

UDC 004.3

SWITCHING ACTIVITY MINIMIZATION FOR XOR GATE DECOMPOSITION

P. BARSKAR, I. MURASHKO

*Department of information technologies Pavel Sukhoi State
Technical University of Gomel, Republic of Belarus*

Keywords: Low Power Design, Logic Gate Decomposition, Switching Activity, Power Estimation, VLSI.

Introduction

Currently, due to rapid progress in the field of semiconductor integrated circuit technology, in particular the transition to nanoelectronic technologies, new tasks arise for the logical synthesis of computing devices implemented on the basis of these technologies. One of such tasks is the development of methods for designing digital devices with low power consumption [1]. Actuality of this task is determined by the following main factors [2]–[4]:

- the presence of many applications (portable personal computers, mobile communications, digital audio and video equipment), which must combine high speed with low power consumption;
- the need to reduce the level of energy consumption in order to achieve the necessary duration of autonomous work;
- the need to reduce the power consumption to solve the problem of heat dissipation, as this determines the weight and size of the devices;
- the need to reduce power consumption in order to solve the problem of performing effective testing of digital devices (studies show that during testing, power consumption, and accordingly, dissipated power, can increase two to three times [5], [6]).

Main goal of this work is decomposition algorithm for low power XOR-based circuit's synthesis. The fundamental idea based on calculation all possible switches in the circuits. This allows you to take upper and lower bounds of switching activity for the circuit. It is main advantage of our work.

A model for analyzing of switching activity

To address the area of power estimation and optimization, we would revisit the basic power consumption and dissipation for CMOS circuit can be differenced as [5]:

$$P = 0,5V_{dd}^2 f_{clk} \cdot \sum_{i=1}^n C_i^L \cdot WSA_i, \quad (1)$$

where C_i^L is the physical capacitance at the output of the node; V_{dd} is the supply voltage; WSA (weighted switching activity) is the average number of output transitions per clock cycle; f_{clk} is the clock frequency; n is a number of nodes.

We will be used zero delay models where gate delays are assumed to zero. Also we assume that load capacitance of every nodes is equal [7]. We also assume that supply voltage and clock frequency is constant. Thus we must calculate only weighted switching activity for every node for estimation of power consumption circuits.

We focus on estimating switching activity in the combinational logic of the given circuits. In the zero delay model, all gates have zero delay and, therefore, they switch instantaneously. Using the zero delay model ignores glitches in the circuit and therefore, power dissipation due to glitches is taken into account [5].

Time diagram for glitch to don't change in the logical states of output (Fig. 1, a) and time diagram of the change in the logical states of its inputs and output (Fig. 1, b).

