
RELATIVE TIMING BASED VERIFICATION OF CONCURRENT SYSTEMS

MARCO ANTONIO PEÑA BASURTO

Llicenciat en Informàtica, Technical University of Catalonia, Barcelona, Spain 1993

Llicenciat amb grau en Informàtica, Technical University of Catalonia, Barcelona, Spain 1995

Department of Computer Architecture

Technical University of Catalonia

Barcelona (Spain), February, 2003

**A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor en Informàtica**

To Nuria and Pau

Para mi no hay emoción comparable a la que produce la actividad creadora, tanto en ciencia como en arte, literatura u otras ocupaciones del intelecto humano. Mi mensaje, dirigido sobre todo a la juventud, es que si sienten inclinación por la ciencia, la sigan, pues no dejará de proporcionarles satisfacciones inigualables. Cierto es que abundan los momentos de desaliento y frustración, pero éstos se olvidan pronto, mientras que las satisfacciones no se olvidan jamás.

—Severo Ochoa

It has long been my personal view that the separation of practical and theoretical work is artificial and injurious. Much of the practical work done in computing, both in software and hardware design, is unsound and clumsy because the people who do it do not have any clear understanding of the fundamental principles underlying their work. Most of the abstract mathematics and theoretical work is sterile because it has no contact with the real computing.

—Christopher Strachey

CONTENTS

LIST OF FIGURES	xi
LIST OF TABLES	xv
ABSTRACT	xvii
ACKNOWLEDGMENTS	xix
1 INTRODUCTION	1
1.1 Introduction	2
1.2 Formal methods	3
1.2.1 Formal methods in the design process	4
1.3 Formal verification	6
1.3.1 Verification versus simulation	7
1.3.2 Main approaches to formal verification	8
1.4 Formal verification of timed systems	10
1.5 Overview of the contributions	11
1.6 Structure of the thesis	14
2 MODELS FOR CONCURRENT SYSTEMS	17
2.1 Introduction	18
2.2 Transition systems	18
2.3 Timed transition systems	21
2.4 Lazy transition systems	24
2.5 Petri nets	27
2.5.1 Labeled Petri nets	31
2.6 Conclusions	33
3 VERIFICATION OF TIMED SYSTEMS	35
3.1 Introduction	36
3.2 Quantitative timing information	37
3.3 Timed automata	39
3.4 Timed specifications	42
3.4.1 Temporal logic	42
3.4.2 Timed temporal logic	45

3.5	Verification of timed systems	46
3.5.1	Clock regions	46
3.5.2	Region automata	48
3.5.3	Zone automata	48
3.5.4	Difference-bound matrices	49
3.5.5	Discussion	50
3.6	Petri net-based methods	51
3.7	Conclusions	52
4	VERIFICATION WITH RELATIVE TIMING	53
4.1	Introduction	54
4.1.1	Relative Timing	55
4.2	Overview	57
4.3	Trace semantics	62
4.3.1	Traces and languages	62
4.3.2	Trace-based verification	63
4.3.3	Enabling compatibility	66
4.4	Event structures	70
4.4.1	Timing analysis on event structures	74
4.5	Enabling-compatible product	77
4.5.1	State-based representation of a CES	77
4.5.2	Refining the reachability space by timing constraints	79
4.6	Verification methodology	81
4.6.1	Iterative refinement	83
4.6.2	Off-line timing analysis of failures	85
4.6.3	Incorporation of relative timing constraints	86
4.6.4	Back-annotation	87
4.6.5	Correctness	87
4.6.6	Convergence	88
4.7	Conclusions	90
5	EXPERIMENTAL RESULTS	93
5.1	A brief introduction to TRANSYT	94
5.1.1	Representation of LzTSs with boolean algebras	95
5.1.2	TRANSYT input format	98
5.2	An example with forward unfolding	102
5.2.1	Model of a timed PN	104
5.2.2	Verification	106
5.2.3	Discussion	111
5.3	Verification of complex-gate decompositions in speed-independent circuits	112
5.3.1	Speed-independent circuits	112
5.3.2	Experimental set-up	113
5.3.3	The <code>sbuf-read-ctl</code> controller	114

