7

CONCLUSIONS

That was Ender's gift to us, to free us from the illusion that any one explanation will ever contain the final answer for all time, for all hearers. There is always, always more to learn.

—Orson Scott Card - Children of the Mind, 1996

Now this is not the end. It is not event the beginning of the end. But it is, perhaps, the end of the beginning.

—Winston Churchill - The End of the Beginning, 1943

Summary

This chapter presents the conclusions of the thesis. In first place, an overview of the generalities of the work is provided by reviewing the material included in each chapter of this document. Next, the contributions of the work to different aspects of the formal verification of complex timed systems are analyzed. The last section describes a number of open issues for future developments and research.

7.1 Introduction

This thesis presents a new approach for the formal verification of safety properties in timed systems. The correct operation of such systems not only depends on functional properties but also on assumptions about the delays of the components of the system and the response times of the environment in which the system operates. The approach presented is theoretically sound and has been automated into a software CAD/CAV tool.

Formal verification uses deductive methods in order to prove if a system satisfies a given set of well-defined properties. Formal techniques have received increasing attention in recent years, mainly due to two factors: the high complexity of nowadays systems, and the high cost of correcting errors at the end of the design cycle or even once a system is yet in the market. These aspects were reviewed in Chapter 1 as an important motivation for this work.

In order to formally verify a system, a precise and unambiguous model of the system is required. Such model must capture the subset of relevant aspects of the system which are of interest for the verification. Several models have been proposed for timed systems, yielding also to different verification approaches. In this work, *transition systems* are the model of choice (see Chapter 2). On the other hand, Chapter 3 reviews some of the most relevant approaches in the area of verification of timed systems.

The verification of timed systems poses serious complexity problems. Although efficient techniques have been devised to overcome the complexity issue, well-known symbolic methods cannot be easily applied. Since most methods for verification rely on the computation of the complete state space, the combinatorial state explosion problem becomes exacerbated by the time dimension. As a consequence, the practical applicability of the resulting verification methods is often restricted to small systems, or to systems with particular characteristics that fit well with a given verification method.

The theory that supports the verification approach proposed in this thesis was introduced in Chapter 4. The approach extends the conventional symbolic model checking methods to the verification of timed systems. *Timed transition systems* are used as the underlying formalism for timed systems under the *continuous-time* paradigm. Instead of computing the exact timed state space, the *relative timing* paradigm is used to abstract exact time information from the representation. Hence, *lazy transition systems* are used, which represent the ordering relations between events in the timed domain by explicitly distinguishing between their enabling and their actual firing conditions. This simple yet powerful model allows the representation of the timed domain of a system using efficient symbolic methods.

The verification approach has been fully implemented in an experimental tool called TRANSYT. The tool can handle hierarchical and distributed modular systems which can inter-operate by a variety of communication mechanisms. TRANSYT proved its function-

ality as well as the validity of the overall verification approach in Chapters 5 and 6. In Chapter 5, a number of timed asynchronous circuits with up to several millions of untimed states were verified with reasonable CPU and memory resources. The experiments covered the verification of complex-gate decompositions in quasi-speed-independent asynchronous circuits, and the verification of circuits optimized for area and/or speed using relative timing assumptions. Additionally, in Chapter 6, compositional verification methods were combined with the basic verification approach in order to tackle the size/complexity issues involved in the verification of complex timed systems. Thus, abstractions, assume-guarantee reasoning and mathematical induction were used to prove the correctness the IPCMOS architecture.

7.2 Contributions

This thesis proposes a novel approach for the formal verification of timed systems. The thesis contributes to this field of research both by providing an original theoretical framework as well as a software tool that implements it. The verification approach is based on two fundamental facts that we want to remark:

- The observation that the set of traces of a transition system can be covered by a set of partial orders. This fact allows to reduce the verification problem to that of: the timing analysis over small sets of events from which timing constraints that prove the correctness or incorrectness of a system can be derived; and the incorporation of such constraints into the system along an incremental refinement process.

- The fact that *relative timing* allows to represent the timed domain of a system in an efficient way using symbolic methods. When considering precise delay bounds in timed systems, the complexity blow-up often causes verification to become an intractable problem, even for small systems. Instead, relative timing considers the *effect* of delays in a system in terms of relative ordering of events.

