Further Improvements in the Boolean Domain

Further Improvements in the Boolean Domain

Edited by

Bernd Steinbach

Cambridge
Scholars
Publishing

Further Improvements in the Boolean Domain

Edited by Bernd Steinbach

This book first published 2018

Cambridge Scholars Publishing

Lady Stephenson Library, Newcastle upon Tyne, NE6 2PA, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-5275-0371-2
ISBN (13): 978-1-5275-0371-7

Contents

List of Figures

List of Tables

Foreword

Further Improvements in the Boolean Domain contains some of the latest innovations with regard to the theory and application of algebraic methods over the Boolean domain. Algebras involving the Boolean domain have been studied and used by philosophers, scientists, mathematicians, and engineers since at least the time of Aristotle's development of the syllogism. In the past century, electrical and electronic artifacts that utilize switching elements have been extensively modeled with switching algebras or binary-valued algebras due to the advent of digital computation and communication. Although many theorists and practitioners have studied and used methods in the Boolean domain, new and useful results continue to emerge as the information age continues to evolve. This useful compilation of further improvements continues this tradition.

The book is organized into three parts titled: "Extensions in Theory and Computations", "Digital Circuits", and "Towards Future Technologies". These three parts are further divided into five separate chapters that provide results in areas ranging from theoretical concerns to those that are applicable to modern design and implementation challenges such as automated synthesis and reliability. Emerging computational paradigms based upon reversible functions and quantum mechanical phenomena continue to utilize frameworks in the Boolean domain, further underscoring the need for continued improvements in this area of discrete mathematics.

The first part of the book is devoted to theory and computation. Chapter One contains several new theoretical results including the relationship of Boolean equations to problems in the class NP. A recent area of interest is the study of the class of functions known as index generation functions. New theoretical characteristics are provided for these functions that have many useful applications in data networks and memory. Approximate computing encompasses the use of functions that are not precisely equivalent to those they approximate. The use of approximate functions can lead to significant efficiencies although a corresponding loss in precision accompanies their use and this topic is considered. Spectral methods have been the subject of both practical and theoretical concern for many years although new results continue to emerge and some of the latest results are provided in a survey of applications. Next, the topic of finite topologies is considered with the interesting approach of using a relational algebraic framework provided by the RelView computer algebra system. Chapter One concludes with a subsection devoted to the application of partially defined logic to the important and timely area of asynchronous circuit design.

The second Chapter of the book is concerned with accelerated computations. Performance continues to be a major concern and new results in the Boolean domain are applied to achieve performance enhancement. Bent functions are those that exhibit maximal nonlinearity and are known to have desirable characteristics when employed in certain classes of cryptographic algorithms. While bent functions are desirable to use in these circumstances, their enumeration and discovery remains a hard problem that motivates the development of new architectures for that purpose. An approach based upon FPGAs for the purpose of finding such functions is described and its effectiveness is analyzed. A second approach for generating bent functions combines a random method with GPU computational cores. Next, the subject of an arithmetic code known as the AN code is considered. AN codes are nonlinear and find their application in error detection at the hardware level. Once again, a GPU-based analysis and experimentation environment is described that allows for the computation of AN code distance distributions and SDC probabilities. The final contribution in Chapter Two considers the situation wherein associated forms of Boolean functions are often preferable to normal forms in terms of the literal count; however, the associated forms are not necessarily orthogonal. Ternary vector lists (TVLs) are presented and a means for using them to find orthogonal associated forms is provided and validated with experimental results.

The next part of the book is devoted to digital circuits and is comprised of Chapter Three which is concerned with synthesis, visualization, and benchmarks, and, Chapter Four which is concerned with reliability and linearity.

