A Constraint Programming Approach to Model ATPG Related Problems

Francisco Azevedo & Pedro Barahona
{fa,pb}@di.fct.unl.pt
New University of Lisbon

Introduction

- **Scope**: Combinatorial problems in Electronics Computer Aided Design (ECAD), namely in the field of Automatic Test Pattern Generation (ATPG)
- **Goal**: Model different satisfaction and optimisation problems for combinational circuits with a set *D* of diagnoses. (where a diagnosis *d* is a set of faults, and each fault is a stuck line.)
- **Approaches**:
  Specific tools --> low flexibility
  General Problem solvers (e.g. SAT) --> duplication of circuits
  CLP: Constraint Logic Programming --> our approach

Problems

- **Diagnosis**. Find an input test pattern i that differentiates two diagnoses $\{d_1, d_2\} = D$, i.e. find an input that yields a different value, in at least one output bit, for diagnoses d_1 and d_2;
- **Maximisation**. Find an input i which is a test pattern for a maximum number of diagnoses in D;
- **Minimisation**. Find a minimal set S of input test patterns that cover all diagnoses in D.

An 8-Valued Logic for Differential Test Pattern Generation

- Given input I, an unexpected output O is observed in the circuit => alternative theories.
- 3 circuit models (a normal one plus 2 diagnostic theories):

  N: no faults
  T_1: circuit with faults F_1
  T_2: circuit with faults F_2
- To differentiate diagnoses we keep track of dependencies:

X the signal truth value is X, independent from faults F_1 and F_2.

d_1-X depends on F_1, but not on F_2. If F_1 occurs, signal is *not(X)*; otherwise it is X.

d_2-X similar to the previous, with F_1 and F_2 swapping roles.

m-X depends on both diagnoses F_1 and F_2. If any of them occurs, signal is *not(X)*

An 8-Valued Logic for Differential Test Pattern Generation

- Truth value *Z* of a “bit” with a normal model and two different theories

N	0	0	0	0	1	1	1	1
T_1	0	0	1	1	0	0	1	1
T_2	0	1	0	1	0	1	0	1
Z	0	d_2-0	d_1-0	m-0	m-1	d_1-1	d_2-1	1

d-values thus differentiate theories. Such a value is needed in an output bit to obtain a differential test pattern.

Boolean Operations

- Physical output of an xor-gate with inputs *m-1* and *d_1-0*

	m-1	d_1-0	m-1 $\oplus$ d_1-0
N	1	0	1
T_1	0	1	1
T_2	0	0	0

- Xor Truth table in 8-valued logic

xor	**m-0**	**d_2-0**	**d_1-0**	**0**	**1**	**d_1-1**	**d_2-1**	**m-1**
m-0	0	d_1-0	d_2-0	m-0	m-1	d_2-1	d_1-1	1
d_2-0	d_1-0	0	m-0	d_2-0	d_2-1	m-1	1	d_1-1
d_1-0	d_2-0	m-0	0	d_1-0	d_1-1	1	m-1	d_2-1
0	m-0	d_2-0	d_1-0	0	1	d_1-1	d_2-1	m-1
1	m-1	d_2-1	d_1-1	1	0	d_1-0	d_2-0	m-0
d_1-1	d_2-1	m-1	1	d_1-1	d_1-0	0	m-0	d_2-0
d_2-1	d_1-1	1	m-1	d_2-1	d_2-0	m-0	0	d_1-0
m-1	1	d_1-1	***d_2-1***	m-1	m-0	d_2-0	d_1-0	0

Modelling Alternative Diagnostic Theories

- Output of an S-buffer (stuck-at-0 in T_1 and stuck-at-1 in T_2) for input m-0

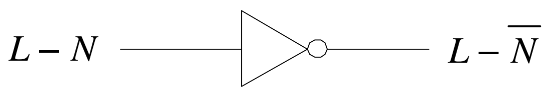
	m-0	S-buffer (T1/0,T2/1) output
N	0	0
T_1	1	0
T_2	1	1

- Truth table for the 8 different types of S-buffer (possible faulty lines)

