Hiding Memory Elements in Induced Hierarchical Verification of Speed-Independent Circuits *

Vida Vakilotojar and Peter A. Beerel
EE-Systems Department
University of Southern California
Los Angeles, CA 90089

Abstract: The goal of induced hierarchical verification techniques is to automatically create hierarchy in an originally flat circuit in order to decompose the verification problem (i.e., checking hazard-freedom and conformance to a specification) into that of verifying a set of smaller sub-circuits. Existing induced hierarchical verification techniques for speed-independent circuits are limited because the output of any memory element (e.g., Muller C-element) must be a sub-circuit output (i.e., memory elements cannot be hidden). Consequently, these techniques have exponential complexity in the number of memory element outputs in the circuit. In this paper, we prove that this limitation is not fundamental. Specifically, we develop a theoretical framework for induced hierarchical verification of speed-independent circuits and show that, under certain well-defined conditions, sub-circuits can be formed which may contain memory elements whose outputs are internal to the sub-circuit. The key step of the approach is to obtain a *safe abstraction* of the behavior of a set of *observationally-sufficient external signals.* We prove that such a safe abstraction can drive the reduction of the verification problem into one of verifying a set of smaller sub-circuits. We show how a class of safe abstractions can be derived using an efficient partial order technique. Although generally applicable, we show the result is particularly easy to apply and useful when verifying whether sequential decompositions in speed-independent circuits preserve hazard-freedom.

1 Introduction

Asynchronous circuits have many potential advantages over synchronous circuits, including the potential for lower power and higher average-case performance. However, the lack of a global clock makes ensuring hazard-freedom in an asynchronous circuit a primary concern, and the high concurrency that these circuits often exhibit makes verifying them a computational challenge. Consequently, efficient verification techniques for asynchronous circuits are in great demand. In this paper, we focus on the verification of speed-independent circuits, a class of asynchronous circuits that are hazard-free regardless of the delays assumed by the individual circuit elements.

The verification of speed-independent circuits has been given significant attention in the literature. Dill proposed a trace theoretic framework in which he formulated the notion of trace conformance of speed-independent circuits [9]. Trace conformance is a safety property of speed-independent circuits which checks whether the circuit can generate outputs that are unexpected by its specification. Ebergen and Gingras introduced the notion of completeness with respect to a specification which is stronger than trace conformance in that it requires the circuit to be able to exhibit all the behaviors defined by the specification [10]. Gopalakrishnan et al. proposed a similar notion of strong conformance [12].

It is important to note that both Dill's work [9] and Ebergen and Gingras's work [10] *support* hierarchical verification of circuits. Specifically, if a sub-circuit of a circuit has been successfully verified against a specification, the sub-circuit can be modeled by its specification rather than by its implementation when verifying the whole circuit. This feature is very useful since a specification is typically much easier to represent in a computer than the behavior of the corresponding implementation. This feature, however, does not apply to an originally flat circuit in which sub-circuits do not have specifications. Thus, for a flat circuit, both methods require the entire circuit state space to be analyzed. Unfortunately, the concurrency inside the circuit typically leads to an exponential number of signal interleavings that need to be analyzed, making the verification problem computationally challenging.

Numerous techniques have been proposed to speed-up the verification process of a flat circuit. McMillan proposed a

* This research was supported in part by SRC Contract no. 98-DJ-486, and a gift from the Intel Corporation.

partial order approach based on a technique called *Petri-net unfolding* [18]. While very successful on some scalable examples, the worst-case complexity is in fact no smaller than that of standard reachability analysis algorithms. Burch et al. proposed BDD-based techniques to implicitly analyze the circuit's state space [7]. While successful on some examples, the techniques do not improve the worst-case complexity of the algorithm. Lastly, Roig et al. introduced a modified symbolic breadth-first search algorithm which resulted in significant run-time improvements for some circuits, but again, the complexity of their algorithm stays the same [21].