A fundamental operation in digital circuit synthesis is that of decomposition. A particular form of decomposition, namely vectorial bi-decomposition for lattices is described in detail in the first contribution of Chapter Three. The next contribution takes a somewhat philosophical view and considers the use of visualization as a tool in hardware/software design with both a survey of present methods and predictions about the future of this area and its corresponding potential impact. The subject of complemented circuits and their role in logic synthesis is described with emphasis placed upon the minimization problem and experimental results provided to validate the approach. A large percentage of digital circuit data-paths include arithmetic circuitry with the multiplier being a common element. An approach for the design of such multipliers based upon the use of the Fourier transform is described and example multiplier designs using both regular and saturated arithmetic are provided. The state assignment problem is considered next with respect to the criterion of minimizing power dissipation. A heuristic approach to the state assignment problem for low power is provided with an accompanying example to illustrate the method. Simulation is a basic need in digital circuit design and analysis and is often used in a stand-alone manner, or in support of other digital circuit engineering tasks. Discrete event modeling is considered and a syntax is provided based on both partially and totally specified propositions. The final contribution of Chapter Three is concerned with the use of benchmark circuits for the purpose of evaluating new approaches in digital circuit engineering tasks. A history and analysis of many common benchmark circuit collections is provided as well as an analysis of their performance characteristics when used in a variety of different digital circuit engineering tasks.

Chapter Four is also included in the digital circuits section of the book and is comprised of three contributions. The first contribution is concerned with security oriented codes that are referred to as low complexity high rate robust codes. The motivation for the use of these types of codes is to overcome the effects of adversaries that may be employing side channel or other types of attacks. The next section is aimed toward increasing reliability through decomposing a circuit into linear and non-linear portions. A degree of linearity is introduced whereby the measure can be used to guide a bi-decomposition of a candidate circuit. The final contribution of Chapter Four is concerned with partially specified functions and describes how such functions can be linearized.

The third and final part of the book is concerned with future technologies and is comprised of four contributions. The first contribution is concerned with reversible circuit synthesis via the use of functional decision diagrams (FDDs). Reversible circuit design is also considered in the second contribution; however this time a probabilistic approach in the form of an evolutionary algorithm is used. Although irreversible function classification has a rich history, the classification of reversible functions has not been studied to a similar depth. The next contribution is concerned with the classification of reversible circuits and provides several definitions and theorems. The final contribution moves from reversible logic into the more general realm of quantum operators and considers various decompositions for the $\mathtt{C}^n\mathtt{F}$ gate as derived from the $\mathtt{C}^n\mathtt{NOT}$ gate.

Mitchell A. Thornton

Southern Methodist University, Dallas, Texas, USA
June 2017

Preface

Digital systems significantly contribute to the progress in almost all areas of our life. Boolean variables and functions are used to describe such systems. These variables can only carry two different values: 0 and 1. This is the smallest possible number and contributes to both a high reliability and a simpler production in comparison to systems with a higher number of different basic values. This book presents further improvements regarding a large number of problems by 36 authors from the international Boolean domain research community. Basic versions of the contributions of this book have been published in the proceedings of the 12th International Workshop on Boolean Problems [320].

Improvements in the Boolean domain require both progress in theory and powerful tools which utilize the new theory. The first part of this book deals with methods, algorithms, and programs for these aims.

Solutions of many Boolean problems exponentially depend on the number of variables. Hence, we are faced in the Boolean domain with the most complex problems. In addition to the well-known CD-SAT-formulas which are restricted to conjunctions of disjunctions (clauses), the more compact CDC-SAT-formulas are introduced where the variables of the disjunctions are replaced by the conjunction of Boolean variables. Due to the improved power of SAT-solvers and the high performance of ternary vectors and further concepts implemented in the XBOOLE system, Boolean problems can be solved that have a complexity far beyond any human possibilities. Hence, it remains to find a proper description of the problem using Boolean variables, Boolean functions, and Boolean equations. Using many examples reaching from combinatorics on the chessboard, over several covering problems, to different graph coloring problems, the creation of models represented by Boolean equations as a unifying instrument have been demonstrated.

