LOW-POWER SYNTHESIS OF COMBINATIONAL CMOS CIRCUITS

Dmitry Cheremisinov, Liudmila Cheremisinova

***Abstract**: An approach to logic synthesis using CMOS element library is suggested, it allows to minimize the area and the average value of power consumption of microcircuit implemented on CMOS VLSI chip. The case of synthesis of combinational CMOS networks is considered when, for the purposes of energy estimation during the synthesis process, the static method based on probabilistic properties of input signals is used. The synthesis is comprised of the technology independent phase where logic minimization and decomposition are performed on the Boolean functions equations and the technology dependent phase where mapping to a physical cell library is performed.*

***Keywords**: power consumption, low-power synthesis, CMOS circuit.*

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

Introduction

In the VLSI (Very Large Scale Integration) chip design performance, area and cost were historically the major considerations. However, in the last years power consumption has become the major issue in electronic research, it is being given increased weight age in comparison to area and speed because of three main reasons: increasing use of portable and battery operated electronic devices which have limited battery life; continuous increase in chip density resulting in VLSI circuits that contain up to hundreds of millions of transistors and topicality of high performance computing resulting in VLSI circuits that have clock frequencies in the GigaHertz range. So, 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.

Static CMOS logic style is used for the vast majority of logic gates in digital integrated circuits because they have technological parameters and good power dissipation characteristics. Many ASIC methodologies allow only complementary CMOS circuits, custom designs use static CMOS for 95% of the logic [Zimmermann, 1997]. In CMOS based digital circuits, the most part of energy is dissipated during charging and discharging of node capacitances. To reduce the power dissipation, internal load

capacitances and switching activities of circuit gates must be lowered. Thus, at the stage of logic optimization the majority of the overall energy savings were achieved by minimizing the switching activities in the circuit. 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. And 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 found to consume more power than expected.

In the paper we consider a task of optimization of multilevel CMOS networks intended to obtain not only minimal area representation network but one optimized according to total gate switching activity, which helps in reducing the average power dissipation of the circuit. Techniques and program tools are suggested which should minimize the average power dissipation during technology-independent and technology-dependent phases of combinational logic synthesis.

Power dissipation in CMOS circuits

The power dissipation of CMOS circuits results from three parts: static, short-circuit and dynamic components. A small amount of current flow is termed the static component and is due to leakage currents (due to spurious currents in the non-conducting state) subthreshold currents. Short-circuit power happens briefly during switching and it usually accounts for 15%-20% of the overall power dissipation [Balasubramanian, 2007].

Dynamic power dissipation, also called as the switching power is related to a node capacitor which is charged and discharged. The dynamic component of power consumption normally dominates in CMOS system-on-chip and accounts for roughly 75% of the total power consumption. The dynamic power dissipation of a synchronous CMOS circuit with n gates is represented by the following approximation [Benini, 2002]:

$$P_{dyn} = \frac{1}{2} V_{dd}^2 f_{clk} \sum_{i=1}^{n} E_i C_i ,$$

where V_{dd} is the supply voltage, f_{clk} clock frequency, n the number of nodes in the circuit, C_i the node output capacitance, E_i estimates node switching activity and is called as the node transition density [Najm, 1994], that is the average number of logic transitions per a second. E_i is the transition probability at signal i, i.e., the probability that there is a $1 \rightarrow 0$ or a $0 \rightarrow 1$ transition on signal i from one clock cycle to the next.

At logic synthesis level the dynamic dissipation, that is the major source of power dissipation in static CMOS circuits, can be minimized by means of reducing switching activity (i.e. logic transitions from 0 to 1 or from 1 to 0 made by circuit nodes) in a designed logic circuit without changing its functionality. Taking out of context all the constants for used technology and capacitances that are unknown during logical synthesis we may estimate power consumption by the sum of values of switching activities of circuit nodes.

Logic synthesis transforms a circuit step by step, and each step optimizes with respect to the cost function. One transformation step (e.g., decomposition) changes only a small part of a circuit. The general power minimization strategy at logic level is to decrease $\sum_{i=1}^{n} E_i C_i$ at all stages of logical synthesis.

Switching activities will be parameters of the cost function, and after each optimization step, they must be re-estimated. Thus, an estimation of the switching activity should be accurate and fast. Accurate estimation is necessary to guide the optimization process in a proper manner. Fast estimation allows applying a large number of optimization steps and thus also contributes to the design quality.

