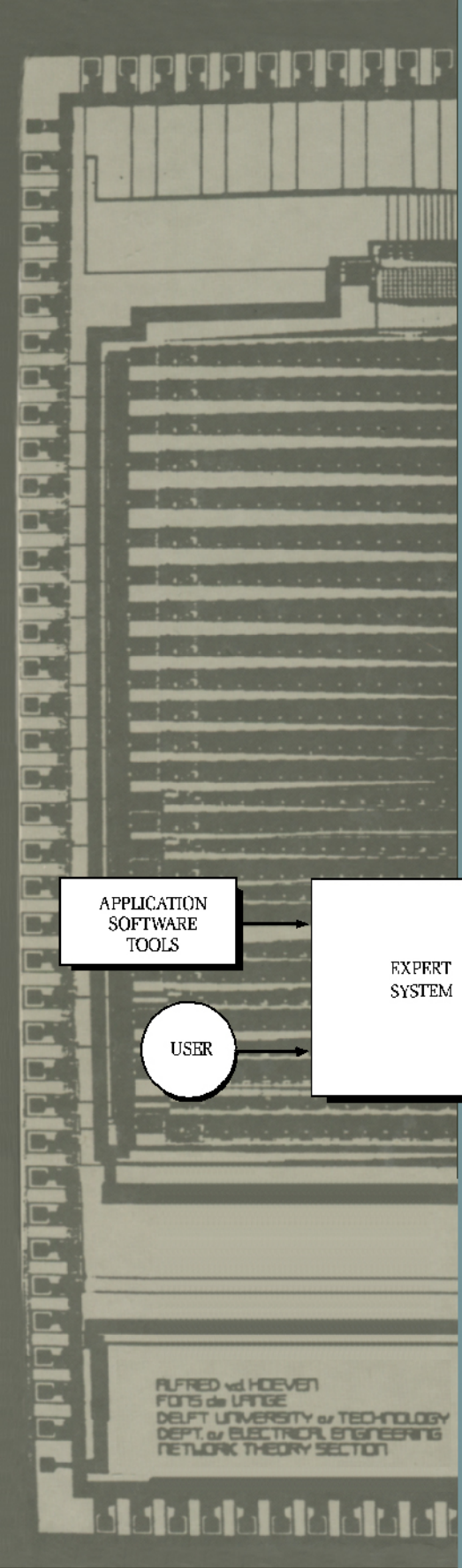
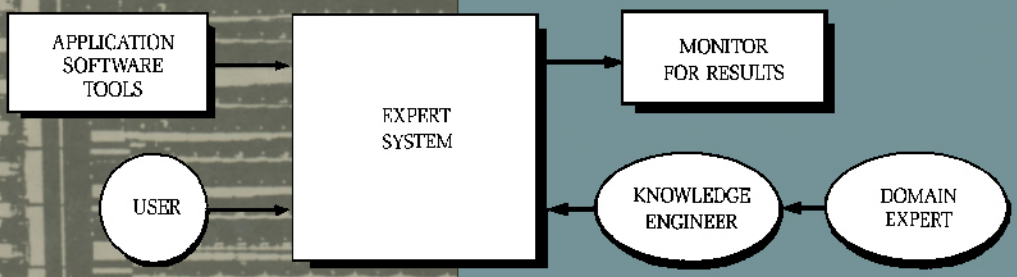


HANDBOOK OF VLSI CHIP DESIGN AND EXPERT SYSTEMS



A. F. SCHWARZ

Handbook of VLSI Chip Design and Expert Systems

A. F. Schwarz

Faculty of Electrical Engineering
Delft University of Technology
Delft, The Netherlands

ACADEMIC PRESS

Harcourt Brace Jovanovich, Publishers
London San Diego New York
Boston Sydney Tokyo Toronto

This book is printed on acid-free paper.

Academic Press Limited
24–28 Oval Road
London NW1 7DX

United States edition published by
Academic Press Inc.
San Diego, CA 92101

A catalogue record for this book is available from the British Library

ISBN 0-12-632425-5

Printed and bound in Great Britain by Hartnolls Ltd, Bodmin, Cornwall

PREFACE

This book consists of three interrelated parts: (1) Computer-Aided Design of VLSI Digital Circuits and Systems (CADCAS), (2) Artificial Intelligence (AI) and Expert Systems, and (3) Applications of AI / Expert Systems to CADCAS. The field of CADCAS has experienced a remarkable growth and development during its short existence. It primarily depends on a set of well-formalized algorithms, which means that the many procedures for performing subtasks in the design cycle follow fixed preprogrammed sequences of instructions. The ultimate aim in CADCAS is to automate the design process so that human-introduced errors are avoided or at least minimized.

As system complexity increases, chip design becomes more and more a multidisciplinary enterprise. The production of a complex chip is the result of individual contributions from many experts, including system design experts, logic designers, circuit analysts and designers, device engineers, technologists and application-specific designers. Designing and producing complex chips requires a hierarchical top-down approach with several levels of abstraction, starting from the highest system level going down to the lowest fabrication level. Each expert working at any level uses a specific data description and representation associated with this level. Chip production requires the interdisciplinary teamwork of specialists, implying adequate communication among specialists and unambiguous translation of descriptive languages between the subsequent levels of the design cycle. This integration problem appears to be intractable when exclusively algorithmic methods are used. Knowledge-based methods may offer a solution to this problem.

In the past decade, progress in artificial intelligence, with an emphasis on expert systems, has been overwhelming. In addition to the algorithmic approach as applied in CADCAS, expert systems provide us with a knowledge-based approach to problem solving. This approach, which is based on heuristics and rules contained in a knowledge base, has been adopted in a great variety of applications, e.g., in medical diagnosis. Though CADCAS as well as expert systems are growing to maturity, each in its particular domain, their integration (that is, the application of expert systems in CADCAS) is still in its infancy. Books on pure CADCAS are available in abundance; so are books on AI and expert systems and its applications. However, publications on applications of expert systems in CADCAS are scattered among a multitude of monographs, specialized journals and conference proceedings. Books published so far are restricted to specific subtasks of the chip design cycle, for example, either for high-level design, routing or testing. The present book is an attempt to encompass, in a single monograph, all important issues emerging when expert

systems are applied to microelectronic chip design.

