TEST GENERATION FOR ESTIMATING POWER CONSUMPTION OF SEQUANTIAL CIRCUITS

Liudmila Cheremisinova

Abstract*: The reliability of electronic circuits is closely related to the power dissipated by them. Tools for evaluating the power consumption of sequential circuits is becoming a primal concern for designers of low-power circuits. The problem of estimation of the projected power, consumed by the CMOS sequential circuits, by means of its simulation is discussed. The task of forming test sequences of input actions to estimate the average circuit consumption is considered for the case when for the circuit automaton description in the form of Finite State Machine is available. The proposed method of test generation is based on special graph models of sequential circuits that allow formalizing the process of generating test sequences.*

Keywords*: power consumption, low-power design, test generation.*

ACM Classification Keywords*: B.6.1 Logic design: Design Style – Sequential circuits; B.7.3 Integrated Circuits: Reliability and Testing – Test Generation*

Introduction

Power consumption has become the major issue in electronic research since about 1990, it is being given increased weightage in comparison to area and speed. The minimization of power dissipation has become a task of critical concern with the advent of high density integrated circuits and portable micro-electronic devices. In the first case the heat generated by integrated circuits begins to exceed the ability of packaging to dissipate it, and in the second case new portable applications (such as notebook computers, cellular phones) require high speed, yet low power consumption. Low power ASIC design results in improved reliability and increased battery life. The Semiconductor Industry Association technology roadmap [SIA, 2014] has identified low power design techniques as a critical technological need in semiconductor industry today.

The development of methods and software tools that can help designers to optimize digital circuits for power consumption has received increasing attention. Accurate and efficient power estimation during design phase is required. The appropriate tools must have efficient means to estimate the power consumed by a circuit on different design phases. At present an increasing attention is focused not only

on transistor-level design but on higher levels of abstraction because early power estimation is important in VLSI circuits, because it has a significant impact on the reliability of the circuits under design. In the process of optimizing circuits for low power a designer is interested in knowing the effects of specific design techniques on the power consumption of the projected circuit. With the relevant information about power characteristics designer can redesign or correct a circuit in early design stages if it is stated that it can consume more power than expected.

Currently, the simplest and most direct power estimation can be done by circuit simulation when the monitoring of the power supply current is done [Benini, 2002]. Power consumption values are determined which depend on the given test sequence. So, using simulators, power is measured for a specific set of input patterns (often chosen randomly) referred to as test sequence. There are circuit-level power estimators available as commercial tools. For example, the most known SPICE [Nagel, 1973]. But the simulation results are highly related to the input patterns given to the circuit [Kang, 1986]. Simulation methods suffer from two major drawbacks. First, they are very time consuming, especially for large circuits (because to produce a meaningful power estimate the required number of simulated vectors is usually high). Second, it needs to know the set of input patterns when the power for a designed circuit embedded in a large system is to be calculated. Thus, the calculated power may be erroneous because some of input patterns used for estimation may never occur during normal circuit operation.

Using simulators, power is measured for a specific set of input vectors, and can be referred to average power consumption. Many investigations were focusing on the average power estimation [Arasu, 2013; Chou, 1996; Ghosh, 1992; Najm, 1994; Wang1, 1996]. The proposed methods are not only simulation-based but probabilistic methods are very popular too [Chou, 1996; Ghosh, 1992; Najm, 1994].

Power and switching activity estimation for sequential circuits is significantly more complex task than that for combinational circuits because power value depends not only on input test patterns but on the state the circuit is in. Although the problem of estimation of power in VLSI circuits is essential for determining the appropriate packaging and cooling techniques, optimizing the power and ground routing networks, there are a limited number of papers devoted to the problem of average and maximum power estimation of sequential circuits.

In the paper the task of average switching activity estimation for CMOS synchronous sequential circuit is considered when its automaton description in the form of Finite State Machine (FSM) is available. The methods of generating test sequences are based on finding out directed paths of special type for proposed graph models generated by FSM state transition graph (STG). The paths are closely related with input patterns for simulating the sequential circuit for power estimation.

In the proposed approach, we construct test sequences that, as we expect, allow obtaining the estimate of average power consumption and:

1) we are not interested in detail of the target circuit structure and use only its global structure – STG of FSM;

2) we expect the good correlation between switching frequency used for test sequence estimate and the actual switching frequency during normal mode of circuit operation;

3) the process technology is not taken into account.