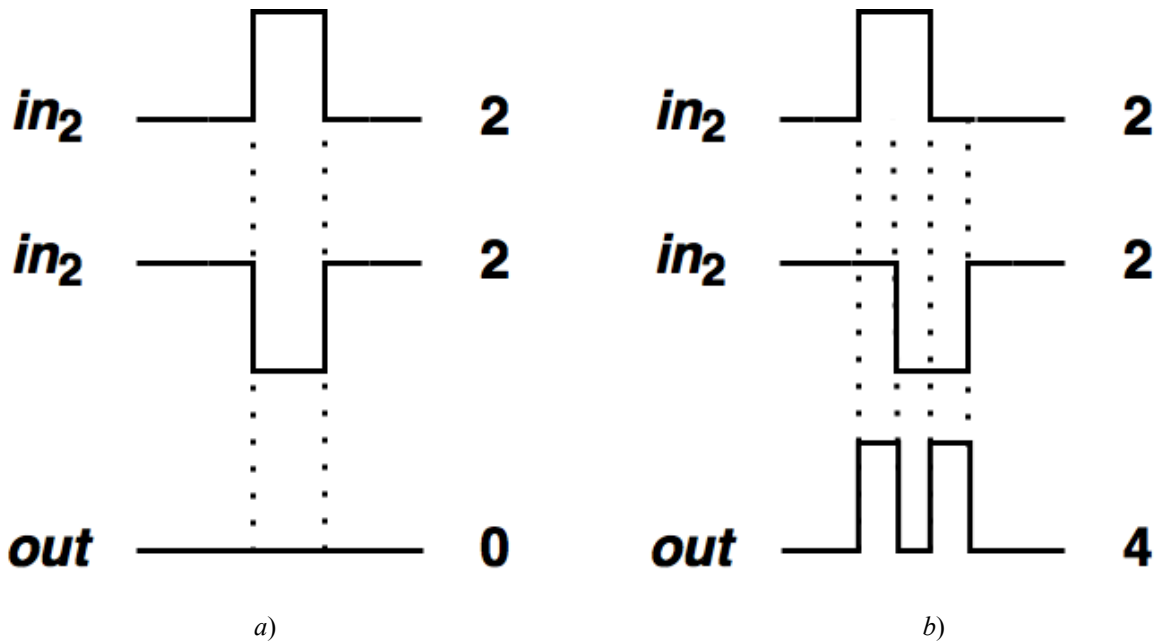


Fig. 1. Time diagram: a – for glitch to don't change in the logical states of output; b – change in the logical states of output

We are consider two-input XOR gate. Fig. 1, a shows the signal transitions at the primary input nodes of the network. The same time points are in normalized units. The signal of all inputs (in_1 and in_2) switches simultaneously at the same time point. But we obtain no switches at the output.

Fig. 1, b show case when switch was occurred at different time. Any input has 2 switches but output switching growing in 4. Thus all switches were transferred into output.

Problem for decomposition

Main task of technology decomposition is present multi-input XOR gate as network of XOR gates, with smallest number of inputs. Shown examples decomposition 7-input XOR gate with type of three-input gates into a network trees (Fig. 2). This circuit has the same hardware costs but different switching activity. Therefore they have different energy consumption during work.

Let's find switching activity both circuits. For simplify calculation let us assumed that WSA of all inputs is equal to 1. WSA of output is equal sum of WSA of all inputs. This value we will be considered only one times (we will use only WSA output). For examples (Fig. 2) output switching activity is equal 7. We find WSA circuits as sum WSA of all output:

$$WSA_{out} = \sum_{i=1}^k WSA_{out_i} \tag{2}$$

As has shown in [1], exist many different variants of decomposition multi-input XOR gate. This variant has equal number of inputs and equal number of gates but different switching activity (Fig. 2). Let us calculate total output switching activity using equation (2) for all network node (Fig. 2, a): $WSA_{out} = WSA_{out_1} + WSA_{out_2} + WSA_{out_3} = 3 + 3 + 7 = 13$. Also we made this calculation for Fig. 2, b: $WSA_{out} = 3 + 5 + 7 = 15$. Both circuits have equal num-

ber of gates. But first circuit (Fig. 2, a) has been respectively switching activity in 1,15 time's less than second circuit (Fig. 2, b). Thus there are problem finding such variant of decomposition which have minimal switching activity.

