DOI 10.1007/s10598-019-09445-y

Computational Mathematics and Modeling, Vol. 30, No. 2, April, 2019

COMPUTING OBSERVABILITY OF GATES IN COMBINATIONAL LOGIC CIRCUITS BY BIT-PARALLEL SIMULATION

D. V. Telpukhov,[1] **V. V. Nadolenko,**[2] **and S. I. Gurov**[3]

UDC 004.052.32

The article considers vector computation methods (bit-parallel simulation) for determining the observability of combinational logic gates. The computations produce an ODC (observability don't care) set of all gates for a given set of circuit states. These results make it possible to evaluate the probability of logical masking of a random circuit fault. The methods are compared by accuracy and time costs using testing results for ISCAS '85 benchmark circuits.

Keywords: observability, bit-parallel simulation, combinational circuits, ODC.

Introduction

The observability of gates in a combinational circuit constitutes the strongest [1] and most difficult to estimate [2] error masking mechanism – logical masking. This mechanism is triggered when a fault occurs on a unobservable gate G, i.e., the signals on the primary circuit outputs are independent of the signal on the output of the gate G. In the general case, logical masking depends on the circuit state (the values of the signals on all its nodes). The state of a correctly operating combinational circuit in turn is uniquely determined by the set of input signals. Therefore, observability of the gate G (i.e., the probability that the logical masking mechanism is not triggered by an inversion fault on the given gate) is determined by the formula

$$P_{O_G} = \sum_i P_{X_i} * O_G^i, \tag{1}$$

where X_i is the i th input vector, O_G^i is the observability of the gate G with the given input vector (0 or 1), summation is over all input vectors.

Computing all gate observabilities from (1), we can then estimate the fault-tolerance of the entire circuit [3].

Bit-Parallel Simulation

In this article, we apply bit-parallel simulation – a Monte Carlo method that saves and simultaneously processes multiple circuit states [4]. Instead of a single logic value, each node in a chain is assigned the signature

$$signature_G = \left[f_G(X_1) f_G(X_2) \ldots f_G(X_N) \right], \tag{2}$$

[1] Head of the ICDM Department, Institute for Design Problems in Microelectronics (IPPM RAS), Moscow, Russia; e-mail: nofrost@inbox.ru.

[2] Institute for Design Problems in Microelectronics (IPPM RAS), Moscow, Russia; e-mail: vl777nd@list.ru.

[3] Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Moscow, Russia; e-mail: sgur@cs.msu.ru.

Translated from Prikladnaya Matematika i Informatika, No. 59, 2018, pp. 94–110.

where $X_1,\ldots,X_N$ are input patterns (vectors of signal values on the circuit inputs), f_G is the node logic function.

We first use the input vectors to specify the signatures of the primary circuit inputs. To this end, we form the matrix of column vectors $X_1,\ldots,X_N$ in which the rows are the signatures of the corresponding inputs:

$$\begin{array}{lccccc} & X_1 & X_2 & \cdots & X_N \\ signature_{in1} & x_{11} & x_{21} & \cdots & x_{N1} \\ signature_{in2} & x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ signature_{inK} & x_{1K} & x_{2K} & \cdots & x_{NK} \end{array} . \tag{3}$$

Then we traverse the circuit gates in topological order. The signature of a gate output is computed by bitwise application of its logic function to the input signatures. In this way, we perform full logical simulation of the circuit.

Then for each gate G and each input vector X_i we determine O_G^i (see (1)) and construct the vector of observabilities – the ODC (observability don't care) mask [5]:

$$ODC_G = \left[O_G^1 O_G^2 \ldots O_G^N \right]. \tag{4}$$

With uniformly distributed appearance probability of the input vectors $X_1,\ldots,X_N$, we obtain from (1)

$$P_{O_G} = \frac{w(ODC_G)}{\left|ODC_G\right|}, \tag{5}$$

where $w(ODC_G)$ is the Hamming weight of the Boolean vector ODC_G, i.e., the number of ones in this vector, $\left|ODC_G\right|$ is the length of the vector.

Evaluation of ODC Masks

We consider five methods for the evaluation of ODC masks: inversion error simulation [6], back propagation [5], fast error simulation [7], module approach [8, 9], and partial simulation.

The first method is the benchmark, because it evaluates observability from its basic definition. The second method is quite fast, but its error is highly sensitive to the circuit structure: it may reach 3%–5% in the presence of reconvergent paths or 10% and higher in the presence of functionally redundant parts in the circuit. The remaining methods combine fault simulation and back propagation in different ways. Fast simulation produces an exact result in less time than full simulation. The module approach and partial simulation strike a compromise between speed and accuracy.

In what follows, we describe each method in detail, highlighting its strengths and weaknesses.