Guest Editor’s Introduction

Guest Editor's Introduction

Special Issue of VLSI Design: Decomposition in VLSI Design

LECH JÓŹWIAK
Guest Editor

The term *composition* means the act or state of arrangement of various or similar parts (things, actions, functions, etc.) into a proper relation in order to form a whole (an aggregate). This term is also used to describe the nature or constitution of an aggregate, i.e. the way in which the parts form the whole. The term *decomposition* means the act or state of disintegration or breakdown of a whole into a system of constituent parts (simpler aggregates or elements). These two notions are very general and can be used in a lot of various contexts.

In the context of problem solving, decomposition can mean breaking down a complex problem into a system of simpler sub-problems. The solution to the original problem can then be achieved by solving the sub-problems and composing the sub-problem solutions into an aggregate solution. Of course, the sub-problems cannot always be solved independently. They often require appropriate coordination and this can take various forms ranging from solving the sub-problems in a certain order (when using the results of earlier solved problems for solving the subsequent sub-problems) through intercommunication and synchronization between the sub-problems to introducing and solving special coordination sub-problems. In the area of system design, "decomposition" and "composition" are used in the above sense for design problem solving.

The structure design problems form a very important class of system design problems. A system *structure design problem* consists of providing a structure composed of sub-systems that, when implemented, realizes the specified system's behaviour, satisfies certain constraints and optimizes specific objectives. The structure design breaks down a complex system into a structure of cooperating sub-systems, which can in turn be further broken down to the simpler sub-systems that can be directly implemented with the elements at the designer's disposal. The resulting system structure can consist of physical and virtual sub-systems; however, each virtual sub-system must be finally implemented with, or composed of, some physical elements. Of course, it is not necessary to express the system structure only in the atomic concepts. Some higher level concepts not only may, but should be used, realizing the reuse of earlier (sub-)designs, properly organizing and accelerating the design and fabrication processes to result in more "matured" products. Therefore, the system structure design can finish with some higher level concepts. Of course the designer must have or know their implementation or their implementation must be possible to be constructed automatically with CAD tools or it must be delivered by a separate design. From the above, it can be directly concluded that the structure design problems are basically specific decomposition problems. Therefore, the terms structure design and *structural decomposition* are interchangeable.

Every design process of a purposive system begins with the general aim of design, i.e. a design problem. The design process aims to solve the problem by providing a design of a system (problem solver) which matches the behavioural and parametric characteristics required, can be directly implemented and can be developed, produced, used and disposed according to the requirements imposed by the original problem. Therefore, the design, fabrication, production, usage, and disposal processes, together with the designed system, can be considered as a complex problem solver for the original problem, and the designed system can be considered as a "kernel" problem solver. Each of these processes can in turn be decomposed into many sub-processes. The solution to the original design problem is therefore obtained by finding an appropriate decomposition into a system of sub-solutions for certain sub-problems. The designed system itself solves a "kernel" part of the design problem by exposing the required behaviour and matching the required parametric characteristics. However, since the designed system is composed of a number of sub-systems, it solves the "kernel" design problem in a decompositional manner. Each sub-system solves a certain sub-problem and all sub-systems cooperating together solve the whole problem by exposing the external aggregate behaviour which matches the originally required behaviour.

From the above, it can be concluded that decomposition is a general and basic activity and state of a real-life systemic problem solving. In particular, it is a general and basic activity and state of a system design process.

Decomposition is not a new phenomenon—people have been doing it perhaps forever. What is new is the necessity to perform complex decompositions extremely effectively and efficiently. Today's systems require too high quality, are too complex, and involve too many aspects to be designed ad hoc. The design processes must be well-organized, systematic and coherent and they have to involve very precise, efficient and factual decision making. Therefore, research which aims to develop the theoretical foundations of decomposition, as well as the decomposition methods, techniques and tools, is of primary importance. This research should provide solid theoretical fundamentals and methods which will enable decompositions to be performed in a well organized, systematic, and coherent manner and also to allow for growing automation of this basic and general design activity. In fact, only automation can enable effective and efficient exploration of the vast space of possible decompositions.

This special issue is devoted to decompositions in VLSI design. VLSI design is understood here in its broadest meaning, i.e. as the design of VLSI circuits as well as design of systems composed of VLSI circuits and software. Most of the papers in this issue are devoted to decompositions in logic design.

The history of logic design is not so old as the history of decomposition. As an engineering discipline it started perhaps with the development of the first mechanical digital calculation machines by Pascal and Leibnitz in seventeenth century and first mechanical digital controllers for controlling looms by Falcon around 1720. Since then, various mechanical and electro-mechanical calculation machines and controllers have been developed. The first electronic computer, ENIAC, was built by Eckart and Mauchly at the University of Pennsylvannia in 1946, but the first reasonably powerful electronic computer (0,25 MIPS) was LARC of Univac built in 1960. However, a real revolution in logic design came with the invention of the transistor by Shockley, Bardeen and Brattain in 1947, and the usage of semiconductor technology for implementing logic circuits as integrated circuits. Due to the rapid development of semiconductor technology, the complexity of integrated circuits grew very quickly from SSI to VLSI and ULSI.

