A New Variable Testability Measure

M. Jamoussi, B. Kaminska, D. Mukhedkar
Department of Electrical and Computer Engineering
Ecole Polytechnique de Montréal
P.O.Box 6079, Station A
Montréal, Canada (H3C 3A7)

Abstract- **In this paper, we propose a new Variable Testability Measure (VTM) for implementing testability at the high-level synthesis stage of the design process of integrated circuits. This new approach, based on binary decision diagrams, representing fully functional blocks of a circuit, and on their cyclomatic testability measures. It manipulates dataflow blocks to predict whether the circuit is testable and the vector set required to test it.**

1 Introduction

In recent years, the use of silicon compilation and other standard cell design tools has changed the way digital systems are designed. As a result, more and more systems are being designed at the functional level, with little gate-level design being explicitly performed. So, an appropriate measure can be developed which efficiently represents knowledge about functional-level testability.

This paper addresses an approach for implementing testability in the high-level synthesis process, and particularly at the functional-level stage. Then, a new Variable Testability Measure (VTM) is defined and used to evaluate the dataflow testability in an advanced step of the design process.

2 Variable Testability Measure

We are interested in the testability of integrated circuits as early as possible in their design process.

Usually, testability implementation is left until after the design is completed, which requires greater effort at later stages. Testability analysis tools, such as SCOAP [7], which are supposed to support testability incorporation during the design stages, actually provide poor predictions of testability and do not suggest how and what test methodologies should be applied.

2.1 High-Level Synthesis

Synthesis involves finding a structure that implements the behavior, the constraints and the goals of a given system. Generally, synthesis may be considered at various levels of abstraction because designs can be described at various levels of details. High-level synthesis [9] is the type of synthesis that begins at what is often called the algorithmic level.

It takes an abstract behavioral specification of a digital system and finds a register/transfer-level structure to realize the given behavior.

The system to be designed is usually represented at the algorithmic level by a programming language such as PASCAL [10] or ADA [6], or by a hardware description language similar to a programming one, such as VHDL [8]. The first step in high-level synthesis is usually the compilation of the formal language describing the system behavior into an internal representation.

The next two steps in synthesis transform core behavior into structure: scheduling and allocation. They are closely interrelated and interdependent. Scheduling consists in assigning the operations to so called control steps, fundamental sequencing units in synchronous systems and corresponds to a clock cycle. Allocation consists in assigning the operations to hardware. Finally, the design has to be converted into real hardware. Lower-level tools such as logic synthesis and layout synthesis complete the design.

The advantages of implementing testability as early as possible in the design process are a testable design, a reduced test cost and an earlier detection of intestable blocks. To incorporate testability in high-level synthesis as a constraint of the design specifications, a new concept of Variable Testability Measure is introduced providing information if the circuit is testable and permits a good prediction of the test-vectors set for a graph-representation of a circuit.

2.2 Variable Testability Measure

A new method for gathering the testability information at the dataflow-design stage called the Variable Testability Measure (VTM) is introduced. It permits an easy propagation of the information about functional-block testability and indicates that testability problems are very easily dealt with high-level synthesis.

Using a hierarchical abstraction principle, VTM will be able to provide two kinds of informations: first, whether or not each subcircuit and then the whole circuit are testable; second, the minimum number of test vectors required for a specific block of the circuit, and then for the whole circuit. This approach is based on the binary decision diagram [1] and the cyclomatic testability measure [2].

The Binary Decision Diagram (BDD) is a method for defining, analyzing, testing, and implementing large digital functions. It provides a complete implementation-free description of the functions involved. One of the areas in which these diagrams can be particularly useful is that of test generation, i.e. finding a set of inputs able to confirm that a given implementation performs correctly. Finally, BDDs may be directly interconnected to define still larger functions.

