Combining Formal Verification and Timing Analysis

(a request for a research grant)

Oded Maler
VERIMAG

January 30, 2002

Abstract

The goal of this project is to extend the domain of applicability of formal verification methodology from functional design toward more lower-level performance sensitive design. We intend to develop new methods for circuit timing analysis based on the *timed automaton* model while taking special care of the scalability requirements implied by the size of industrial-size circuits. We hope that such methods will give better results than static timing analysis methods which are currently in use, and that they could be applied beyond the scope of these methods, for example, to cyclic circuits.

1 Background

The importance of timing analysis in the design of circuits is a commonly-accepted fact, as is the importance of functional verification as a tool to achieve total coverage of all possible behaviors of a design, beyond the capabilities of simulation techniques. Our aim is to apply techniques, inspired by formal verification, to the timing domain using timed automata (TA) to model gates with bi-bounded inertial delays. There are several reasons why formal verification has not yet proliferated to timing analysis:

1. In most companies there is a separation between those dealing with timing and other performance issues (they are closer to designers and have an EE culture) and those working on functional formal verification (coming from a CS background).

2. Timing models are more refined and complex and as such the cost of their analysis is much higher. In particular, the verification of timed automata is considered to be hopelessly intractable (as was verification a decade ago) and less ambitious validation methods with lower complexity are preferred.

3. It takes some time, especially for people with a concrete engineering background, to grasp the technicalities of timed and hybrid automata, and the existing literature on the topic does not always help them by focusing, sometimes, on complex aspects with marginal practical relevance.

The proposed project intends to change this situation by:

1. Develpoing a new user-friendly methodology for modeling circuits with delays as timed automata in a transparent way.

2. Implementing new techniques for analyzing large circuits by successive abstraction and approximation.

3. Applying these techniques to large benchmarks examples in order to find better estimation of the maximal time to stabilization of combinational circuits.

To explain the attrativeness of our approach, let us give a rough sketch of the way timing analysis if currently being performed, while maintaining a separation between the *logical* functionalities of a circuit and the *delay* properties of its components. For example, in the design of clocked synchronous circuits, once we are convinced that the cycle time is large enough for all gate outputs to stabilize, we can perform functional verification of the circuit by ignoring gate and wire delays and treating the whole circuit at the level of abstraction of an *untimed sequential machine*. The common method for determining the stabilization time is by summing up the accumulated delays along the longest path from inputs to latches. Such analysis can be viewed as abstracting away from the logical aspects of the circuit, because it will give the same results for all circuits having the same topology, regardless of the actual identity of their gates.

It is well-known, however, that in reality logic and timing do interact and that certain paths will never be excercised due to logical constraints and that the topological delay is only an upper-bound on the stabilization time. To obtain better estimates many methods for detecting “false paths” have been invented using a variety of techniques. We propose an alterntaive to this practice that, we hope, will lead to better estimates and could be applied to more complex models involving bounded delay uncertainties and cycles in the circuit.

