits based on Test Elimination*, Procs of ITC98, 1074-1083, 1998.
- 5 F. Azevedo and P. Barahona. *Generation of Test Patterns for Differential Diagnosis of Digital Circuits (Extended Abstract)*, in CP'98, M. Maher and J.-F. Puget (Eds.), Springer, p. 462, 1998. Long version: ERCIM/COMPULOG Workshop on Constraints, K. Apt, P. Codognet, E. Monfroy (Eds.), 1998.
- 6 C. Gervet, *Interval Propagation to Reason about Sets: Definition and Implementation of a Practical Language*, Constraints, vol. 1 (3), Kluwer Academic Pub, 191-244, 1997.
- 7 H. Simonis. *Test Generation using the Constraint Logic Programming Language CHIP*, 6th Int. Conf. on Logic Programming, MIT Press, 101-112, 1989.