
**HIGH LEVEL SYNTHESIS OF ASICs
UNDER TIMING AND
SYNCHRONIZATION
CONSTRAINTS**

**THE KLUWER INTERNATIONAL SERIES
IN ENGINEERING AND COMPUTER SCIENCE**

**VLSI, COMPUTER ARCHITECTURE AND
DIGITAL SIGNAL PROCESSING**

Consulting Editor
Jonathan Allen

Latest Titles

Hardware Annealing in Analog VLSI Neurocomputing, B. W. Lee, B. J. Sheu

ISBN: 0-7923-9132-2

Neural Networks and Speech Processing, D. P. Morgan, C.L. Scofield

ISBN: 0-7923-9144-6

Silicon-on-Insulator Technology: Materials to VLSI, J.P. Colinge

ISBN: 0-7923-9150-0

Microwave Semiconductor Devices, S. Yngvesson

ISBN: 0-7923-9156-X

A Survey of High-Level Synthesis Systems, R. A. Walker, R. Camposano

ISBN: 0-7923-9158-6

Symbolic Analysis for Automated Design of Analog Integrated Circuits,

G. Gielen, W. Sansen,

ISBN: 0-7923-9161-6

High-Level VLSI Synthesis, R. Camposano, W. Wolf,

ISBN: 0-7923-9159-4

*Integrating Functional and Temporal Domains in Logic Design: The False Path
Problem and its Implications*, P. C. McGeer, R. K. Brayton,

ISBN: 0-7923-9163-2

Neural Models and Algorithms for Digital Testing, S. T. Chakradhar,

V. D. Agrawal, M. L. Bushnell,

ISBN: 0-7923-9165-9

Monte Carlo Device Simulation: Full Band and Beyond, Karl Hess, editor

ISBN: 0-7923-9172-1

The Design of Communicating Systems: A System Engineering Approach,

C. J. Koomey

ISBN: 0-7923-9203-5

Parallel Algorithms and Architectures for DSP Applications,

M. A. Bayoumi, editor

ISBN: 0-7923-9209-4

Digital Speech Processing: Speech Coding, Synthesis and Recognition

A. Nejat Ince, editor

ISBN: 0-7923-9220-5

Sequential Logic Synthesis, P. Ashar, S. Devadas, A. R. Newton

ISBN: 0-7923-9187-X

Sequential Logic Testing and Verification, A. Ghosh, S. Devadas, A. R. Newton

ISBN: 0-7923-9188-8

Introduction to the Design of Transconductor-Capacitor Filters,

J. E. Kardontchik

ISBN: 0-7923-9195-0

The Synthesis Approach to Digital System Design, P. Michel, U. Lauther, P. Duzy

ISBN: 0-7923-9199-3

Fault Covering Problems in Reconfigurable VLSI Systems, R. Libeskind-Hadas,

N. Hasan, J. Cong, P. McKinley, C. L. Liu

ISBN: 0-7923-9231-0

**HIGH LEVEL SYNTHESIS OF ASICs
UNDER TIMING AND
SYNCHRONIZATION
CONSTRAINTS**

by

David C. Ku

Redwood Design Automation
&
Stanford University

and

Giovanni De Micheli
Stanford University



Springer Science+Business Media, LLC

Library of Congress Cataloging-in-Publication Data

Ku, David C., 1964

High level synthesis of ASICs under timing and synchronization constraints / by David C. Ku and Giovanni De Micheli.

p. cm. -- (Kluwer international series in engineering and computer science ; 177)

Includes bibliographical references and index.

ISBN 978-1-4419-5129-8 ISBN 978-1-4757-2117-1 (eBook)

DOI 10.1007/978-1-4757-2117-1

1. Application specific integrated circuits--Design and construction--Data processing. 2. computer-aided design. I. De Micheli, Giovanni. II. Title. III. Series: Kluwer international series in engineering and computer ; SEC 177.

TK874.6.K8 1992

621.381'5--dc20

92-10571

CIP

Copyright © 1992 by Springer Science+Business Media New York
Originally published by Kluwer Academic Publishers in 1992
Softcover reprint of the hardcover 1st edition 1992

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photo-copying, recording, or otherwise, without the prior written permission of the publisher, Springer Science+Business Media, LLC.

Printed on acid-free paper.

To our loving families.

