Modelling Digital Circuits Problems with Set Constraints

Francisco Azevedo and Pedro Barahona
{fa,pb}@di.fct.unl.pt
Departamento de Informática, Universidade Nova de Lisboa
2825-114 Caparica — Portugal

Abstract. A number of diagnostic and optimisation problems in Electronics Computer Aided Design have usually been handled either by specific tools or by mapping them into a general problem solver (e.g. a propositional Boolean SAT tool). This approach, however, requires models with substantial duplication of digital circuits. In Constraint Logic Programming, the use of extra values in the digital signals (other than the usual 0/1) was proposed to reflect their dependency on some faulty gate. In this paper we present an extension of this modelling approach, using set variables to denote dependency of the signals on sets of faults, to model different circuits problems. We then show the importance of propagating constraints on sets cardinality, by comparing Cardinal, a set constraint solver that we implemented, with a simpler version that propagates these constraints similarly to Conjunto, a widely available set constraint solver. Results show speed ups of Cardinal of about two orders of magnitude, on a set of diagnostic problems.

1 Introduction

A number of problems in Electronics Computer Aided Design (ECAD) has been widely studied and they are still the subject of active research, with a variety of approaches. The evolution of the area, with new technologies and continuous new requirements and needs, makes it a suitable application field for CLP [12], whose usefulness was already exemplified and discussed [17].

One particular sub-area of ECAD that deserves plenty of attention is that of Automatic Test Pattern Generation (ATPG), which aims at checking whether a circuit is faulty or not. In this context, a digital circuit (e.g. a VLSI chip) is regarded as a black box, performing some function, and one has only access to its inputs and outputs. The basic problem consists of finding an input test pattern for a specific faulty gate, i.e. an input that makes the output dependent on whether the gate is faulty or not. In general, one is not interested in the basic problem but rather in some related and more complex problems.

One such problem is the generation of minimal sets of test patterns, i.e. in finding sets of test patterns with minimum cardinality that cover all the possible faults in a digital circuit. A related problem is finding maximal test patterns, i.e. those that maximise the number of faults they unveil. A third problem, diagnosis, aims at generating patterns for a circuit that would produce different outputs for different sets of faulty gates. The problem is not only interesting in itself, but has possible applications on the related optimisation problems.
These problems have usually been handled either by specific tools or by modelling them in some appropriate form to be subsequently dealt with by a general problem solver (e.g. a Boolean SAT-based solver [15]). Current techniques to deal with the problem of diagnosis try to generate input vectors that cause different outputs in two circuits. In [9] one tries to detect one fault assuming the circuit has the other; in [8] one tries to detect one fault without detecting the other; finally, in [14] one tries to detect both, and after undetect one of them. The complexity of the diagnosis increases significantly with the extra circuitry involved, though, and modelling the above referred optimisation problems into a SAT solver poses a more challenging problem with respect to the multiplication of circuits [13], and the associated combinatorics.

As an alternative to Boolean satisfiability, a CLP system presented in [16] adds two extra values to the usual Boolean 0/1 values to code the dependency of a digital signal on the (faulty) state of a gate. With this extension, a test pattern is an input of the circuit that yields an output with one such extra value. In [2], this idea was adopted and extended further for diagnostic problems, by introducing a logic whose 8 values were used not only to represent dependency on a faulty gate, but also to discriminate the dependencies between two sets of faulty gates. Nevertheless, this 8-valued logic does not allow the modelling of ATPG related optimisation problems.

In this paper we discuss an alternative approach using CLP over sets as a unifying modelling framework for all these ATPG related problems, whereby the dependency on sets of faults is modelled by explicit consideration of these sets in the signals that are carried throughout the digital circuit.

Although avoiding the duplication of circuitry required by the Boolean approach, the domains of the variables in this new modelling become more complex, requiring the handling of set constraints, and their efficient constraint solving. Conjunto [6] was a first language to represent set variables by set intervals with a lower and an upper bound considering set inclusion as the partial ordering. Consistency techniques are then applied to set constraints by interval reasoning [3]. This language, implemented as an ECLiPSe [4] library, represented a great improvement over previous CLP languages with set data structures [6].

Conjunto makes a limited use of the information about the cardinality of set variables. The reason for this lies in the fact that it is in general too costly to derive all the inferences one might do over the cardinality information, in order to tackle the problems Conjunto had initially been designed for (i.e. large scale set packing and partitioning problems) [7]. Nonetheless, and given their nature, we anticipated that some use of this information could be quite useful and speed up the solving of ATPG related problems. We thus developed a new constraint solver over sets with two versions. The first fully uses constraint propagation on sets cardinality; the other uses a more limited amount of constraint propagation, similar to that used in Conjunto. In the following we will refer to these versions as Cardinal and „Conjunto“, respectively.

In this paper we present a formal definition of Cardinal and show that, in a preliminary evaluation with diagnostic problems in digital circuits, it has a significant speed up (over 100 times, in average) over „Conjunto“. The paper is organised as follows. Section 2 addresses the modelling of ATPG problems with set constraints. Section 3 describes Cardinal. Section 4 presents some implementation issues as well as preliminary results. Section 5 summarises conclusions and discusses further work.