An index generation function is a function which maps a binary input pattern to a unique non-zero integer index value. Such a pattern may represent a virus to be detected or a packet to be routed. The number of Boolean variables needed to distinguish between all patterns determines the size and cost of the hardware needed to realize such a function. Assuming that the k patterns must be detected then m variables are needed, where $\lceil log_2\ k \rceil \leq m \leq k-1$. Using an experimental approach it has been found that the minimum number of variables needed in the realization of an index generation function can be expected to be closer to the lower bound than to the upper bound, especially when k is large. Hence, most index generation functions can be realized using inexpensive conventional memory. Furthermore, it has been found that balanced columns are of benefit to the search for minimum distinguishing sets, especially when k is small.

Significant improvements in terms of performance and energy efficiency can be achieved when instead of an exact implementation an approximate one is realized. Approximate computing is a technique that relaxes the requirement for an exact equivalence between the specification and the implementation of circuits. This approach can be used, e.g., when the limited perceptual capabilities of humans do not require an exact numerical computation. The quality of an approximation is measured using an error metric that compares the approximated function with the original one. Several error metrics are explored with the result that the synthesis for approximate computing with precise error bounds is a difficult task. The derived challenges have a need for strong methods in computing precise errors as well as heuristics methods.

Spectral techniques based on various spectral transforms in different algebraic structures provide the foundations for the approaches for classifying Boolean functions, detecting their hidden properties, or reducing the computation effort. A comprehensive review of the origins and evolution of spectral techniques provide the readers with a very useful basis for research in the areas of design of digital systems and signal processing. Many references to books or articles in journals support this research. This review indicates that both serious tasks and restricted resources are incitements for scientific progress and practical applications. It can be expected that spectral techniques furthermore contribute to improvements in the Boolean domain.

Topology is a fundamental branch of mathematics that explores the properties of mathematical structures. Basics of topology are geometry and set theory. The descriptive set theory that explores operator algebras, computability, mathematical logic, as well as harmonic analysis belong to the wide field of applications of topologies. Due to the focus on computational problems finite topologies are explored. It is shown how objects and concepts from finite topologies can be modeled using relations, how related tasks can be expressed using the language of relation algebra and how the RelView system can be used to compute and visualize solutions. The efficient implementation of this tool allows for experiments with very large topologies.

The clocked synchronization of digital systems ensures their deterministic behavior to the price of a certain inefficiency. Analog systems work efficiently but suffer under an ambiguous behavior. It is a challenge to couple the advantages of these types of systems to create efficient deterministic circuits. Key issues in this field of research are partially defined functions, their models, and utilization within the design process. A new formal methodology is suggested that warrants the match between the partially specified functions and real world asynchronous feedback structures. This dual-rail approach combines the benefits of both traditional types of systems and can even be used for safety critical systems.

Due to the exponentiation complexity of almost all Boolean problems efficient tools for their solution are needed. As an example of a very hard Boolean problem the computation of the number of bent functions has been selected. Bent functions are the most non-linear functions that can be used in cryptography to resist linear attacks. In a previous work an expensive reconfigurable computer was used to speed-up this calculation by about 60,000 times. The utilization of both a deeper knowledge about bent functions and the application of a circular pipeline on a much cheaper Field Programmable Gate Array (FPGA) result in an additional speed-up of more than two orders of magnitude.

The computation of bent functions is also the topic of other research. The key idea of this approach consists of the random generation of a function of a certain even number of variables and the check to see whether it is a bent function. Due to the very small fraction of bent functions the generation is executed in the Reed-Muller domain, where the search spaces for bent functions can be restricted be means of several theorems. The fast Reed-Muller transform has been used to compute the truth vector of a Boolean function. Utilizing the GPU an additional speedup of up to three orders of magnitude has been reached for bent functions of up to ten variables.

An important practical problem is the detection and correction of one or more bit flips (errors) in data words, for which data coding is typically exploited. There is always the risk that bit flips change valid code words into other valid code words, which prohibits both error detection and correction. In order to minimize this risk non-systematic, non-linear AN-codes are used. The letter A indicates an integer constant used to encode the data word N, which is usually also an integer number. Here, the error detection capability is influenced by both the parameter A and the data type of N. To estimate the risk of undetectable bit flips, we need to compute the distance distribution between the codewords for each possible value of A depending on the width k of the data words. This computation is a big challenge which has time complexity in the order of 4^k. Efficient multi-GPU implementations have been developed to solve this problem and determine preferable values of A.

