DESIGN METHODOLOGIES AND SOME ASPECTS OF VLSI AND ULSI

A. MÖSCHWITZER

Department of Information Technology,
Technical University of Dresden

Received January 13, 1988

Abstract

This is a review and considers first technical aspects and drawbacks for realizing very large scale and wafer scale integrated circuits. Among these are small geometry effects and interconnection delay. Some solutions are considered. Next, design methodologies for VLSI are discussed, considering the hierarchy of VLSI systems. Finally the influences for students education are remarked and a design system consisting of CAD tools for logic synthesis and simulation, network-and device simulation, automated layout generation and layout-verification are described.

1. VLSI, ULSI and WSI which way it goes

Recent development of microelectronics to VLSI has led to major changes in nearly all technical disciplines. The rapid growth of available transistors per chip makes it possible to implement new functions and systems on silicon chips and wafers. This development is strongly dependent on advanced semiconductor technology and highly intelligent CAD environment [1]. The physical phenomena will limit the device miniaturization (down to 0.1...0.2 μm), the interconnection delays will limit the chip area and speed. Among other technologies (silicon, bipolar, GaAs—HEMT and MESFET) CMOS will be the most promising for future VLSI and USLI. With feature sizes of $\delta=0.2$ μm $10^7...10^8$ devices will be possible on a single chip at the end of this century (Fig. 1).

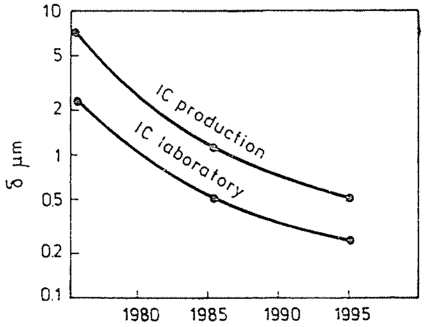


Fig. 1. Development of minimum feature size

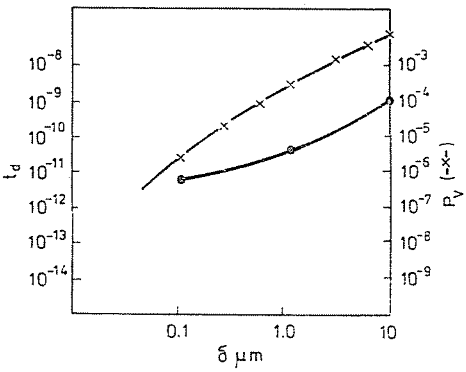
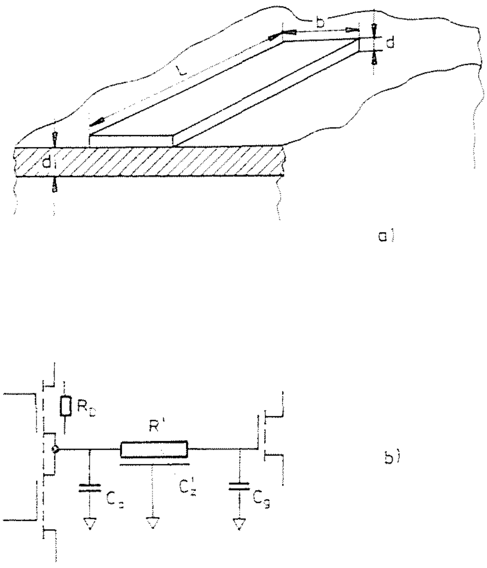


Fig. 2. Gate delay — and power consumption per gate —*x*— as a function of minimum feature size of CMOS devices

Power and delay will decrease down to some μW and ps, respectively (Fig. 2).

For CMOS device miniaturization there are some drawbacks. The most importants are as follows:

— short channel effects (inrease and decrease of the threshold voltage),
— leakage currents (weak inversion and punchthrough),
— snap back and latch up,
— drain and source series resistances.



$$t_{d\,total} = R_D\,(C_a + C_z'\,L + C_g) + \underbrace{\frac{1}{2}R'C_z'\,L^2}_{\text{interconn.}} + R'\,L\,C_g$$

Fig. 3. Interconnection line a) structure; b) model

A serious problem is coming up with the increase of the delay at interconnection lines (Fig. 3).

