Preface

The up-to-day state of the art in the computer science is characterized by three major factors. The first factor is a development of ultra complex VLSI such as "system-on-programmable chips" having billions of transistors and hundreds of millions of equivalent gates. The second factor is a development and widespreading hardware description languages (HDL) such as VHDL and Verilog that permits to capture a design with tremendous complexness. The third factor is an existence of different computer-aided design (CAD) tools. It allows designing very complex projects in the satisfactory time-to-market. These three factors affected significantly the modern process of hardware design. Now the hardware design is very similar to the development of computer programs. An application of HDLs together with CAD-tools allows concentrating the designer's energy on the basic problems of design, whereas a routine work remains the prerogative of computers.

Tremendous achievements in the area of semiconductor electronics has turned microelectronics into nanoelectronics. Actually, we observe a real technical boom connected with achievements in nanoelectronics. It results in development of very complex integrated circuits, particularly in the area of field programmable logic devices. Our book targets field-programmable gate arrays (FPGA). First FPGA chips were introduced by Xilinx in 1985, i.e. exactly 30 years ago. They were used mostly as devices implementing simple random and glue logic. Up-to-day FPGAs have up to 7 billion of transistors. So, they are so huge, that it is enough only one chip to implement a rather complex digital system including a datapath and a control unit. Because of the extreme complexity of FPGA chips, it is very important to develop effective design methods targeting their particular properties. It means that the design methods should be technology-depended.

As it is known, any digital system can be represented as a composition of a datapath and a control unit. Logic circuits of operational blocks forming a datapath have regular structures. It allows using standard library elements of CAD tools (such as counters, multibit adders, multipliers, multiplexers, decoders and so on) for their design. A control unit coordinates interplay of other system blocks producing a sequence of control signals (microoperations). These control signals cause

executing required actions by a datapath. As a rule, control units have irregular structures. It makes process of its design very sophisticated. Many important features of a digital system, such as hardware amount, performance, power consumption and so on, depend to a large extent on characteristics of its control unit. Therefore, to design competitive digital systems with FPGAs, a designer should have fundamental knowledge in the area of logic synthesis and optimization of logic circuits of control units. As experience of many scientists shows, design methods used by standard industrial packages are far from optimal. Especially it is true in the case of designing complex control units. It means that a designer could be forced to develop his own design methods, next to program them and at last to combine them with standard packages to get a result with desired characteristics. To help such a designer, this book is devoted to solution of the problems of logic synthesis and reduction of hardware amount in control units. We discuss the case when a control unit is represented by the model of finite state machine (FSM). The book contains many original synthesis and optimization methods based on the taking into account the peculiarities of FPGA chips and an FSM model in use. One of the peculiarities of FPGA chips is existence of embedded memory blocks (EMB). We try to implement with EMBs as much of control unit's circuits as it is possible. It results in reducing the irregular part of the control units described by means of Boolean functions. It permits decreasing for the total number of look-up table (LUT) elements in comparison with logic circuits based on known models of FSM. Also, it decreases the number of interconnections in the resulting circuits. In turn, it makes the problem of place-and-routing much simpler. The third benefit is the reducing power dissipation in comparison with FSM circuits implemented only with LUTs. In our book, control algorithms are represented by graph-schemes of algorithms (GSA). This choice is based on obvious fact that this specification provides the simple explanation of the methods proposed by the authors.

To replace LUT-based subcircuits in FSM logic circuits by EMB-based parts, it is necessary to diminish the number of arguments in corresponding systems of Boolean functions. To do it, we propose using the methods of structural decomposition, such as: (1) the replacement of logical conditions; (2) the encoding of the collections of microoperations; (3) the transformation of FSM objects, when the internal states are represented as functions of microoperations and vice versa. Also, we use the existence of the classes of pseudoequivalent states of Moore FSMs to compress structure tables and diminish the number of state variables. It simplifies the system of input memory functions and, therefore, decreases the number of EMBs in the resulting FSM circuit. We combine this approach with using EMBs for implementing the system of output functions (microoperations). It allows a significant decreasing for the number of LUTs, as well as eliminating a lot of interconnections in the FSM logic circuit. It saves area occupied by the circuit and diminishes the resulting power dissipation. Of course, it leads to more sophisticated synthesis process than the one targeting only LUT-based circuits.

The process of FSM logic synthesis is reduced to a transformation of a control algorithm into some tables describing the behaviour of FSM blocks. These tables are used to find the systems of Boolean functions, which can be used to implement

logic circuits of particular FSM blocks. In order to implement corresponding circuits, this information should be transformed using data formats of particular industrial CAD systems. We do not discuss this step is in our book. Our book contains a lot of examples showing design of FSMs with using the proposed methods. Some examples are illustrated by logic circuits.

The book contains Introduction, eight chapters and Conclusion. It is written by a research group from University of Zielona Góra (Poland). Professors Alexander Barkalov and Larysa Titarenko wrote Introduction, two first chapters and Conclusion. Chapters 3 and 4 are written by Ph.D. Kamil Mielcarek. Chapters 5 and 6 are written by Ph.D. Grzegorz Bazydlo. Chapters 7 and 8 are written by Ph.D. Malgorzata Kolopienczyk.