To reduce the verification complexity, Beerel et al. proposed a two-phase approach in which first functional correctness (i.e., *complex-gate equivalence*) of the circuit was verified and then behavioral properties (i.e., *hazard-freedom*) was checked [3, 4]. The key to their technique is that the behavior of some of the circuit signals could be safely approximated, exponentially reducing the time and space complexity of the verification problem in many examples [3, 4]. Later, Roig et. al proposed a hierarchical approach which also had the advantage of approximating the behavior of some of the circuit signals [22]. Since our proposed technique is most directly related to these latter two works, we describe them in more detail.

The first step in both Beerel et al.'s and Roig et al.'s algorithms is to create a *complex-gate circuit* which effectively *induces* hierarchy by making the signals internal to the complex-gate *hidden*. The state space of the remaining *external* signals, is then analyzed using standard reachability analysis techniques. In Beerel et al.'s technique, an analysis of this state space is used to deduce hazard-freedom of the internal hidden signals. In Roig et al.'s technique, projections of this state space are used as the environment of the complex gates to verify the hidden signals. The key disadvantage of both techniques, however, is that the set of external signals needs to include all memory element outputs (i.e., memory elements cannot be hidden). Since most asynchronous circuits are dominated by memory elements, the number of external signals can still be large and their state space can be too large to analyze. It is this limitation that this paper addresses.

2 Overview of the approach

We observe that analyzing the state space of the complex-gate circuit is equivalent to analyzing a subset of the interleavings of signals of the original circuit. Specifically, this analysis is equivalent to dictating that gates which drive hidden signals have zero-delay and thus hidden enabled signals fire before any enabled external signal.

Motivated by this observation, we develop a theoretical framework for induced hierarchical verification in which, under specific conditions, the set of external signals need not include all memory element outputs. The key result of the paper is the characterization of a *safe abstraction* of the behavior of a set of *observationally-sufficient external signals* that can drive the transformation of the task of verifying a circuit into one of verifying a set of smaller sub-circuits. Our basic verification task is to verify that a closed circuit is failure-free (all input transitions to each circuit element are *legal*). Similar to [9], conformance of an open circuit to its specification can be verified by checking failure-freedom of the closed system composed of the open circuit and the mirror of its specification.

2.1 Safe abstractions and observational sufficiency

We define a safe abstraction of the behavior of a set of external signals E of a circuit to be a behavior (a set of traces), which may under-approximate the behavior of the signals in E if the circuit is not failure-free, but, exactly characterizes the behavior of the signals in E if the circuit is failure-free. We have proved that given such a safe abstraction, the circuit is failure-free iff all of its sub-circuits are failure-free, where a sub-circuit is obtained by composing a circuit block (of the circuit partitioned by the set of external signals) with an environment element appropriately derived from the safe abstraction.

We use transition relations to represent safe abstractions. Consequently, we say that a set of external signals E is *observationally sufficient* if there exists a transition relation over E which does not generate any behavior not present in the original circuit (i.e., the behavior generated by the transition relation is a safe abstraction of E). We call such a transition relation a *safe transition relation.*

2.2 Hierarchically verifying failure-freedom and conformance

Once found, a safe abstraction over a set of external signals drives the transformation of the task of verifying failure-freedom of a circuit into one of verifying failure-freedom of a set of smaller sub-circuits. In particular, verification of failure-freedom of each sub-circuit is performed by checking the conformance of the corresponding circuit block to a specification derived from the safe abstraction, as we now describe. A specification consists of input and output alphabets as well as a description of success and failure behaviors over the inputs and outputs.The output alphabet of the derived specification is taken to be the set of outputs of the circuit block, and its input alphabet is set to the remaining external signals. Traces present in the safe abstraction are considered to be successes. Any other trace is considered to be a failure. This means that traces that include a transition on an output signal which is not present in the safe transition relation are considered to be failures. To check the conformance of the circuit block to the derived specification, the closed circuit composed of the circuit block and the mirror of the specification (which acts like an *environment element*) is checked for failure-freedom. Note that the mirror of the specification is found by merely switching its input and output alphabets (as in [9]).