5.3.4	Model of an STG	115
5.3.5	Model of the circuit	117
5.3.6	Specification of properties	118
5.3.7	Verification	121
5.3.8	Results and discussion	125
5.4	Verification of relative timing assumptions	127
5.4.1	Synthesis of asynchronous circuits with relative timing assumptions	128
5.4.2	The VME bus controller	131
5.4.3	Models and properties	134
5.4.4	Verification	137
5.4.5	Discussion	142
5.5	Conclusions	144
6	COMPOSITIONAL VERIFICATION	147
6.1	Introduction	148
6.2	The IPCMOS architecture	149
6.2.1	IPCMOS pipelines	149
6.2.2	Strobe circuit	152
6.2.3	Reset circuit	153
6.2.4	Valid circuit	155
6.2.5	The environment modules	155
6.2.6	About complexity	156
6.3	Compositional verification	156
6.3.1	Framework	158
6.4	Verification of IPCMOS pipelines	159
6.4.1	Verification strategy	159
6.4.2	Abstractions	161
6.4.3	Assume-guarantee verification	162
6.5	Verification of a stage	165
6.5.1	Modeling CMOS circuits	165
6.5.2	Modeling IPCMOS circuits	169
6.5.3	Verification results	170
6.6	Conclusions	173
7	CONCLUSIONS	175
7.1	Introduction	176
7.2	Contributions	177
7.3	Future research	179
A	TIMING ANALYSIS	183
A.1	Introduction	184
A.2	Timing analysis on acyclic graphs	186

B	ON THE ENABLING-COMPATIBLE PRODUCT	189
B.1	Enabling-compatible product	190
B.1.1	State-based representation of a CES	190
B.1.2	Refining the reachability space by timing constraints	191
B.2	Symbolic representation	192
B.2.1	Encoding of a LzTS	193
B.2.2	Encoding the state space of a LzCES	193
B.3	Computation of the new transition relations	194
B.3.1	Transitions entering the timed domain	194
B.3.2	Staying inside the timed domain: no synchronization	195
B.3.3	Staying inside the timed domain: synchronization	195
B.3.4	Transitions re-entering the timed domain	196
B.3.5	Exiting or staying outside the timed domain	196
B.3.6	New transition relation	196
B.3.7	Lazy events	196
B.3.8	Initial state	197
C	VERIFICATION-RELATED COMMANDS	199
C.1	Failure analysis	200
C.1.1	The <code>add_fail</code> command	200
C.1.2	The <code>check_fails</code> command	200
C.1.3	The <code>print_fails</code> command	201
C.2	Analysis of delay relations	201
C.3	The <code>uverif</code> command	202
C.4	The <code>tverif</code> command	203
C.4.1	Output	204
C.4.2	Construction of the failure trace	209
C.4.3	Construction of the LzCES	209
C.4.4	Timing analysis	211
C.4.5	Miscellaneous	211
C.4.6	Summary of the <code>tverif</code> command	212
	REFERENCES	215
	GLOSSARY OF SYMBOLS	227

LIST OF FIGURES

1	INTRODUCTION	1
1.1	Automaton modeling a modulo 3 counter.	4
1.2	Iterative design flow.	6
1.3	The theorem proving approach.	9
1.4	The model checking approach.	10
2	MODELS FOR CONCURRENT SYSTEMS	17
2.1	An example of transition system.	19
2.2	An example of timed transition system.	21
2.3	Portion of the timed state space of a TTS.	23
2.4	An example of lazy transition system.	25
2.5	Relations among the main notions related to transition systems.	27
2.6	An example of Petri net.	28
2.7	Reachability graph of a Petri net	31
2.8	Petri net of Figure 2.6 with labeled transitions.	32
3	VERIFICATION OF TIMED SYSTEMS	35
3.1	Three representations of time.	38
3.2	An example of timed automaton.	40
3.3	A simple automaton with atomic propositions.	43
3.4	Regions for two clocks.	47
3.5	Region automaton.	49
3.6	Zone automaton.	50
4	VERIFICATION WITH RELATIVE TIMING	53
4.1	Relative timing in the synthesis of circuits.	56
4.2	Example of verification with relative timing.	59
4.3	Example of verification with relative timing (first iteration).	60
4.4	Example of verification with relative timing (second iteration).	61
4.5	From traces to language refinement.	63
4.6	Relation between runs and traces.	64
4.7	Enabling and disabling in a trace.	65