Using simplified lumped model for the line and the capacitances as well [2] we arrive at the following delay time for line Length L ($\sim$ chip size)

$$t_d = \frac{1}{2} R' C'_Z L^2 \qquad (1)$$

with the resistance per unit length

$$R' = \frac{\varrho}{db}$$

and the capacitance per unit length

$$C'_Z \approx \frac{\mathscr{E}_i}{d_i} b + 2\,\mathscr{E}_i \ln\left(1 + \frac{d}{d_i}\right) \qquad (2)$$

For $d = d_i = b = \delta$ we get $C'_Z \approx 3\mathscr{E}_i$ and for $R' = \frac{\varrho}{\delta^2}$ we arrive at

$$t_d \approx \frac{3}{2} \mathscr{E}_i \varrho \frac{L^2}{\delta^2} \qquad (3)$$

or 125 ps $(L^2/\mathrm{cm}^2)/(\delta^2/\mu\mathrm{m}^2)$ for aluminium inter-connection. With scaling down the minimum feature size and scaling up the chip size L from (3) we see a dramatic increase in delay time for VLSI. So we have to have some ideas to solve this problem to get good working and fast running VLSI parts.

Among these are:

— high conductivity inter-connection materials,
— avoid fringing capacitors,
— subdivision of the lines into parts connected with amplifiers (repeater),
— 3-dimensional integration,
— new architectures (e.g. array architectures).

Another approach to VLSI and ULSI is *wafer scale integration* (WSI). There are considerable advantages and some drawbacks, too. One can get reduced cost and more efficient cooling due to system level interconnection and higher degrees of integration. Due to interchip wires we have enhanced reliability and increased speed (small size and capacitance of wires) and no pin out overhead. So high speed portions of the system can be put in close proximity.
Among the disadvantages are

— redundancy and processing overhead,
— yield problems.

2. Design methodologies

Lets have a look first at Fig. 4 which shows the hierarchy of an VLSI-System. With millions of transistors you can make up a *supersystem* consisting of several processors (universal, floating point, speech and pattern recognition and memories). This supersystem can be subdivided into *systems* containing several thousands of transistors. The system is made up with *macrocells* containing several hundreds of transistors and consisting of *standard cells.* The lowest level of the hierarchy is the physical lay-out level.

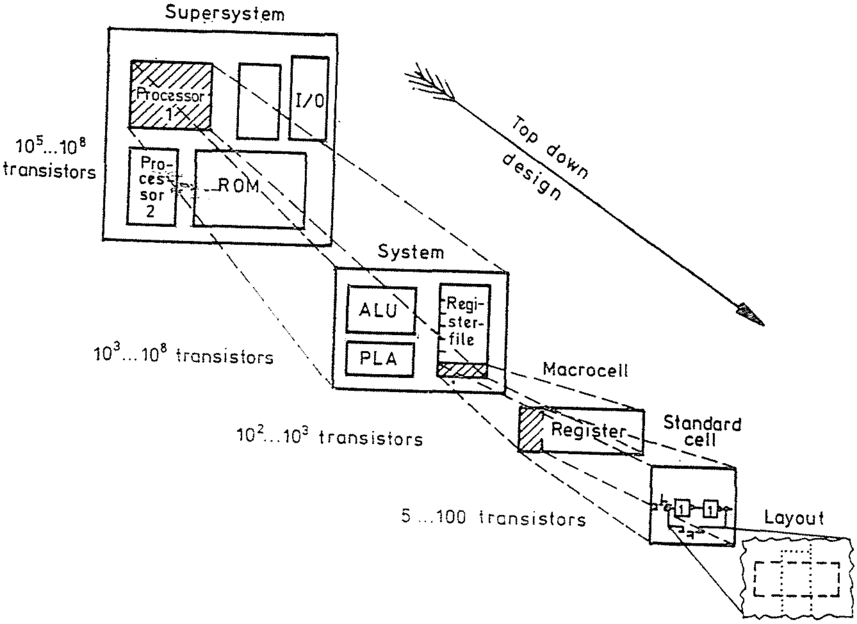


Fig. 4. The hierarchy of a VLSI chip

Design methodologies must take this hierarchy into account and follow the mentioned guidelines [2], [9].