Once the test sequences have been determined circuit-level simulation should be performed to determine accurately the associated values of average power consumption.

Basic definitions

Let any input pattern of the test sequence (looked for) be a Boolean vector from the Boolean space of dimension *n*, where *n* is the number of the circuit inputs.

The difference between combinational and sequential circuits is the memory elements issue. The states of sequential circuits cannot be assigned to arbitrary values but only to reachable ones. If the initial state is initialized to any arbitrary value during the power estimation, then the power value will be wrong since unreachable states are not allowed. The simulation-based power calculation procedure is comprised of three phases: generation (may be randomly) of a sequence of input patterns to be tested; simulation of the tested circuit on the sequence of input patterns estimating power dissipation on each clock cycle and then calculation of the average value of power dissipation. In the case of sequential circuit, the initial sequence of input patterns should start from some reachable state (it may be reset state). So, there are the following difficulties of usage of simulation-based method for the power estimation:

1) the need to generate such a sequence of input patterns that ensures energy normal mode of circuit operation (otherwise we do not get estimate of the average power consumption);

2) the simulation process is very time consuming because of the great number of simulated vectors for large circuits to produce a meaningful power estimate;

3) the necessity to initialize the tested sequential circuit, to start simulation from a reachable state;

4) baffling complexity of the task because a sequential circuit can be considered as a series of combinational circuits with different initial reachable states.

At the level of logic design a gate-level netlist is generated often from a FSM description, so a circuit operation is reflected by an appropriate FSM structure. We make an assumption that the sequential circuit automaton description in the form of FSM state transition graph (STG) is available. We seek for test sequence of input vectors that are the candidates to be tested for the average power dissipation in sequential circuit. It is proposed the test sequence is derived from augmented STG of the given FSM.

STG is a directed graph whose vertices correspond to the automaton states s_i, and its arcs – to the transitions between the states. Any arc of the graph between the states s_i and s_j is marked with an input pattern $\boldsymbol{x}_{ij}$ (a set of values of input variables $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$) which causes the corresponding transition from the state s_i into s_j. It is assumed that the automaton STG is strongly connected, i.e. for any pair of states always there exists a sequence of input pattern that brings the automaton from one state to another. Figure 1 shows an example of such a STG, each arc of the graph is labeled by a pair: the arc number / the input pattern $\boldsymbol{x}_{ij}$.

