PREFACE

The present dissertation is based on some results of the investigations carried out by the author at IIT, Kharagpur during 1988-90 when he was on study leave from his parent organization, Bharat Heavy Electricals Limited, Bangalore, India.

The work presented in the thesis can be broadly divided into two areas, namely logic design using multiplexers (MUXs) and the testing of digital circuits. Although there is no apparent relation between the two areas, both the topics have been included here because a characterizing parameter of Boolean function called **Ratio Parameter** has been used in both the cases.

PART I. LOGIC DESIGN USING MULTIPLEXERS

With the increasing complexity of integrated circuit technology, the problems of designing integrated circuits are turning out to be more and more complex, which necessitates advanced tools to reduce the considerably growing costs and design efforts. This has motivated researchers to develop suitable algorithms for automated synthesis of digital circuits.

Automated synthesis of the logic circuits, optimized for speed and area, is one of the major challenges for computer aided design (CAD). Research works done over the past two decades have led to efficient methods for implementing combinational logic in optimal two-level form using programmable logic arrays (PLA's). However, in many situations PLA implementation does not give efficient realization with respect to speed and/or area [BRA-2]. In such situations, multilevel realization may be a viable alternative. In recent years, an increasing level of research has been pursued on multilevel logic synthesis.

But in the context of VLSI synthesis, there exist other parameters, which are also very important. It is known that in a typical VLSI chip 50 to 60 per cent of the chip area is used up for interconnection (routing). Therefore, regularity in the interconnection, which has significant influence on the routing area, is very important. Secondly, low fan-in and fan-out of the gates are impotant from the viewpoint of speed and area occupied by the MOS devices. Most of the failures in VLSI have been reported to be due to drive problems. Low fan-out reduces failure rate and improves reliability of VLSI circuits. Lastly, if a circuit is modular and comprises regular interconnection of one or few modules (cells), its design time will be low, and it will facilitate automated design. In view of these, regular structures, such as, PLAs, cellular networks and systolic arrays are attractive for VLSI realization.

In this context, the use of MUXs in the realization of digital circuits is another possibility. Long back MUXs have been recognised as universal logic modules [YAU-2] of digital circuits. Some works have also been reported on logic design with single multiplexer and network of multiplexers [TAB-1], [ALM-1], [MAN-1], [TOS-1], [PAL-1], [LLO-1], etc. But most of the existing algorithms are neither computer-oriented nor suitable for even a moderate number of input variables. As a result, these are unsuitable for present day IC technology. So, we feel that the true potential of multiplexers as a general purpose building block of digital circuits has not yet been fully explored.

In view of this, the work was carried out to develop suitable algorithms for automated synthesis of digital circuits by single MUX of minimum size and optimal or near optimal tree network of multiplexers. Moreover, the algorithms should be suitable for relatively large number of variables.

PART II. TESTING OF DIGITAL CIRCUITS

Increasing complexity of digital circuits is rendering their testing more and more difficult. Various techniques have been developed to cope with the problem. These techniques can be broadly classified into three categories; Deterministic, Probabilistic and Exhaustive. One major shortcoming of the deterministic method is prohibitively long and complex test generation overhead. The problem of probablistic method (random or pseudorandom) is its low fault coverage.

In exhaustive testing all 2^n input vectors are applied to the circuit under test (CUT). The advantages of this technique are extremely high fault coverage and easy test generation. However, on-chip storage of the correct output sequence requires too much memory (2^n). Hence, instead of checking the outputs after the application of each test vector, the output responses are compressed into a much smaller form called **signature** and at the end it is compared with the corresponding value of the fault-free circuit. This is called **compact testing**. The aim is to reduce the number of bits that must be examined to determine whether the circuit under test is faulty. But, some information is always lost by this process resulting in lower fault coverage.

The choice of a signature is influenced mainly by two factors: (1) the hardware overhead and (2) the loss of "effective fault coverage". A number of signatures have been proposed, namely parity [TZI-1], syndrome [SAV-4], transition count [HAY-5], etc. These approaches reduce the hardware overhead significantly for collecting the output responses. However, their fault coverage is not very high. This has motivated us to look for alternative signatures having higher fault coverage even at the expense of increased hardware overhead.

THESIS ORGANIZATION

The results of the study have been presented in two parts. The first part comprises seven chapters (Chaps. 1 to 7) concerned with the logic design with MUXS. Part two consists of four chapters (Chaps. 8 to 11) dealing with the testing of digital circuits.

Chapter 1 includes preliminaries and brief review of related work in the field of general logic design and multiplexer based logic design. The advantages of MUXs over commonly used gates for logic design have been discussed here. Also, ratio parameter, which is a characterizing parameter of Boolean functions has been defined. For an n-variable function f, the set of parameters

$$< N_1/D_1,\ N_2/D_2,\ \ldots,\ N_{n-1}/D_{n-1},\ N_n/D_n >$$

is called ratio parameter (RP) of f, where N_i and D_i are the number of 1's and 0's, respectively in the x_ith column of the minterm table containing true minterms of f.