4.8	Circuit with a potential disabling.	67
4.9	Enabling-compatible and non-enabling-compatible mapping.	68
4.10	Symmetric vs. asymmetric disabling.	71
4.11	Timing analysis over a CES and resulting lazy CES.	74
4.12	Graphs of reachable configurations and enablings.	78
4.13	Step by step refinement by enabling-compatible product.	82
4.14	Flow of the verification methodology.	83
4.15	Main algorithm of the relative timing-based verification approach.	84
4.16	Algorithm for the derivation of a LzCES from a trace.	84
4.17	Generation of the sufficient shortest suffix of a trace.	85
4.18	Example of a nodal and a not nodal point.	89
5	EXPERIMENTAL RESULTS	93
5.1	A binary-encoded LzTS.	98
5.2	TRANSYT input file for the LzTS of Figure 5.1.	100
5.3	<i>Yoneda's</i> example.	103
5.4	Transition of an PN with its input and output places.	104
5.5	<i>Yoneda's</i> example: first and second refinements.	107
5.6	<i>Yoneda's</i> example: third and forth refinements.	108
5.7	<i>Yoneda's</i> example: counterexample trace proving incorrectness.	110
5.8	System vs. specification verification scheme.	113
5.9	Input-output interface of the <code>sbuf-read-ctl</code> controller.	114
5.10	STG specification of the <code>sbuf-read-ctl</code> controller.	116
5.11	Two implementations of the <code>sbuf-read-ctl</code> controller.	118
5.12	TRANSYT input file for the circuit of Figure 5.11 (b).	119
5.13	Three refinements for the <code>sbuf-read-ctl</code> controller.	124
5.14	An example of relative timing assumptions.	129
5.15	VME bus controller.	132
5.16	Timing assumptions for the synthesis of the VME bus controller.	133
5.17	Implementation of the VME bus controller with timing assumptions.	134
5.18	TRANSYT input files for the VME bus controller.	135
5.19	First four refinements for the VME bus controller.	141
5.20	Last four refinements for the VME bus controller.	143
6	COMPOSITIONAL VERIFICATION	147
6.1	General block-level interlocking scheme.	149
6.2	Linear IPCMOS architecture.	150
6.3	Detailed 2-stage IPCMOS pipeline and waveform of its behavior.	151
6.4	Two-phase handshake mechanism.	152
6.5	The <i>strobe</i> circuit in detail.	153
6.6	The <i>reset</i> circuit in detail.	154

6.7	The <i>valid</i> circuit in detail.	155
6.8	STGs modeling the pulse-based behavior of the <i>IN</i> and <i>OUT</i> modules.	155
6.9	Assume-guarantee verification using abstractions.	157
6.10	Pipeline verification using abstractions.	160
6.11	STGs modeling the abstractions A_{in} and A_{out} .	161
6.12	Scheme of the <i>guarantee</i> part of the verification.	163
6.13	LzCES used to prove correctness of the <i>strobe switch</i> circuit.	170
A	TIMING ANALYSIS	183
A.1	Classes of timing analysis problems.	184
A.2	Timing analysis on an acyclic graph.	187
C	VERIFICATION-RELATED COMMANDS	199
C.1	An example of DOT file for a failure trace.	205
C.2	An example of DOT file for a LzCES.	207
C.3	An example of DOT file for the lazy state space of a system.	208

LIST OF TABLES

5	EXPERIMENTAL RESULTS	93
5.1	Failure situations in <code>sbuf-read-ctl</code> along the verification.	122
5.2	Experimental results for the verification of asynchronous circuits.	125
5.3	Failure situations in the VME bus controller along the verification.	139
6	COMPOSITIONAL VERIFICATION	147
6.1	Summary of the results for the 5 steps of the verification.	164
6.2	Model of the <i>strobe</i> circuit.	166
6.3	Model of the <i>strobe switch</i> circuit.	166
6.4	Model of the <i>reset</i> circuit.	167
6.5	Model of the <i>reset switch</i> circuit.	167
6.6	Model of the <i>valid</i> circuit.	167

ABSTRACT

The thesis presents a new theory and methodology for the formal verification of safety properties in timed systems. The correct operation of such systems not only depends on a set of functional properties but also on certain assumptions about the delays of the components of the system and the response times of the environment in which the system operates. The verification of this type of systems typically involves several computationally hard problems. In particular, the combinatorial state explosion problem becomes exacerbated by the time dimension.

The theory that supports the proposed verification approach extends the conventional BDD-based symbolic methods to the verification of timed systems, modeled by means of *timed transition systems*. The theory is based on the *relative timing* paradigm, which instead of considering exact time differences in the occurrence of events, considers the effect of delays in terms of relative orderings between events. For example, in order to guarantee that a *race* is not propagated in a digital circuit, it is often sufficient to check that certain signal switches before another, instead of identifying the exact instants of time in which both signals switch. Moreover, the timing information does not need to be computed for the overall system, but only *locally* for the part of the system involved in the proof or disproof of a given property. This is possible thanks to a crucial observation, that the set of executions of a transition system can be covered by a set of partial orders. As a consequence, only a subset of the events of the system is involved in the proof of a property and the timing analysis can be carried out very efficiently.