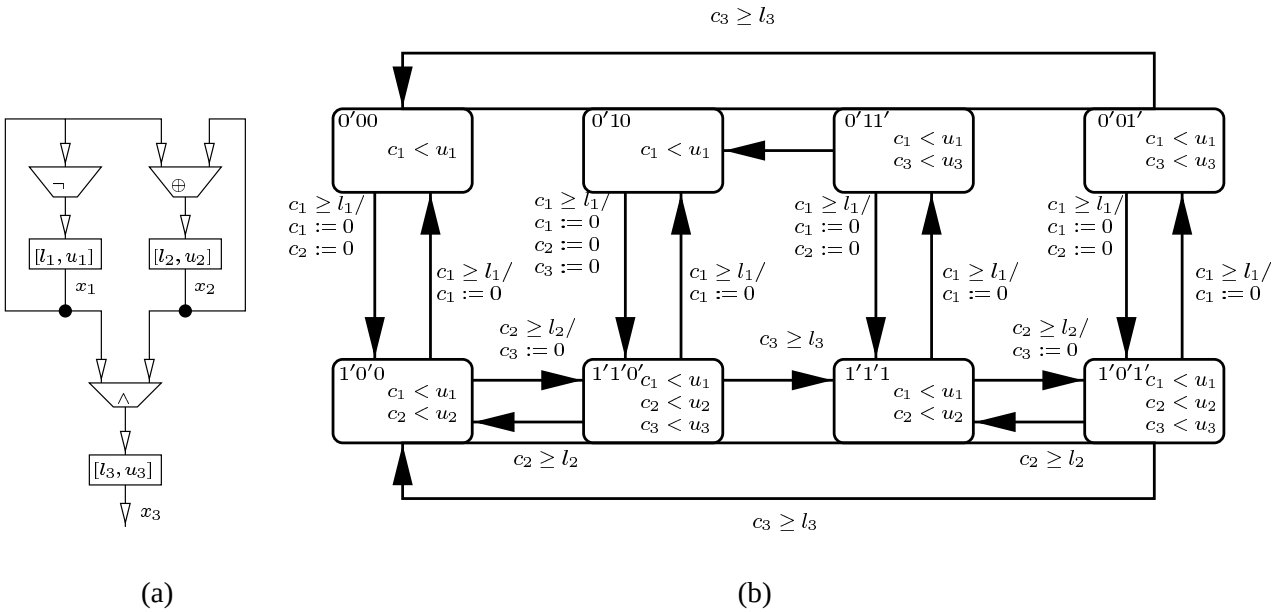


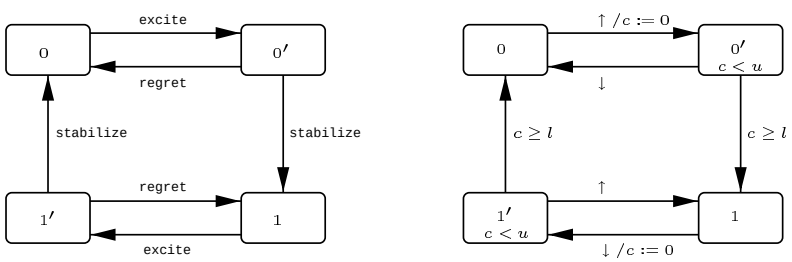
Figure 1: (a) A circuit with an inventer, a XOR gate and an AND gate; (b) The timed automaton model for the circuit.

2 The Modeling Approach

We consider circuits as networks of gates, each of which decomposed into an instantenuous Boolean function and an inertial bi-bounded (non-determinisitc) delay, as the example in Figure 1-(a). The automaton of Figure 2-(b) depicts all the (untimed) behaviors of the delay element: from a stable state (0 or 1) where the input and output agree, the `excite` transition, triggered by a change in the input, puts the automaton in an unstable state ($0'$ or $1'$). This state represents an intermediate phase where the physical process that will eventually lead to switching of the output has been initiated. From there two continuations are possible: either the output switches and the automaton takes a `stabilize` transition (from $0'$ to 1 or from $1'$ to 0) and the new value becomes visible to other gates to which it is connected, or the input changes again before stabilization and the automaton takes a `regret` transition back to a stable state.[1]

The delay parameters $[l, u]$ indicate roughly that 1) every change in the input that persists for u time must be propagated to the output; and 2) every change in the output must be preceded by a change in the input that has persisted for at least l time. Note that this definition allows infinitely-many output signals for a given input. Timed automata model this feature using a fictitious clock variable c which is reset to zero while moving to an excited state and its value determines the possiblity to stay there ($c < u$) or to stabilize ($c \geq l$) (see Figure 2-(b)). While composing the automata for all gates and delay elements of the circuit, a global timed automaton is obtained whose semantics is the set of *all* possible behaviors of the circuits, for all possible choices of delays. A behavior of such a timed automaton consists of an alternating sequence of state transitions and time intervals during which the clock values grow uniformly. The automaton can stay in a state as long the clocks satisfy the staying condition of this state and can make a transition whenever the clocks satisfy the transition guards.

[1] Within the limits of this proposal we do not give more refined models that are closer to the physical reality. Such models will be treated within the project.



(a) (b)

Figure 2: (a) An untimed model of the intertial delay buffer; (b) A timed model.

When a transition excites a state variables it resets its associated clock.

The automaton for the circuit example appears in Figure 1-(b) where the discrete states correspond to the values of x_1, x_2 and x_3. Note, for example, that x_1, being connected to an inverter, is always unstable and that x_2 is always unstable when $x_1 = 1$. In a state where x_i is unstable, its corresponding clock c_i measures the time since excitation. Clearly this automaton captures the logical constraints on the dynamics of the circuit and the clocks restrict further the behavior of the circuit by eliminating behaviors which are inconsistent with the timing constraints. For example, a behavior such as

$$(0'00) \rightarrow (1'0'0) \rightarrow (0'00)$$

where x_1 switches from 0 to 1 and back without a change in x_2 is impossible if $u_2 \leq 2l_1$.

The theory of timed automata shows that, in principle, such models admits the same powerful anaysis techniques as the untimed models used in functional formal verification. In other words, it is possible to check whether some desired property of a circuit holds for all possible input signals and for all possible choices of delay values.[2] In particular, for circuits which stabilize, it is possible to verify that they *always* stabilizae within a given amount of time.