Input= / Dependency	m-0	d_2-0	d_1-0	0	1	d_1-1	d_2-1	m-1
$T_1/0$, $T_2/0$	0	0	0	0	m-1	m-1	m-1	m-1
$T_2/0$	d_1-0	0	d_1-0	0	d_2-1	m-1	d_2-1	m-1
$T_1/0$	d_2-0	d_2-0	0	0	d_1-1	d_1-1	m-1	m-1
$T_1/0$, $T_2/1$	**d_2-0**	d_2-0	d_2-0	d_2-0	d_1-1	d_1-1	d_1-1	d_1-1
$T_1/1$, $T_2/0$	d_1-0	d_1-0	d_1-0	d_1-0	d_2-1	d_2-1	d_2-1	d_2-1
$T_1/1$	m-0	m-0	d_1-0	d_1-0	1	1	d_2-1	d_2-1
$T_2/1$	m-0	d_2-0	m-0	d_2-0	1	d_1-1	1	d_1-1
$T_1/1$, $T_2/1$	m-0	m-0	m-0	m-0	1	1	1	1

Generic Modelling

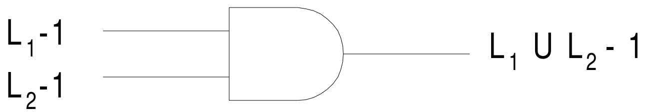
- Normal Gates



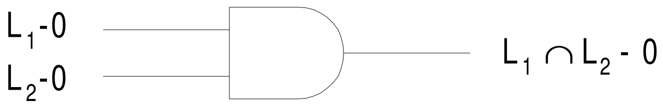
L is a set of dependencies;
N is a Booleean value

e.g. *{{f/0,g/0},{f/1,h/0},{i/1,j/1}}-0*

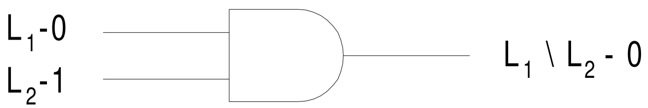
normally is 0 but if either both faults {f/0,g/0} (i.e. *f* stuck-at-0 and *g* stuck-at-0), or {f/1,h/0} or {i/1,j/1} occur, the actual value is 1. Signal *{}-N* thus represents a constant value *N*, independent of any fault.



with two 'ones' as normal inputs, it is enough that one of them is wrong to obtain a wrong output, so we have the union of dependencies;



two 'zeros': it is necessary to invert both to invert the output, so we have the intersection of the dependencies;



0 and 1: to obtain a wrong output (i.e. 1), the 0 input must be inverted to 1,and the other input must remain 1, i.e. it must not be inverted, which justifies the set difference.

The output normal value is just the conjunction of the normal inputs.

Generic Modelling

- ## S-Buffers

  (an S-buffer *g* has associated a set L_S of the possible diagnoses where *g* appears as stuck)

$L_{S0} = \{diag \in D:\ g/0 \in diag\}$
$L_{S1} = \{diag \in D:\ g/1 \in diag\}$
$L_S = L_{S0} \cup L_{S1}$
(obviously, $L_{S0} \cap L_{S1} = \varnothing$)

In	Out
$\varnothing$-0	L_{S1}-0
$\varnothing$-1	L_{S0}-1
L_i-0	$L_{S1} \cup (L_i \setminus L_{S0})$ - 0
L_i-1	$L_{S0} \cup (L_i \setminus L_{S1})$ - 1

Domain Reduction to Sets

- Pairs *L-N* inconvenient since both the set and the Boolean values can be variables, making it hard to handle
- Signal transformation to sets

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

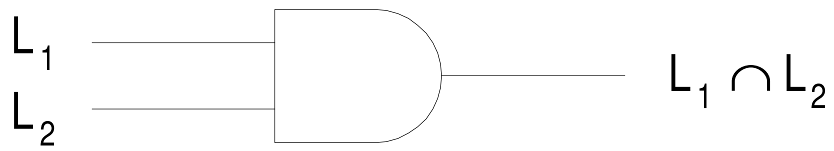
- *transf* is not a bijective function, since $transf(L-0) = transf(\overline{L}-1) = L$ Still it is useful to work with, and possible to express our goal and obtain the desired result