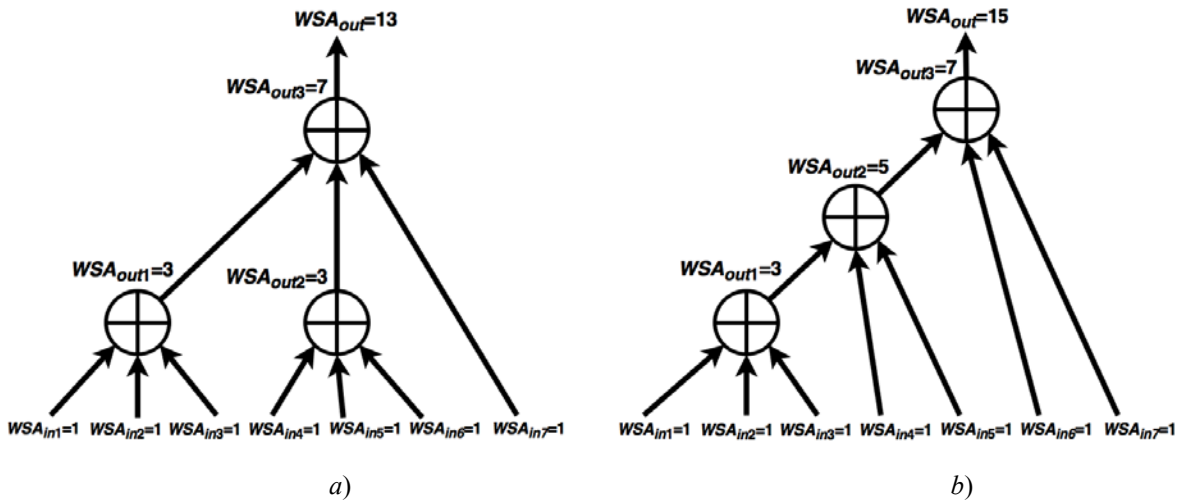


Fig. 2. Examples of 7-input XOR gate decomposition:
a – minimal switching activity; b – maximal switching activity

Estimation of minimum switching activity

In implementing the multi-input *XOR* gate with type of *d*-input *XOR* gates, the switching activity its will depend on the different type of circuit. We analyze the minimum values of switching activity (WSA_{\min}) of *n*-input *XOR* gate realized various trees on a *d*-input *XOR* gates (where all input logic signals are changed at different time) for the arbitrary number inputs. At the same time we defined switching activity of all inputs are same and equal to $x = 1$.

Let *n* is number of inputs multiple-input *XOR* gate, *d* is input type of *XOR* gate, *k* is number of *XOR* gates, which is needed for decomposition. Then *k* was found as:

$$k = \left\lceil \frac{n-1}{d-1} \right\rceil, \quad (3)$$

where $\lceil x \rceil$ – is nearest integer, greatest or equal *x*.

Now we consider example. Let $n = 7$, $d = 3$, then $k = \left\lceil \frac{7-1}{3-1} \right\rceil = \lceil 3 \rceil = 3$. Thus for decomposition 7-input *XOR* gate it is necessary to use 3 three-input gates.

Let us consider the derivation of the expressions for estimating the minimum switching activity. In [1] was proposed formula of minimum switching activity for $d = 2$. We find new formula for general case when *d* is equal any natural numbers ($d = 2, 3, 4, \dots$):

$$WSA_{\min} = n \lceil \log_d n \rceil + \left\lceil \frac{n - d^{\lceil \log_d n \rceil}}{d-1} \right\rceil. \quad (4)$$

Let us consider example for sixteen-input *XOR* gate ($n = 16$), which is implemented on three-input *XOR* gate ($d = 3$). Then total switching activity is:

$$WSA_{\min} = 16 \lceil \log_3 16 \rceil + \left\lceil \frac{16 - 3^{\lceil \log_3 16 \rceil}}{3-1} \right\rceil = 43.$$

Thus if we know number of inputs *XOR* gate (*n*) and number of inputs basic *XOR* gates (*d*) we can calculate *WSA* of decomposition tree very quickly.

Algorithm for synthesis XOR-based circuits with minimum switching activity

We can propose new algorithm for decomposition multi-input *XOR* gate.

Input: *n* – number of inputs of *XOR* gates; *d* – type of *XOR* gate (number of inputs basic *XOR* gate, which used for decomposition); *x* – *WSA* all inputs is the same and equal to 1.

Output: decomposition tree presented in the recursive list form: (*x*₁ + *x*₂ + ... + *x*_{*d*}), where *x*_{*i*} is input switching activity or children tree presented in same list form.

Step 0. Find number of gates (*k*) for decomposition tree, using (3).

Step 1. Find the output switching activity *WSA*_{out} = *nx*.

Step 2. Imagine the *WSA*_{out} as a sum of *n* numbers *WSA*_{out} = *n*₁ + *n*₂ + ... + *n*_{*d*} in such a way that the following restrictions are fulfilled:

– All numbers is divided by *x*; in form of natural numbers.

– We are using table to find input switching activity for child node in combination of *d* numbers. Sort this number in descending order (*n*₁ ≥ *n*₂ ≥ ... ≥ *n*_{*d*}). Then we search in this numbers value is equal to *x* = 1 (or *x* = 0) and fixing this node. If number not equal 1 we repeat steps 2 of the algorithm for this node.