Similarly, we check conformance of a speed-independent circuit to a given specification by checking failure-freedom of the circuit composed with the mirror of the specification in our hierarchical fashion. The only requirement is that the specification need have the form of a transition relation. Note that more common specification formats such as Petri-nets and STGs can be modeled by transition relations (see, e.g., [22]). In some cases, the transition relation may have signals that do not correspond to input or outputs of the circuit (e.g., signals that represent Petri-net places). We consider any such signal as an input of both the specification and the circuit, and assume that transitions of such signals can never lead to a failure in either the specification or the circuit.

2.3 Checking observational sufficiency of a set of external signals

Unfortunately, it may sometimes be computationally expensive to determine whether a set of circuit signals is observationally-sufficient. We therefore restrict our search for an observationally sufficient set of external signals to the special cases in which the set is a superset of the set of all outputs of non-deterministic circuit elements. We then use properties of speed-independent circuits to show that, for such a set of external signals, it is possible to check its observational-sufficiency at the same time we compute the safe abstraction of its behavior. Moreover, we derive this safe abstraction by analyzing only a subset of interleavings of all hidden signals.

More specifically, the safe abstraction can be found by performing a special form of reachability analysis on the circuit as follows. We start the analysis from any state of the circuit where all internal signals are stable, referred to as an *internally stable state*. From such a state, the firings of all enabled external signals are explored. After each external signal firing, only one possible interleaving of hidden signals leading to the next internally stable state is explored. The next internally stable state of the circuit is found by finding the stable state of each hidden signal. Note that a hidden signal may fire more than once before becoming stable. Once the next internally stable state of the circuit is found, any newly enabled external signal w is identified, and added to the set of enabled external signals. This process is repeated until the whole external state space of the circuit is explored.

We have proved that a necessary and sufficient condition for observational sufficiency is that the same external signal transitions must be enabled at all internally stable states which are explored as explained above and which have the same external encoding. If this condition is violated, however, a new set of external signals must be chosen and the verification procedure must repeat.

We believe that symbolic reachability analysis techniques using BDDs can also be used for this modified reachability analysis. That way, sets of states, e.g., those reachable by all possible interleavings of enabled external signals, can be processed simultaneously. The key advantage of our approach is that the BDDs need to depend only on external signals. For example, to find the stable values, we can use one such BDD for each hidden signal to keep track of the signal's stable value in each external state. Thus, compared to traditional symbolic approaches which explore the reachable state space of all circuit signals, the BDDs can be exponentially smaller, possibly leading to dramatic run-time savings. Moreover, we can use symbolic trajectory evaluation techniques to reduce the number of iterations of reachability analysis in finding stable values of hidden signals, obtaining further speed ups.

2.4 Implications on model checking of speed-independent circuits

The application of our hierarchical verification techniques for speed-independent circuits/systems is not limited to verifying conformance and failure-freedom. Once failure-freedom of the circuit is hierarchically established, it is guaranteed that the behavior of the external signals is exactly characterized by our safe abstraction. Consequently any path-based property that refers to only external signals can be verified using the safe abstraction. This observation is consistent with the fact that model checking of speed-independent circuits can benefit greatly from partial order techniques because all signals are *independent* [24,11]. Also, if all the signals involved in a given property of interest fall into a single sub-circuit, then the property of interest need to be checked only in that sub-circuit. This suggests that our work may sometimes lead to faster ways of performing sym-