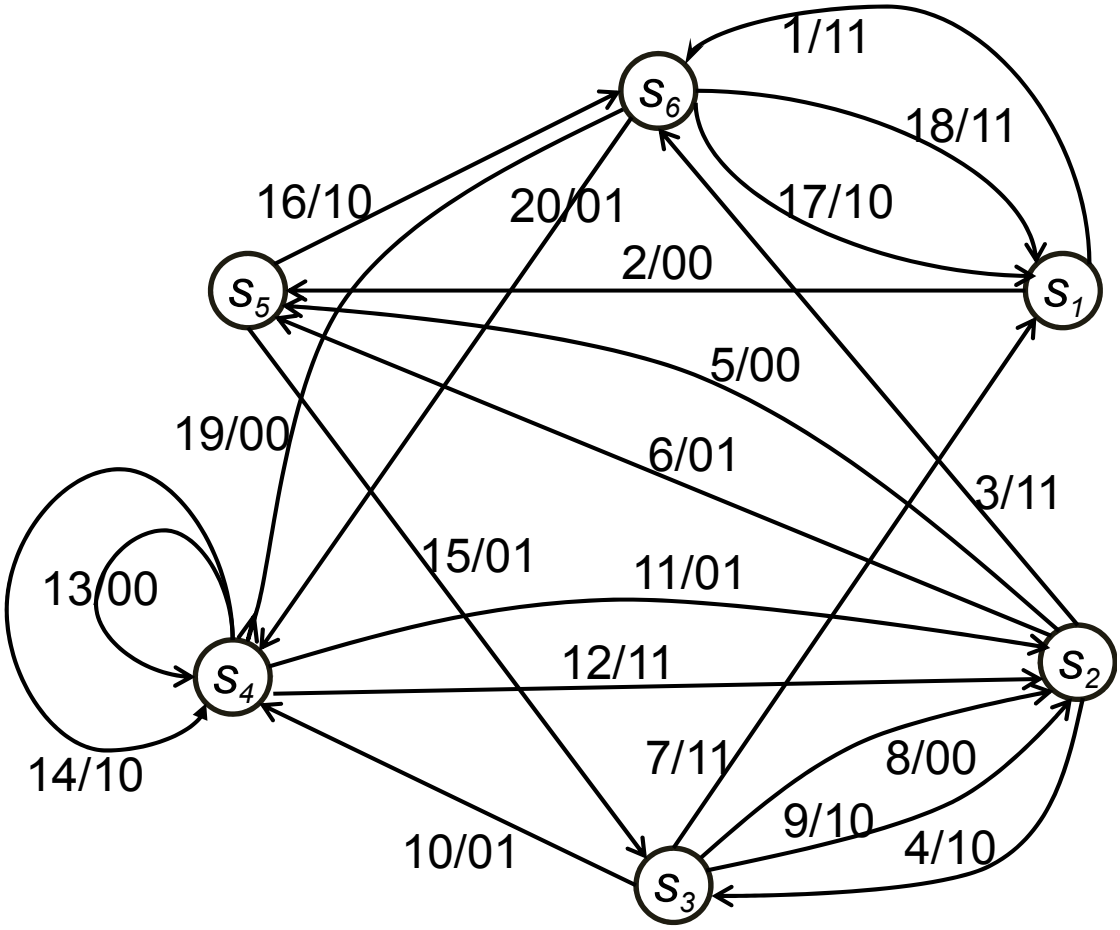

Figure 1. Automaton state transition graph

Let T_i denote test sequence in the form ($\boldsymbol{s}^i$, $\boldsymbol{x}_1^i$, $\boldsymbol{x}_2^i$,…), where $\boldsymbol{s}^i$ is a FSM internal state represented by a Boolean vector of memory element states, $\boldsymbol{x}_j^i$ is a Boolean vector of input variable values representing a FSM input pattern feeding circuit at the *j*-th clock cycle. The values of $\boldsymbol{s}^i$ and $\boldsymbol{x}_1^i$ initialize the circuit at the first clock cycle, before the process of estimating the series of switching's in the circuit. During the simulation of the circuit under the test T_i , the sequence ($\boldsymbol{s}^i$, $\boldsymbol{x}_1^i$, $\boldsymbol{s}_1^i$, $\boldsymbol{x}_2^i$, $\boldsymbol{s}_2^i$, …) of automaton states changes is generated.

Problem statement

For adequate estimation of power consumption a large number of input patterns should be considered to allow making statistically significant conclusions with this test sequence. And the test sequence should correspond to a normal mode of test circuit operation to provide calculating average power consumption. If the conditions of the circuit use are not known, the most effective toll will be pseudo-random test sequence of the exhaustive search, which must satisfy to following demands:

1) it includes every possible ordered pair of input patterns;

2) adjacent elements in the test sequence are presented exactly once.

The minimum size of such a test sequence is $2^n(2^n - 1) + 1$ (that follows from the algorithm of enumeration of oriented pairs in finite sets [Zakrevskij, 2010]). This estimate, however, can be achieved only for the case of combinational circuits. For the circuits with memory, this problem is much more complicated, since it is necessary to take into account changes in the states of the memory elements and their attainability during sequential circuit operation.

So, when simulating the circuit to estimate its average energy consumption it is desirable to analyze its responses to every possible change of its inputs. It is advisable to consider all possible ordered pairs of input combinations that are valid at the normal mode of the circuit functioning.

When testing sequential circuit, input patterns following one after another are separated by different states of memory elements. In other words, it is necessary to test not only pairs but various three element fragments ($\boldsymbol{s}^i$, $\boldsymbol{x}_1^i$, $\boldsymbol{x}_2^i$) which are allowable by the given automaton STG. For example, the arc 7 (Figure 1) corresponding to the automaton transition under input condition $x_1 x_2$ (corresponding to Boolean vector 1 1) is immediately preceded with the arcs 4 or 15 (corresponding to automaton transitions that move up the automaton into the state s_3), i.e. the test sequence should include three element fragments (s_2, x_1 $\bar{x}_2$, s_3, $x_1 x_2$) and (s_5, $\bar{x}_1 x_2$, s_3, $x_1 x_2$) corresponding to pairs of transitions 4, 7 and 15, 7.