— Hierarchical structure and modularity,
— regularity and uniformity,
— memory oriented logic (PLA, RAM, ROM),
— flexibility,
— design of fast circuits with low power consumption
— design for testability and built-in self test design automation (CAD).

To perform this top-down methodology we need a design environment consisting of hard-and software tools working mainly in an on-line manner with a unified data base with interactive options (Fig. 5).

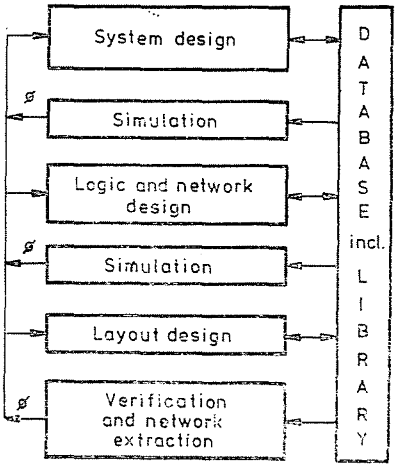


Fig. 5. Steps in VLSI chip design

3. Influence on students' education and design system

VLSI demands a steadily growing community of engineers which are able to implement their system ideas into silicon. They must be trained in some fundamentals of semiconductors, logic design with transistors (e.g. CMOS), register transfer operation of macrocells (e.g. contained in a library) [8] and handle effective methods to describe the problem in the algorithmic level and last but not least to know about and run CAD- and simulation tools [2].

In Fig. 6 some design steps are shown which are supported with programs developed at our university [2].

Starting from the system level we describe the problem with flow diagrams and information on data and instruction format. From this we get the architecture of the data path (system resources, macrocells) and the vector of the controlprimitives of the macrocells. According to the statement in 2. on design methodologies we design a very regular control path with micro sequenzer, or logic design we use a library of regular CMOS-structures (UL, PLA, ROM, LATCHES, RESISTERFILES, etc.) and simulate it with an event oriented transistor based static logic simulator LSINET [4] or for dynamic simulation with our network analysis program MISNET [7]. The device models we get from an overall process and device simulation system called PROMOS (s. appendix) [6].

The synthesis (minimization and folding) of PLA, ROM (and even for random logic) may be performed with a program called REKOS [2].

The layout can be performed automatically with the compiler called CALMOST [5], and at the end of the design process the layout verification with LSISIMULATOR [3] takes place. LSISIMULATOR is a design rule checking and network extraction tool. So we have feedback to upper levels of design verification.