The main objective of writing this book is to present the reader with the use of expert systems in every possible subtask of VLSI chip design as well as in the interrelations between the subtasks. Usually, the entire design cycle consists of a set of algorithmic procedures whose sequential steps are controlled by the human designer who also performs the intelligent tasks which cannot be done by the computer. A typical session of designing microelectronic chips consists of running a set of automatic programs, knit together by a human designer sitting in front of a workstation. Knowledge-based tools can be used to replace the human designer in performing the heuristic and intelligent decisions to be made. This is implemented by storing non-algorithmic design expertise acquired from recognized human design experts into the knowledge base of expert systems.

Only recently have heuristic knowledge-based methods been considered applicable to CADCAS. In my view, future CADCAS procedures will be mainly algorithmic, while heuristic AI methods will be used as a complement to the algorithmic methods in those parts of the design cycle which lend themselves very well to a heuristic approach. Future design systems will involve both algorithmic and heuristic methods. Generally speaking, expert systems in CADCAS can be applied both to specialized subtasks and to the integration aspect of the design cycle. The different character of each subtask requires a different kind of expert system. For example, an expert system used for high-level design is different from an expert system used for routing or testing. The kind of expert system to be used depends on the subtask for which it is used. Therefore, this subtask has first to be defined, the algorithmic approach should be outlined, and then the question can be answered as to whether an expert system can be used. A desirable goal of using expert systems is to shift the burden of the remaining human contributions to the computer so that the entire design cycle is fully automatic. To this end, an integration of the algorithmic and the essentially different knowledge-based procedures into one coherent system is necessary. This dynamic area on the frontier of microelectronics research is not yet well established with many problems requiring further study.

CADCAS and expert systems have grown so rapidly that it is difficult to keep abreast of the latest developments in the field. There is a need for emphasizing fundamentals in order to give the book a lasting value. At the same time, a presentation of the state of the art (though it changes in the course of time) is necessary to give the reader a sufficient amount of knowledge as to where we are now so that we can assess future developments on a reasonable basis. To that end, this book gives extensive surveys of representative expert systems for CADCAS purposes as published in the literature.

This book is intended to be an introductory guide through the maze of articles published on the use of expert systems in VLSI chip design. Readers who want to probe deeper into a subject will benefit from the extensive list of references at

the end of the book. In summary, the main ingredients of this book are as follow:

- CADCAS is discussed focusing on the most important methods which will influence future developments. I will not restrict myself to only one specific subtask of CADCAS, but place particular emphasis on the integration of the separate subtasks.
- A review of the growing field of expert systems will be given as an introduction in order to help the reader to conceive the possible uses of expert systems to CADCAS. Due attention will be paid to the programming languages and to the personal computers which can be used in expert systems.
- Applications of expert systems in specific subtasks as well as in the integrated CADCAS system will be described with illustrative examples.

This scheme of organization is quite natural considering the fact that the AI approach complements rather than replaces the algorithmic approach to CADCAS. The objective of the book is to give the reader

- a basic conceptual and methodological understanding of all that is involved in chip design and a clear insight into fundamental principles in CADCAS so that possible future trends can be evaluated.
- sufficient knowledge of AI, expert systems and personal computers to be able to understand their use in CADCAS.
- an insightful grip and knowledge as to where, why and how expert systems can be employed in subtasks of CADCAS and in the integrated chip design system.

The targeted readership consists of three categories:

- Electronics engineers and computer scientists involved in any subtask of CADCAS (chip design) in need of more insight into the interrelations of the various subtasks in the design cycle.
- AI scientists and expert system designers who are interested in the applicability of expert systems to CADCAS and, last but not least,
- university professors and undergraduate or graduate students who want to acquaint themselves with the latest developments in this highly active research field.

Acknowledgments

This book should be considered as a tribute to all researchers in the field of CADCAS and AI whose work has brought the field to a remarkably high level of sophistication. In describing their methods and tools, I have carefully referenced the original publications, wherever appropriate.

I am deeply indebted to Professor Patrick Dewilde and his staff of the Delft University's Network Theory Section, who provided the stimulating environment in which I could work on the book. I have greatly benefited from the critical reviews and comparative studies of recent developments in CADCAS and expert systems undertaken by students. Erik Platzbecker was particularly helpful whenever text-processing problems occurred.

I am obliged to the publisher, Academic Press, especially to editor James Gaussen, who handled the final editing and production process. I wish to express my appreciation to the staff of the Production Department of Academic Press who transformed my camera-ready copy into this fine book.

Last but not least, my special thanks go to my wife, Nicoline, for her understanding and patience during years of composing and editing of the text.

Dolf Schwarz

CONTENTS

Chapter 1

VLSI CHIP DESIGN

1.1 COMPUTER-AIDED CIRCUIT AND SYSTEM DESIGN

1.1a Impacts of the Rapidly Changing Technology

Introduction

The impressive advances in digital computers and communication engineering have been made possible by the various achievements in electronic design and semiconductor technology. The first three generations of computers are usually associated with the changing electronic hardware components, which were successively vacuum tubes, discrete transistors and integrated circuits. The advent of the integrated circuit (IC), which ushered in the *microelectronics* era, has initiated the emergence of *chips* with ever-increasing integration densities. In about two decades, the manufactured chips have reached the density of *Very-Large-Scale Integration* (VLSI), which implies more than 40000 transistors per chip. At present, a chip may contain millions of transistors.

The growing density in a chip goes hand in hand with an increase in the complexity of the system functions which a chip can perform. Systems have gradually become more comprehensive in scope and more complex in conception and design. As systems become larger and more complex, they strain the capabilities of the designers in that the immense masses of detail that the design emcompasses can no longer be comprehended. As a consequence, we witness changes in some of the older concepts and methods of digital-circuit design, while new approaches to solving the complexity problem emerge. The fourth generation of computers, which was developed in the 1980s, is characterized by an abundant use of VLSI and distributed computing systems. The combined use of microelectronics, artificial intelligence and automatic development tools will beget the fifth-generation computer, which is expected to flourish in the 1990s.

The emergence of complex logic systems requires a *systems approach* to design in addition to the structural *circuit-design approach.* This systems approach, which stresses the functional behavior of a system rather than its structure, has led us to consider higher levels of abstraction, such as the register-transfer level and the system level.