The representation of a Boolean function as an orthogonal list of ternary vectors allows us to use such a TVL for both a disjunctive form and an antivalence form. The knowledge that each binary vector cannot be covered by more than one ternary vector of such an orthogonal TVL is an additional advantage. A special order in which the ternary vectors are selected from a TVL in disjunctive form to compute the needed orthogonal difference leads to a shorter number of ternary vectors in the resulting TVL. The preprocessing steps of absorption and sorting of the ternary vectors by increasing numbers of dashes additionally contribute to both a shorter time needed to compute an orthogonal TVL and the smaller number of ternary vectors in the result.

Improvements in the Boolean domain considerably affect the development and application of digital circuits. Due to the extensive use of such circuits in almost all areas of our daily life we immediately notice this progress. Digital circuits have been developed and applied over several decades. Hence, one could think all problems about them have already been solved. The second part of this book explores new insights in this field and confirms the continuous progress in appropriate research and applications.

Bi-decomposition is a powerful synthesis method for combinational circuits. This methods splits a given function into two simpler functions which control the inputs of an AND-, an OR, or an XOR-gate such that the given function appears on the output of this gate. The simplification of the decomposition functions is reached in the case of the strong bi-decomposition by a smaller number of variables that the decomposition functions are depending on. Strong and weak bi-decompositions are sufficient for a complete synthesis approach. The decomposition functions of recently suggested vectorial bi-decompositions are simpler than the given function because of the independence of the simultaneous change of several variables. The generalized theory of derivative operations for lattices of the second level has been utilized for vectorial bi-decompositions of such lattices and furthermore reduces the needed chip-area, power, and delay of the synthesized circuits.

An interesting analysis about the visualization in both the hardware and software domains come to astounding and alarming results. While software visualization is an active field of research that supports the software designer with many helpful visualization techniques, the hardware visualization is in a state of a “lost world”. Almost unchanged over several decades are graph views that are used to show how parts of the hardware are interconnected and waveform views visualize the signal changes over time. In the context of growing system designs, where both hardware and software contribute to the solution, innovative tools are needed for Hardware/Software Co-Visualization. The answer to the question “why” there is such a discrepancy between hardware and software visualization approaches can help to remove obstacles and encourage engineers and scientist to fill the recent gap in this field.

Traditional aims in circuit design consist of the synthesis of smaller and faster circuits for a given function. A new contribution to improve the reached limits of these aims is the synthesis of complemented circuits. This approach utilizes the differences in the space and delay needed that can exist between a circuit that realizes the given function and a circuit of the complement of this function. The benefits of the new complemented circuits result from the common use of both the function and its complement as well as the utilization of given and created don't cares. The theoretical basis of this approach is the utilization of Boolean relations which are explored for all ten operations depending on two-inputs. Comprehensive experimental results for both the exact and heuristic synthesis of more than 100 benchmark functions show that this new approach improves the known results of three-level circuits in many cases.

Next to addition, multiplication is a frequently used arithmetic operation in digital circuits. While several approaches of optimized adders are known, the possibilities to optimize multipliers are not as yet completely utilized. There are two types of multipliers. Assuming n bits for each of the two input values, in regular arithmetic the output of the multiplier contains $2n$ bits, but in saturation arithmetic the output is restricted to n bits. A comprehensive exploration of circuit structures of multipliers in both regular and saturation arithmetic leads to up to 33% faster circuits in comparison to the multipliers synthesized by the commercial tool Synopsys. The sources for this improvement are the use of a monolithic multiplier block of a size of around 4×4, the concatenation of fitted intermediate results, and a restricted tree of adders. Unfortunately, the reached speed up of monolithic multiplier blocks of a size of 4×4 or 5×5 requires about two to five times more area.