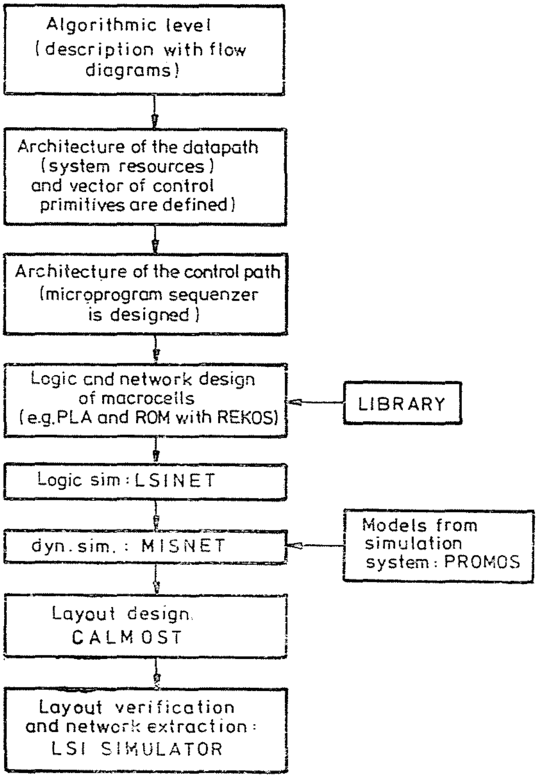

Algorithmic level (description with flow diagrams)
Architecture of the datapath (system resources) and vector of control primitives are defined)
Architecture of the control path (microprogram sequenzer is designed)
Logic and network design of macrocells (e.g.PLA and ROM with REKOS)
LIBRARY
Logic sim: LSINET
dyn. sim.: MISNET
Models from simulation system: PROMOS
Layout design CALMOST
Layout verification and network extraction: LSI SIMULATOR

Fig. 6. Flow diagram and used CAD tools for processor chip design (education project)

Acknowledgement

Dr. Sándor Török from Technical University Budapest has contributed very much to preparing this review of my talk for publishing. I appreciate it very much. I also will express my thanks to Professor K. Tarnay inviting me to contribute to the seminar "Problems of microelectronics" held in october 1985 at the Chair of Electron Devices.

References

1. Proceedings VLSI 85 Tokyo August (1985).
2. Möschwitzer, A.—Rössler, F.: VLSI-Systeme, Berlin. VEB Verlag Technik (1987).
3. Rössler, F. et. al.: LSISIMULATOR-ein Leistungsfähiges Programm System zur Funktionsverifikation hoch- und höchstintegrierter Schaltkreise Nachrichtentechnik /Elektronik *34* № 6 p. 213—124. (1984),
4. Hecker, W. Rössler, F. Möschwitzer, A.: LSINET-ein neues Logik- und Timingsimulationsprogramm für LSI- und VLSI-Schaltkreise Nachrichtentechnik /Elektronik *34* № 6, p. 214—218, (1984).

5. Rottler, O. Schüffny, R. Möschwitzer, A.: CALMOST- ein Siliconcompiler für NMOS-Schaltkreise, Nachrichtentechnik /Elektronik *33* № 1, (1986).
6. Schaarschmidt, J.: Ph. D-thesis, Technical University Dresden (1984).
7. Diener, K. H.: Misnet — ein Netzwerkanalyseprogram für MOS-Schaltkreise WZE (1975).
8. Mikroelektronikai berendezés-orientált áramkörök tervezése. Edited by Tarnay, K. Budapest (1984). In Hungarian.
9. Mead, C. Conway, L.: Introduction to VLSI systems Amsterdam (1980).

Appendix A.: PROMOS

The overall device and circuit simulation system PROMOS [6]
Consists of four main programs (see Fig. A)

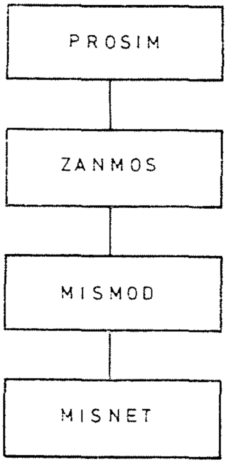


Fig. A Overview on the overall process, device and network simulation system PROMOS

The first is a two-dimensional process simulator PROSIM. Here we have input data as process variables (e.g. diffusion temperature, diffusion time, dopants, dose for ion implantation, lay-out geometries etc.) and we get results as impurity profiles, oxide thicknesses geometries of (moving) surfaces, which are now used in the second part, the two-dimensional device simulator ZANMOS [6]. This program calculates all electrical characteristics for static and dynamic operation of MOS-transistors in strong and weak inversion and punchthrough as well. Great attention is paid to investigating several effects for small geometry transistors channel length below 1 μm. The data coming out of this program are used to dimension network models for several modes for operation. So we have got by now some highly sophisticated but rather easy to use network models including weak inversion, punchthrough, special effects for improving analysis of CMOS analog circuits, breakdown, hot electron effects.

These models are available then in our network analysis program *MISNET* which is a powerful classical nonlinear network analysis program solving the implicit nonlinear differential equation system with Gear Algorithm, Newton iteration and sparse matrix technique [7].