The Cyclomatic Testability Measure (CTM) is a method for predicting and determining a minimum set of test vectors for graphs of combinational and sequential circuits in the early conceptual stage of the design process. This approach is based on the cyclomatic number [3], and the BDD. If we note V the CTM of a given BDD called G, V is computed by the following equation:

$$V(G) = e - n + 3 \quad (1)$$

where e and n are the numbers of edges and nodes in the graph G respectively.

Finally, it was shown that the CTM, called V, of a circuit composed of p subcircuits and represented by a BDD called G, is computed by the following equation:

$$V(G) = V(G_1) + \sum_{i=2}^{p}(V(G_i) - 2) \quad (2)$$

where $V(G_1)$ is the CTM of the subcircuit containing the entry node represented by the BDD, called G_1. $V(G_i)$ corresponds to the i^{th} subcircuit represented by its BDD, called G_i, $(i = 2, ..., p)$.

2.3 Definition of VTM

The concept of VTM is a further development of the Cyclomatic Testability Measure. The VTM is defined as follows: Consider a functional block having its input and output variables given on n bits, such as an adder or a multiplier. VTM is a coefficient assigned to each bit of the input and output variables of this functional block. The VTM of a bit means the minimum number of test vectors required to test it. In this effect a variable given on n bits has n different VTMs, one for each bit.

The VTM permits treatment of functional blocks having their inputs and outputs given on various numbers of bits, while the CTM treats blocks with single outputs and inputs given on one bit each of them.

The advantage of the VTM is the easy composition of various blocks, which permit testability incorporation in the dataflow of the synthesis process.

3 Use of VTM in High-Level Synthesis

Throughout this paragraph, we try to use this new measure for some common functional primitives. Next, we will see how this new measure is involved in the evaluation of the testability of a circuit in its high-level synthesis stage.

3.1 Computation of VTM for some functional blocks

We try to determine the VTMs of the outputs of some functional-level circuit primitives such as comparators, adders, multipliers, logical operators , multiplexors which basically form the data flow of a circuit. Three cases are discussed in this section.

For the cases of functional primitives studied, we note A and B the inputs of these blocks, each one is given on n bits: $A_nA_{n-1}..A_i..A_2A_1$ and $B_nB_{n-1}..B_i..B_2B_1$ respectively. This notation means that A_n,A_{n-1},..,A_i,..,A_2,A_1 and B_n,B_{n-1},..,B_i,..,B_2,B_1 are the binary encoding of A and B respectively. We also note a_1,a_2,..,a_i,..,a_n and b_1,b_2,..,b_i,..,b_n the VTMs of A and B respectively.

3.2 A Comparator

Let us consider the case of a comparator of two variables A and B, given on n bits. The logical value of the output S is 1, while A is greater than B ($A > B$). If not, the value of S is 0.

In figure 1, we present the BDD describing this functional block. According to equations (1) and (2), the VTM of S, noteed s, is:

$$s = (4 * n + 3) - (a_n + b_1) + 2 * \sum_{i=1}^{n}(a_i + b_i + 4) \tag{3}$$

So the comparator of two variables, given on n bits each one, is tested with a minimum number of test vectors found by equation (3).

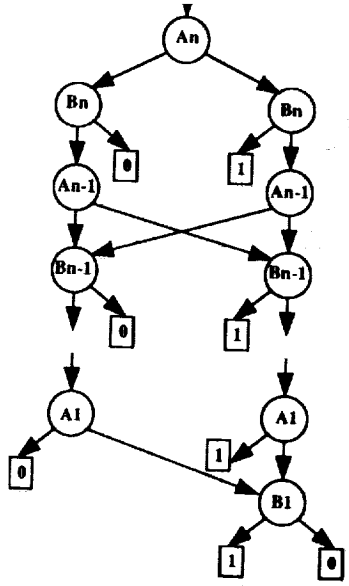

Figure 1: BDD of a Comparator

3.3 A Multiplexor

Let us study now the case of a multiplexor of two variables. The inputs A, B are given on n bits while the control signal C on one bit. The output S, which is $S_nS_{n-1}..S_i..S_2S_1$, is equal to A if C is true, or else it is equal to B.