The test sequence ($\boldsymbol{s}^i$, $\boldsymbol{x}_1^i$, $\boldsymbol{s}_1^i$, $\boldsymbol{x}_2^i$, $\boldsymbol{s}_2^i$, …) for calculating the energy consumption of a circuit implementing an automaton description should satisfy the following conditions:

1) tinternal state $\boldsymbol{s}^i$ should be reset state as any test scenario can run from the test circuit reset state;

2) test sequence should consist of alternating input patterns and automaton internal states, which are provided by a traversal of all the arcs of the STG (not necessarily once), starting with a given internal state $\boldsymbol{s}^i$;

3) test sequence should include every possible triple mentioned above.

The existence of such a sequence ($\boldsymbol{s}^i$, $\boldsymbol{x}_1{}^i$, $\boldsymbol{s}_1{}^i$, $\boldsymbol{x}_2{}^i$, $\boldsymbol{s}_2{}^i$, …) is provided by the assumption that an automaton STG is strongly connected. The question is how to find it in an efficient manner.

Graph models to search test sequence

The input data for generating the test sequence is a connected directed graph $G = (V, E)$ corresponding to the initial automaton STG. It is multidigraph that may have multiple arcs and sometimes loops.

It would seem, the task of forming the test sequence could be reduced to searching the shortest directed walk (as an open finite alternating sequence of vertices and arcs) that visits each arc in the directed graph G at least once.

The task can be stated as the Chinese postman problem for the case of digraphs: given a directed graph, find the shortest closed walk that visits all the arcs at least once. Indeed, the traversal of every arc of the multidigraph is the necessary condition. But since one of the distinctive features of an automaton model is that there is typically a large number of possible “next actions” at every vertex in the graph $G = (V, E)$ and we would like to test these combinations too. The Chinese Postman (and its variations) solutions guarantee visiting every arc, but not every arc combination of the length 2. The last demand is greatly difficult to perform solving the Chinese Postman problem on the graph $G = (V, E)$.

The idea of the proposed solution of the problem consists in the use of another automaton graph model, which allows easy to deal with combinations of the length 2 of arcs of the graph $G = (V, E)$. Such a graph model proves to be the line graph $L(G)$ of the graph G. The vertexes of the line graph $L(G)$ [Harary, 1969] correspond to the arcs of $G = (V, E)$ and $L(G)$ represents the adjacencies between arcs of G. If $G = (V, E)$ is a directed graph, its line graph is directed too. The line digraph $L(G)$ has one vertex for each arc of G. Two vertices of the line digraph $L(G)$, that correspond to the directed arcs from p to q and from u to v in $G = (V, E)$, are connected by an arc from $(p\ q)$ to $(u\ v)$ in the line digraph when $q = u$. That is, each arc in the line digraph $L(G)$ represents a length-two directed path in $G = (V, E)$ or a pair of automaton transitions.

The directed graph $G = (V, E)$ under consideration has 20 arcs, therefore the line graph $L(G)$ will have 20 vertexes too. The adjacency matrix $\boldsymbol{R}$ of the line graph $L(G)$ (Figure 1) is shown in Table 1. The

columns and rows of the adjacency matrix are associated with of the line graph vertexes labeled the same manner as the corresponding arcs of the graph G. The matrix element $r_i{}^j \in \boldsymbol{R}$ is equal to 1 if there exists an arc coming from the i-th vertex of the graph $L(G)$ to the j-th one (which corresponds to the presence of an ordered pair of arcs i and j in the graph $G = (V, E)$. The last column and row of the adjacency matrix indicate the values of out-degrees d^+ and in-degrees d^- of vertexes associated with the corresponding rows and columns of the matrix.

Table 1. The adjacency matrix ***R*** of the line graph $L(G)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	d^+
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	4
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	2
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	4
4	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	4
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	2
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	2
7	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
8	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
8	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
10	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	4
11	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
12	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
13	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	0	0	0	0	2
14	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	0	0	0	0	2
15	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	4
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	4
17	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
18	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
19	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	4
20	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	4
d^-	3	3	4	4	4	4	2	2	2	2	5	5	3	3	3	3	3	3	3	3	

Thus, if we find such a walk in the digraph $L(G)$ that passes through all its arcs at least once, we ensure that all three element fragments ($\boldsymbol{s}^i$, $\boldsymbol{x}_1{}^i$, $\boldsymbol{x}_2{}^i$) generated by STG of FSM are considered. So, the task of finding a test sequence to estimate the average power dissipation of the sequential circuit will be solved. The walk should begin with one of the arcs proceeding from the initial automaton state (reset

state) that is zero encoded. For our example, this arc may be the arc 1 in the graph $G = (V, E)$ and so, the start point should be vertex v_1 of the digraph $L(G)$. The desired test sequence $(s_1, \boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \boldsymbol{x}_3^t, \ldots)$ will consist of input patterns assigned to passable vertexes of the line graph $L(G)$ (and corresponding to the arcs of the digraph $G = (V, E)$.