Chapter 2 discusses logic design with single multiplexer. An algorithm has been developed for the realization of a Boolean function by a single MUX of minimum size based on ratio parameter. This algorithm is an extension of the algorithm proposed earlier by Pal [PAL - 1]. In this approach, ratio parameter has been conveniently used to formulate the necessary and sufficient conditions of MUX realizability of a given function. Algorithms for both completely specified and incompletely specified functions have been presented.

Chapter 3 is concerned with the combinational logic design with cascade network of MUXs. An iterative algorithm based on the ratio parameter has been developed to realize combinational circuits for completely specified functions using cascade network of 2-to-1 MUXs, if realizable. The algorithm has also been extended for incompletely specified functions [RK-1], [RK-7].

Chapter 4 considers logic design with tree network of 2-to-1 MUXs. This is a generalization of the algorithm used for the synthesis of cascade network. The function is iteratively expanded about suitable control variables such that optimal or near optimal tree network is obtained. A cascade network may be obtained as a special case of a tree network [RK-3], [RK-10]. This algorithm has also been extended for incompletely specified functions.

The main advantage of the above algorithms for single multiplexer, cascade and tree network of multiplexers is that no prior minimization of the Boolean function is necessary. In all the cases the ratio parameter has been used to reduce the search space to arrive at an optimal or near optimal solution. The algorithms are efficient compared to the existing methods and very suitable for machine computation. These algorithms have been implemented in a HP-9000 computer system using C-language and tested. However, memory requirement and run time increase with the number of variables because of the rapid increase in the number of minterms.

Chapter 5 discusses logic design with multiplexers from algebraic representation of a Boolean function. As the number of minterms increases with the number of variables, minterm based algorithms are not very suitable for functions of a large number of variables. In such cases, algorithms based on more concise representations, such as cube notation [BRA-1], are more appropriate. An algorithm has been developed to realize a function given in cube notation by a tree network of MUXs. Here, some heuristics have been used to reduce the search space [RK-11].

In **Chap. 6** an algorithm has been developed to realize the combinational part of a sequential circuit using MUXs. It has been assumed that the given sequential circuit is given in the form of a state table. A suitable scheme for state assignment has been developed to obtain an optimal/near optimal tree network of 2-to-1 mutiplexers.

Chapter 7 considers the testing of multiplexer network. The testability of a logic circuit realized by a network of 2-to-1 MUXs has been considered. It has been established that circuits realized with a tree network of 2-to-1 MUXs are easily testable. A most interesting feature is that no separate test generation nor any memory storage for the test set is required. No additional control inputs nor gates are to be added to make the MUX network testable. Another important aspect is that the test set is dependent on circuit topology. A simple testing scheme for test application has been proposed.

Chapter 8 presents a brief overview of related work on testing of digital circuits. It discusses fault diagnosis, design for testability and test response compression techniques for exhaustive testing.

Chapter 9 presents ratio parameter (RP) as a signature for exhaustive testing of combinational circuits. It has been found that when RP is used as signature, fault coverage is higher than any other existing signature. However, the hardware overhead involved in using it as signature is relatively high. Several alternative approaches have been suggested to make an RP-untestable circuit into an RP-testable one. The methods are based on simple modifications of the circuits [RK-8], [RK-5].

In **chapter 10** the scope of using cumulative RP as signature has been explored. The parameter

$$<(N_1+N_2+ \ldots +N_{n-1}+N_n)/(D_1+D_2+ \ldots +D_{n-1}+D_n)>$$

represented as $<N_c/D_c>$ is called cumulative RP (CRP) of the function f represented by CRP(f). This signature is derived from ratio parameter and needs less hardware for storing compared to ratio parameter testing. Also the fault coverage obtained using

CRP as signature is high compared to other existing techniques with comparable hardware requirement but it is less than that obtained using ratio parameter [RK-6], [RK-9].

Chapter 11 considers yet another signature called Input Transition Count. The input transition count (ITC) of an n-variable function is the number of logical transitions (from 0 to 1 and vice versa) for each variable in the input for all the true minterms, when the inputs are applied in increasing binary value. Thus

$$ITC\ (f) = < TC(x_1),\ TC(x_2),\ \ldots,\ TC(x_n) >$$

where x_1, x_2, ..., x_n are the input variables and $TC(x_i)$ is the transition count of the x_ith variable in the true minterms of the function. Many of the faults which are not covered by RP are covered by this signature and as a result its fault coverage is quite high compared to other existing techniques with comparable hardware requirement.