Contents

1	Introduction	1
1.1	Overview of High-level Synthesis	2
1.2	Motivation and Issues in High-level Synthesis	3
1.2.1	Example of an ASIC design	4
1.2.2	Requirements of ASIC synthesis	5
1.3	Research Objectives	6
1.3.1	Organization of synthesis system	7
1.3.2	New synthesis algorithms	9
1.4	Summary of Related Work	13
1.4.1	Overview of existing approaches	13
1.4.2	Critical review and research contributions	15
1.5	Assumptions and Limitations	16
1.6	Outline of the Book	16
2	System Overview	19
2.1	Modeling Hardware Behavior	19
2.1.1	Models	21
2.1.2	Constants and variables	24
2.1.3	Language features	26
2.1.4	Timing semantics	31
2.1.5	A HardwareC example	36
2.2	Overview of the Synthesis Flow	42
2.2.1	Hercules: Behavioral synthesis	44
2.2.2	Hebe: Structural synthesis	44
3	Behavioral Transformations	47
3.1	Behavioral Intermediate Form	48

3.2	User-driven Behavioral Transformations	49
3.3	Automatic Behavioral Transformations	51
3.3.1	Reference stack	53
3.3.2	Combinational logic coalescing	58
3.4	Summary of Behavioral Transformations	60
4	Sequencing Graph and Resource Model	61
4.1	Basic Definitions	62
4.1.1	Hierarchy in the model	64
4.1.2	Data-dependent delay operations	66
4.1.3	Sequencing graph examples	68
4.2	Timing Constraints	69
4.2.1	Properties of constraint graph	73
4.3	Hardware Resources	75
4.3.1	Functional units	76
4.3.2	Registers	77
4.3.3	Multiplexers	80
4.4	Summary of Sequencing Graph Model	80
5	Design Space Exploration	83
5.1	Design Space Formulation	85
5.1.1	Background	85
5.1.2	Defining the design space	90
5.2	Concurrency Factor	93
5.2.1	Computation for flat graphs	94
5.2.2	Computation for hierarchical graphs	97
5.2.3	Conflict-free resource allocation	101
5.3	Design Space Exploration Strategy	102
5.3.1	Exact search strategy	103
5.3.2	Heuristic search strategy	104
5.4	Summary of Design Space Exploration	110
6	Relative Scheduling	113
6.1	Introduction to Scheduling	117
6.2	Relative Scheduling Formulation	118
6.2.1	Well-Posedness of timing constraints	121
6.2.2	Properties of relative schedule	127
6.2.3	Relevant and irredundant anchor sets	130
6.3	Algorithms for Relative Scheduling	141

6.3.1	Finding anchor sets	142
6.3.2	Checking well-posed	144
6.3.3	Making well-posed	145
6.3.4	Removing redundant anchors	146
6.3.5	Iterative incremental scheduling	148
6.4	Analysis of Algorithms	156
6.4.1	Analyzing making well-posed	156
6.4.2	Analyzing iterative incremental scheduling	158
6.5	Summary of Relative Scheduling	161
7	Resource Conflict Resolution	163
7.1	Conflict Resolution Formulation	164
7.1.1	Objective in conflict resolution	166
7.2	Constraint Topology	166
7.2.1	Orientation and polarization	169
7.2.2	Properties of polarizations	173
7.3	Algorithms for Conflict Resolution	174
7.3.1	Heuristic ordering search	176
7.3.2	Exact ordering search	179
7.4	Summary of Conflict Resolution	181
8	Relative Control Generation	183
8.1	Adaptive Control Approach	187
8.1.1	Basic adaptive control strategy	187
8.1.2	Simple adaptive control implementation	191
8.1.3	Precise adaptive control implementation	195
8.1.4	Analysis of adaptive control	201
8.2	Relative Control Approach	204
8.2.1	Relative control for non-anchors	205
8.2.2	Relative control for anchors	209
8.2.3	Analysis of relative control	211
8.3	Summary of Control Generation	212
9	Relative Control Optimization	213
9.1	Control Optimization Criterion	214
9.1.1	Defining the cost function	216
9.1.2	Control optimization objective	216
9.2	Redundancy in Synchronization	217
9.2.1	Making Anchors Redundant	218

9.2.2	Prime versus Non-prime Anchors	221
9.3	Control Optimization Approach	223
9.3.1	Minimize Prime Anchor Sets	224
9.3.2	Partition to Resynchronize Operations	226
9.3.3	Making the Graph Taut	229
9.3.4	Example	232
9.4	Analysis of Control Resynchronization	234
9.5	Summary of Control Resynchronization	235
10	System Implementation	237
10.1	Overview of Hercules	239
10.1.1	Data organization in Hercules	240
10.1.2	Command organization in Hercules	241
10.2	Overview of Hebe	243
10.2.1	Command organization in Hebe	245
10.2.2	Design subsystems in Hebe	247
10.3	Summary of System Implementation	252
11	Experimental Results	253
11.1	Design Experiences	254
11.1.1	Ethernet co-processor	254
11.1.2	Digital audio I/O chip	258
11.1.3	Bidimensional discrete cosine transform	260
11.1.4	Raster line design	262
11.1.5	Error correcting code	264
11.1.6	Greatest common divisor	268
11.2	Synthesis of Benchmark Examples	270
11.3	Summary of Experimental Results	273
12	Conclusions and Future Work	275
12.0.1	Summary of Major Results	275
12.1	Future Work	277
12.1.1	Current limitations	278
12.1.2	Future directions	278
	References	281
	Index	291

