TEHCNOLOGY MAPPING OF TIMED ASYNCHRONOUS CIRCUITS

by

Curtis A. Nelson

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Electrical and Computer Engineering

The University of Utah

December 2003

Copyright © Curtis A. Nelson 2003

All Rights Reserved

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a dissertation submitted by

Curtis A. Nelson

This dissertation has been read by each member of the following supervisory committee and by majority vote has been found to be satisfactory.

Chair: Chris J. Myers

Ken Stevens

Erik Brunvand

Reid R. Harrison

Behrouz

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

FINAL READING APPROVAL

To the Graduate Council of the University of Utah:

I have read the dissertation of Curtis A. Nelson in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the Supervisory Committee and is ready for submission to The Graduate School.

Date

Chris J. Myers
Chair, Supervisory Committee

Approved for the Major Department

Chris J. Myers
Chair/Dean

Approved for the Graduate Council

David S. Chapman
Dean of The Graduate School

ABSTRACT

This paper presents an efficient method for verifying hazard-freedom in timed asynchronous circuits. Timed circuits are a class of asynchronous circuits that utilize explicit timing information for optimization throughout the entire design process. In asynchronous circuits, correct operation requires that there are no hazards in the circuit implementation. Therefore, when designing an asynchronous circuit, each internal node and output of the circuit must be verified for hazard-freedom to ensure correct operation. Current verification algorithms for timed asynchronous circuits require an explicit state exploration often resulting in state explosion for even modest sized examples. The goal of this work is to abstract the behavior of internal nodes and utilize this information to make a conservative determination of hazard-freedom for each node in the circuit. Experimental results indicate that this approach is substantially more efficient than existing timing verification tools. These results also indicate that this method scales well for large examples in that it is capable of analyzing circuits in less than a second that could not be previously analyzed. While this method is conservative in that some false hazards may be reported, our results indicate that the number of false hazards is small.

To all the insults that have made me who I am

CONTENTS

ABSTRACT . . . . . **iv**

LIST OF TABLES . . . . . **viii**

LIST OF FIGURES . . . . . **ix**

CHAPTERS

1. INTRODUCTION **1**

1.1 Synchronous Design Flow 3
1.2 Asynchronous Design Flow 4
1.3 Verification 5
1.4 Synchronous Technology Mapping 6
1.5 Asynchronous Technology Mapping 11
1.5.1 Delay Insensitive Circuits 11
1.5.2 Speed Independent Circuits 11
1.5.3 Fundamental Mode Circuits 13
1.5.4 Timed Circuits 15
1.6 Contributions 16
1.7 Dissertation Overview 18

2. CIRCUIT SPECIFICATIONS AND SEMANTICS **20**

2.1 Logic Expressions 20
2.2 Time Petri Nets 22
2.3 Netlist 24
2.4 State Graphs 25
2.5 Design Flow Completion 27
2.6 Another Example 29
2.7 Summary 30

3. DETERMINATION OF HAZARD-FREEDOM **31**

3.1 Formal Definitions 32
3.2 Checking equivalence 32
3.3 Finding stable states 39
3.3.1 Untimed stabilization 41
3.3.2 Timed stabilization 42
3.4 Checking for hazard-freedom 47

4. DECOMPOSITION **49**

5. MATCHING AND COVERING **50**

6. EXPERIMENTAL RESULTS **51**

7. CONCLUSIONS **54**

7.1 Future Work 54

REFERENCES **56**

LIST OF TABLES

6.1 Comparison of standard benchmarks against other timing verification tools. 52

6.2 Comparison for decomposed netlists. 53

LIST OF FIGURES

1.1 Synchronous design flow. 3

1.2 Asynchronous design flow. 4

1.3 (a) 4-input AND gate. (b) A non-symmetric decomposition. (c) A symmetric decomposition. 8

1.4 (a) 4-input AND gate. (b) A non-symmetric decomposition. (c) A symmetric decomposition. 10

1.5 (a) 3-input AND function. (b) A circuit netlist that is hazardous under the speed-independent model. (c) A circuit netlist that is hazard-free under the speed-independent model. (d) State graph for 3-input AND function. 12

1.6 (a) 3-input AND function. (b) A circuit netlist that is hazardous under the speed-independent model. (c) A circuit netlist that is hazard-free under the speed-independent model. (d) Petri net for the 3-input AND function. 14

2.1 Expanded timed asynchronous technology mapping design flow. 21

2.2 (a) POS form of the logic expressions. (b) A CMOS implementation with weak feedback. 22

2.3 (a) An example TPN. (b) A circuit netlist that is hazardous under the speed-independent model. (c) A circuit netlist that is hazard-free under the speed-independent model. (d) A state graph. (e) A complex gate equivalent circuit. 24

2.4 State graph for the network of Figure 2.3(c). 27

2.5 (a) A Petri net of a C-element. (b) A netlist of a C-element. (c) A state graph of a C-element. (d) A complex gate equivalent netlist of a C-element. 29

3.1 Top-level algorithm for verification. 32

3.2 Algorithm for finding stable states. 40

3.3 Untimed stabilization algorithm. 41

3.4 Timed stabilization algorithm. 43

3.5 Algorithm to update the zone. 44

3.6 Zone creation and evolution. 45

3.7 Timed stabilization recursion. 46

3.8 Algorithm to check acknowledgment. 47
3.9 Algorithm to check monotonicity. 48