Conventional methods for the verification of timed systems rely on the computation of the exact timed state space of the system as the first step of the analysis. Although efficient techniques have been devised to overcome the complexity issue (*e.g.* difference bound matrices), symbolic methods cannot be easily applied. Thus, the combinatorial time-state explosion problem often limits the applicability of such methods to moderate-size systems.

Instead, the approach proposed in the thesis relies on an incremental refinement of the untimed state space of the system, so that timing information is incorporated as soon as it is needed. The timing information is derived by an efficient *off-line* timing analysis over small sets of events. The refined state space is captured under the model of *lazy transition*

system, which allows an efficient representation of the timed domain using conventional symbolic methods. 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. Moreover, the 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.

A key feature of the proposed verification approach is that not only proves or disproves the correctness of a timed system. 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. This feature, which allows to bridge the gap between verification and design, constitutes another differential aspect of our verification approach when compared to other equivalent verification methods.

The verification approach has been fully implemented in an experimental CAV tool called TRANSYT. The tool can handle hierarchical and distributed modular systems which can inter-operate by a variety of communication mechanisms. TRANSYT has successfully proved its functionality as well as the validity of the overall verification approach, by verifying a number of timed asynchronous circuits with up to more than 10^6 untimed states. The experiments cover, for example, the verification of: complex-gate decompositions in quasi-speed-independent asynchronous circuits, delay-reset domino circuits, pulse-based systems, circuits optimized for speed using timing assumptions, etc. Additionally, compositional verification methods have been 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 have been used to prove the correctness the IPCMOS architecture. It is a scalable pipelined architecture which is aimed to the interconnection of different clock zones in a system.

Thanks to the rather theoretical nature of the proposed verification approach, its potential applicability covers a wider range of systems than those cited above, 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.

ACKNOWLEDGMENTS

If I have seen further than others, it is because I was standing upon the shoulders of giants.

—Isaac Newton - Letter to a friend, 1676

The first person that deserves my acknowledgment is my supervisor Jordi Cortadella. He introduced me to the world of asynchronous circuits, Petri nets, formal verification, etc. back in 1993 when I was still an undergraduate student. His deep insight into the subject of this thesis provided me a lot of helpful suggestions. Without his guidance and kind encouragement this thesis would have never been possible.

I am specially indebted to my other supervisor, Enric Pastor. His continuous support and friendship have helped me to overcome the (technical and personal) difficulties during the critical phases of this work.

My gratitude also to Alex Kondratyev and Alexander Smirnov for their contributions to the theoretical soundness and the practical implementation of this work, respectively. And to Luciano Lavagno, Alex Yakovlev and Alexander Taubin, for their kindness hosting me in respective visits to the *Politecnico di Torino*, the University of Newcastle and the University of Aizu. The numerous insightful discussions with them about different research topics have contributed this thesis in a number of ways.

Thanks to the other members of the CAD/VLSI group at the Department of Computer Architecture of the Technical University of Catalonia —Rosa Badia, Fermín Sánchez and Josep Carmona— and former members —Oriol Roig and Enric Musoll. Along these years they have provided me with the kind of environment that makes work a much more pleasant experience.

Thanks also to the reviewers at Department of Computer Architecture —Rosa Badia, Antonio González and Antonio Juan Hormigo— and the external reviewers —Abelardo Pardo and Supratik Chakraborty. They read carefully the preliminary versions of this thesis giving me valuable suggestions on the contents and the presentation of this work. And the members of the thesis committee for the effort they put into judging this thesis.

On the institutional and industry side I would like to acknowledge: the Ministry of Science and Technology of Spain under contracts CICYT-TIC 95-0419, CICYT-TIC 98-0410-CO-01 and CICYT-TIC 2001-2476-CO-03; the ACiD Working Group under contracts ESPRIT-7225, ESPRIT-21949 and IST-1999-29119; and Intel Corporation. They all are gratefully acknowledged for funding this research.

On the personal side my family deserves infinite gratitude: my father Antonio, my mother Adoración and my brother Jose. Thanks a lot for your efforts on infusing me the values of a good education and supporting all my studies along the years.

And the most effusive thanks to Nuria. During all this time she has been my best friend, providing lots of emotional support and love, and patiently suffering the numerous moments of solitude she has been forced to because of my work. This thesis is dedicated to her and our beloved son Pau.

Finally I would like to express my thanks to everyone I have not cited above but has helped me, directly or not, in the long way until this thesis has been completed. Thanks a lot to you all.