A Technique to Automate STG Analysis and Refinement for CSC and Normalcy

Alexandre Smirnov

Abstract

One of the formalisms used for specifying the control dominated circuits' behavior is Signal Transition Graph (STG) - an interpreted Petri Net where events represent the transitions of circuit signals. There is a number of properties the specification must satisfy for being implementable by speed-independent (SI) circuit composed of logic gates. These are widely investigated in a number of works.

This work mostly concerns the STG property called normalcy. Normalcy has gained the circuits designers' attention as an STG property, which enables implementing the specified behavior by a circuit consisting of monotonic gates. The use of the later (mostly the negative ones like NAND, NOR) in CMOS technology is prospective for lowering the power consumption, area and/or latency thanks to their inherent characteristics. As well the monotonic implementation contributes to the circuit robustness eliminating the zero input inverter delay assumption.

Normalcy is expressed in terms of the STG states for every particular signal of the STG. There are two conditions either of which being applied for a particular signal ensures its implementation by positive or negative monotonic gate. Much like the Complete State Coding (CSC) property (one of those required for STG implementability by SI circuit composed of logic gates) only the signals, which behavior is to be implemented (non-input) are constrained. CSC and normalcy are closely related and it can be shown that normal STG satisfies CSC.

The number of problems appears in the course of checking refining the specification for normalcy. They are: the state space size of a bounded STG is exponential from the size of STG; the number of distinct solutions for refining STG for normalcy is also exponential; the solution quality criterion significantly depends on the latency constraints between the particular transitions of the specification therefore simply minimizing the number of auxiliary signals introduced does not always mean a good solution.

With the above problems in mind the technique for refining the STG for normalcy is designed customizable and interactive to tackle the solution optimality. Exponential complexity of the analysis and solution search is also mostly avoided with the use of heuristics. These assets however do not come free. The technique is relying on the STG concurrency relations. To make these relations clean and unique the class of STGs the technique can be applied to is limited to those we call canonical. Canonical STGs are supposed to be consistent, live, safe and satisfy the Unique State Factorization (USF) property. Informally the latter implies the uniqueness of the concurrency relations regardless of the choice made by the system. This limitation features the complete information on the state of all STG signals (0, 1, undefined due to concurrency) provided a given place is marked. Besides simplifying the analysis and refinement this limitation contributes to the polynomial synthesis technique designed some time ago in the same framework.

USF enables structural STG analysis and refinement, avoiding generation of the whole state space. The fact that the object on which all operations are performed is the original STG (unlike other techniques employing the derived state graph, unfolding e.t.c) besides reducing complexity enables displaying the detected normalcy and CSC conflicts on the original STG as well as interaction with the designer in the course of the complete solution search.

In this work we: define and justify the class of STG (called canonical) handled by the technique; demonstrate the subset of STGs that can be equivalently transformed to the canonical form; present the theoretical background as well as the most essential issues of the analysis and refinement technique and finally illustrate it with an example.