Appendix B.: CALMOST

CALMOST is able to convert NSGT—MOS circuits of some thousands of transistors into a layout.

As input you can use transistor-net-list and mixed-level gate-oriented notation as well. The gates will be resolved by the program into transistor lists. The design rules are given to the program as alternable parameters. The designer has the option to define bonding pads for input and output and the ratio of the chip dimensions as well. After that an automated placing and routing is carried out according to the minimal sum of all interconnections (excluding that to the PIN's which are preferred through a "high weight" [5]). The program works highly interactive. So the designer has some option to modify and improve what is going on.

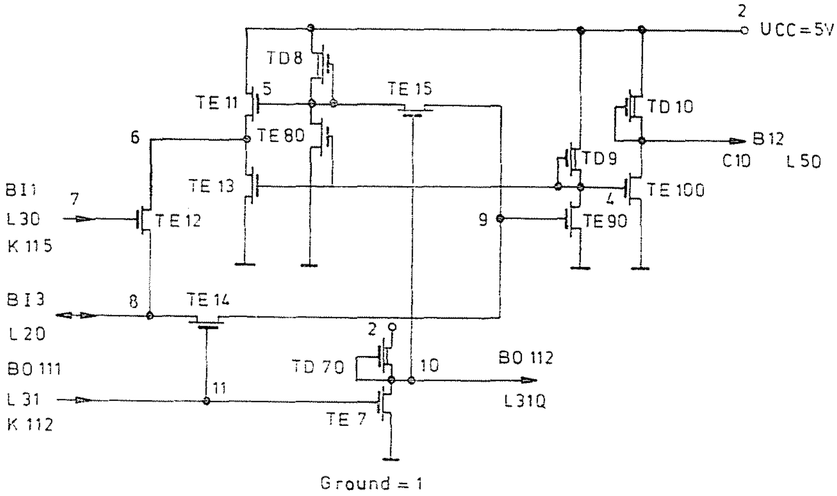


Fig. B.1. Network example

As an example you can see a layout of the circuit (given in Fig. B.1.) in Fig. B.2 which was automatically designed by *CALMOST* demanding a *CPU*-time of a minicomputer of approximately 45 seconds.

Prof. A. Möschwitzer — Technical University of Dresden

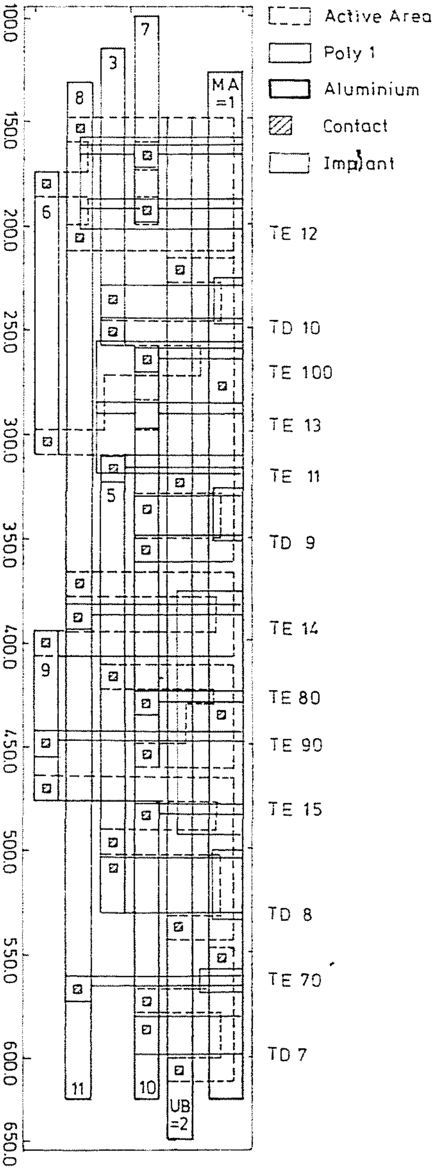


Fig. B.2. Layout of the example B.1 done automatically by CALMOST