The verification approach can be briefly summarized as follows. Rather than computing the exact timed state space of the system, the approach starts with a simple approximation in which time is not taken into account. If the property under verification is satisfied in the untimed approximation, it will be also satisfied in the timed domain of the system, and the verification concludes. Otherwise, a counterexample trace is built which reproduces a violation of the property. Timing information is then used to try to refuse the counterexample. Thus, an efficient *off-line* timing analysis is performed on an *event structure* that covers the counterexample trace. If the counterexample persists, the verification concludes. On the contrary, the system is refined with the relative timing information derived from the timing analysis. This process repeats until an actual counterexample is found or the property is proved correct. Therefore, the proposed approach relies on a

series of incremental refinements of the state space of the system, so that the complexity due to the timing information is incorporated only when it is needed.

The idea of using event structures for timing analysis was already proposed in [KBS02]. However, no algorithm was presented that can handle a general class of transition systems for verification. On the contrary, the approach proposed in this thesis is applicable to systems modeled by timed transition systems without restrictions. For example, no requirement is imposed about the causality relations between events or about the types of choice allowed.

We want also to remark that the use of the proposed approach for the verification of untimed systems does not involve any additional overhead with respect to the conventional symbolic methods for such type of systems.

The key features of the presented work on the verification of timed systems can be summarized by the following topics:

Relative timing. The use of relative timing allows to avoid the computation of the exact timed state space of the system, which is a common practice of model checking methods for timed systems. Instead, in the proposed approach the timed behavior of events is captured by means of partial orders that represent simple facts as if an event happens before another, *i.e.* relative temporal relations.

Symbolic representation. As a consequence of the previous topic, the state space of the system can be represented and managed using symbolic methods with proved efficiency such as BDDs. This allows a natural extension of traditional symbolic model checking techniques for untimed systems into the timed systems domain of application.

Local timing analysis. No global timing analysis is done for the whole system. Instead, the timing analysis is performed locally for a set of failure traces that are covered by a partial order. Therefore, only a subset of the events of the system is involved and the timing analysis can be carried out very efficiently.

Incremental timing information. Although timed systems provide delays for all the events in the system, often many of the constraints imposed by such delays are not required for the correctness of the system. Because of the iterative nature of the proposed verification approach, timing information is only considered in an *on-demand* basis, as long as it is required to prove the infeasibility in the timed domain of a set of failure traces.

Iterative refinement. As a result of the previous topic, the untimed state space of the system is refined incrementally as long as new timing information is taken into account. This incremental nature of the approach provides a good way to obtain at least partial results even on systems for which complete solutions could be too complex to compute. As a consequence, the approach can be potentially applied to bigger systems or to systems

with more level of detail, than those that can be handled by similar methods for the verification of timed systems.

Back-annotation. A key feature of the proposed verification approach is that it not only proves or disproves the correctness of a timed system with respect to a set of properties. If the system is correct the set of relative timing relations used for the proof are provided. Such relations constitute a set of sufficient timing constraints that guarantee the correctness of the system. On the other hand, if the system is incorrect, a counterexample failure trace is provided. The most important aspect of all this feedback is that it can be used as valuable *back-annotation* information along the design process. Hence, bridging the gap between design and verification. This feature constitutes another differential aspect of our verification approach when compared to other equivalent verification methods.

Automated. The verification approach has been fully implemented into the experimental tool TRANSYT. The tool has proved its functionality as well as the validity of the overall verification approach, by verifying a set of different types of timed asynchronous circuits with millions of untimed states. TRANSYT is available for public download from `http://research.ac.upc.es/VLSI/transyt/transyt.html`.

Compositional methods. Compositional verification provides promising techniques to tackle the complexity in the verification of large and complex systems. In this thesis, compositional methods has been combined with the relative timing-based verification approach in order to tackle the size/complexity issues involved in the verification of complex timed systems. Thus, abstractions, assume-guarantee reasoning and mathematical induction have been used to prove the correctness of a scalable pipelined architecture (IPCMOS). The use of the relative timing-based verification approach has been crucial to prove the correctness of such a complex system. Although some other parametrized systems have been verified in the past, this is the first case in which delay information and refinements down to transistor level of an actual industrial system have been provided.

Finally say that the size/complexity of the systems that can be formally verified is still far from the industry-desired goals. Nevertheless, this thesis has shown that with the proposed methods, relevant systems can be successfully verified.

7.3 Future research

Several issues remain open for future developments of the proposed verification approach. Some of them are related to improvements of the current implementation as well as possible developments to enrich the features of the approach. Whereas other relate to new theoretical challenges.

BDD blow-up. Although BDDs are a good data structure for the representation of symbolic boolean information, they often suffer from a memory blow-up during the inter-

mediate computations, thus limiting the applicability of certain algorithms. Therefore, it would be desirable to experiment with other data structures which provide similar benefits than BDDs and allow better manipulation of bigger sets of states.