Design Automation

The increasing complexity of logic systems forces us to search for methods and tools which automate the design process. This has led to the concept of design automation. Strictly speaking, *design automation* (DA) implies the automatic generation of an electronic circuit or system, given a specification of design requirements. However, except for restricted applications and layout structures, a fully automatic design procedure from initial design concept to the final chip production is unattainable, if ever desirable. For this reason, design automation will be defined here as the art of utilizing the digital computer to automate as many steps as possible in the design of integrated circuits and systems. Design automation can be identified as *Computer-Aided Design of Circuits and Systems* (CADCAS) with automation in mind [Sch87].

Design automation exploits the basic capabilities of computers to perform complex calculations and to handle huge amounts of data with a high accuracy and speed. These capabilities are enhanced by new data-storage techniques and the use of interactive graphics systems, which facilitate human-machine communication. The primary incentive of design automation is the reduction of the required investments entailed in the design efforts:

a. Substantial reduction in the turnaround time, i.e., the time between design specification and product delivery.
b. Reduction in errors made during the entire process from initial design to product manufacture.
c. Fewer demands on the designer's skills. Human experience can be exploited for improving complex designs in an interactive way.
d. Reduction in production costs and increased manufacturing yield and hence a higher probability of initial success.

Disadvantages of design automation are the relatively inefficient usage of silicon area and the lower chip performance compared to full-custom design.

Papers on virtually all aspects of design automation have been presented at several conferences. Starting in 1964, the Design Automation Conference, sponsored by the ACM and the IEEE Computer Society, is held annually in the USA at the end of June. The Proceedings of this conference give a good picture of how design automation has evolved through the years. Similar conferences are being held in the USA as well as in Europe. Several books and survey articles on computer-aided design of VLSI chips have been published [Ein85, Fic87, Sch87, Dil88, DiG89, Sha89].

Expert Systems

It is generally agreed that human intervention in approprate stages of the design cycle will always be desirable. Chip design involves two main approaches to problem solving: (a) the algorithmic approach, and (b) the knowledge-based approach. The algorithmic approach, substantiated by design automation or

CADCAS, primarily depends on well-formulated algorithms. This means that the procedure of solving a problem follows a fixed preprogrammed sequence of steps, ultimately leading to the solution. The aim is to automate the design process as much as possible so that human-introduced errors are avoided or at least minimized.

The second approach involves those activities which are not amenable to algorithmic methodology and which therefore have been relegated to human intervention. These activities are based on knowledge (in the form of heuristics and rules) derived from many years experience acquired from a selected group of human experts. Only recently have attempts been made to implement hardware systems which can take over the activities thus far performed by human designers. The knowledge in such systems is stored in a knowledge base, while a reasoning procedure is incorporated into the system. Such a system is now known under the name *expert system.* A main objective of this book is to survey the possibilities of using expert systems in the design of VLSI chips. It must be emphasized that expert systems are to be used as complements rather than as replacements of human designers.

Rapid Changes in VLSI Technology

Complex system design requires adequate computer-aided design tools. When the IC technology changes (which it does at a dramatic pace), the design tools will have to be changed accordingly. The advances of VLSI technology and the exploitation of design automation tools have made possible the manufacture of digital systems with smaller sizes and with much lower power consumptions and production costs and higher speeds than ever before. The main problem that designers of complex systems have to face can be phrased as follows: Given a rapidly changing technological environment, how can we develop and maintain design tools which remain useful and at least easily updatable as technology progresses?

Without special measures being taken, such as the use of CADCAS and DA tools, the chip-design time will increase exponentially as circuit complexity increases. In the literature, many solutions to the complexity problem in the area of circuit or system design, simulation, testing and layout are attempts to keep up with the rapid changes in technology. The most important *impacts of VLSI technology* on the design of complex digital systems will be summarized below:

a. *New Design Methodologies.* To cope with the increasing complexities in VLSI sytems, new system-design methodologies must be developed. Major issues are the hierarchical approach to system design (involving the partitioning of system design into smaller, more manageable parts), a structured method of software development and the use of an integrated design system with updating facilities. There is an urgent need for technology-independent CADCAS or DA tools for complex system designs.

b. *Regular Structures and Design Automation.* Increasing system complexities can be handled by using regular system structures. This facilitates the exploitation of design automation, which in turn is evolving as a major contributor to further technological advances. Design automation and VSLI systems usually involve predefined structures, notably regular structures, such as RAMs, ROMs, PLAs and gate arrays. An advantage of regular structures is the efficiency of design and the ease of design testability and verification.

c. *Hardware-Software Tradeoffs.* The decreasing costs of hardware components, made possible by VLSI technology, force the designers to consider tradeoffs between the high performance and speed of hardware implementations and the programming flexibility of software to perform various tasks. As a consequence of the increasing costs of software development, the designer is compelled to use more sophisticated software-development tools. On the other hand, the decreasing costs of hardware encourage the development of new multiprocessor system architectures, different from the Von Neumann architectures. This causes a movement of software functions into hardware.

d. *Emergence of New Products.* A milestone in the history of IC technology is the fabrication of a complete computer on a chip. The dramatic reduction in physical size, power consumption and production cost of LSI and VLSI chips (including microprocessors) has made new products proliferate, thereby opening the door to new markets. There is a growing demand for applications in the home, factory, office and just about everywhere else.

e. *Revival of Customized Designs.* Many semiconductor manufacturers used to be reluctant to produce low-volume customized ICs because of low profits. The explosion of new application areas for LSI and VLSI chips and the advances in CADCAS and DA systems for IC design have led to reduced development costs and time scales for design. As a consequence, semicustom designs or Application-Specific Integrated Circuits (ASICs) are now showing significant cost benefits, even at low-volume production. Many approaches to custom design with a short turnaround time have been proposed with great success.

f. *New Computer Architectures.* The increasing complexity of problems encountered in advanced applications has an impact on the computer architecture. The trend is to diverge from the general-purpose computer toward a cooperative set of processors each dedicated to specific tasks and from the traditional Von Neumann machine toward parallelism, both in function and control.

g. *Expert Systems.* Non-algorithmic tasks in problem solving, hitherto a domain of human experts, are being performed by expert systems. Proposed applications of expert systems in VLSI chip design is the main topic of the

present book.

h. *Neural Networks.* While problem solving using traditional computers is based on exploiting the inherent properties of computers (e.g., high-speed processing of computational tasks), there is a trend to let hardware systems perform many intelligent tasks in various applications by using neural networks, which mimic the structure and performance of the human brain. Possible applications of neural networks in chip design are being investigated. At least, the problem of how to implement them as VLSI chips falls within the scope of this book.

