VLSI 83

VLSI design of digital systems

edited by
f. anceau and e. j. aas

VLSI 83

VLSI Design of Digital Systems

Proceedings of the IFIP TC 10 / WG 10.5 International Conference on
Very Large Scale Integration
Trondheim, Norway, 16–19 August 1983

edited by

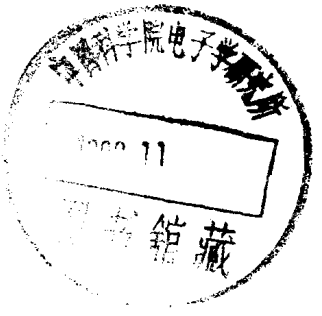
F. ANCEAU

University of Grenoble
France

and

E. J. AAS

University of Trondheim
Norway



1983

NORTH-HOLLAND
AMSTERDAM • NEW YORK • OXFORD

ISBN: 0 444 86751 1

Published by:
ELSEVIER SCIENCE PUBLISHERS B.V.
P.O. Box 1991
1000 BZ Amsterdam
The Netherlands

Sole distributors for the U.S.A. and Canada:
ELSEVIER SCIENCE PUBLISHING COMPANY, INC.
52 Vanderbilt Avenue
New York, N.Y. 10017
U.S.A.

Library of Congress Cataloging in Publication Data

IFIP TC 10 / WG 10.5 International Conference on
Very Large Scale Integration (1983 : Trondheim,
Norway)
VLSI '83.

Bibliography: p.
Includes index.
1. Integrated circuits--Very large scale integration
--Congresses. I. Anceau, F. II. Aas, E. J. (Einar J.)
III. IFIP TC-10. IV. IFIP WG 10.5. V. International
Federation for Information Processing. VI. Title.
VII. Title: V.L.S.I. '83.

TK7874.I328 1983 621.381'73 83-13347
ISBN 0-444-86751-1

PRINTED IN THE NETHERLANDS

VLSI 83

VLSI Design of Digital Systems

IFIP TC 10 / WG 10.5 International Conference on
Very Large Scale Integration
Trondheim, Norway, 16-19 August 1983

Sponsored by

The International Federation for Information Processing (IFIP)
The Norwegian Institute of Technology (NTH)
A/S Elektrisk Bureau
Norsk Data A/S
Standard Telefon og Kabelfabrik A/S
Tandberg Data A/S
Lehmkuhl Elektronikk A/S
SINTEF
IBM Norway
and
The Royal Norwegian Council for Scientific and Industrial Research (NTNF)

International Programme Committee

F. Anceau (chairman), J. Borel, H. De Man, N. C. De Troye, D. Eglin, W. L. Engl, E. Hörbst, I. Höglund, H. T. Kung, H. Lawson, Y. Lundh, S. Michaelson, J. C. Mudge, G. Musgrave, J. P. Roth, C. Trullemans, C. Séquin, J. D. Ullman and J. Vuillemin

Organizing Committee

A. Halaas (chairman), K. Bratbergsengen, K. A. Ingebrigtsen, K. Malvig, H. Sørbye, E. J. Aas, O. Aaserud and T. Jørgensen

General Chairman – O. Landsverk
Conference Secretariat – P. Ueland

NORTH-HOLLAND
AMSTERDAM • NEW YORK • OXFORD

PREFACE

The rather dramatic advances in integrated circuit technology have resulted in the well-known exponential growth of the maximum number of transistors that can be put on one chip. A similar progress in maximum frequency and computational power is observed. Most important, this trend is intended to continue at least through the 80's.

The dynamic enhancement of VLSI power, described above, has a great influence on the design of digital systems. This influence will take several forms.

The most obvious exploitation of VLSI technology is the integration of existing conventional systems. Major cost reductions and increased reliability for mass-produced standard systems may be achieved. As a result, digital systems will proliferate in many products, and play an important role in the changes which are taking place in our society.

More challenging perhaps, is the new generation of systems that will be designed. This was previously inconceivable due to size, cost or human resourcefulness. Think of real-time image processing, pattern recognition and intelligent, knowledge-based systems.

The ever increasing complexity of VLSI systems requires new and better design methodologies, novel architectures, more powerful CAD tools for design capture, evaluation, verification and realization, and new approaches to the growing problem of testing.

The programme committee had three main goals in mind when arranging this conference: that it should be interesting to both the industrial and the academic community, that it should bridge the gap between computer scientists and electronic designers, and that it should cover a wide range of disciplines related to VLSI. We feel that the present programme meets these goals.

We would like to take this opportunity to express our sincere gratitude to the members of the programme committee for the job they have done. We would also like to thank all referees for their great efforts in ranking the approximately 100 papers submitted to the conference. Of these, 31 papers of high quality were selected.

The VLSI 83 conference is a successor to the VLSI 81 conference in Edinburgh, after which a new working group within IFIP, WG 10.5 (VLSI) was formed. The working group is operated through the IFIP Technical Committee 10 (TC 10), and is aimed at all aspects of the interaction between semiconductor fabrication and digital system design.

This working group plans to organize a bi-annual series of international conferences, the first one being VLSI 83 in Trondheim. The next conference in the series is planned to take place in Japan in 1985.

F. Anceau
Editor

E.J. Aas
Editor

CONTENTS

*This paper was submitted at too late a date to be included in its correct session position. It can be found at the end of this volume under the Additional Submissions section.