The tool KRONOS (and its "compiled" successor OPENKRONOS) has been developed at VERIMAG within the last decade and is considered as one of the leading tools in the domain. It accepts as an input a set of interacting timed automata and a property to be verified, and by computing reachable states (in a state-space that includes the set of possible clock valuations) it gives a yes/no answer. KRONOS has been applied to the verification of embedded real-time software and communication protocols, to solve scheduling problems and to verify some circuit benachmarks. While some of the experimental results with circuits were encouraging (verification of an 18-stage asynchronous buffer or of a combined Multiplexor and Latch Domino circuit) it became clear that some new development is needed in order to scale up timing verification, and this is exactly the goal of the proposed project.

3 Project Description

We intend to develop a methodology, supported by tools, for timing analysis of large-scale circuits. As a testbed for this approach we will first restrict ourselves to acyclic combinational circuits where the inputs change only at the beginning and all outputs eventually stabilize. The gates of such circuits

[2]Note that by all possible choices we mean that each time a delay element switches it may choose a different number in its $[l, u]$ range – this gives a much better coverage than simulation of "corner cases".

admit a partial-order and the corresponding timed automata are acyclic and admit special analysis techniques. The essence of our approach is summarized bellow.

Let k be the maximal number of gates in a circuit such that its corresponding timed automoaton can be verified by OPENKRONOS (currently between 15 and 20 gates depending on the particular details of the circuit). Starting from the inputs find a sub-circuit of size not larger than k and build its corresponding timed automaton. For each output of this sub-circuit use OPENKRONOS to find lower- and upper-bounds on the time in which it will switch its values (it can do it more than once). From this information, a simplified timed automaton model for each output can be constructed, having the following two properties: 1) It is small (few states and one clock); 2) It is an over-approximation of the circuit model in the sense that all possible behaviors of the circuit can be generated by the automaton. Then pick the next sub-circuit and verify it using the approximate models of its inputs, computed in the previous step. This procedure is repeated until the primary outputs are reached.

The evident advantage of this method is that it can tackle a circuit of any size in a reasonable time. Due to the conservative nature of the approximation, the maximal stabilization time computed by this technique might be larger than the actual maximum, but in most cases it will be smaller than the bound computed by static timing analysis.[3] The technique can be tuned in many ways, for example by choosing different granularities of the the approximated automata (e.g. make two separate automata for two outputs or, if they are very correlated, make a model of both), different methods of partitioning the circuit, measuring absolute or relative time in the approximate model, and more. Our first goal is to implement these ideas and test them on ISCASS95 benchmarks in order to prove their applicability.

4 Milestones

Time (months)	Milestone
$T_0 + 3$	translating circuits from BLIF format to timed automata
$T_0 + 6$	development of partitioning algorithms
$T_0 + 9$	implementing the approximation technique
$T_0 + 12$	tests on the benchmarks
$T_0 + 18$	development of techniques for cyclic circuits
$T_0 + 24$	final results and conclusions

5 Organiztion of Work and Budget

The work will be performed by the timed systems group of VERIMAG under the supervision of Oded Maler. Other members of VERIMAG who will participate are Eugene Asarin, Sergio Yovine, Stavros Tripakis and Marius Bozga, all having a proven record in the theory and application of timed automata. Amir Onueli from Weizmann Institute, a close collaborator of the group, will also participate in the project. At least two PhD students will be dedicated to this project and if budget will allow, an engineer or a post-doc will be hired. The work will be conducted using the computational infra-structure of VERIMAG (including 4 workstation donated recently by Intel). The estimated budget is 50K$ per year, i.e. 100K$.

[3] In some preliminary experiments we tested the technique on a parity circuit with 16 input and 61 XOR gates and obtained, within ten seconds, a result which is 73% of the sum of the delays along the longest path.

6 The Proposers

VERIMAG, headed by Jospeh Sifakis, is an academic laboratory associated with the CNRS (national center for scientific research), UJF (university of Grenoble) and INPG (Grenoble engineering school). It is one of the world leading laboratories in formal verification and related areas. The *timed and hybrid systems group*, led by Oded Maler, specializes in the theory and application of timed automata and in the development of their corresponding verification tools. It includes Sergio Yovine, Stavros Tripakis and Marius Bozga, the major developers of KRONOS and OPENKRONOS and Eugene Asarin, an expert in the mathematics of timed behaviors. The group is currently involved in several national and European projects around timed and hybrid systems.