Chapter 1 provides some basic issues connected with finite state machines and field programmable logic devices. The basic models of Mealy and Moore FSMs are presented. The classical design methods are presented for three kinds of control units: microprogrammed automata, microprogram control units and compositional microprogram control units. Main methods of hardware reduction are given for finite state machines implemented with field programmable logic devices, such as the replacement of logical conditions, encoding of collections of microoperations and encoding the fields of compatible microoperations.

Chapter 2 is devoted to application of field programmable gate arrays in design of logic circuits of FSMs. The basic features of FPGA are analysed. It is shown that embedded memory blocks allow effective implementing systems of regular Boolean functions. Next, the basic problems of FSM design are considered. Different state assignment methods are analysed in details, as well as the methods of functional decomposition. At last, there are discussed the methods of hardware reduction for FPGA-based FSMs.

Chapter 3 presents the original methods of hardware reduction based on the transformation of object codes of Mealy FSMs. The principle of object code transformation (OCT) is introduced. Two types of objects are introduced: internal states and collections of microoperations. Two types of basic models of Mealy FSMs with OCT are described. The design methods are proposed for the EMB-based FSMs with transformation of the states into the collections of microoperations. Next, the design methods are shown allowing the transformation of the collections of microoperations into the states. The models of FSMs with the replacement of logical conditions and OCT are discussed. At last, the analysis of the proposed methods is executed giving conditions of their application.

Chapter 4 deals with the original methods of hardware reduction based on the transformation of object codes of Moore FSMs. Two types of basic models of Moore FSMs with OCT are described, as well as EMB-based structures corresponding to these models. The design methods are proposed for the EMB-based FSMs with transformation of states into the collections of microoperations. Next, the design methods are shown allowing the transformation of the collections of

microoperations into the states. The models of FSMs with the replacement of logical conditions and OCT are discussed. The additional hardware reduction is achieved due to using the classes of pseudoequivalent states.

Chapter 5 deals with optimization of logic circuits of Moore FSMs based on using two and three sources of codes of classes of pseudoequivalent states (PES). First of all, the application of this method for CPLD-based FSMs is discussed. Next, the models with two sources of class codes are discussed and corresponding design methods are proposed. This approach requires the usage of a multiplexor to choose a particular source. Also, the models with three sources of class codes are discussed and corresponding design methods are proposed. It is shown how the replacement of logical conditions can be used in multisource models of FSMs. At last, it is shown that the hardware reduction can be obtained due to increasing the number of class variables.

Chapter 6 is devoted to hardware reduction based on using many directions of input memory functions in Moore FSMs. First, the hardware reduction methods are proposed for the two-directional Moore FSMs. They are based on the special state assignment allowing the decreasing for the number of literals in sum-of-products representing input memory functions. Next, the design methods are proposed for the three-directional Moore FSMs. It is shown that the number of directions can be increased. It leads to simplifying the input memory functions in comparison with the single-directional models. The last part of the chapter is devoted to combining the replacement of logical conditions with many directions of state codes.

Chapter 7 deals with design of Mealy FSMs based on using embedded memory blocks. The methods of trivial EMB-based implementation of logic circuits of Mealy FSMs are discussed. In this case, only one EMB is enough for implementing the circuit. Next, the optimization methods are discussed based on encoding of the collections of microoperations and replacement of logical conditions. Also, the methods are discussed based on encoding of the rows of FSM structure table. All these methods lead to two-level models of Mealy FSMs. Next, these methods are combined together for further optimizing the hardware amount in FSM logic circuits. The last section considers different methods proposed for diminishing the hardware amount in LUTer implementing the block of replacement of logical conditions. The chapter includes a lot of tables with results of investigations of proposed methods for the standard benchmarks.

Chapter 8 deals with design of Moore FSMs based on using embedded memory blocks. The methods of trivial EMB-based implementation of logic circuits of Moore FSMs are discussed. In this case, only a single EMB is enough for implementing the logic circuit. Next, the optimization methods are discussed based on the structural decomposition leading to two-level models of FSMs. It is shown how to use the classes of PES for decreasing the number of EMBs in the final circuit. The last section considers different methods proposed for diminishing the hardware amount in LUTer implementing the block of replacement of logical conditions. It is shown that at least 17 different models can be used for optimizing the LUTer.

We hope that our book will be interesting and useful for students and Ph.D. students in the area of Computer Science, as well as for designers of modern digital systems. We think that proposed FSM models enlarge the class of models applied for implementation of control units with modern FPGA chips.

Zielona Góra
May 2015

Alexander Barkalov
Larysa Titarenko

http://www.springer.com/978-3-319-24200-2

Logic Synthesis for FPGA-Based Finite State Machines

Barkalov, A.; Titarenko, L.; Kolopienczyk, M.; Mielcarek, K.; Bazydlo, G.

2016, XIV, 280 p. 205 illus. in color., Hardcover
ISBN: 978-3-319-24200-2