i. *Need for a New Breed of Chip Designer.* A complete logic system can be integrated in the form of a very small physical chip, which is actually a microelectronic circuit capable of performing logic functions. Due to this fact, the borderline between system designers and circuit designers is becoming blurred. A modern circuit or system designer is necessarily concerned with (and hence should develop a basic foundation of knowledge in) all aspects of the design cycle from the highest system level down to the fabrication level. This statement does not suggest that everyone involved in chip design and manufacture must do everything. The need for system designers, circuit designers, device engineers and chip manufacturers who are expert in their own fields remains. The point is that a basic knowledge of the entire design cycle facilitates intelligent and effective interactions between various specialists and a better appreciation of each of the other's problems.

j. *Integrated Workstations for Chip Design.* The fast rate of change in semiconductor technology has created a scarcity in chip designers who are proficient in a multidisciplinary role. As a consequence, there is a trend to develop powerful workstations which integrate the whole spectrum of the design cycle and expert systems which incorporate design knowledge and experience from expert human designers. Though such systems are required to be technology-independent, regular updates will remain necessary and chip designers as indicated in i. are well qualified to exploit such systems for increasing the design quality and productivity. Computer communication networks allow designers to access remote databases and to exchange design information with colleagues working on the same project.

Consequences of the VLSI Explosion

The tremendous impacts of VLSI technology outlined above pose substantial problems to chip designers. The more rapid the technological changes are, the more imperative it is that we should master basic principles which have a lasting value. Fortunately, we may still rely on a number of principles and concepts, which have withstood the ravages of time. The advent of new technologies does not automatically make all the existing methods and tools of design obsolete. A

new VLSI tool is often either an extension or an adjunct to an existing design tool.

For all that, considerable ingenuity must be employed in developing a methodology which is able to cope with the rapid changes in technology. Design systems must be updated on a continuing basis to accomodate changes in chip technology, customer requirements and critical time schedules. A major goal in IC design and manufacture is to produce the largest possible percentage of error-free chips that will work upon first operation. With this in mind, the ability to develop adequate tools for design verification and testing may be a rate-limiting factor constraining the potential growth in the complexity of logic systems [Mea80].

The scarcity of skilled personnel able to cope with the rapidly changing technology has forced different companies to join efforts with the objective of advancing the state of the art in chip technology by improving the planning and coordination of system research and development. In the USA, the Semiconductor Research Cooperative proposed by the Semiconductor Industry Association coordinates university research, while the Microelectronics and Computer Technology Corporation supports joint efforts among industrial companies. In Japan, the Ministry of International Trade and Industry provides financial support to research projects in the microelectronics field. In Europe, the Commission of the EEC (European Economic Community) provides finances for launching innovative projects, in particular through the Microelectronics Program and ESPRIT (European Strategic Progamme of Research and Development in Information Technology).

1.1b The Hierarchical Approach to Chip Design

From SSI to VLSI

There has been tremendously fast progress in microelectronics starting with MOS technology and later followed by bipolar technology. This has led to the development of more complex digital systems with increased device density in a chip. Since digital systems are composed of logic gates, the *integration density* of integrated circuits is usually expressed in terms of *equivalent gates per chip*, where each equivalent gate (EG) is assumed to contain four transistors. Based on the integration density, the following classification of IC technologies is commonly used:

SSI (Small Scale Integration), up to 10 EG/chip;
MSI (Medium Scale Integration), between 10 and 10^2 EG/chip;
LSI (Large Scale Integration), between 10^2 and 10^4 EG/chip;
VLSI (Very Large Scale Integration), more than 10^4 EG/chip.

As the integration density increases, more and more functions can be

incorporated into one chip. This has led to the introduction of standard modules in more advanced forms of integration: SSI → MSI → LSI → VLSI. Consequently, circuit primitives can be defined according to the level of integration density. When an interconnected set of transistors, which are primitives in a transistor circuit, is used to form a logic gate, and several types of logic gates are defined, it is expedient to base a logic-circuit design on logic gates than on transistors. Logic gates form a higher level of primitives than transistors. Similarly, interconnected sets of gates form functional units which can be taken as primitives for designing a logic system. The development of higher-level primitives leads to the problem of modeling and simulation at different levels of detail.

Complex high-density LSI and VLSI circuits require a hierarchical approach to design which implies simulation at various levels. The levels of detail (or abstraction) in simulation are directly related to the model primitives which are selected.

Hierarchical Levels of a Complex Logic System

A complex logic system requires a top-down hierarchical design approach, starting at a high system level and working down to the mask-artwork and fabrication level. At the higher levels, most emphasis is placed on the *behavioral* aspects of the target system, i.e., the ultimate hardware realization of the system. As one proceeds to lower levels, more and more *structural* information is added to the design in progress, until at the end the design can be put into fabrication.

With a view to Fig. 1.1, let us describe the various abstraction levels which are encountered during the entire design cycle.

System Level or Algorithmic Level

This level serves to specify the behavioral and performance requirements of the system according to customer needs. System analysis is related to the applications for which the design is intended. A feasability study is conducted taking into account present-day physical constraints. The global behavior, particularly the information flow in the target system, is evaluated and a suitable design methodology is selected aimed at an optimal solution in terms of performance and throughput. The system level is partitioned in terms of *system primitives*, such as central processing units, memory modules, buffers and peripheral devices. Parameters of interest to be studied include information rates, memory capacities and system costs.

Register-Transfer Level

At this level, *data flow* and *control flow* are of interest. The register-transfer primitives are registers which are operated upon to perform all kinds of data processing ranging from simple data transfer to complex logic and arithmetic operations. This level deals with detailed data path and control algorithms

without emphasizing the constraints imposed by a particular implementation technology. The functional design of the system can be simulated, thereby enabling microprograms etc. to be evaluated.