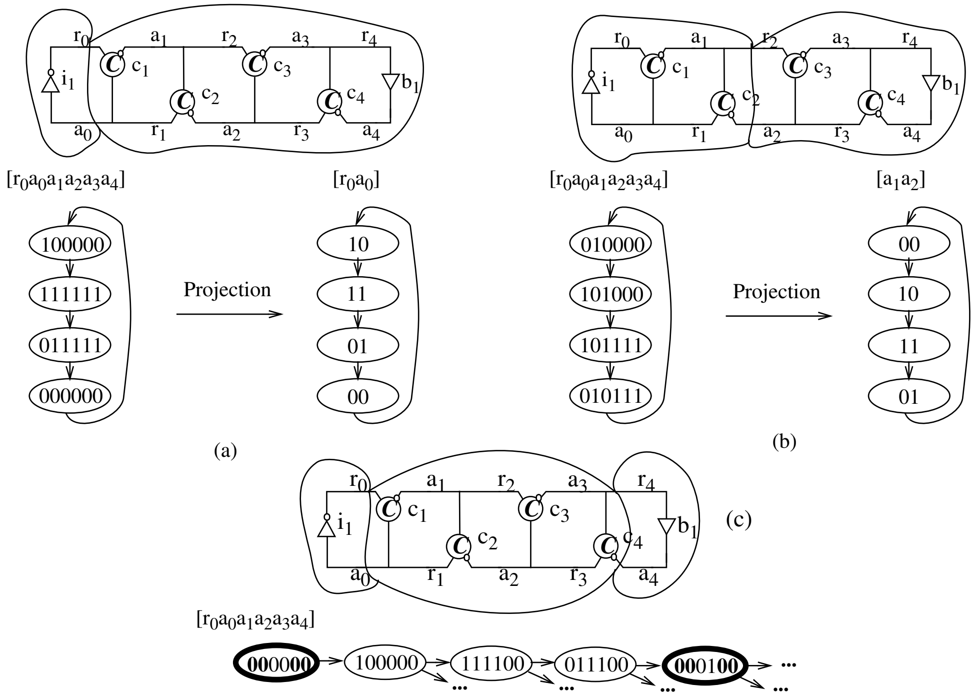

Fig. 1: Sequences of internally stable states of three different partitions of the FIFO, and projections of those behaviors.

bolic model checking of speed-independent circuits.

3 An illustrative example

This section illustrates our hierarchical verification procedure and its potential power in the analysis of speed-independent circuits through an example circuit, a four-stage FIFO controller in its abstract environment, depicted in Figure 1. Although somewhat trivial, the circuit is interesting because it is an extreme example of a circuit dominated by memory elements (Muller C-elements) and the achievable reduction in analysis complexity is significant. The circuit also will help illustrate the impact of different partitions of a circuit on the existence and efficiency of a safe abstraction of its external behavior.

Figure 1 shows three different possible partitions of the circuit. In each case, the possible sequences of transitions of external signals from internally stable states are found as follows: (1) the initial state must be internally stable; that is no hidden signals are enabled, (2) from each reachable state, all of which should be internally stable, all possible next states reachable by an external signal transition and subsequently settling of all hidden signals are found.

Consider the transition from state 101000 to 101111 in Figure 1(b). In the internally stable state 101000, the only external signal that can fire is a_2. Once a_2 rises, a sequence of hidden signal transitions on a_3 and a_4 occurs, which leads to the internally stable state 101111 where the only enabled signal is a_1, an external signal.

Projecting the derived sequence of states (behavior) of Figure 1(b) onto the set of external signals $\{a_1, a_2\}$, results in a safe abstraction which can be used to derive specifications for the two circuit blocks generated by the external signals. The safe abstraction obtained appears like that of an inverter to the right block, and that of a buffer to the left block, which is as expected. The problem of verifying the four-stage FIFO controller, whose actual state space is rather large, is now broken to that of two two-stage FIFO controllers, and the underlying state space exploration is much easier than exhaustive state exploration.