The multiplexor can be considered in this case as a set of elementary multiplexors, where the i^{th} bit of the output is given as follows, ($i = 1, ..., n$):

$$S_i = C.A_i + \bar{C}.B_i \tag{4}$$

The CTM of the BDD describing S_i is equal to 4. Then, supposing a_i,b_i,c,s_i the VTMs of A_i,B_i,C,S_i respectively, according to equations (1) and (2), s_i is computed as follows, ($i = 1, ..., n$):

$$s_i = a_i + b_i + c + 2 \tag{5}$$

Finally, if A and B are primary inputs, which means $a_i = b_i = 2$, $(i = 1, ..., n)$, according to (5), s_i is given as follows:

$$s_i = c + 2 \tag{6}$$

3.4 An (n,n) Adder

Let us consider now, an adder of two variables A an B given on n bits, called an (n,n) adder. The sum S is given on $(n+1)$ bits and is $S_{n+1}S_n..S_i..S_2S_1$. Considering the elementary full-adder, we have:

$$S_i = A_i \bigoplus B_i \bigoplus R_{i-1} \tag{7}$$
$$R_i = M(A_i, B_i, R_{i-1}) \tag{8}$$

where S_i is the i^{th} bit of the sum and R_i the carry out of this addition. The CTMs of the BDDs describing S_i and R_i are 7 and 5 respectively. Then, if s_i, r_{i-1} and r_i are the VTMs of S_i, R_{i-1} and R_i respectively, according to equation (2) s_i and r_i are computed as follows, $(i = 2, ..., n-1, n)$:

$$s_i = 7 + (a_i - 2) + (b_i - 2) + (r_{i-1} - 2) \tag{9}$$
$$r_i = 5 + (a_i - 2) + (b_i - 2) + (r_{i-1} - 2) \tag{10}$$

In the case of the half-adder, we have:

$$S_1 = A_1 \bigoplus B_1 \tag{11}$$
$$R_1 = M(A_1, B_1, 0) \tag{12}$$

Then, supposing s_1 and r_1 the VTMs of S_1 and R_1 respectively, s_1 and r_1 are computed as follows:

$$s_1 = 4 + (a_1 - 2) + (b_1 - 2) \tag{13}$$
$$r_1 = 3 + (a_1 - 2) + (b_1 - 2) \tag{14}$$

Noting a_p and b_p the VTMs of A_p and B_p respectively $(p = 1, ..., i)$, the VTM s_i of S_i, the i^{th} bit of S, is:

$$s_i = \sum_{p=1}^{i} (a_p + b_p) + (2 - i) \tag{15}$$

OPERATOR	K (2 bits)	K (3 bits)
Addition	12	23
Subtraction	12	23
Comparator	7	11
Multiplexor	8	12
Multiplication	9	24
AND / OR	3	3

Table 1: Operator-Cost Coefficients

3.5 Objective function

It is common practice to use the cost function in the high-level synthesis, considering delay, area for testability constraints [5]. In this work, we propose a new objective function able to estimate and evaluate particularly the testability and the area constraints of a circuit. In a first stage, we propose to define a coefficient K for each functional primitive as the sum of its output VTMs, while it only has primary inputs. the value of K depends essentially on the bit number of the given functional-primitive variables.

In the case of logical operators or specific blocks where variables are given on one bit such as AND, OR gates, K is the output VTM. Table 1 gives the costs of some common functional blocks operating with variables given on 2 bits and 3 bits respectively. This coefficient will be, for a given functional primitive, its cost coefficient in the objective function introduced bellow:

3.6 Definition

Let us consider a circuit C composed of n-connected functional primitives. Given the i^{th} functional block ($i = 1, ...n$), let us assume:

m_i: the sum of the bit numbers of its outputs.
$a_{i,p}$, $(p = 1, ..., m_i)$: the VTMs of this block outputs.
K_i: the cost of this block.