Searching for the test sequence

As stated above the problem is to find the shortest walk in digraph $L(G)$, which passes through each digraph arc at least once. And the task is to find the shortest walk among all visiting each digraph arc. It is clear that the minimum walk length is achieved when each arc is traversed exactly once. Such a decision may take place in the special case when $L(G)$ is Euler graph (rare graph type). In other cases, the desired walk will contain repeated arcs. The goal is to minimize the number of repeated passages of the arcs. The solved problem is similar to the problem of the Chinese postman and its variation for digraphs – the New York Street Sweeper Problem [Bodin, 1983]. The difference between the mentioned problems and considered in the paper is that we do not need necessary to obtain a cycle and the weights of all arcs are equal in our case.

It is well known that Chinese postman problem (and its variations) is NP-hard. So, there appears a modest chance to find out such a solution for digraphs of high dimension. As it is easy to see, each vertex of degree k in the original graph $G = (V, E)$ creates $k(k\text{-}1)/2$ arcs in the line graph $L(G)$. This means that transforming a "thick" graph into a line one results in considerable increasing its complexity.

For the tasks of the practical dimension it is advisable to use approximate methods of constructing the shortest walks. We will form the desired path in the digraph $L(G)$ starting with some initial vertex v_k (by agreement, it is the v_1) of the graph and selecting at each step one of the arcs outgoing from of v_k that is reached at the step.

Before the start of the shortest walk in digraph $L(G)$ let make a copy $\boldsymbol{C}$ of the matrix $\boldsymbol{R}$ to have the possibility to keep record of arcs that are passed already and repass them only in necessary cases. Then let introduce the sequence D of vertexes that will constitute the walk under consideration. First D has the only initial vertex v_k (by agreement initially $k = 1$) then in the process of walk construction D will appended.

Step 1. The vector $\boldsymbol{s}_k$ is set to be equal the row $\boldsymbol{c}_k \in \boldsymbol{C}$ or, if $\boldsymbol{c}_k = \boldsymbol{0}$ ($\boldsymbol{0}$ vector), to the row $\boldsymbol{r}_k \in \boldsymbol{R}$.

Step 2. From the set of vertices corresponding to the units in the vector $\boldsymbol{s}_k$ (that succeed from the vertex v_k) such a vertex v_l is chosen that corresponds to the row $\boldsymbol{c}_l \in \boldsymbol{C}$ that has the largest weight (the number of units). But here the following cases can take place: A) There exist several such rows (they have the same weight), then for each of them the disjunction of rows $\boldsymbol{c}_i$ mentioned in it is formed

and the resulting vertex v_l will be chosen with accordance to weights of these disjunctions if they differ in weights. Otherwise we could form new disjunctions or simply choose the first of the rows $\boldsymbol{c}_l$ that generates one of disjunctions with the largest weight. B) All the rows $\boldsymbol{c}_l$ generated by $\boldsymbol{s}_k$ have the weight 0, then the rows $\boldsymbol{r}_l \in \boldsymbol{R}$ are considered that correspond to the units in the vector $\boldsymbol{s}_k$ and for them Step 2 is executed.

Step 3. After selecting the vertex v_l the element $c_k^l \in \boldsymbol{C}$ is reset to zero, the index number l is appended to the sequence D of passed vertices and the index number k is set to l. If the resulting matrix $\boldsymbol{C}$ has unit elements, the process of the walk forming continues with the Step 1. Otherwise, the desired shortest walk is found.

The vertices of the digraph $L(G)$, listed in the obtained sequence D, correspond to the numbers of arcs of the graph $G = (V, E)$. The input patterns assigned to the appropriate automaton transitions give the desired test sequence.

Let us pay attention to the considered directed graph $G = (V, E)$ and its line graph $L(G)$. Peeking at the values of out-degrees d^+ and in-degrees d^- of vertexes associated with the corresponding rows and columns of the adjacency matrix $\boldsymbol{R}$ of the line graph $L(G)$ we make conclusion that the graph $L(G)$ is not Euler's one (and not semiEuler). So, the walk will contain recurring arcs. In such a case, it is necessary to minimize the number of repetitions using the proposed procedure.