The existing techniques for determining switching activity of nodes in a Boolean network [Najm, 1994] can be divided into two classes: statistical techniques (also called dynamic techniques) and probabilistic (or static) techniques. Statistical techniques simulate the circuit repeatedly until the power values converge to an average power, based on statistical measures. The methods are computationally intensive and are not feasible for iteratively updating switching activities as the network changes.

Probabilistic techniques propagate input statistics (probabilities) through the circuit to obtain the switching probability for each gate in the circuit. Though both the above techniques exist, the static techniques enable a quick approximate estimate of the power consumption of a digital integrated circuit at the logic level, without the need for extensive simulation.

Further in the process of logic synthesis we use the following two fast simple enough estimations of switching activity based on spatial independence assumption (circuit inputs and internal nodes are assumed to be independent). These techniques can introduce errors in the cases when input events are not independent. But empirical data shown that these errors are about hundredth parts of the calculated signal probabilities for nodes.

1) The first estimation technique is based on a zero delay model and so on assumption of signal's temporal independence assumption. For this case we are given the signal probabilities P_i (the average fraction of clock cycles in which the signal value is a logic 1) of signals on CMOS gates e_i outputs. The following equation is used to estimate the switching activity of g_i:

$$E_i = 2 P_i (1 - P_i). \quad (1)$$

If the input probabilities to a network are provided then they are propagated through to evaluate the probabilities at each node. For example, the output signal probabilities for $n(e)$-input AND and OR gates can be computed such a manner:

$$P_e^{\wedge} = \prod_{i=1}^{n(e)} P_i ; \quad P_e^{\vee} = 1 - \prod_{i=1}^{n(e)} (1 - P_i) ; \quad (2)$$

2) The second estimation technique applies a real delay model, so it computes power due to glitches too. In this case we are given transition densities A_i (the average number of signal transitions per a second) of signals. If a gate implements the function $f(x_1, x_2, \ldots x_n)$ and its inputs x_i are independent then the transition density of its output is computed as [Najm, 1994]:

$$A_f = \sum_{i=1}^{n} P(\frac{dy}{dx_i}) A_{xi} , \quad (3)$$

where the Boolean difference of the function y with respect to x_i is defined as

$$\frac{dy}{dx} = y(x = 1) \oplus y(x = 0) .$$

For a synchronous circuit with a clock period T, the relationship between transition density and switching activity is

$$A \geq \frac{E}{T} .$$

Low-power logic synthesis

In the process of logic synthesis an abstract form of desired circuit behavior (system of Boolean functions) is turned into a design implementation in terms of logic gates of CMOS cell library. The synthesis is comprised of two stages: the technology independent phase where logic minimization and decomposition is performed on the Boolean functions equations with no regard to physical properties and the technology dependent phase where mapping to a physical cell library is performed. The power dissipation of the mapped circuit is highly dependent on the structure obtained at the technology independent phase. Therefore, a power conscious design must consider power at all stages of the design process.

It is found that area minimized solutions has, most of the time, a lower power dissipation, and moreover, reducing power estimates (such as switching activity) in the synthesis process could increase the power dissipation of resulted hardware. This may be due to an increase in any of the other power components, besides switching power. Therefore, we use area criterion (for example, the number of literals) as the first one in all techniques and only then switching activity criterion.

At each synthesis phase, the problem is formulated as: given a system a set of Boolean equation (specifying two- or multi-level AND-OR circuit) and signal probabilities P_i for each input, find an implementation such that the estimates of the resulting circuit area and total gate switching activity are minimized.

Two-level minimization for low power

Two-level networks, or system of disjunctive normal forms (DNFs), are logic networks in which the primary inputs feed AND gates whose outputs are inputs to OR gates. We denote n network primary inputs as $x_1, x_2, \dots x_n$ (among which there are ones corresponding to input signals and their complements), m AND gates as $k_i = x_1 x_2 \dots x_{ni}$ and r OR gates as $f_i = k_1 \vee k_2 \vee \dots \vee k_{mi}$. If a gate implements the function $f(x_1, x_2, \dots x_n)$ and its inputs x_i are independent then the transition density of its output is computed as in the formula (2) where P_x is the equilibrium signal probability of a logic signal x defined as the average fraction of time that the signal is 1, $A(x_i)$ is the transition density of x_i that is supposed to be given before logic optimization.