The objective function is defined as follows:

$$f = \sum_{i=1}^{n} K_i * (\sum_{p=1}^{m_i} a_{i,p}) \tag{16}$$

The function given by equation (16) shows a trade-off between the circuit area (functional primitives used), and the number of test vectors or, in other words, the test time. One of our goals then, by using this new measure in high-level synthesis, can be expressed as a question of minimizing this objective function.

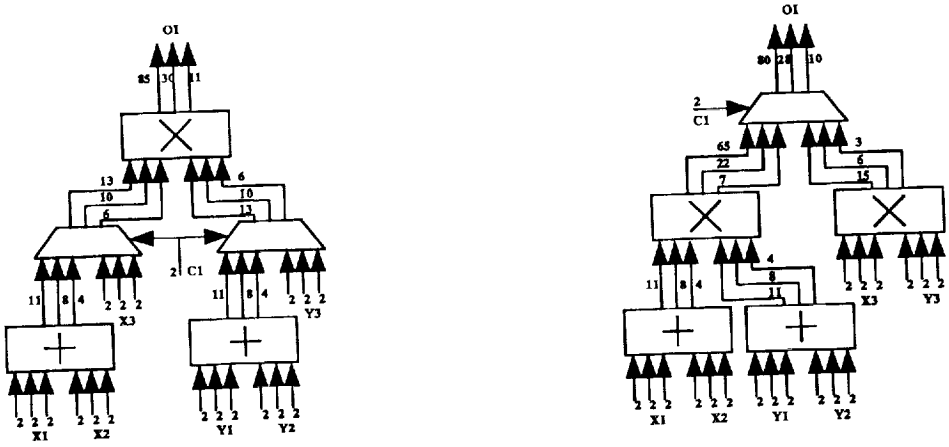

Figure 2: (a) Example circuit (b) Example circuit modified

The example presented in figure 2(a) is modified in 2(b) without changing the main task of the circuit. We notice lower VTM values against a greater silicon area due to a second multiplier used in figure 2(a). This transformation shows the trade-off discussed above.

4 Conclusion

In this paper we have proposed a new Variable Testability Measure (VTM) with the aim of incorporating testability at the functional-level stage, considered as an advanced step in the design process of a circuit. Our approach is essentially based on the Binary Decision Diagram (BDD) and the Cyclomatic Testability Measure (CTM). We have proposed an objective function to estimate circuit-testability cost.

References

[1] S. B. Akers, Binary Decision Diagram, *IEEE Transactions on Computers*, Vol. C-27, pp. 509-516, No.6, June 1978.

[2] B. Ayari & B. Kaminska, A Cyclomatic Testability Measure, *Proceedings of the Canadian Conference on VLSI*, Ottawa, pp. 8.2.1-8.2.10, 1990.

[3] C. Berge, Graphs and Hypergraphs, North-Holland, 1973.

[4] M. A. Breuer & T. H. Chen, Automatic Design for Testability Via Testability Measures, *IEEE Transactions on Computer-Aided Design*, Vol. CAD.4, pp. 3-11, January 1985.

[5] M. I. Elmasry & C. H. Gebotys, VLSI Design with Testability, *Design Automation Conference*, pp. 16-21, 1988.

[6] E. F. Girczyc, Automatic Generation of Microsequenced Data Paths to Realize ADA Circuit Description, PhD Thesis, Carleton University, July 1984.

[7] L. Goldstein & E. L. Thigpen, SCOAP: Sandia Controllability / Observability Analysis Program, *IEEE Design Automation Conference*, 1980.

[8] A. Lowenstein & G. Winter, VHDL 's Impact on Test, *IEEE Design & Test Computers*, V3 n2, pp. 48-53, April 1986.

[9] A. C. Parker, Tutorial on High-Level Synthesis, *IEEE Design Automation Conference*, pp. 330-336, 1988.

[10] Trickey & H. Flamel, A High-Level Hardware Compiler, *IEEE Transactions on Computer-Aided Design*, Vol. CAD-6,2 pp. 259-269, March 1987.