Preface

Computer-aided synthesis of digital circuits from behavioral level specifications offers an effective means to deal with the increasing complexity of digital hardware design. This book addresses both theoretical and practical aspects in the design of a high-level synthesis system that transforms a behavioral level description of hardware to a synchronous logic-level implementation consisting of logic gates and registers.

The present level of maturity of synthesis techniques for digital circuits has been proven by the large number of designs achieved by means of computer-aided tools. Synthesis from behavioral descriptions can leverage the expressive power of high-level models and shorten the design time of a circuit. High-level optimization algorithms can be extremely powerful in making decisions about the macroscopic structure of a circuit, with beneficial effects on the overall performance, area and testability.

Synthesis techniques provide a competitive edge in the design of *Application Specific Integrated Circuits* (ASICs), which represent a growing sector of the semiconductor market. In particular, high-level synthesis can be extremely useful to ASIC design, because of the paramount importance of short time to market.

This book addresses specific issues in applying high-level synthesis techniques to the design of ASICs. This complements previous results achieved in synthesis of general-purpose and signal processors, where *data-path* design is of utmost importance. In contrast, ASIC designs are often characterized by complex *control* schemes, to support communication and synchronization with the environment. The combined design of efficient data-path and control-unit is the major contribution of this book.

Three requirements are important in modeling ASIC designs: *concurrency*, *external synchronization*, and *detailed timing constraints*. The objective of the research work presented here is to develop a hardware model incorporating these requirements as well as synthesis algorithms that operate on this hardware model.

The contributions of this book address both theory and implementation of algorithms for hardware synthesis. The following novel synthesis algorithms have been developed: *relative scheduling*, a scheduling formulation that supports external synchronizations and detailed timing constraints; *constrained conflict resolution*, a method to resolve resource conflicts under timing constraints; *relative control synthesis*, a control generation approach for relative scheduling; *relative control optimization*, a novel optimization strategy that minimizes the size of the control implementation under timing constraints.

In addition, we developed a hardware description language called *HardwareC* with particular attention to ASIC design specification issues. The system implementation of the algorithms is provided by programs *Hercules* and *Hebe*, which serve as the front-end to Stanford's Olympus Synthesis system. They have been applied to the synthesis of benchmark examples and to some chip designs at Stanford University.

D. C. Ku
G. De Micheli

Acknowledgements

The high-level synthesis research described in this book is part of an on-going synthesis research project at Stanford University called the *Olympus* synthesis system. We would like to acknowledge and thank the many people who have contributed to this research.

We would like to acknowledge the significant contributions of David Filo to the project. He co-developed the theory behind the control resynchronization optimization, described in Chapter 9 of this book. He implemented control optimization and register folding in program Hebe. Other supporting tools of the *Olympus* suite were developed by Frédéric Mailhot, who was responsible for the logic-level framework, which included technology mapping, logic simulation, and the SLIF interchange format and by Thomas Truong, who wrote a simulator for SIF and a graphic display package.

Rajesh Gupta spent a vast amount of time generating design and benchmark examples in HardwareC, working with Claudionor Coelho on the network co-processor and with Thomas Truong on the second version of the DAIO chip. Dave Kastle designed the MAMA decoder chip. Michiel Ligthart designed the first version of the DAIO chip. We are indebted to them for their tenacious work in using the prototype versions of the synthesis system. We also thank Polly Siegel, Jerry Yang and Maurizio Damiani for their suggestions and criticisms.

We would like to thank Daniel Weise and Teresa Meng for their comments and criticisms, and Joe Kostelec, Uzi Bar-Gadda and Rindert Schutten of Phillips Research Laboratory for their industrial perspectives. Most important of all, we would like to thank our families for their love, understanding, and encouragement. None of this would have been possible without their support.

This research was sponsored by a fellowship provided by Phillips/Signetics, by NSF/ARPA, under grant No. MIP 8719546, and by AT&T and DEC jointly with NSF, under a Presidential Young Investigator Award program.