1985 Chapel Hill Conference on Very Large Scale Integration

Edited by Henry Fuchs

83

73.715/1023
C 462

1985 Chapel Hill Conference on Very Large Scale Integration

Edited by Henry Fuchs

1988 5

COMPUTER SCIENCE PRESS

8850135

Copyright © 1985 Computer Science Press, Inc.

Printed in the United States of America.

All rights reserved. No part of this book may be reproduced in any form, including photostat, microfilm, and xerography, and not in information storage and retrieval systems, without permission in writing from the publisher, except by a reviewer who may quote brief passages in a review or as provided for in the Copyright Act of 1976.

Computer Science Press, Inc.
1803 Research Boulevard
Rockville, Maryland 20850

1 2 3 4 5 6 90 89 88 87 86 85

Library of Congress Cataloging in Publication Data

Chapel Hill Conference on Very Large Scale Integration (1985)
1985 Chapel Hill Conference on Very Large Scale Integration.

Includes index.
1. Integrated circuits—Very large scale integration —Congresses. I. Fuchs, Henry, 1948– . II. Title
TK7874.C43 1985 621.3819'5835 85-7819
ISBN 0-88175-103-0

1985 Chapel Hill Conference on Very Large Scale Integration

PREFACE

The last five years have witnessed a revolution in computer design in universities and in many companies, away from building with standard IC's to building integrated systems with fully custom VLSI components. This approach has caused a major reorientation in all phases of realization of computing sytems — from fabrication lines, building a wider variety of more special purpose IC's, to design tools for handling large fully custom circuits, to machine architectures with a closer coupling (for example) between memory and processing elements. A worker using this more integrated approach can take advantage of a wider variety of design alternatives and thus can more effectively globally optimize designs than can a colleague restricted to a range of technical alternatives. (This approach is perhaps most effectively promulgated in the popular text by Carver Mead and Lynn Conway, *Introduction to VLSI Systems.*) This wider range of capabilities is difficult to acquire in the current technical atmosphere of ever increasing specialization. The series of conferences, of which this one is the seventh annual, has served as the major forum for supporting and enhancing this integrated approach to the study and realization of computing systems in VLSI. It has brought together leading researchers from universities, research laboratories, and government agencies in an informal atmosphere to discuss the newest developments in the wide range of technical areas that are involved.

The pace of progress over the last six years, the tenure of this series of conferences, has indeed been phenomenal — working custom IC's, a major achievement just a few years ago, are considered commonplace for beginning students now. Indeed with the acceptance of the relative ease of realizing a functioning chip, a new plateau has been reached. Now new challenges present themselves — what exactly does this new capability buy us? Just what is the effect of VLSI on the nature of computation, on algorithm design, on the design of full-size computing systems? These new challenges are the special emphases of this year's conference. Increasingly researchers are coming up against new limits when attempting to build full-speed useful-size computing systems with VLSI — the design automation algorithms are insufficient for the large size of chips needed, the design environment does not encompass the really difficult areas, those outside the chip, and the techniques for designing for testability and fault tolerance aren't sufficiently robust to be useful. These and other related issues are the ones being faced by the advanced researchers who have participated in this series of conferences.

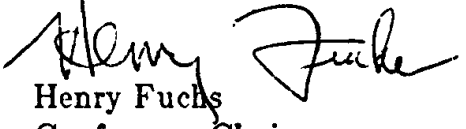
此为试读,需要完整PDF请访问: www.ertongbook.com

The 25 papers in this volume were selected from 79 submitted to the conference Program Committee. Although they represent a broad range of outstanding new work, they were selected solely on quality, without regard for "representation" or "balance". The selection process was based on evaluation of each paper by usually three individual Program Committee members. Often outside reviews were also solicited.