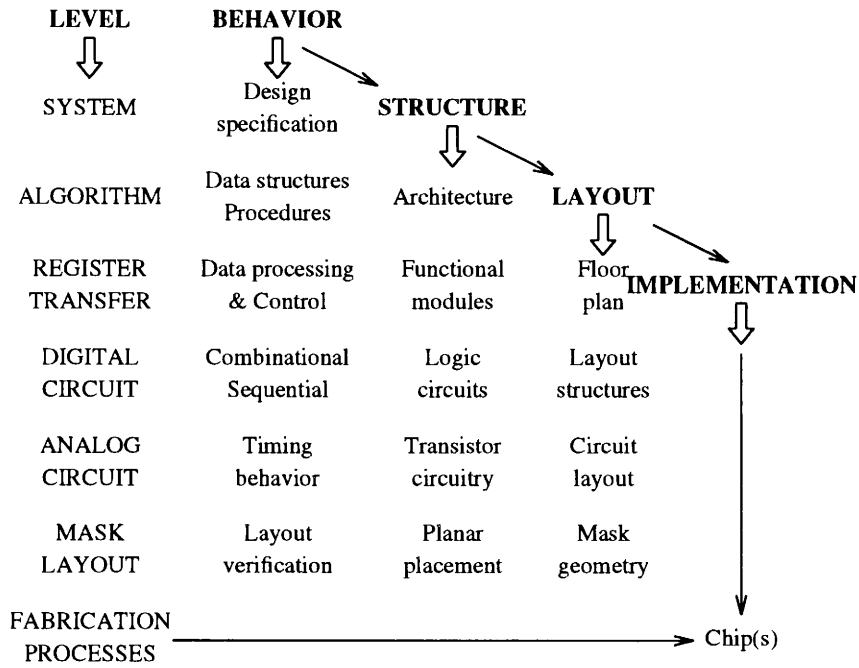


Fig. 1.1 Hierarchy of a complex logic system

Digital-Circuit Level

The construction of the target system from gates and flip-flops is emphasized. At this stage, the structure of the circuit implementation becomes visible. The choice of the implementation technology (bipolar, NMOS, CMOS, etc.) may be made at this level, provided that one is not restricted to a fixed technology at the outset. For the purpose of simulation at this level, models of gates and flip-flops can be defined with different levels of accuracy. The *Switch Level*, relevant to the MOS technologies, can be considered as a logic-gate level as far as the logic values (e.g., 0, 1 and *U*) are concerned. At the same time, the switch level can be considered as a form of circuit-element level, where all circuit elements (usually transistors) are represented by very simple models: voltage-controlled switches.

Analog-Circuit Level

The target system has taken the form of a transistor circuit in the selected technology. The circuit primitives are transistors, capacitors, resistors, etc. In an

integrated circuit, the main primitives are transistors. Hence, it is important to be thoroughly acquainted with the behavior of transistors in a circuit. A characteristic feature of this level is that the transistors can be represented by very accurate models. The major limitation of circuit simulation is the size of the network that can be handled (at most a few hundred transistors). When the circuit to be simulated is large, some form of partitioning into suitable subcircuits is necessary. The main advantage of circuit simulation is that extremely accurate transistor models can be used, taking into account the effect of any desirable parameter. As a result, realistic simulations are possible. When we want to simulate large circuits, we may employ macromodels which are models in a simplified form.

Physical-Device Level
At this level, the physical properties of the transistor as a semiconductor device are studied. In particular, the effect of physical contraints on the electrical behavior of the transistor is of great importance. Simulation at the device-physics level allows one to study the effect of physical and geometric parameters of the transistors on the terminal behavior of the transistors or transistor circuits. This level is intimately related to the layout design and the fabrication technology.

Mask-Artwork and Fabrication Level
It is assumed that the target system is a transistor circuit, that is, we do not consider mechanical peripheral devices which are also parts of the complete digital system. To allow the transistor circuit to be implemented in the desired integrated-circuit form, a set of masks is required. The mask artwork which defines the set of masks is obtained as the result of the layout design. The symbolic layout level may precede the geometric mask-artwork level. To verify the correctness of the layout, this layout is thoroughly examined to check for violation of the specified mask-design.

Top-Down Design
When a very complex system is to be designed, it is natural to adopt a *top-down approach*, which proceeds from the top level with system simulations down to the implementation level, resulting in the ultimate fabrication of the desired integrated circuit.

The top-down approach implies partitioning a large problem into manageable subproblems which can be solved in an appropriate manner. Let us illustrate this approach by considering the design of a complex digital logic system. In general, the design of a complex system proceeds in a top-down manner, after which the implementation takes place in a bottom-up manner. This is clarified by the flowchart of Fig. 1.1. This chart demonstrates the hierarchical character of the design cycle. The different levels of hierarchy allow a complex system to be

designed in a systematic way that is amenable to human understanding and compatible with computer processing capabilities.

The hierarchical levels in the lefthand column of Fig. 1.1 are in conformity with the top-down approach that a human designer uses. The other columns show the corresponding software and hardware features which are relevant at each particular level. The *design specification* includes the definition of the data or signal processing function to be realized. The upper two levels (system level and register-transfer level), which deal with system functions without considering physical implementation details, will be discussed in Chapter 5. A large part of this book will be concerned with the lower levels, which are distinguished from the upper two levels by the presence of a physical structure, as an interconnection of functional units (functional level), as a logic-gate circuit (gate level) or transistor circuit (circuit-element level), or as a pattern of mask primitives (implementation level). The hierarchical top-down approach focuses the design successively on the system behavior, the data and control flow at the register-transfer level, the logic-circuit behavior, the analog-circuit performance and finally the mask artwork, which defines the planar mapping of the complete transistor circuit.

A software capability (e.g., simulation) is provided for each stage of design. The software tool associated with a particular level is tailored to the distinctive requirements and desirable design verifications which are characteristic of this level. Note that simulation at intermediate levels or at multiple levels can be successfully accomplished. For example, a timing simulation is a simulation mode which attempts to combine the speed of gate-level simulation and the accuracy of simulation at the circuit-element level (see Subsection 6.2c). Mixed-mode simulators have been developed which permit simultaneous simulation of different parts of the circuit at different levels (logic, timing and circuit-element levels).

The top-down approach may include *backward loops* which represent repeating design cycles. For example, when the design of the complete mask artwork is finished, useful information regarding the corresponding logic circuit, including calculated logic delays, can be extracted from it. The extracted circuit can be submitted to a logic or timing simulator, or even for better accuracy to a circuit simulator. This circuit extraction allows the simulation of the actual circuitry, including parasitic and other device parameters which are revealed during the mask-artwork analysis.

The hierarchical top-down approach which passes through all design stages, as indicated in Fig. 1.1, is not pursued in all cases of complex design. When we restrict ourselves to special structures, it is possible to move from the logic-function specification directly to the mask artwork, sometimes referred to as *silicon compilation* (Subsection 5.3b).