In the first step of the procedure the first row of the adjacency matrix $\boldsymbol{R}$ is considered. The corresponding vertex v_1 is adjacent to the vertexes of v_{17}, v_{18}, v_{19} and v_{20} (as follows from the first row of the matrix $\boldsymbol{R}$). Two vertices v_{19} and v_{20} (or rows of $\boldsymbol{R}$) of the great weight are equal. So, the first one may be chosen.

Repeating the procedure of constructing a walk to the end, we get the walk of the length of 85, while the digraph $L(G)$ contains 64 arcs only. Therefore, the obtained walk repasses through 21 arcs. A part of the obtained walk includes the following arcs:

$$D = \{1,19,11,3,20,12,4,8,3,19,12,5,15,9,6,16,17,1,20,11,4,10,13, \ldots\}.$$

Accordingly, the initial fragment of the corresponding obtained test sequence to estimate the average power consumption of considered sequential circuit begins with the state $\boldsymbol{s}_1 = 000$ of memory elements and consists of the following input patterns:

{11,00,01,11,01,11,10,00,11,00,11,00,01,10,01,10,01,11,01,01,10,01,00,…}

Conclusion

The task of estimating average power consumption for sequential circuits is simplified when its initial automaton description is known. In this case it is shown how test sequence of input patterns that ensures estimation of energy consumption in normal mode of circuit operation can be found out.

The suggested graph models are an excellent instrument to solve the problems of generating test sequences for power consumption of sequential circuits for which there exists an initial automaton description. The process of forming such a test sequence can be viewed as traversing walks through the line digraph of the circuit automaton model.

Acknowledgement

The paper is published with partial support by the project ITHEA XXI of the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

Bibliography

[Arasu, 2013] S.A.K. Arasu, F.E.Josy, N.Manibharathi, K.Rajasekaran. BPNN Based Power Estimation of Sequential Circuits. In: Intern. Journal of Advanced Research in Computer Science and Software Engineering, 2013, vol. 3, no. 11, pp. 256–260.

[Benini, 2002] L. Benini, G.De. Micheli. Logic Synthesis for Low Power. In: Logic Synthesis and Verification. Ed. S. Hassoun, T. Sasao and R.K. Brayton. Boston, Dardrecht, London: Kluwer Academic Publishers, 2002, pp. 197–223.

[Bodin, 1983] L. Bodin. A Model for Municipal Street Sweeping Operations. In: Discrete and System Models (Modules in Applied Mathematics, 1983. V. 3). 1983, pp. 76-111.

[Chou, 1996] T. Chou, K. Roy. Accurate Power Estimation of CMOS Sequential Circuits. In: IEEE Trans. VLSI Systems, vol. 4, no. 3, 1996, pp. 369–380.

[Ghosh, 1992] A. Ghosh, S. Devadas, K. Keutzer, J. White. Estimation of Average Switching Activity in Combinational and Sequential Circuits. In: 29th ACM / IEEE Design Automation Conference. Tech. Dig, June 1992, pp. 253–259.

[Harary, 1969] F. Harary. Graph theory. Addison-Wesley Publishing Company, 1969. 274 p.

[Kang, 1986] S.M. Kang. Accurate simulation of power dissipation in VLSI circuit. In: IEEE Journal of Solid-State Circuits, 1986, vol. 21, no. 5, pp. 889–891.

[Nagel, 1973] L.W. Nagel, D.O. Pederson. Spice (simulation program with integrated circuit emphasis). In: Technical Report UCB/ERL M382. EECS Department, Berkeley: University of California, April 1973.

[Najm, 1994] F.N. Najm. A survey of Power Estimation Techniques in VLSI Circuits. In: IEEE Trans. on VLSI, 1994, no. 12, pp. 446–455.

[SIA, 2014] Semiconductor Industry Association (SIA) ITRS Roadmap, available: http://www.sia-online.org/backgrounders_itrs.cfm.

[Zakrevskij, 2010] A.D. Zakrevskij. Minimization of enumeration of oriented pairs. In: Tanaevskie Chtenia: 4th Intern. Scientific Conf., Minsk, March 29 – 30, 2010, pp. 58–62.

Authors' Information

Liudmila Cheremisinova *– The United Institute of Informatics Problems of National Academy of Sciences of Belarus, principal researcher, Surganov str., 6, Minsk, 220012, Belarus; e-mail: cld@newman.bas-net.by*

Major Fields of Scientific Research: Discrete mathematics, Logic design automation