Acknowledgements: This conference represents the work of many individuals. The Program Committee members worked many long hours reviewing each of the submitted papers. The outside referees submitted thoughtful reviews often within very short deadlines. (Their names are listed on a following page.) The local member of the Program Committee, Arnold Rosenberg, put in many additional hours during the entire reviewing and selection process. Local department colleagues, Peter Calingaert and Kye Hedlund, gave advice and counsel on innumerable topics on a moment's notice. The chairs of past conferences in this series, Paul Penfield of MIT and Chuck Seitz of Caltech, were valuable sources of helpful advice and encouragement. The staff at these places, especially Barbara Lory of MIT and Phyllis Weiss of Caltech, saved us from disaster by supplying us with the past conference mailing lists and forms and procedures. We thank the sponsors of this conference, NSF Computer Engineering Section (Dr. Bernard Chern, Director) and the Department of Computer Science of UNC-Chapel Hill, for their support of this conference. We also thank the Microelectronics Center of North Carolina (MCNC) and its staff, especially Clyde Kelley, for their kindness and generosity in hosting the reception and tour of MCNC. We gratefully acknowledge Dan Murray and Norm Zeck of Xerox Coporation for the cover illustration. Finally, we are deeply grateful to Bobette Eckland, the one-person organizing staff of this conference, for the many months of dedicated effort on every aspect of this conference.

The Cover Illustration is a plot of part of a Pixel-planes 4 chip described in "PIXEL-PLANES: Building a VLSI-Based Graphic System" in these proceedings. The left side shows several nodes of the binary "multiplier tree", several one-bit pixel-processors, and part of the memory grid.

We hope you enjoy the conference and these proceedings.



Henry Fuchs
Conference Chair

March 14, 1985
Chapel Hill, North Carolina

PROGRAM COMMITTEE

Randal E. Bryant, CMU
George Cox, Intel Corp., Hillsboro, OR
Henry Fuchs, UNC-Chapel Hill (Chair)
Lance A. Glasser, MIT
John Hayes, U of Michigan
H. T. Kung, CMU
John K. Ousterhout, UC-Berkeley
Paul Penfield, MIT
Franco Preparata, U of Illinois
Arnold Rosenberg, Duke
Charles L. Seitz, Caltech
Thomas G. Szymanski, AT&T Bell Labs, Murray Hill
Nick Tredennick, IBM, T J Watson Research Center

REFEREES

Gregory D. Abram, UNC-Chapel Hill
George Agule, Raytheon, Bedford, MA
John Austin, UNC-Chapel Hill
Gary Bishop, AT&T Bell Laboratories, Holmdel, NJ
Stephen Brookes, CMU
Tam-Anh Chu, MIT
Roger B. Dannenberg, CMU
David Dill, CMU
John G. Eyles, UNC-Chapel Hill
Allan L. Fisher, CMU
Gordon Hamachi, UC-Berkeley
Dan Hammerstrom, Intel Corp., Hillsboro, OR
Kye Hedlund, UNC-Chapel Hill
Stephen Heller, MIT
Steve Hornick, U of Illinois
Mike McIlrath, MIT
Tom Knight, MIT
Hae-Seung Lee, MIT
Tom Leighton, MIT
Mark Matson, MIT
Robert Mayo, UC-Berkeley
Bruce Musicus, MIT
Sid Penstone, Queen's U, Kingston, Ontario
Jack Raffel, MIT
Alberto Sangiovanni-Vincentelli, UC-Berkeley
Majid Sarrafzadeh, U of Illinois
Walter Scott, UC-Berkeley
Carlo Sequin, UC-Berkeley
C. G. Sodini, MIT
R. Tammassia, U of Illinois
Christopher Terman, MIT
Ioannis Tollis, U of Illinois
Hank Walker, CMU
Manfred Wiesel, Intel Corp., Santa Clara, CA
John Wroclawski, MIT
Richard Zippel, MIT
Charles Zukowski, MIT

AUTHORS

CONTENTS

此为试读,需要完整PDF请访问: www.ertongbook.com

Hot-Clock nMOS

Charles L. Seitz, Alexander H. Frey,[1] Sven Mattisson,[2] Steve D. Rabin, Don A. Speck, Jan L. A. van de Snepscheut[3]

Department of Computer Science
California Institute of Technology, Pasadena

1. Also affiliated with IBM Los Angeles Scientific Center.

2. At Caltech 1982-84; now at the Applied Electronics Department, Lund Institute of Technology, Sweden.

3. At Caltech 1983-84; now at the Department of Mathematics and Computing Science, Groningen University, The Netherlands.