Oded Maler holds a PhD title from Weizmann Institute (1990) and is currently a "directeur de recherche" at the CNRS (the equivalent of a professor in the research hierarchy). Together with Amir Pnueli he conceived the modeling approach described in this proposal and was involved in many efforts to improve the performance of KRONOS and to adapt it to circuit analysis and to the solution of scheduling problems. He coordinated the European project VHS (Verification of Hybrid Systems), currently coordinates the project CC (Control and Computation) ans is the scientific coordinator of the new project AMETIST (Advanced Methods in Real Time Systems) whose goal is to advance timing technology based on timed automata.

In addition to the concrete technical results, the project will enhance the relations between Intel and VERIMAG, a European research laboratory of a strategic importance, and will facilitate Intel's access to novel research results that might be applicable in the future to other domains such as embedded systems, low-power design, instruction scheduling and CAD for analog circuits.

References

[ABK+97] E. Asarin, M. Bozga, A. Kerbrat, O. Maler, A. Pnueli and A. Rasse, Data-Structures for the Verification of Timed Automata, in O. Maler (Ed.), *Proc. HART'97*, LNCS 1201, 346-360, Springer, 1997.

[ACM02] E. Asarin, P. Caspi and O. Maler, Timed Regular Expressions, *Journal of the ACM* (to appear), 2002.

[AM01] Y. Abdeddaïm and O. Maler, Job-Shop Scheduling using Timed Automata in G. Berry, H. Comon and A. Finkel (Eds.), *Proc. CAV'01*, 478-492, LNCS 2102, Springer 2001.

[AM99] E. Asarin and O. Maler, As Soon as Possible: Time Optimal Control for Timed Automata, in F. Vaandrager and J. van Schuppen (Eds.), *Hybrid Systems: Computation and Control*, LNCS 1569, 19-30, Springer, 1999.

[AMP95] E. Asarin, O. Maler and A. Pnueli, Symbolic Controller Synthesis for Discrete and Timed Systems, in P. Antsaklis et al (Eds.), *Hybrid Systems II*, LNCS 999, 1-20, Springer, 1995.

[AMP98] E. Asarin, O. Maler and A. Pnueli, On the Discretization of Delays in Timed Automata and Digital Circuits, in R. de Simone and D. Sangiorgi (Eds), *Proc. Concur'98*, LNCS 1466, 470-484, Springer, 1998.

[BJMY02] M. Bozga, H. Jianmin, O. Maler and S. Yovine, Verification of Asynchronous Circuits using Timed Automata, *Proc. TPTS'02*, 2002.

[BMPY97] M. Bozga, O. Maler, A. Pnueli, S. Yovine, Some Progress in the Symbolic Verification of Timed Automata, in O. Grumberg (Ed.) *Proc. CAV'97*, 179-190, LNCS 1254, Springer, 1997.

[BMT99] M. Bozga, O. Maler and S. Tripakis, Efficient Verification of Timed Automata using Dense and Discrete Time Semantics, in L. Pierre and T. Kropf (Eds.), *Proc. CHARME'99*, 125-141, LNCS 1703, Springer, 1999.

[DOTY96] C. Daws, A. Olivero, S. Tripakis, and S. Yovine, The Tool KRONOS, in R. Alur, T.A. Henzinger and E. Sontag (Eds.), *Hybrid Systems III*, LNCS 1066, 208-219, Springer, 1996.

[MPS95] O. Maler, A. Pnueli and J. Sifakis, On the Synthesis of Discrete Controllers for Timed Systems, In E.W. Mayr and C. Puech (Eds.), *Proc. STACS '95*, LNCS 900, 229-242, Springer 1995.

[MP95] O. Maler and A. Pnueli, Timing Analysis of Asynchronous Circuits using Timed Automata, in P.E. Camurati, H. Eveking (Eds.), *Proc. CHARME'95*, LNCS 987, 189-205, Springer, 1995.

[MY96] O. Maler and S. Yovine. Hardware Timing Verification using KRONOS, In *Proc. 7th Israeli Conference on Computer Systems and Software Engineering*, Herzliya, Israel, June 1996.

[NTY00] P. Niebert, S. Tripakis and S. Yovine, Minimum-time reachability for timed automata, *Proc. 8th Mediterranean Conference on Control and Automation*, 2000.

[TKYBS98] S. Tasiran, S. P. Khatri, S. Yovine, R.K. Brayton, A. Sangiovanni-Vincentelli, A timed automaton-based method for accurate computation of circuit delay in the presence of cross-talk, *Proc. FMCAD'98*, 149-166, LNCS 1522, Springer, 1998.

[Y97] S. Yovine, Kronos: A verification tool for real-time systems, *International Journal of Software Tools for Technology Transfer* 1, 123-133, 1997.