The abstract behavior of the partition of Figure 1(a) is similar to that of Figure 1(b). Although this behavior can successfully help in verifying the failure-freedom of the left block, it does not reduce the complexity of verifying the right block and hence the whole circuit. Thus, the partitioning of Figure 1(a) is not an efficient one.

Finally, Figure 1(c) shows an example of a partition which is not observationally sufficient; i.e., whose derived behavior does not have a safe transition relation. Since the state graph is large, only a portion of it, containing an instance of violation of the necessary condition for observational sufficiency, is shown. It can be seen that while the external signal a_3 is enabled to rise in the internally stable state **00**01**00**, it is not enabled in **00**00**00** which has the same external encoding (the

Fig. 2: Decomposition of a four-input generalized C-element into a two-input and a three-input generalized C-element [6].

external state is shown in bold face). In fact, the two states are different because of the number of pending tokens in the chain. Hence, this partition will not yield a safe abstraction, because the transition relation of its projected behavior is not safe.

4 Applications of the approach

The practicality of this theory depends on how common are those circuits for which there exists an observationally sufficient set of external signals such that a large number of memory elements can be hidden, and how difficult finding such a set is. Although we believe that this technique will be widely applicable to many circuits, we have not yet explored techniques to identify candidate external signal sets. We believe that in many cases designers and synthesis tools can provide candidate external signal sets. For example, recently proposed technology mapping tools for speed-independent circuits generate many internal memory elements that, once identified, can be hidden by our tool.

The principal challenge with technology mapping of speed-independent circuits is that the decomposition of gates can introduce hazards. For this reason, traditional synthesis algorithms had to rely on the existence of virtual cell libraries which contained gates of unlimited fanin. The most recent work has focused on the sequential decomposition of memory elements (which is more powerful than combinational decomposition) [6,8,15]. Figure 2 shows an example of the decomposition of a high-fanin state holding circuit element during technology mapping [6]. In some cases, the decomposition is more complex in that the newly made hidden node must have additional connections, called *acknowledgment wire forks*, that branch off to other parts of the circuit to ensure hazard-freedom [8,15]. Such transformations can be done at many parts of the circuit leading to many internal C-elements.

The key observation we make is that for the decomposition to be correct the behavior of the original signals must not be altered by the decomposition. Consequently, the original set of external signals (before decomposition) is observationally-sufficient. In other words, we can hide all internal memory elements introduced by sequential decomposition. Thus, for this application, a good set of external signals can be easily provided for the verification tool by the technology mapping tool.

From a practical standpoint, this means that we can verify circuits derived using sequential decomposition exponentially faster than traditional algorithms. More specifically, while existing algorithms must explore all interleavings of the outputs of the newly introduced sequential elements, we need to analyze only one.

5 Sequential circuits with internal memory elements versus cyclic combinational circuits

The necessary and sufficient conditions for observational sufficiency of a set of external signals, imply that each individual circuit block must be *valid*. This section defines this notion of validity and relates it to previous induced hierarchical verification efforts [3,4,21,22] as well as the notion of *cyclic combinational circuits* discussed by Malik [25] and Wolf et al. [26].

For technical reasons and to simplify the discussion, the internal signals of a valid block must not oscillate; i.e., they eventually settle if the external signals are held fixed (this is a restriction not present in [25]). Now, for any external (input/output) state of a circuit block, the set of consistent (having the same external signal values) internally-stable states can be partitioned in the following way: two consistent internally-stable states are in the same equivalence class if they have the same set of outputs enabled. If the environment of the circuit block interacts with it in such a way that for any reachable external state, whenever such a state is revisited the internally-stable state of the circuit block (obtained by holding the external signals fixed and letting the internal signals stabilize) falls in the same equivalence class, then, the circuit is valid.

A *cyclic combinational circuit* [25] is a circuit which logically acts combinationally but has structural loops. It is easy to show that such circuit blocks are valid (assuming that no component within the block oscillates). Validity, however, is different than cyclic combinationality because it also allows a *sequential* block (i.e., not combinational) whose next state of the outputs *depend* only on the current state of the inputs and outputs, but no other internal signals. In other words, sequential blocks that have no “hidden internal state” are valid.