The power consumption becomes a more and more important limitation factor for very large scale integrated circuits. The thermal leakage power causes a temperature rise that constrains the circuit behavior and requires additional equipment for heat transmission away from the device. Low power consumption is also welcome for a long period of use of a mobile device until the next charge of the rechargeable battery. The power consumption is caused by the switching elements. One contribution to reduce the power consumption consists of the state assignment of an automaton such that a minimal number of switching elements must be changed for the needed transitions. A basic model and an heuristic algorithm for this task will be explained. This approach can be used for asynchronous finite state machines and leads to race-free circuits of low power consumption.

Digital systems are realized by logic gates and flip-flops. Many synthesis approaches are known to find a circuit structure of these building blocks for a given behavior. Transistors are the real switching elements used within the logic gates and flip-flops. A more fine granular modeling technique has been suggested that directly allows us to use transistors as basic building blocks of digital systems. The theoretical foundation is constituted on the definition of both partial and total operations of the implication and equivalence. This approach can be uniformly used on several levels of abstraction: the global behavior represented by a directed graph, the more concrete signal flow graph which can be seen as the most abstract structural view of a arbitrary circuit, the even more concrete signal flow plan that consists of modules of partially defined behaviors, down to the transaction level modeling.

The complexity of digital circuits requires the use of design automation tools. Consequently, new synthesis procedures are implemented in such tools to improve the structure of the designed circuits. The only way to compare the properties of several synthesis tools is the synthesis of a set of circuits based on the same descriptions of these benchmark circuits. This general approach has been used over several decades and different benchmark sets were published and used for this purpose. However, the circuit implementations have been changed over the years and influenced the creation of new benchmark sets. A prudent approach led to a comprehensive collection of benchmark circuits for logic synthesis and optimization. The benefit of this collection is a unique description of benchmarks of many sources presented in a cleaned, flattened form and split into connected components.

Secret information stored within a hardware system is the target of side-channel attacks such as the differential fault analysis. Such attacks try to inject faults into the system that alter the output. Knowing the injected faults and the associated output signals the wanted secret keys can be calculated. The faults can be injected within the communication channel. Security oriented codes for the transmitted data can be used to detect injected faults and prevent such side-channel attacks. The few known codes have drawbacks regarding the error masking probability as well as the cost of their implementation. New suggested low complexity, high rate robust codes are the shortened quadratic-sum, triple sum, and triple-quadratic-sum codes.

These codes close the mentioned gaps and are able to detect any error in the transmitted data with non-zero probability.

The decrease of circuit structures to a few nanometers has the benefit of both a reduced area and a reduced power consumption but unfortunately increases the appearance of faults caused by outer influences like cosmic radiations. Hence, fault tolerant techniques must be used to improve the reliability of digital circuitry. One of these techniques is the extension of the circuit by redundancy such that an error correction becomes realizable. Low density parity check codes can be used for this purpose and were successfully applied to improve the reliability of XOR-only logic network. This requires a split of the circuit into a linear and a non-linear part. Methods to synthesize circuits where the linear part is separated from the non-linear part are explored for adders of different sizes. Both the strong and the vectorial bi-decompositions contribute to this aim of synthesis.

Another application of a linear circuit is the transformation of incompletely specified Boolean functions of n variables into a Boolean space of m variables where $m < n$. Such a transformation is possible when the function values are specified only for k input patterns and the value of k is much smaller than 2^n. Applications of this task are, e.g., the design of on-line real-time control systems or built-in self-test equipment. The number of variables m of the target space should be as small as possible. An efficient method of finding a linear transform that is injective for the k specified input patterns will be explained. Using the knowledge of the coding theory and the theory of finite fields a lower bound has been found which strongly reduces the search space for such a transformation. This lower bound depends on both the number of variables n and the number of specified input patterns k. The provided results have interesting connections to linear error-correcting codes.

The third part of this book deals with problems that the Boolean domain will be faced with in the future. The continued reduction of the size of the switching elements brings us closer (and close) to the level of single atoms and quantum logic. Completely different physical laws must be considered in this field. The exploration of reversible circuits can be seen as a bridge between traditional circuit structures and circuits realized using future quantum technologies.