**This paper was not submitted in time for publication within this volume.

***This was a panel discussion session which took place at the conference site; no material from which is included within this volume.

Session 1

OFFICIAL OPENING & KEYNOTE SPEAKERS

VLSI '83
F. Anceau and E.J. Aas (eds.)
Elsevier Science Publishers B.V. (North-Holland)

Structural and Behavioral Composition of VLSI

Carver A. Mead
Computer Science Department
California Institute of Technology
Pasadena, California 91125
USA

VLSI design requires all of the complexity management discipline associated with complex software systems, but without the underlying simplicity of a single sequential machine. Not only must we deal with the problems of enormous concurrency, but we must map the entire design onto a physical medium, with real constraints on space, time, and energy imposed by the laws of physics.

It has been clear for a number of years that identifying functional elements of a large system and providing clean interfaces at their boundaries is the only way to avoid a combinatorial explosion of complexity in the design. In the VLSI world this technique for complexity management has been carried to its logical extreme in the well-known structured design methodology. Here the functional elements are identified one-to-one with structural units. Each structural unit, in turn, corresponds to a single contiguous area on the chip. Thus it is characteristic of a structured design that the signals on any set of "connectors" at the boundary of such a functional element ("Block") are related by a clean functional description. We have given a general model of concurrent computation such that formal semantics can be defined for such a functional block [4][5]. If a system is constructed in such a way that it corresponds to this model, many properties can be guaranteed. In particular, the behavior of higher-level blocks built up by composing lower-level ones can be derived from the behaviors of the more primitive blocks in a precise way. In general this property is difficult to come by, and requires a great deal of discipline. Intuitively, this discipline corresponds to what we think of as good engineering practice – In particular, a well- defined and consistent timing convention, and well defined data types. If these disciplines are followed, the behavior of the block can be described by a set of state transition functions which map from inputs and internal stored state to outputs and next state [4]. Furthermore, any legal interconnection of several such blocks can itself be described as a block. The behavior of such a composite block can be derived from the behaviors of the component blocks by a single universal fixed-point algorithm [5]. A universal hierarchical simulator (UHS) has been constructed using this algorithm. It has been used to simulate systems from the transistor circuit level to high-level communicating processes. In contrast to standard "mixed mode" simulators, the UHS works on hierarchical levels appropriate to the design, rather than those pre-defined in the program itself. At the lowest level, an extension of the formalization of conductance networks given by [3] is used. [4] gives the first formal semantics for complete MOS circuits and thus enables the treatment of circuits in a hierarchical context.

Silicon Compilation [6][1] is defined as the generation of a chip from a functional description of the chip. For the term to have any meaning, it is necessary that the functional description be well defined. It follows that the behavioral description of a compilable chip must be representable as a block in the sense given above. The recursive definition of a block can thus be given: A block is either a primitive block, or a legal composition of blocks. A sensible compiler strategy used by many workers is based directly on the structured design methodology. Each construct in the definition language corresponds to a functional element implemented as a block in the sense defined above. Primitive blocks correspond to geometric areas on the silicon. Composite blocks correspond to an interconnection of several primitive or composite blocks. Any well-defined chip specified in this way can be simulated directly from its definition. An elegant implementation of this technique is described in [10], where the definition language was a subset of LISP. Simulation was carried out by the LISP interpreter, acting as the UHS. In other languages it is also possible to embed a UHS and let the language system do the work. For a true UHS, it is necessary to have a fast and convenient representation for a process. Each process must contain both functional and data attributes. The data must be as long-lived as the entire simulation, as they represent the stored state within each block. Connections are also represented as processes. Typical representations of a process are as an instance of a class in Simula, a module in Mainsail or Modular Pascal, or a suspendable function in ICL [2]. A record can be used in any language which has a procedure or function data type, or the equivalent. For example, in C an appropriate representation would be a structure containing both the data attributes of the process, and function pointers to the functional attributes.

A Silicon Compiler generates either mask geometry or simulation by passing the functional description to a general compilation mechanism. As the final step in chip generation, this mechanism generates the polygons and rectangles that implement the mask geometry of the corresponding block. Which geometrical objects are generated is determined by the procedures and data within the nested block definition. If the mechanism is generating simulation, it similarly instantiates processes which represent the functional elements within the block. Once the simulation is compiled, it is executed by the UHS. This approach has enormous advantages over conventional practice. Simulation is exceedingly fast, even on small machines. There is no need to recover high-level design intent, such as signal names and types, from a low-level representation such as mask geometry. The same mechanism which composes structural entities to form the structure at a higher level is used to compose process descriptions into those of composite processes. Once the underlying mechanism is proven correct, it is no longer necessary to check each chip by painfully crawling over the fully-instantiated output. It is this direct correspondence between the behavioral and physical structure which distinguishes a Silicon Compiler from programs which are merely faster ways to generate and analyze mask geometry.

New fabrication technologies appear every year; higher density, higher speed, lower power. A chip compiled into one technology must be compatible, at the functional level, with the same source code compiled into another technology. A 68000 in CMOS should execute the same instruction set as a 68000 in NMOS. It follows that the longevity of a compiler is not vested in one particular cell library, but rather in the ability to make the identical functional blocks efficiently in new technologies as they emerge. The