In addition to the required simulations for verifying the correctness of a

circuit design, we have to consider the *testing problem* in order to facilitate system maintenance in the field. Just like the simulation problem, the testing problem should be solved in a top-down manner. Functional tests at successively lower levels may detect the location of a faulty component. VLSI chip design, design verification, testing and layout design will be dealt with in Chapters 5 through 8.

1.1c Design Methodologies

Introduction

By a *design methodology* we may include the whole body of selected procedures or techniques and the various hardware and software tools employed in solving a design problem. The advances in device technology, software engineering and system architectures as well as the available design tools and automatic or interactive design facilities have generated a plethora of system-design methodologies which present an overwhelming problem to the novice in the field. This is serious, since there is no single design approach that can be applied to all possible design cases with optimal results. This situation cries out for developing a classification of design methods and tools so that each individual design approach can be evaluated in the proper perspective.

To avoid ambiguity, let us first introduce frequently used terminology associated with the design of complex chips.

Customer, Designer and Manufacturer

When chips are to be manufactured, we may identify three distinctive persons or groups of persons, which we simply call the customer, the designer and the manufacturer. The *customer* wants a chip to be designed and manufactured. The design activity required to meet the customer's specification is called *customization*. The *designer* performs the circuit and system design according to the customer's requirements. Finally, we need a chip *manufacturer* to implement the circuit and system in the form of the desired chip. The three categories (customer, designer and manufacturer) need not reside at different locations. In big companies, such as IBM, AT&T Bell Labs and Philips, all three categories are employed in-house. The customer and designer may be identical so that we have the customer/designer on the one hand and the chip manufacturer on the other hand.

For reasons of profit, semiconductor companies were formerly mainly concerned with the manufacture of standard chips, ranging from simple off-the-shelf chips to microprocessors and large-scale memories. Gradually, these companies are producing more and more final products, such as digital watches, video games, educational toys and even microcomputers. At the same time,

system companies (such as IBM, Honeywell, NCR, Burroughs, Sperry Univac and Control Data Corporation), which were founded as mainframe-computer designers, are now equipped with extensive in-house chip-manufacturing facilities. The above trend of bringing together the system-design and chip-manufacturing activities within one company is sometimes called *vertical integration.* The increasing demand for custom designs has created the opposite trend, which will be discussed below.

Silicon Foundries

The past decade has shown a trend toward transferring the fabrication process to separate semiconductor companies which commit themselves exclusively to the manufacture of the chips, which are the physical implementations of customized designs. These companies are called *silicon foundries.* The idea is that one company (or customer/designer) is engaged in the IC design process resulting in the complete set of masks, which is used by another company (the silicon foundry) to produce the desired chips. The reason for the emergence of silicon foundries is obvious. A system designer cannot afford to set up a fabrication line which requires a large starting investment in the equipment needed along with the required skills and experience.

Leaving the chip manufacture to a silicon foundry is not devoid of difficulties. The mere fact that the system company (customer/designer) and the silicon foundry are separate organizations may bring up several questions. In what form should the designer submit the chip-design specifications (e.g., layout-description files for pattern generation) to the chip manufacturer? To what extent is the manufacturer willing to provide detailed information on the processing technology, including the latest developments? In any case, the system company and the silicon foundry should reach agreement on several aspects of design and implementation. Particularly, the performance of an IC design must be accurately verified before committing it to fabrication. For this purpose, many silicon foundries make circuit-simulator models and parameters for their technologies available to the customer. Design verifications must be performed with the proper device parameters and process specifications pertaining to the implementation technology which is supported by the silicon foundry. The layout descriptions must meet the design rules imposed by the technology. The large capital involved in chip production justifies testing the prototype circuit thoroughly. To this end, a special integrated transistor circuit is usually fabricated on the same wafer as the customer's prototype circuit.

There is a wide variety among silicon foundries as to the implementation technology, the integration density, the system performance and the minimum amount of chips that can be fabricated. Some silicon foundries accept small-volume orders, while others require a minimal level of commitment. Important issues in dealing with silicon foundries are the the implementation technology

used, the production costs, the simulation facilities and post-fabrication services.

Automated Design

From the viewpoint of design automation, we can distinguish manual design and automated design. *Manual design* proceeds by human methods and the layout design is carried out as a handcraft. The experience of the chip designer plays an essential role. In *automated design*, automatic synthesis methods are employed and the masks are produced automatically by appropriate layout design progams. Manual design is aimed at achieving compact layouts, which may have an irregular structure. It produces the best results in terms of high performance and silicon area utilization. The handcrafted design is profitable for either high-volume production of standard circuits or full-custom designs which require high circuit performance. When the circuit becomes too complex or when a small turnaround time is required, we have to resort to using automatic design tools. Automated design typically differs from manual design in that the layout structure is restricted to standard cells, PLAs, gate arrays or other structures which are amenable to easy automation of the design process. Regular layout structures tend to alleviate the interconnection problem.

Application-Specific Integrated Circuits

An IC design is referred to as *(semi)custom design* or *Application-Specific Integrated Circuit* (ASIC) *design* when the IC package is designed according to the particular specification of an individual customer. As mentioned earlier, the customer and chip designer may be the same person or group of persons. The custom design approach may take several forms. A simple approach is to utilize a manufacturer's standard chips, which are interconnected on a thin-film or thick-film substrate and encapsulated in a single package.

In a full-custom design, all the steps of the design cycle are dictated by the customer's requirements, which are unique in terms of the hardware implementation and the circuit performance. Full-custom design is the right choice when a high circuit performance is mandatory, either in terms of a very high operating speed, an extremely low power consumption or any other desirable property required by the particular application. Special care is exercised with respect to the layout structure and the diagnosability. The use of manual layout techniques imposes strong demands on the ingenuity and the experience of the chip designer. In order to achieve the optimum compromise between the high circuit performance of a full-custom design on the one hand and the constraints imposed by the implementation technology on the other hand, there should be a close liaison between the customer/designer and the chip manufacturer.

When the number of chips required is low, the long design time inherent in the full-custom approach may make the total design cost prohibitive. There is a considerable need for customized designs which must be completed on a rigid

time scale. A small turnaround time can be achieved by using *automatic design tools*. A custom design which makes use of automatic design tools is often referred to as *semicustom design*. ASIC design is usually connected with semicustom design rather than with full-custom design. Let us look closer at the trend toward using automatic design tools, which is characteristic to ASIC design.