Abstract: "Hot-Clock *n*MOS" is a style of design that has advantages in circuit energetics and performance. When the application of this style is carried to its limits, an *n*MOS chip is powered entirely from its clock signals. There are savings in area, delay, and power, even when the bootstrap circuits of this style are used together with conventional circuitry. We have used this technique in numerous small projects and test structures, and in 3 substantial projects fabricated through MOSIS.

1. Energetics

How is the power required by and dissipated on a MOS chip used? Even in CMOS technology, in which the static power is negligible, "dynamic" power is required to charge and discharge capacitances:

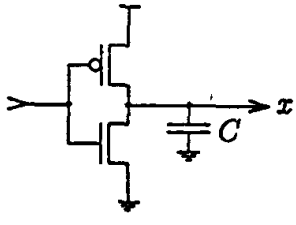


Figure 1

The research described in this paper was sponsored in part by the Defense Advanced Research Projects Agency, ARPA Order number 3771, monitored by the Office of Naval Research under contract number N00014-79-C-0597; and in part by IBM Corporation.

8850135

Each time the signal x is to change $0 \rightarrow 1$, the power supply must provide a quantity of charge CV_{dd} at potential V_{dd}, hence energy CV_{dd}^2. Half of this energy ends up stored in the capacitance C, and the other half is dissipated in the p-channel transistor. When x is to change $1 \rightarrow 0$, the charge stored on C is conducted through the n-channel transistor into the ground terminal, and the stored energy is dissipated in the transistor. Thus in a full cycle $0 \rightarrow 1 \rightarrow 0$ of signal x, energy CV_{dd}^2 must be supplied to the chip, and is dissipated in the transistors that drive this signal. The fundamental motivation behind hot-clock *n*MOS is to get around this "inevitable" dissipation of power on a high-complexity chip.

If one were to try to spot the places on a MOS chip where most of the dynamic power goes, it would be in the drivers of relatively large capacitances – long and/or highly loaded wires – that are driven at relatively high frequencies. Examples of such signals are control lines and data buses in instruction and arithmetic processors; word and bit lines in RAMs; literal and implicant lines in large PLAs, and output pads. By virtue of their capacitance, these are also signals that are difficult to drive with small delay.

Many of these signals are naturally driven in synchrony with one of the clock signals, so another possibility is to drive them through some approximation to an ideal switch:

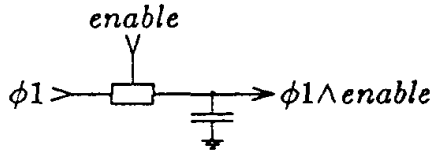


Figure 2

Here we assume that the output is initially 0, and that the *enable* signal changes only during $\phi 2$ in a two-phase non-overlapping clocking scheme. The output is then $\phi 1 \wedge enable$. If the switch were ideal, the circuit would introduce no delay, the output being just a gated replica of the input clock. Also, the switch turns on only when there is no voltage across it, and off when there is no current flowing through it, so even if it did exhibit some non-zero resistance or conductance while switching, it dissipates no power in changing state.