The theoretical roots of logic design can be found in the works of Georges Bool which date from around 1850, but just a century later Boolean algebra produced the first practical applications in the form of a switching network theory for relay networks. In the 1960s and 1970s, Boolean algebra was used for a two-level AND-OR network minimization (gate networks and PLAs). At that time, it became clear that for circuits of the required complexity, logic design cannot be performed manually and subsequently the first logic synthesis tools emerged.

A two-level AND-OR implementation consists of a decomposition of a Boolean function into a number of sub-functions being AND functions. Each of these sub-function is then implemented with a separate AND circuit. Computation results of all these AND circuits are aggregated by using a circuit that implements an OR function. By the early 1960s, some people have already realized that the two-level implementation is a very special decomposition case which is not necessary the best one and therefore they started to look for more general decompositions. The first theoretical works in the field of a combinational circuit decomposition were presented by Ashenhurst and Curtis and, in the field of sequential circuit decomposition, by Hartmanis. These works were very important but they did not produce a complete decomposition theory and technology for use in practice. The first reasonably practical solution for the multiple-level AND-OR synthesis was proposed in 1982 and it was the algebraic division method of Brayton for a multiple-level logic synthesis. It was not an ideal solution and remained a synthesis with AND and OR elements, but the vicious circle of the two-level logic synthesis as a unique synthesis form was broken.

Although logic designers have been building circuits for years, they have realized that advances in microelectronic technology are outstripping their abilities to make use of the created opportunities. Modern microelectronic technology gives opportunities to build digital circuits of huge complexity and provides a huge diversity of logic building blocks. The element libraries contain not only ANDs, ORs, NOTs, NANDs, and NORs, but many various gates. The field programmable logic families include various types of FPGAs, PLDs, and CPLDs. AND-OR-based synthesis is not sufficient. If used for target implementation structures, which considerably differ from the AND-OR structures, it results in the necessity of technology mapping and this does not guarantee an optimal solution if the initial network is constructed without any regard to future implementation. The opportunities given by microelectronic technology cannot be effectively exploited. This has created a strong stimulus for developing a general decomposition theory and methodology, which will enable a logic designer to cope with various logic building blocks and any sort networks. It also stimulates development of specific decomposition methods for targets differing from the AND-OR networks.

The first paper of this special issue presents a general decomposition theory for sequential and combinational circuits. It shows that all the known decomposition models are special cases of general decomposition model and explains how to use the general decomposition theory for the digital circuit synthesis and design verification. The theory presented in the paper constitutes the theory of correct digital circuit structures and forms a sound base for the construction of decomposition algorithms. The logic design methodology based on this theory guarantees "correctness by construction," enables very effective and efficient post-factum verification and makes possible extensive examination of the space of correct circuit structures in relation to the actual constraints and objectives.

The second paper presents a method for decomposition of a sequential machine into a network of interacting sequential machines and a method for correctness verification of the component machine network by a special kind of simulation. The decomposition model that it uses is a special case of the general decomposition model presented in the first paper (output bit decomposition with the state and output behaviour realization). The decomposition method is devoted to logic implementations with the two-level AND-OR networks (e.g. composed of PLAs). It attempts to reduce the complexity of each partial machine while keeping the number of machines close to a minimum.

The next four papers deal with the combinational logic synthesis. The first of these analyses and compares a general decomposition-based approach and a division-based approach for the multiple-level logic synthesis. It shows that the division-based synthesis is a very specific case of the general decomposition-based synthesis and discusses the implications of this. In addition, the particular synthesis methods are analyzed and their strengths and weaknesses are discussed. This shows among others that the division-based methods can still be improved. This paper also gives a good introduction to the following three papers.

The next paper describes a general decomposition-based method for FPGA-based synthesis. This method decomposes the multiple-output Boolean functions into the FPGA networks using the parallel (input and output bit parallel) and serial (input bit parallel) decompositions and by trying to find their appropriate combination. The benchmark results presented show that this method is very effective for the circuits based on look-up tables.

The following paper introduces several new families of decision diagrams for representing multi-output Boolean functions and discusses their application in multi-level logic synthesis. The new families include the known from literature binary decision diagrams (BDDs) and functional decision diagrams (FDDs) as special cases. Used for synthesis, they result in diagrams with less nodes than BDDs or FDDs and allow for usage of AND, OR, and EXOR gates for circuit realization.

The last paper on the combinational logic synthesis presents a co-generation of a fine-grain FPGA architecture and an algorithm for optimal synthesis of combinational circuits with this architecture. Traditionally, the FPGA devices are designed first and the synthesis methods for these devices are subsequently created. Therefore, the difficulty of synthesis is not properly considered during design of the device. This can result in an unnecessary complex synthesis. The approach presented in this paper is to create a good FPGA architecture which will enable a relatively easy synthesis by the simultaneous design of the device and algorithm.