Selecting a Design Methodology

Which actual design strategy to choose in a particular design case depends on various factors, including the kind of application, the production volume, the implementation technology and the CADCAS tools which are available. Manual layout, standard-cell, gate-array and block layouts can be employed for virtually all kinds of applications. The PLA structure is useful for implementing combinational logic and finite-state machines and hence is suitable for implementing the control path of the system. The standard floorplan technique in the form of a silicon compiler is a special form of the hierarchical block design approach, which is useful for designing microprocessors.

The advent of the microprocessor has opened up two *design style* options:

1. The *hardware-oriented* design style, which involves the design of a logic system and the implementation of dedicated hardware components required to construct the logic system.
2. The *microprocessor-based* design style, which uses a general-purpose data processor (notably a microprocessor) as the central component of the system so that the design can be completed only by developing the hardware interfaces and the software needed for the system to perform the prespecified system function.

Option 1. can be divided into two different design approaches:

a. The *distributed* or *hard-wired* custom design style, which emphasizes manual procedures to achieve a high circuit performance.
b. The *structured* semicustom design styles, which imply the use of structured layouts and automatic methods for achieving short turnaround times, even at the expense of the circuit performance and chip-area utilization.

Option 2. with the microprocessor as the basic component is flexible in that different tasks can be performed merely by changing the controlling software.

An alternative classification of design styles may be based on the *system architecture*. Among other styles, we may discern:

a. The *distributed* composition of logic modules, which are SSI, MSI or LSI building blocks.
b. The *bus-oriented* system architecture, which uses registers and other system components for data processing and storage, all of which are attached to buses which function as communication media between the different parts of the system.

c. *Parallel* and *pipeline* architectures, which use a number of processors, as opposed to the normal Von Neumann architecture.

The ultimate design style to be selected must provide an efficient solution tailored to the particular customer's requirements. The total design costs may include the costs for conventional programming and microprogramming, the design costs for hard-wired methods or the costs for using automatic design tools when using standard cells, gate arrays or PLAs. A special methodology which has gained a widespread popularity in the past decade will be described below.

CADCAS and Expert Systems

The increasing complexity of microelectronic chips has stimulated designers to develop automatic tools which perform the various tasks involved in chip design [Sha89]. Undoubtedly, these tools facilitate and speed up the design process, at least in a large number of individual design steps in the hierarchical design cycle as shown in Fig. 1.1.

A characteristic feature of these tools is that they execute a fixed algorithm, that is, given the specific method for solving the problem at hand, a prescribed procedure of executing a sequence of instructions is followed, leading to a fixed solution, if there is one. In fact, the design specification (including performance criteria and design constraints) define a potential solution space, within which a particular design is selected after performing a number of trade-offs between counteracting design parameters (e.g., operating speed and power dissipation). In the hierarchical design cycle, a number of intelligent decisions have to be made by the human designer. It is the aspiration of the chip designer community to conceive a design methodology that, given a design specification, produces an optimal design without any human intervention. This implies that all tasks involved in the entire design cycle, including the intelligent decision making, must be executed solely by the computer.

In recent years, much effort has been spent to the development of expert systems (see Chapter 4), which solve problems by utilizing expert knowledge (in the form of facts and rules) captured in the knowledge base of the system. Expert systems employ reasoning methods in the way humans do. While various tasks are performed by executing algorithms, intelligent tasks not amenable to algorithmic problem solving can be relegated to expert systems. Although some human intervention can never be dispensed with, the introduction of expert systems into the set of CAD tools will bring the design process nearer to the point of full automation. The use of expert systems not only reduces the turnaround time, but also explains and justifies why a particular solution is proposed.

An introduction to expert systems will be given in Subsection 3.3b and Chapter 4. The applications of expert systems in VLSI chip design, verification, testing and layout design are dealt with in Chapters 5 through 8, after a brief

introduction of conventional algorithmic design. It should be emphasized that expert systems are not to replace conventional algorithmic approaches to design, but rather as a complement to them.

Implementation Technologies

A rich variety of implementation technologies and circuit-design techniques are available to the integrated-circuit designer. The decision as to which implementation technology or logic family will be chosen for the circuit implementation is governed by the intended applications of the integrated circuit. In some cases, the designer is committed to using the implementation technology available within the company. Let us consider some technologies.

Bipolar technology. The propagation delay of a logic gate is proportional to the node capacitances and is inversely proportional to the transconductance of the gate. The exponential current-voltage relationships in a bipolar transistor lead to a high transconductance and the attendant high current-drive capability. A bipolar transistor can thus provide a larger current per unit active area than a corresponding MOSFET device can. This explains the good high-speed quality of bipolar transistors. Bipolar technology is primarily suitable for analog-circuit applications due to the high single-stage gain-bandwidth product. A major problem with Si bipolar devices is the power dissipation [Nin86].

The TTL (Transistor-Transistor Logic) technology has long been a standard technology for LSI circuits. ECL (Emitter Coupled Logic) is still being used due to its high-speed properties.

MOSFET technology. The simplicity of MOSFET circuitry and fabrication along with its amenability to further size reduction makes MOSFET technologies very well suited to VLSI fabrication of logic circuits and memories. The high input impedance of a MOSFET gate alleviates the loading problem, allowing a large fanout to be used. Especially, silicon-gate MOSFETs [Mye86] hold promise for the future due to the reduced threshold voltages and speeds as high as 100 MHz. The ease of designing MOSFET circuits has given impetus to the rapid development of LSI and VLSI systems. MOSFET technology is responsible for several landmarks in the history of microelectronics: the pocket calculator, the MOS memory and the microprocessor. The first commercially available microprocessor (manufactured in 1971) was the four-bit Intel 4004. MOSFETs are divided into N-channel and P-channel transistors.

CMOS technology. Complementary-MOS technology is acquiring a growing stake in the world markets for microelectronic chips. These are some of the most striking features of CMOS technology:

a. CMOS chips are praised for their extremely low power consumption. At moderate speeds, the power dissipation is less than a microwatt per gate. CMOS is a good choice when low standby power is imperative, notably in battery-operated applications.

b. CMOS inverters have a nearly ideal DC transfer characteristic. CMOS noise immunity is very much better than those of other technologies.
c. The above properties are valid for a wide operating range of power supply voltages, i.e., CMOS circuits do not need expensive close-tolerance power supplies. The body effect, which degrades the characteristics of non-complementary static inverters, is of no account in CMOS inverters.
d. The fact that CMOS inverters are ratioless facilitates the design of CMOS circuitry. The term "ratioless" is related to the length-width ratios of the transistor channels.