For power, the switching statistics of the inputs and conjunctions must be taken into account. We will estimate the signal activity by transition density (2). In the case of the conjunction $k = x_1x_2\dots x_n$ the transition density is computed easy enough:

$$\frac{dy}{dx_i} = x_1\dots x_{i-1}\dots x_{i+1}\dots x_m ,$$

$$A_k = \sum_{i=1}^{n} P_{x1}\dots P x_{i-1} P_{xi+1}\dots P_{xn} A_{xi} .$$

Assume for the sake of simplicity that transition densities of inputs are the same as their signal probabilities:

$$A_{xi} = A_{\overline{xi}} = P_{xi} .$$

The proposed minimization techniques are extensions of known methods of Boolean function minimization by adding heuristics that turn the minimization process towards lowering the power dissipation in the sought CMOS-circuits. The results of computer experiments are given in [Cheremisinov, 2011], which allow to evaluate power driven minimization influence on power dissipation of resulting CMOS-circuits.

Power consumption of two-level network as whole is estimated by the sum of estimations for all terminals (AND gates outputs and primary inputs) they depend on transition densities (2) depending on signal probabilities. Thus, the proposed methods of minimization of system of Boolean functions try to find out only prime implicants with less switching activities to minimize the sum of estimations for all terminals of two-level network under construction.

The two-level minimization problem that is obtained by appending the don't care conditions to the function of a node is typically solved by a heuristic algorithms. The set of primes of a function can be partitioned into three classes: essential, partially redundant and totally redundant primes. The possible source of power savings during Boolean functions minimization concerns only partially redundant primes. The typical basic operations of any minimization algorithm are: cube reduction and expansion, searching for irredundant cover. So, they should be done such a way to lead potentially to power reduction. In the first, the cubes of the function being minimized are reduced as much as possible; in the second step they are expanded so that as many cubes as possible are covered by other cubes and can therefore be dropped. In the third and final step, an irredundant set of cubes that cover the function is extracted from the set of those that have survived the expansion phase. In expanding a cube, two objectives must be taken into account. The first of then is the quality of the cube taken in isolation. For power, the switching statistics of the inputs must be taken into account. In the case of power minimization, the value of an expansion depends not only on the number of cubes that should be covered, but also on the activity of the cubes that are eliminated (using (2)) because they are covered.

The maximal expansions of a single cube do not depend on the order in which the cubes of cover of network S are processed. However, the order in which the cubes are processed for expansion is important because of the effect it has on which cubes are covered and hence dropped: Expanding a cube too early may prevent another cube from covering it. In order to minimize power consumption, the switching activity of each reduced cube is computed and then they are processed according to increasing switching activities. Thus, the cubes with high switching activity are kept last, in the hope that some other cubes will expand to cover them.

Therefore, in expanding a cube, two objectives are taken into account: 1) to reduce the activity of resulting prime we try first of all to exclude the most active literals; 2) to cover the most active cubes

because they will be eliminated. To ensure the first demand we put literals in order of decreasing their switching activities to analyze for eliminating first the most active literals. To ensure the second demand we compute the switching activity of each cube to be expanded and then process them according to increasing switching activities. Thus, the cubes with high switching activity are kept last, in the hope that some of them will be covered (and hence dropped) when expanding other cubes. In reducing a cube not any but the least active literals are appended to it. When searching for irredundant cover we try to find a set of cubes with the least total cube switching activities.

The developed program set implements a set of methods that give suboptimal solutions of minimization task [Cheremisinov, 2011]. The program set allows varying:

1) the object of minimization: a completely or incompletely specified Boolean function; a system of completely or incompletely specified functions;

2) the method of minimization: interval competing method [Toropov, 2001], modified method ESPRESSO, iterative method, fast-acting one passing method [Cheremisinov, 2011];

3) individual or joint minimization of functions in a system;

4) minimization with taking into account the output polarity assignment, limiting runtime and others.