The *complex gates* formed by collapsing combinational gates into subsequent memory-elements described in previous induced hierarchical verification techniques [3,4,21,22] are also valid, because regardless of the behavior of the environment the next state of the outputs depend only on the current state of the inputs and outputs. Validity, however, can also be determined by the behavior of the environment of a circuit block. For example, if the environment never puts the circuit block in a state where the next state of outputs depends on the internal signals, then the circuit block is valid. We note that a parallel concept was presented in [25], where Malik suggests that a cyclic sequential circuit block may display combinational behavior if none of the input assignments which make it behave sequentially are ever generated by its environment.

The key point is that, compared to previous work [3,4,21,22,25], our notion of validity allows larger blocks to be abstracted by, when possible, hiding internal memory elements and internal cycles. Thus, validity effectively combines our previous notion of complex-gates with the notion of cyclic combinationality.

6 Comparison of the proposed approach with other reduction techniques

Partial order, abstraction, and hierarchical verification techniques have been extensively used in different tools to reduce the complexity of verification [5, 13, 14]. This section discusses the relationship between the traditional usage of such techniques and our proposed induced hierarchical verification approach.

In VIS [5], abstraction is referred to using non-determinism to abstract the behavior of some circuit signals. Specifically, the signals are treated as primary inputs whose behavior is totally unconstrained. This is probably too conservative for our application because such non-determinism would introduce unreachable states which may exhibit hazards, leading to false negatives. In contrast, we refer to an approximation of the behavior of a subset of circuit signals as an abstraction. Moreover, unlike that of VIS, our abstraction never overestimates the behavior of the signals.

The partial order reduction approach to verification of concurrent systems is based on the observation that some checked properties are insensitive to the order in which concurrent or independent events or signals are interleaved [1, 11, 20, 23]. Based on this observation, at each state, only a subset of enabled signals which are *independent* of the rest of the enabled signals, called an *ample* set, would be sufficient for exploration. This often reduces the number of states necessary to be explored, dramatically. Interestingly, our derivation of a safe abstraction can also be viewed as a partial order reduction.

Consider the simpler case of a deterministic speed-independent circuit (no arbitration). In such a circuit, all enabled signals are independent, in the sense that no enabled signal can be disabled by the firing of another enabled signal. Consequently, when deriving a safe abstraction, any enabled hidden signal constitutes a valid ample set. Since we are interested in obtaining the exact behavior of all external signals, we choose either (1) any single hidden signal as an ample set, if any hidden signal is enabled or, (2) all external signals as the ample set, if no hidden signal is enabled. For more general speed-independent circuits (having arbitration and/or circuit elements with internal state variables), we still can use the same partial order reduction technique if we carefully restrict the set of external signals and do not allow any hidden non-deterministic signals. The tricky part of our use of partial orders is that we do not know a priori whether the circuit is speed-independent. Consequently, unlike the typical use of partial orders, our technique may actually underestimate the behavior of the external signals. A key feature of our technique is that if there is any such underestimation, it is always detected in the form of a failure in some sub-circuit with the conclusion that the circuit is not failure-free.

Moreover, the independence of signals in a speed-independent circuit allows us to use a form of symbolic trajectory analysis to find the states in which all hidden signals have stabilized (given fixed external signal values). This symbolic trajectory analysis has two benefits. First, since the hidden signals are independent, non-interleaving semantics in which all enabled hidden signals fire simultaneously can be used (e.g., [23]). This reduces the number of iterations that need to be analyzed. Secondly, the hidden signals need not to be BDD variables (the settled values are always expressed in terms of external signals). Notice that this property is a feature of speed-independence and is not possible in more general settings [1].