Step 3. The algorithm's work ends when all inputs in the circuit (*n*₁, *n*₂, ... *n*_{*d*}) is equal to 0 or *x* = 1.

Finding switching activity for child nodes

| <i>WSA</i> _{out} | | | | | | | | | |
|---------------------------|--------------------------|----------------------|----------------------|--|--|---|----------------------|---|--------------------------|
| <i>D</i> | 2 | | 3 | | | 4 | | | |
| <i>N</i> | <i>i</i> + 1 | | 2 <i>i</i> + 1 | | | 3 <i>i</i> + 1 | | | |
| <i>i mod d</i> | 0 | 1 | 0 | 1 | 2 | 0 | 1 | 2 | 3 |
| <i>n_i</i> | [<i>n/d</i>] | [<i>n/d</i>] | [<i>n/d</i>] | [<i>n/d</i>]+1 | [<i>n/d</i>]+1 | [<i>n/d</i>] | [<i>n/d</i>] | [<i>n/d</i>] | [<i>n/d</i>] |
| <i>n₁</i> | <i>n_i</i> | <i>n_i</i> | <i>n_i</i> | <i>n_i</i> | <i>n_i</i> - 1 | <i>n_i</i> + 1 | <i>n_i</i> | <i>n_i</i> | <i>n_i</i> - 1 |
| <i>n₂</i> | <i>n_i</i> + 1 | <i>n_i</i> | <i>n_i</i> | <i>n_i</i> | <i>n_i</i> + 1 | <i>n_i</i> + 1 | <i>n_i</i> | <i>n_i</i> | <i>n_i</i> - 1 |
| <i>n₃</i> | – | – | <i>n_i</i> | <i>n</i> - <i>n₁</i> - <i>n₂</i> | <i>n</i> - <i>n₁</i> - <i>n₂</i> | <i>n_i</i> + 1 | <i>n_i</i> | <i>n_i</i> | <i>n_i</i> + 2 |
| <i>n₄</i> | – | – | – | – | – | <i>n</i> - <i>n₁</i> - <i>n₂</i> - <i>n₃</i> | <i>n_i</i> | <i>n</i> - <i>n₁</i> - <i>n₂</i> - <i>n₃</i> | <i>n_i</i> + 2 |

Note: *i* is natural number (*i* = 1, 2, 3, ...).

Illustration

Let us consider work of algorithm for case when *n* = 16, *d* = 3 and *x* = 1. Thus *x* = 1, we can't do check for divide in step 2.

Step 0. Find number of gates for decomposition tree, using (3):

$$k = \left\lceil \frac{16-1}{3-1} \right\rceil = \lceil 7,5 \rceil = 8. \text{ Thus we must use 8 three-input gates.}$$

Step 1. Find the switching activity of the output *WSA*_{out} = *nx* = 16 · 1 = 16 (Fig. 3).

Step 2. We know that *WSA*_{out} = *n*₁ + *n*₂ + *n*₃. Taking into account table to finding combination of this nodes input switching activity, we obtain *n*₁ = 8, *n*₂ = 5 and *n*₃ = 3 (node 1 in Fig. 3). The values of *n*₁, *n*₂ and *n*₃ is not equal to *x* = 1 (or *x* = 0), so for each of them we repeat steps 2. Current tree is (8 + 5 + 3).

Step 2. *WSA*_{out} = *n*₁ = 8. Imagine again *WSA*_{out} = *n*₁ + *n*₂ + *n*₃ = 2+3+3 (for node 2 in Fig. 3). The values of *n*₁, *n*₂ and *n*₃ is not equal to *x* = 1 (or *x* = 0), so for each of them we repeat steps 2. Current tree is ((3 + 3 + 2) + 5 + 3).

Step 2. *WSA*_{out} = *n*₁ = 3. Imagine again *WSA*_{out} = *n*₁ + *n*₂ + *n*₃ = 1 + 1 + 1 (for node 7 in Fig. 3). The values of *n*₁, *n*₂ and *n*₃ is not equal to *x* = 1 (or *x* = 0), so for each of them we repeat steps 2. Current tree is (((1 + 1 + 1) + 3 + 2) + 5 + 3).