In Table 1 some experimental results showing the efficiency of the used techniques aimed at switching activity reduction. Here two versions of interval competing method of minimization was investigated; the initial method [Toropov, 2001] and its low-power modification. More details can be found in [Cheremisinov, 2011]. Here:

n, m, k are the number of inputs, outputs and conjunctions of the DNF system under processing;

kmin1, kmin2 and l1, l2 are the number of conjunctions and literals of the minimized DNF systems;

Ps1, Ps2 are switching activity estimations calculated as $\sum_{i=1}^{n} E_i C_i$ *of the minimized DNF systems;*

t1, t2 are computing times of two investigated methods.

When minimizing Boolean systems the following values of signal probabilities were used:

$p_1 = 0.10$; $p_2 = 0.13$; $p_3 = 0.16$; $p_4 = 0.19$; $p_5 = 0.22$; $p_6 = 0.25$; $p_7 = 0.28$; $p_8 = 0.31$; $p_9 = 0.34$; $p_{10} = 0.37$;

$p_{11} = 0.40$; $p_{12} = 0.43$; $p_{13} = 0.46$; $p_{14} = 0.49$; $p_{15} = 0.52$.

Multi-level minimization for low power

The paper focuses on the task of generating multi-level logical network that is best suited for the technology mapping process for CMOS gate arrays. The proposed methods [Cheremisinova, 2013] are targeted minimization of integrated microcircuit area implemented on CMOC chip and total value of gates switching activity. The methods take into account the specific features of the CMOS cell library during technology independent decomposition in terms of chosen primitive base functions (NOTs, ANDs, ORs). And in contrast to existing approaches it is proposed to decompose a Boolean network, to be mapped, into a *k*–bounded network where the number of fan–ins of each node is less than or equal to the greatest fan–in of the gates specified by the structure of the library cells. As the base functions in our case it might to choose elementary functions: NOT, *k*AND, *k*OR realized by component gates of primitive cells.

Table 1. The investigation results

DNF system	DNF system under minimization			Results of usual minimization				Results of low-power minimization			
	n	m	k	k_{min}^1	l^1	P_s^1	t^1	k_{min}^2	l^2	P_s^2	t^2
b12	15	9	431	45	286	61.8653	0.06	44	267	**61.8647**	0.06
in0	15	11	138	111	1348	**385.478**	0.01	111	1351	385.894	0.01
life	9	1	512	84	756	233.603	0.01	84	756	**230.48**	0.01
mlp4	8	8	256	160	1574	355.837	0.03	160	1577	**355.488**	0.03
root	8	5	256	57	430	**93.8343**	0.01	58	393	95.2714	0.01
tms	8	16	30	30	484	69.3707	0.00	30	484	**69.3705**	0.00
z9sym	9	1	420	90	630	**178.538**	0.01	101	707	197.323	0.04
ADDM4	9	8	512	249	2477	613.69	0.09	249	2485	**612.889**	0.09

Factorization technique is a key tool in facilitating multilevel synthesis. Finding a minimum factored expression is a cumbersome task. The proposed heuristic methods consist in the following two phases.

1) Nontrivial factorization of a system of Boolean expressions via extraction of common single- or multiple-term factors (subexpressions) for two or more Boolean expressions. Each time when factoring the system of Boolean expressions (conjunctions or disjunctions), the best factor is chosen according cost and power estimates.

The cost estimate computes the gain in "area": $T_s = c(s)(|U_s| - 1)$, where $c(s)$ is the size of the factor (number of literals) s and $|U_s|$ is the number of factored expressions. Nontrivial factorization supposes that only those factors are used for which we have $|U_s| > 1$ and $c(s)$ exceeds some predefined value.

The power gain of the factor s is proposed to estimate as:

$$P_s = \sum_{z_i \in s} E_{z_i} (|U_s| - 1),$$

where E_{z_i} is switching activity of a literal $z_i \in s$, computed according to (1) and (2).

As the result of the factorization a multilevel representation of a system of Boolean, an expression is found that has minimal number of literals but still it has some "long" conjunctions and disjunctions.

2. Factorization of each separate Boolean function independently via searching for common literals of its expression terms and then partitioning the rest "long" terms. The procedure is based on the following decomposition of a DNF D of every function: $D = k(A) + B$, where D, A, B are some DNFs and k is conjunction. The core of k is some "best" literal which enters in maximal number of conjunctions and which has the maximal switching activity. The last demand allows decreasing the load on the active node and promoting the active node to be as close as possible to the circuit output.