On the other hand, an explosion in the size of the BDDs used to represent the transition relations is produced as the number of refinements increases. The main reason for the explosion is the fact that each transition relation is split into several pieces for each condition of the enabling-compatible product. Each piece is manipulated and then the new transition relation is built by joining the different pieces. Although the enabling-compatible product provides a simple mechanism for the iterative refinement, it complicates the BDD representation of the transition relations at each iteration. As a result, some large systems with complex causality relations cannot be verified due to memory requirements. To alleviate this problem, partitioned transition relations could be used. Partitions would correspond to the different pieces in which a transition relation is split for the composition. We plan to incorporate this improvement in the near future, which hopefully would allow us to handle larger and more complex systems for verification.

Partial orders. In a similar vein than the previous topic, in order to reduce the memory requirements during the verification of big systems, partial order techniques could be combined with symbolic methods for state space representation and exploration. Partial orders have proved their efficiency for that purpose in several contexts [GW91, Pel96, VdJL96, ABH+97, BJLY98].

Symbolic timing analysis. It would be interesting to incorporate symbolic algorithms for timing analysis (*e.g.* [AH99]), such that actual delay values are not required for verification. Instead, the verification can be tuned to discover the appropriate delays that make a system correct for a given property. This would open the proposed verification approach to new fields of application close to the design tasks.

Disjunctive causality relations. Currently, CESs can model only conjunctive causality relations. However, the causality relations in a TS can be more general, involving disjunctive causality or combinations of both. As a consequence, our approach may need several refinements in order to cover with various CESs the different causality relations among a set of events. Therefore, it would be desirable to allow the CESs to incorporate other types of causality relations, so that less iterations of the main verification algorithm would be required. The inclusion of disjunctive causality relations in CESs would require to review the notions of enabling-compatibility, the way timing analysis is carried out in a CES, the enabling-compatible product, etc. Moreover, the resulting CESs might result complicated for back-annotation purposes, and trade-offs might be required about the amount of information allowed in a single CES.

Enabling-compatible product. Another interesting feature to enrich the verification approach would be the possibility to quantify the effectiveness of an enabling-compatible

product before actually performing it. This would allow to choose the best LzCESs at each iteration, so that the biggest number of failure traces are pruned, the least possible state splitting is produced, etc.

Back-annotation. The back-annotation information currently produced by the tool consists of a set of LzCESs that contain the relative timing constraints used along the verification process. Some of those constraints may appear several times in different iterations, thus being redundant. Therefore, it would be desirable to have a mechanism to summarize the set of timing constraints and provide them in a more readable form to the user of the tool.

Convergence. No formal study has been carried out about the convergence of the proposed verification approach in the absence of nodal states. Our intuition indicates that the method should generally converge after a bounded number of iterations that guarantee a precise-enough timing analysis. Similar results have been already obtained in the context of marked graphs, where a bounded number of unfoldings suffice to compute the cycle times of a system (see [NK94]). Such detailed study on the topic is left for future work.

Hierarchical verification. All the presented experiments have been performed without any specific of optimization for the different types of systems handled: timed PNs, timed STGs, digital circuits, etc. On the contrary, just a direct translation from the corresponding models into TTSs has been performed, and the generic algorithms of Chapter 4 have been used. For example, a possible source of optimization for circuits could have been the use of hierarchical verification techniques, based on the automatic abstraction of sets of gates in a circuit into complex ones [RCP95].

Automatic abstractions. The abstractions of different components of a system in the compositional approach of Chapter 6 have been derived manually. Also, the chain of assume-guarantee proofs and the required systems for verification have been built manually. Automatic extraction of timed abstractions and automatic derivation of the subsequent chain of reasoning constitute important topics for future research in this area.

Applications. Thanks to the rather theoretical nature of the proposed verification approach, its potential applicability covers a wider range of systems than those presented in the thesis, such as: custom transistor-level circuits that exploit the technology limits for performance, complex digital structures where synchronization is a crucial issue (*e.g.* dynamic MOS), asynchronous and GALS-type systems, real-time systems, etc. Therefore, it would be rather challenging to expose the proposed methods to such complex systems.

Beyond control-dominated systems. In the field of digital circuits, data-path circuitry is fairly easy to design correctly and become reusable once it is designed. On the contrary, the correct design of custom control circuitry can be a very difficult task. Since

verification of the former type of systems can be carried out much more efficiently with theorem provers such a HOL [GM93], for example, our approach for verification concentrates in the latter type of systems. An interesting approach might be to combine both approaches for the verification of systems composed of control and data-path units, applying each approach to solve problems in their respective area of expertise. Some examples in this direction have recently appeared (*e.g.* [KN02]).