Step 2. $WSA_{out} = n_1 = 3$. Imagine again $WSA_{out} = n_1 + n_2 + n_3 = 1 + 1 + 1$ (for node 6 in Fig. 3). The values of n_1, n_2 and n_3 is not equal to $x = 1$ (or $x = 0$), so for each of them we repeat steps 2. Current tree is $((1 + 1 + 1) + (1 + 1 + 1) + 2) + 5 + 3$.

Step 2. $WSA_{out} = n_1 = 2$. Imagine again $WSA_{out} = n_1 + n_2 + n_3 = 1 + 1 + 0$ (for node 5 in Fig. 3). The values of n_1, n_2 and n_3 is not equal to $x = 1$ (or $x = 0$), so for each of them we repeat steps 2. Current tree is $((1 + 1 + 1) + (1 + 1 + 1) + (1 + 1 + 0)) + 5 + 3$.

Step 2. $WSA_{out} = n_1 = 5$. Imagine again $WSA_{out} = n_1 + n_2 + n_3 = 3 + 1 + 1$ (for node 3 in Fig. 3). The values of n_1, n_2 and n_3 is not equal to $x = 1$ (or $x = 0$), so for each of them we repeat steps 2. Current tree is $((1 + 1 + 1) + (1 + 1 + 1) + (1 + 1 + 0)) + (3 + 1 + 1) + 3$.

Step 2. $WSA_{out} = n_1 = 3$. Imagine again $WSA_{out} = n_1 + n_2 + n_3 = 1 + 1 + 1$ (for node 8 in Fig. 3). The values of n_1, n_2 and n_3 is not equal to $x = 1$ (or $x = 0$), so for each of them we repeat steps 2. Current tree is $((1 + 1 + 1) + (1 + 1 + 1) + (1 + 1 + 0)) + ((1 + 1 + 1) + 1 + 1) + 3$.

Step 2. $WSA_{out} = n_1 = 3$. Imagine again $WSA_{out} = n_1 + n_2 + n_3 = 1 + 1 + 1$ (for node 4 in Fig. 3). The values of n_1, n_2 and n_3 is not equal to $x = 1$ (or $x = 0$), so for each of them we repeat steps 2. Current tree is $((1 + 1 + 1) + (1 + 1 + 1) + (1 + 1 + 0)) + ((1 + 1 + 1) + 1 + 1) + (1 + 1 + 1)$.

Step 3. The algorithm's work ends because all inputs in the circuit (n_1, n_2, \dots, n_{16}) is equal to 0 or $x = 1$.

Example for our work of algorithm is presented in Fig. 3.

This algorithm is allowed to find one variant to find of decomposition tree with minimum switching activity. Now we find total switching activity (WSA_{min}) of decomposition tree which presented on Fig. 3. We must summaries switching activity for all internal nodes and switching activity of output. We didn't count input switching activity. Thus total switching activity of decomposition tree is: $WSA_{min} = 16 + 8 + 5 + 3 + 2 + 3 + 3 + 3 = 43$.