We conjecture that our technique can be considered to be a homomorphic reduction as used in COSPAN [13, 16]. In COSPAN, such a homomorphism simplifies the language containment test between a model and a task by removing irrelevant aspects of the model. We believe that our safe abstraction can be viewed as the result of a homomorphic transformation which simplifies the model of the environment for each sub-circuit. In our framework, the homomorphic system is automatically generated (using our partial order technique) once a set of external signals is given, and the validity of the homomorphism is checked by checking the necessary and sufficient condition for observational sufficiency. Moreover, we believe that this homomorphic reduction is both necessary and sufficient for verifying the non-reduced problem, and consequently does not lead to any false negatives like some homomorphisms in COSPAN.

Our approach is also similar to the more general assume-guarantee paradigm used in *reactive modules* [2]. A property of reactive modules enables the verification of a composition of modules through the verification of each module in an abstract environment followed by the verification of the composition of abstract environments. We believe that our safe abstraction is to some degree analogous to an abstract environment with some differences. The most obvious difference may be that our methodology does not need a separate step of verifying the composition of abstract environments.

Lastly, we should also note that Weih and Greenstreet developed a verification framework for speed-independent circuits with similar characteristics as ours but for a somewhat different purpose [24]. Specifically, rather than verifying speed-independence of a circuit, their goal is to verify *local formulas* of circuits that have already been verified to be hazard-free (i.e., semi-modular). In other words, in a preprocessing stage, they must rely on traditional techniques to verify the speed-independence of the design. Nevertheless, their ideas are similar to ours in that to achieve their goal, they argue that only one interleaving needs to be analyzed.

7 Conclusions

The practical impact of the developed theoretical framework cannot be fully measured until it is incorporated into a verification algorithm and tested on practical circuits. In doing so, many algorithmic issues need to be addressed. For example, while for the technology mapping application, a good set of external signals can be readily supplied by the user, we need to identify good heuristic techniques for the choice of external signals for more general applications. Developing symbolic algorithms for finding external state spaces is another part of our future work. Also, we plan on exploring extensions of our framework to the verification of timed circuits.

Acknowledgments

We would like to thank our reviewers for their comments and feedback which led us to a more thorough comparison of our approach to previous approaches. We would also like to thank David Dill for his valuable comments which helped us generalize our ideas for hierarchical verification of failure-freedom, to hierarchical verification of speed-independent circuits (including conformance checking).

References

[1] R. Alur, R. K. Brayton, T. A. Henzinger, S. Qadeer, S. K. Rajamani. Partial-Order Reduction in Symbolic State Space Exploration. In *Proc. of the 9th Intl. Conf. on Computer-Aided Verification,* Vol. 1254 of LNCS, pp. 340-351. Springer, 1997.

[2] R. Alur, T. A. Henzinger. Reactive Modules. In *Proc. of the 11th Annual IEEE Symposium on Logic in Computer Science* (*LICS* 1996), pp. 207-218.

[3] P. A. Beerel, J. R. Burch, and T. H.-Y. Meng. Efficient Verification of Determinate Speed-Independent Circuits. In *Proc. of ICCAD-93,* pp. 261-267. IEEE Computer Society Press, 1993.

[4] P. A. Beerel, J. R. Burch, and T. H. Meng. Checking Combinational Equivalence of Speed-Independent Circuits. Accepted for publication in *Formal Methods in System Design.*

[5] R. K. Brayton et al. VIS: A System for Verification and Synthesis. In *Proc. of 8th Intl. Conf. on Computer-Aided Verification*, Vol. 1102 of LNCS, pp. 428-432. Springer, 1996.

[6] S. M. Burns. General Conditions for the Decomposition of State Holding Elements. In *Proc. of Intl. Symp. on Advanced Research in Asynchronous Circuits and Systems, Async-96*, pp. 48-57, 1996.