As drawbacks, CMOS circuits require more complex processing and more layout area than NMOS circuits and, in addition, CMOS circuits has the possibility of latch-up.

BiCMOS technology. Merging a CMOS structure with a bipolar transistor leads to the BiCMOS technology which is successfully employed in the design of high-speed static RAMs and gate arrays. This technology offers the system designer the low-noise characteristics of bipolar devices at high speeds, while maintaining active CMOS power-dissipation levels [Cor87].

GaAs technology. Most existing devices are based on the use of silicon (Si) as the semiconductor material. For high speeds, gallium arsenide (GaAs) may offer an attractive alternative [MIL86]. Electrons move faster through GaAs crystal lattice than through Si crystal. Depending on conditions, the electron mobility is 6 to 10 times higher in GaAs than in Si. Since chip density of GaAs technology is rapidly increasing, GaAs becomes a severe contender of silicon. The saturated drift velocity is about twice as high as in Si. On the other hand, the relative effect of interconnection capacitances in GaAs technology on the propagation delays seems to be more severe than in Si technology. The very high intrinsic bulk resistivity of the GaAs substrate minimizes parasitic capacitances and permits easy electrical isolation of multiple devices on a GaAs integrated circuit chip. GaAs technology will prevail at frequency levels greater than 100 MHz. GaAs chips have been employed in the Cray-3 supercomputer.

Though in principle any technology can be used for different applications, a specific technology is particularly suitable for some class of applications: ECL for mainframes, BiCMOS for Superminis, CMOS for minis and personal computers, and GaAs for high-speed applications.

Since integrated circuits started to emerge, feature sizes have shrunk continuously. Submicron technology has already emerged. With minimum feature size, chips are vulnerable to defects and improving the reliability becomes an important consideration.

1.2 CADCAS TOOLS IN CHIP DESIGN

1.2a Design, Simulation and Testing

Analog Circuits

Analog circuit design has a long history, starting with the use of vacuum tubes in the early twenties. At the present time, microelectronic circuits employ bipolar, NMOS, CMOS or BiCMOS technology. It is apparent that circuit design exploits the vast knowledge gained in the past. Therefore, new circuit designs are modifications of existing designs.

Although the bulk of this book is focused on digital circuits and systems, the significance of analog circuits should not be underestimated for two reasons:

1. Analog integrated circuits are being widely used as low-cost and high-quality components in various applications, notably in signal processing, including channel filters, analog-to-digital and digital-to-analog converters, modulators, speech processors, operational amplifiers and equalizers.
2. Even digital circuits exhibit an inherently analog behavior if exact timing analysis of digital signals is performed. Though binary logic is based on the use of two distinct signal levels, the transition from one signal level to the other occurs continuously rather than abruptly.

The field of analog circuits and signal processing is changing dramatically for two reasons:

a. VLSI is now maturing, with emphasis on submicron structures and on sophisticated applications that combine digital and analog circuits on a single chip. Examples are advanced systems for telecommunications, robotics, automotive electronics, image processing, and intelligent sensors.
b. The rapid technological developments that are leading toward single-chip, mixed analog/digital VLSI systems require design strategies that bridge the gap between classical analog design and VLSI. MOS switched-capacitor and continuous-time circuit designs, on-chip automatic tuning, and self-correcting and self-calibrating design schemes are but a few examples of these emerging technologies.

Smart and dependable CAD tools are needed for designing analog and mixed analog/digital VLSI, mixed-mode simulators, analog layout tools, automatic synthesis tools, and the development of analog-hardware languages.

In early days of electronics history, the design of an electronic circuit is intimately related to its hardware implementation. The relatively simple circuitry made it possible to verify design correctness by electrical measurements directly on the circuit nodes, followed by modifications, if necessary. Integrated analog circuits have the characteristic that only input nodes and output nodes are available for applying input signals and observing the output responses. Since the internal nodes are usually not accessible for measuring purposes, design

verification must be carried out on a realistic circuit model, which is appropriately analyzed. The combination of circuit modeling and the subsequent analysis is called *simulation.* Several circuit simulators have been developed, including SPICE2, ASTAP and NAP2 [Sch87]. Most analog circuit simulators are based on the same principle. The circuit is assumed to be composed of an interconnection of circuit components (transistors, resistors, capacitors, etc.).

Unlike design automation of digital systems, which has grown to maturity, analog design automation has made little progress. The reason is that analog circuits are less amenable to design automation than logic circuits. To minimize design time delays, there is a strong need for improved CAD tools that support analog circuit design. One practical approach to analog circuit design is the use of basic subcircuits and amplifiers as building blocks to construct more cpmplex circuits [All87]. Several articles on automated analog-circuit design have been published. See also Subsection 6.3b.

Though analog design automation has by far not reached the advanced stage of digital design automation, analog system design environments are beginning to emerge. An example is IDAC3 (Interactive Design of Analog Circuits, third version) of the Swiss Center of Microelectronics, Neuchâtel [Deg88]. Its key features are multifunctionality, advanced modeling, new simulation approaches, general sizing algorithms, hierarchy and interactive layout-generation capabilities. A workstation-based environment for the fabrication process and the device level is the Process Engineer's Workbench of Carnegie-Mellon University, Pittsburgh [Str88].

Digital Logic Circuits and Systems

Increasing the accuracy of analog-circuit performance requires better control of physical processes. Digital systems can be made more accurate simply by increasing the number of bits for representing parameter and data values. A higher transistor density in digital chips results in falling prices and better performance of the system. Digital chips are able to encompass more and more complex functions on one chip.

A key aspect of VLSI designs is the computational complexity which increases as the system size increases. Some deterministic computational algorithms are of *polynomial-time complexity*, that is, such algorithms can be performed in a computational time $T = c\, n^k$, where c and k are constants. Unfortunately, many VLSI design problems are non-deterministic and substantially more complex. They are said to be *NP complete.* Appropriate heuristics are used to solve NP-complete problems. The classical solution to controlling computational complexity of a design problem has been to identify subsets of the problem which are computationally manageable.

A VLSI design cannot be committed to chip fabrication, unless the design has has been verified to be correct. In special cases, a formal design procedure can