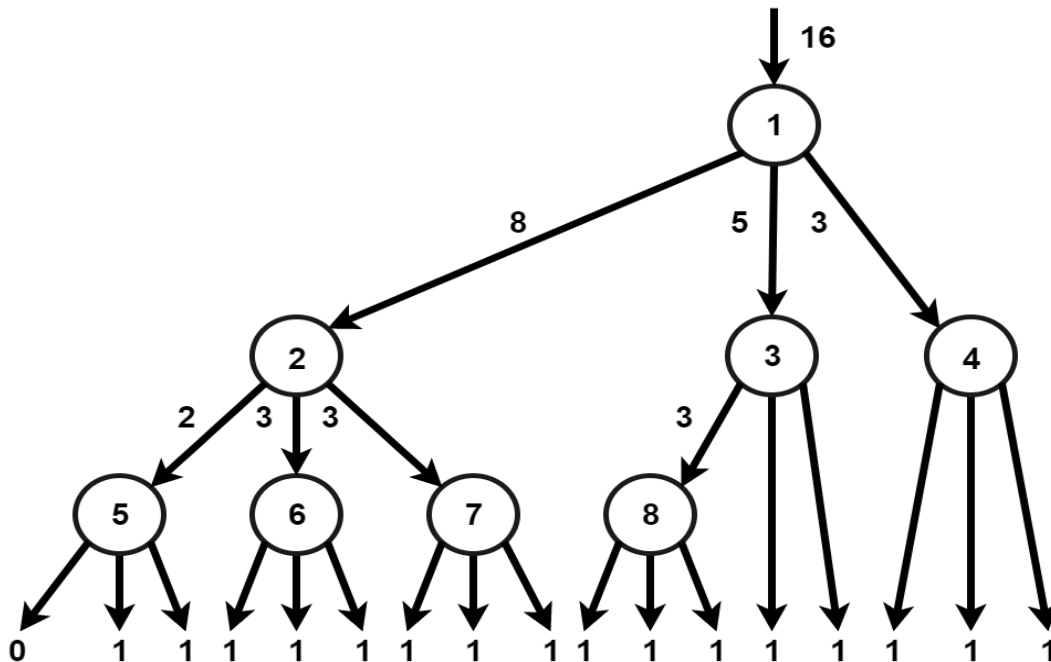


Fig. 3. Example of decomposition 16-input XOR gate on the type of three-input XOR gates with minimal switching activity

Analyze of maximal switching activity

Let us consider calculation maximum switching activity for any type of XOR-based circuits. We need maximum switching activity because we are compared to best case cir-

cuit switching activity to worst case switching activity. We find new formula for general case when d is equal any natural numbers ($d = 2, 3, 4, \dots$):

$$WSA_{\max} = \frac{x}{2} [2(n-1) + k(3-d) + k^2(d-1)]. \tag{5}$$

Let's consider same example for sixteen-input XOR gate ($n = 16$), which is implemented on three-input XOR gate ($d = 3$) and number of gates ($k = 8$). Then maximum switching activity is:

$$WSA_{\max} = \frac{1}{2} [2(16-1) + 8(3-3) + 8^2(3-1)] = 79.$$

In previous section we find minimum switching activity for 16-input XOR gate let us compare this value. As can be showed minimal switching activity is two times less then maximal switching activity.

Comparison results

Fig. 4 show the minimum and maximum switching activity (WSA_{\min} , WSA_{\max}) of n -input XOR gate with the type of d -input XOR gates. Input switching activity is equal 1.

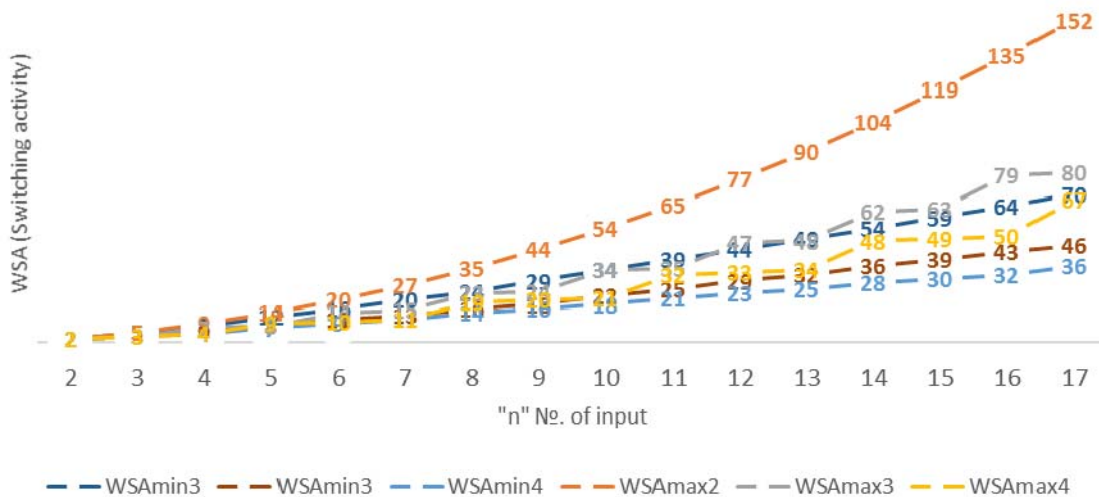


Fig. 4. Comparison between maximal & minimum switching activity

Analysis of the Fig. 4 show that with increasing n and d significantly increases the difference between the minimum and maximum switching activity. If $k \gg 3$ the maximum value increases in proportion to the number of inputs and the minimum switching activity increases as the logarithm of the number of inputs. In order to compare performance of switching activity as the proposed algorithm shows about 70 % average reduction for XOR gate power.