Assume that when this switch is implemented with MOS transistors, it can be modeled as an ideal switch in series with an effective resistance

R, and that the clock transition can be modeled as a ramp from 0V to V_c in time t_r, and from V_c to 0V in time t_f:

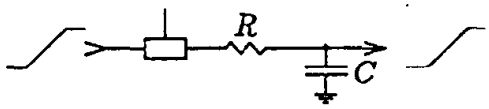


Figure 3

Let the switch be on. If t_r were 0, the full clock voltage would appear initially across the resistance, and the output node would be charged conventionally with the CV_c^2 energy from the clock being split between the output node capacitance and dissipation in the resistance. The case of interest to us is where t_r is some normal and achievable value – say a few ns –, and the sizes of the transistors that form the switch are selected so that R is small enough that the delay through the circuit, approximately RC, is much less than t_r. In this case in which $RC \ll t_r$, the switch is also sized to dissipate little power. The energy dissipated per switching event is closely approximated by:

$$E \approx \frac{RC}{t_r} CV_c^2.$$

Presume that the value of RC is fixed in the design, that is, that the sizes of the switch transistors are determined according to a given $RC \ll t_r, t_f$. Presume also that t_r and t_f are a fixed fraction of the clock period T. The same chip may then be operated at a longer clock period T, resulting in slower operation but also in a smaller *proportion* of the energy supplied in each clock transition being dissipated on-chip. In other words, these circuits exhibit a characteristic in which the total energy required to perform a computation varies as $1/T$. The computation is less costly if one is not in a hurry. This characteristic is at odds with complexity arguments that assert that the cost be expressed in E_{sw} units (§9.10 in [5]), and is interestingly similar to the AT^2 invariance exhibited by many algorithms [8], in which the cost AT of performing a given computation also varies as $1/T$.

This scheme does not really "solve" the power and speed problems of driving the capacitance C. Rather, we have *exported* the problem* elsewhere, namely, to whatever circuit drives the signal $\phi 1$.

* One might compare this technique with a country banning certain polluting industries, but still purchasing the products of those industries from other countries. The country may then be accused of "exporting pollution".

In hot-clock *n*MOS we export the problem off the chip. The off-chip clock driver can then use a technology that is better suited to driving the capacitive clock load than is the high complexity MOS technology. For example, bipolar transistors have much higher transconductance than MOS devices at present feature size, and make excellent clock drivers. A more interesting possibility is to use clock driver circuits that employ inductances. A resonant driver allows an almost lossless transfer of charge from the power supply to the clock capacitance, and then from the clock capacitance back into the power supply. Figure 4 presents an idealized circuit to implement this scheme of saving power in driving capacitive loads:

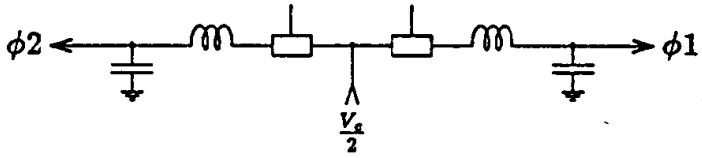


Figure 4

A practical circuit is rather more complicated. This trick is somewhat like "LC logic" (§9.1.3 in [5]), with the L's brought off the chip, and is subject to the same inevitability of some loss in the switching process. However, such techniques could allow VLSI systems – even in *n*MOS technology – to operate at dramatically lower overall dissipation levels than present CMOS circuits, let alone *n*MOS circuits.

2. The Elementary *n*MOS Clock-AND

This technique of exporting the problem of driving large capacitive loads can be applied either to CMOS or to *n*MOS designs. However, it is *n*MOS technology that benefits most from a better way to drive signals to 1 quickly, and that lends itself most readily to a set of elegantly simple clock-powered bootstrap circuits. An *n*MOS circuit that serves as a good approximation of the behavior idealized in Figure 2 is the "Clock-AND":

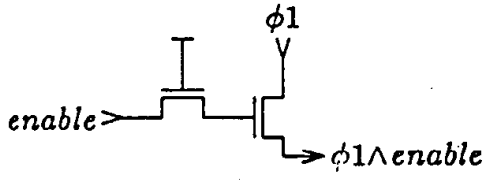


Figure 5

此为试读,需要完整PDF请访问: www.ertongbook.com