The next two papers consider the decomposition problem at the so called system-level of design. Although the heart of the decomposition problem and the philosophy of its solution are independent of the design level (in fact the circuits of the logic design level are lower-level subsystems), each design level has its own specific characteristics which can influence the precise solutions. Perhaps the most important differences between logic design level and system design level are the following:

- the quality metrics at the system level can be more complex and can involve more attributes;
- the design attributes represented by the variables in the quality metrics can be more difficult to estimate at the system level;
- the elementary computation items (atomic behaviours) considered at the system level can be more complex and characterized by more attributes;
- the interfaces between the sub-systems of the higher levels can be more complex than between the sub-circuits.

For the system modeling, both papers use some extended FSM models. The first paper presents an interactive system-level partitioning method and a tool box which helps to perform the partitioning. Partitioning at this level involves mapping of the atomic behaviours to processors and variables to memories, as well as synthesis of communication channels and their mapping to buses. The second paper introduces a formalism and describes a method for factoring the sequential components of the system specification by the interface specification of the components and complimentation.

The last paper of this special issue discusses decomposition in the context of more general problem solving namely, the utilization of decomposition for development of algorithms. It focuses on the algorithms for solving the class of problems where a finite set of objects and a system of constraints are given, and a minimum subset of objects satisfying the specified constraints is to be found. A lot of VLSI design problems can be modeled in such a manner. In particular, the unate and binate covering problems belong to this class. Examples of well-known problems from this class are the following: design of a minimum-cost two-level combinational network, minimum cost TANT network or minimum test set, minimization of Boolean relations and the optimal library-based technology mapping.

We hope that the papers presented in this issue, by providing backgrounds, innovative ideas, stimuli and directions for further work of the research and design community, will influence the field of VLSI design in such a way that the designing of VLSI circuits and systems will proceed in a more coherent, systematic and efficient fashion.

Finally, we would like to acknowledge the encouragement and help obtained from the Editor-in-Chief, George W. Zobrist, in organising this special issue and we would also like to thank all the reviewers who helped to improve the quality of the papers presented in this issue.

Lech Jóźwiak
Eindhoven University of Technology
Faculty of Electrical Engineering
P.O. Box 513
5600 MB Eindhoven
The Netherlands

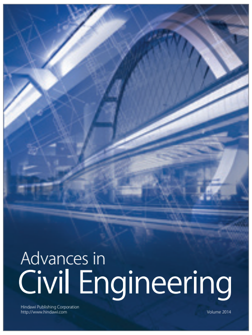
Journal of Engineering

The Scientific World Journal

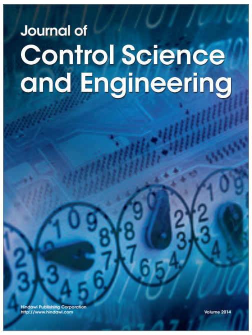
International Journal of Rotating Machinery

Journal of Sensors

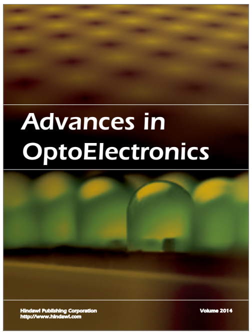
International Journal of Distributed Sensor Networks


Advances in Civil Engineering

Journal of Robotics

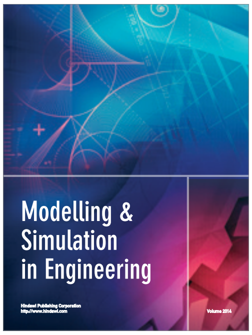

Journal of Control Science and Engineering

Journal of Electrical and Computer Engineering

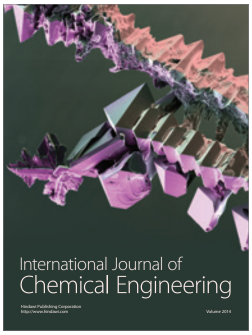

Advances in OptoElectronics

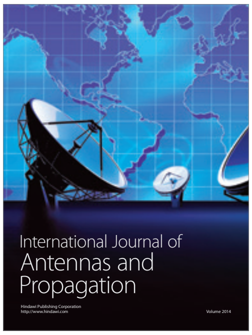
VLSI Design

International Journal of Navigation and Observation

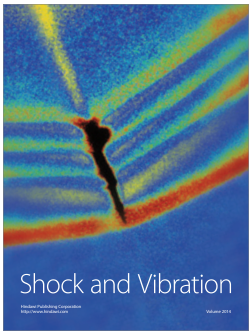

Modelling & Simulation in Engineering

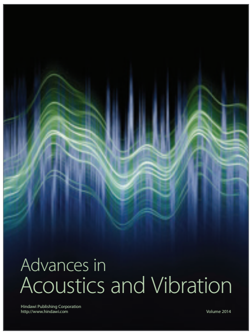
International Journal of Aerospace Engineering


International Journal of Chemical Engineering


International Journal of Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in Acoustics and Vibration