Conclusion

The paper analyzes the power of multi-input XOR gate with d type gates, which along with targeting minimum switching activity and low total power. Formulas for estimation minimum and maximum switching activity during decomposition of multi-input XOR gate was found. An algorithm decomposition of multi-input XOR gate was proposed. It is showing that when we increasing d we obtained decreasing WSA and as result we have decreasing of power. Proposed algorithm can be used in Computer-Aided Design of digital circuits.

References

1. Murashko, I. Power consumption analysis of XOR based circuits / I. Murashko // Informatics. – 2006. – № 1 (9). – P. 97–103 [In Russian].
2. Yeap, G. Practical Low Power Digital VLSI Design. / G. Yeap – Norwell : Kluwer Academic Publishers, 1998. – 212 p.
3. Roy, K. Low Power CMOS VLSI Circuit Design / K. Roy, S. C. Prasad. – NY : John Wiley and Sons, Inc., 2000. – 376 p.
4. Zorian, Y. A Distributed BIST Control Scheme for Complex VLSI Dissipation / Y. Zorian // Proceedings 11th IEEE VLSI Test Symposium (VTS'93), Princeton, NJ, Apr. 6–8, 1993. – IEEE Computer Society Press, 1993. – P. 4–9.
5. Ye, Y. Power Consumption in XOR-Based Circuits / Y. Ye, K. Roy, R. Drechsler // Proceedings of the 1999 Conference on Asia South Pacific Design Automation, Jan. 18–21, 1999. – Wanchai, Hong Kong, 1999. – P. 299–302.
6. Ye, Y. Graph-based Synthesis Algorithms for AND/XOR Networks / Y. Ye, K. Roy // Proceedings of the 34st Conference on Design Automation, Anaheim, California, USA, Anaheim Convention Center, June 9–13, 1997. – ACM Press, 1997. – P. 107–112.
7. Yarmolik, V. A peak-power estimation for digital circuits design / V. Yarmolik, I. Murashko // Fifth International Conference «New Information Technologies», Oct. 29–31, 2002. – Minsk : BSEU, 2002. – P. 34–38.
8. Giacomotto, C. Logic Style Comparison for Ultra Low Power Applications / C. Giacomotto, V. G. Oklobdžija // Techcon «Semiconductor Research Corp», Oct. 2005. – P. 181–190.
9. Lu, Y. H. Power-Aware Operating Systems for Interactive Systems / Y. H. Lu, L. Benini, G. De Micheli // IEEE transactions on very large scale integration (VLSI) systems. – Vol. 10, № 2. – Apr. 2002. – P. 119–134.
10. Iman, S. POSE: Power Optimization and Synthesis Environment / S. Iman, M. Pedram // Techcon «Logic Synthesis for Low Power VLSI Designs» 1998. – P. 199–224.
11. Zhou, H. Exact gate decomposition for low-power technology mapping / H. Zhou, D. F. Wong // Proceedings of the 1997 IEEE/ACM international conference on Computer aided design. – NY : IEEE Computer Society, 1997. – P. 575–580.
12. Kohavi, Z. Switching and finite automata theory / Z. Kohavi, N. K. Jha. – Cambridge : Cambridge University Press, 2010. – 631 p.
13. Lanik, J. On Switching Aware Synthesis for Combinational Circuits / J. Lanik, O. Maler // Haifa Verification Conference «Hardware and Software: Verification and Testing», 28 Nov. 2015. – P. 276–291.
14. Dinesh, B. Delay Minimisation in CMOS Combinational Arithmetic Circuits for Low Power / B. Dinesh, R. Jagadeesh, M. Kathirvelu // IEEE International Conference on Electronics and Communication System. – Coimbatore, India, 14 Feb. 2014. – P. 1–7.