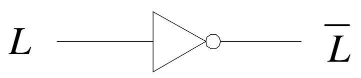
Gates Representation with Sets

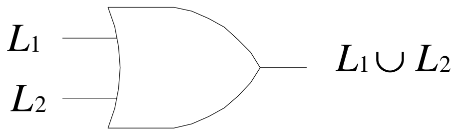
- Normal Gates

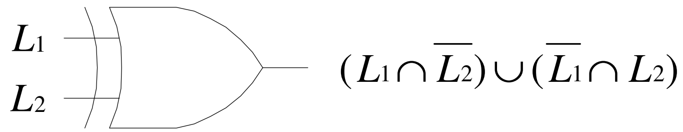
I1	transf(I1)	I2	transf(I2)	I1$\wedge$I2	transf(I1$\wedge$I2)
L_1-1	$\overline{L_1}$	L_2-1	$\overline{L_2}$	$L_1 \cup L_2$-1	$\overline{L_1 \cup L_2} = \overline{L_1} \cap \overline{L_2}$
L_1-0	L_1	L_2-0	L_2	$L_1 \cap L_2$-0	$L_1 \cap L_2$
L_1-0	L_1	L_2-1	$\overline{L_2}$	$L_1 \backslash L_2$-0	$L_1 \backslash L_2 = L_1 \cap \overline{L_2}$

Therefore, $transf(I1) \wedge transf(I2) = transf(I1 \wedge I2)$





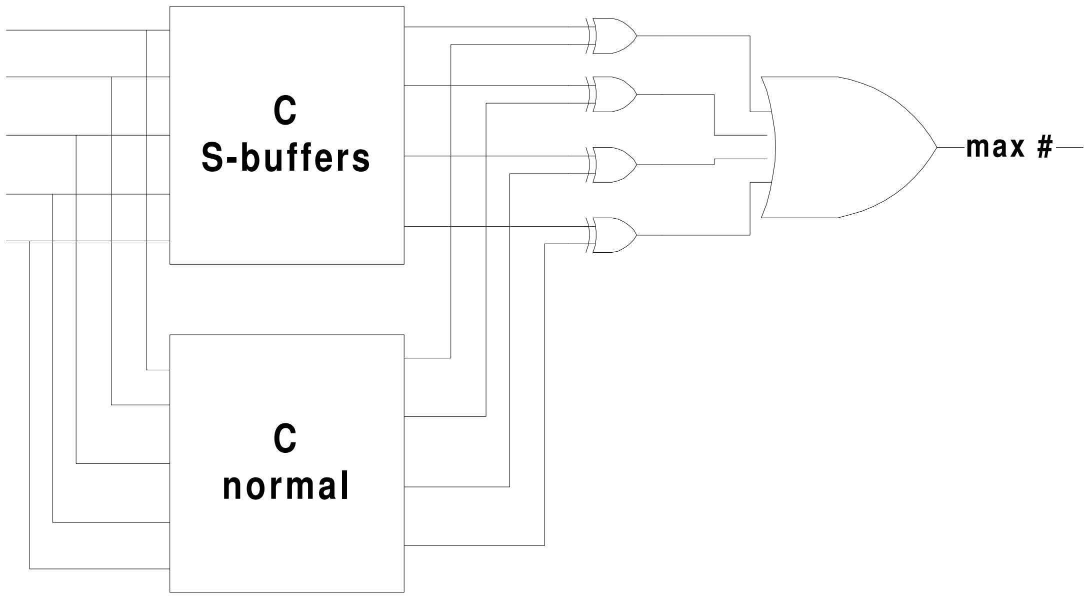




- S-Buffers

Solving

- **Diagnosis**. Simply ensure that a set *L* with cardinality 1 is present at a circuit output bit.
- **Maximisation**. Maximise the union of the correct output dependencies:



- **Minimisation**. Meta-problem involving sets of solutions to set problems. Finding simple test patterns and maximising faults to cover can be used in a branch and bound manner. Still an open problem.

Conclusions and Further Work

- Adequate modelling formalism
- Set Solving under development
- Realistic comparison with alternative approaches