[7] J. R. Burch, E. M. Clarke, D. E. Long, K. L. McMillan, and D. L. Dill. Symbolic Model Checking for Sequential Circuit Verification. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, No. 4, April 1994, pp. 401-424.

[8] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, E. Pastor, A. Yakovlev. Decomposition and Technology Mapping of Speed-Independent Circuits Using Boolean Relations. In *Proc. of Intl. Conf. on CAD-97*, pp. 220-227, 1997.

[9] D. L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits*. ACM Distinguished Dissertation, The MIT Press, 1988.

[10] J. C. Ebergen and S. Gingras. A Verifier for Network Decompositions of Command-Based Specifications. In *Proc. of HICCS*, 1993.

[11] P. Godefroid. *Partial-Order Methods for Verification of Concurrent Systems*. Springer, 1996.

[12] G. Gopalakrishnan. A Correctness Criterion for Asynchronous Circuit Validation and Optimization. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, No. 11, Nov. 1994.

[13] R. H. Hardin, Z. Har'El, and R. P. Kurshan. COSPAN. In *Proc. of 8th Intl. Conf. on Computer-Aided Verification*, Vol. 1102 of LNCS, pp. 423-427. Springer, 1996.

[14] G. H. Holzmann and D. Peled. The State of SPIN. In *Proc. of 8th Intl. Conf. on Computer-Aided Verification*, Vol. 1102 of LNCS, pp. 385-389. Springer, 1996.

[15] A. Kondratyev, J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev. Technology Mapping for Speed-Independent Circuits: Decomposition and Resynthesis. In *Proc. of Intl. Symp. on Advanced Research in Asynchronous Circuits and Systems, Async-97*, 1997.

[16] R. P. Kurshan. *Computer-Aided Verification of Coordinating Processes - The Automata-Theoretic Approach*. Princeton Univ. Press, 1994.

[17] L. Lavagno. *Synthesis and Testing of Bounded Wire Delay Asynchronous Circuits from Signal Transition Graphs*. Ph.D. Dissertation, Univ. California, Berkeley, CA, 1992.

[18] K. L. McMillan. A Technique of State Space Search Based on Unfolding. In *Formal Methods in System Design*, Vol. 6, pp. 45-65, Kluwer Academic Publishers, Boston, 1995.

[19] K. L. McMillan. *Symbolic Model Checking*. New York, Kluwer Academic Publishers, 1993.

[20] D. Peled. Combining Partial Order Reductions with On-the-fly Model-checking. In *Formal Methods in System Design*, Vol. 8, pp. 39-64, Kluwer Academic Publishers, Boston, 1996.

[21] O. Roig, J. Cortadella, and E. Pastor. Verification of Asynchronous Circuits by BDD-based Model Checking of Petri Nets. In *16th Intl. Conf. on Theory and Application of Petri-Nets*, Torino, Italy, June 1996.

[22] O. Roig. *Formal Verification and Testing of Asynchronous Circuits*. Ph.D. Thesis, Univ. of Politecnica de Catalunya, Barcelona, May 1997.

[23] A. Valmari. On-the-fly Verification with Stubborn Sets. In *Proc. of the 5th Intl. Conf. on Computer-Aided Verification*, Vol. 697 of LNCS, pp. 397-408. Springer-Verlag, 1993.

[24] D. T. Weih and M. R. Greenstreet. Verification of Speed-Independent Data-Path Circuits. In *IEE Proceedings-Computers and Digital Techniques*, Vol. 143, No. 5, pp. 295-300, Sept. 1996.

[25] S. Malik. Analysis of Cyclic Combinational Circuits. In *Proc. of ICCAD-93,* pp. 618-625. IEEE Computer Society Press, 1993.

[26] J. R. Burch, D. Dill, E. Wolf, and G. De Micheli. Modeling Hierarchical Combinational Circuits. In *Proc. of ICCAD-93,* pp. 612-618. IEEE Computer Society Press, 1993.