As the result of the factorization, a multilevel k–bounded network will be found.

Technology mapping

The resulting Boolean network consists of some non-overlapping multi-level k–bounded subnetworks with the only output that are then covered. These subnetworks consist of AND, OR, NOT gates. Each

library cell is decomposed into a superposition of the same functions. The next step is the technology mapping, it is performed by replacing subnetworks of the Boolean network with cell library instances.

The idea of the implemented method of technology mapping [Cheremisinova, 2010] is to cover the generated AND-OR network using elements of CMOS sell library. Alongside with the area criterion the energy consumption criterion is used too. The general idea of power saving is the known one – to hide high switching nodes within complex elements. So, the variants of covering are compared on the cost of cover (the ratio of the covered fragment cost according to Quine to the covering element cost) and the total switching activity of gates falling within the library element.

In Table 2 some experimental results showing the efficiency of the used synthesis techniques aimed at switching activity reduction. Here the minimization of DNF systems was fulfilled using the modified ESPRESSO method. The combinational circuits were implemented using elements of the native CMOS library. Two methods of implementation of technology independent synthesis phase have been used. The first of them is based on usual approach where a Boolean network, to be mapped, is decomposed using two input NANDs. And the second method takes into account the specific features of the used CMOS cell library constructs the circuit with factorization techniques mentioned above.

Table 2. *The experimental results of synthesis of combinational circuits*

DNF system	The of inputs,outputs conjunctions	DNF system minimization			Synthesis of multilevel circuit of CMOS library elements					
					The first method			The second method		
		p	*s*	*t*	*p*	*s*	*t*	*p*	*s*	*t*
br1	12, 8, 34	108	279	<1	141,34	668	3	54,11	386	6
br2	12, 8, 35	78,5	212	<1	115,04	526	2	48,29	348	4
in0	15, 11, 138	448,5	1118	<1	438,03	1956	2	229,2	1508	5
in2	19, 10, 137	596,5	1469	<1	451,38	2040	4	248,37	1726	5
mlp4	8, 8, 256	396	979	<1	324,68	1418	3	255,79	1406	6
root	8. 5, 256	150	393	<1	155,84	672	3	141,55	692	5
tms	8, 16, 30	108,5	471	<1	302,98	1276	3	139,1	916	5

z9sym	9, 1, 420	258,00	602	<1	160,31	678	2	136,72	702	5
GenP1	20, 4, 50	315,5	739	<1	548,69	2794	3	228,11	1374	7
GenP2	30, 10, 100	768	1733	1	1331,12	6790	3	631,87	4090	5
GenP3	30, 10, 300	2215,5	5052	14	3807,66	19502	4	1545,63	10418	11
GenP4	30, 8, 400	1798,5	4737	13	4232,16	21060	4	1583,5	9480	11
GenP5	20, 6, 400	1176,5	3029	15	1822,95	8526	4	945,59	5270	6
GenP6	20, 6, 400	1076,5	2948	15	1961,76	9016	5	945,32	5154	7
GenP7	30, 12, 50	244,5	558	<1	328,99	1632	2	252,52	1376	4
GenP9	30, 12, 400	1908,5	4667	7	3442,93	17268	5	1732,14	10366	10
GenP10	30, 12, 700	2314,0	5759	17	3238,65	15420	6	1776,18	9736	18
GenP13	30, 5, 600	2072,0	5492	19	3965,55	18948	8	1577,43	8762	44
GenP14	30, 5, 500	1710,5	4291	13	2618,67	12546	4	1343,04	7396	9
GenP15	30, 5, 400	2184,5	5213	7	3820,06	19416	4	1708,22	10628	12
GenP17	30,10,400	1610,5	4011	9	2785,49	13702	4	1530,39	8776	12
GenP22	24, 7, 790	2356	6223	19	4010,45	18762	10	1852,17	10588	18

Here, in Table 2 p and s are the switching activity and the area (estimated by the transistor number) of the circuit. t is program execution time in seconds on the computer with processor Intel i5-2400@3,1 GHz and 2,99 ГБ.

From the experimental results follows that taking into account the specific features of used CMOS cell library during technology independent decomposition allows synthesize better circuit variants. More details can be found in [Kirienko, 2015].

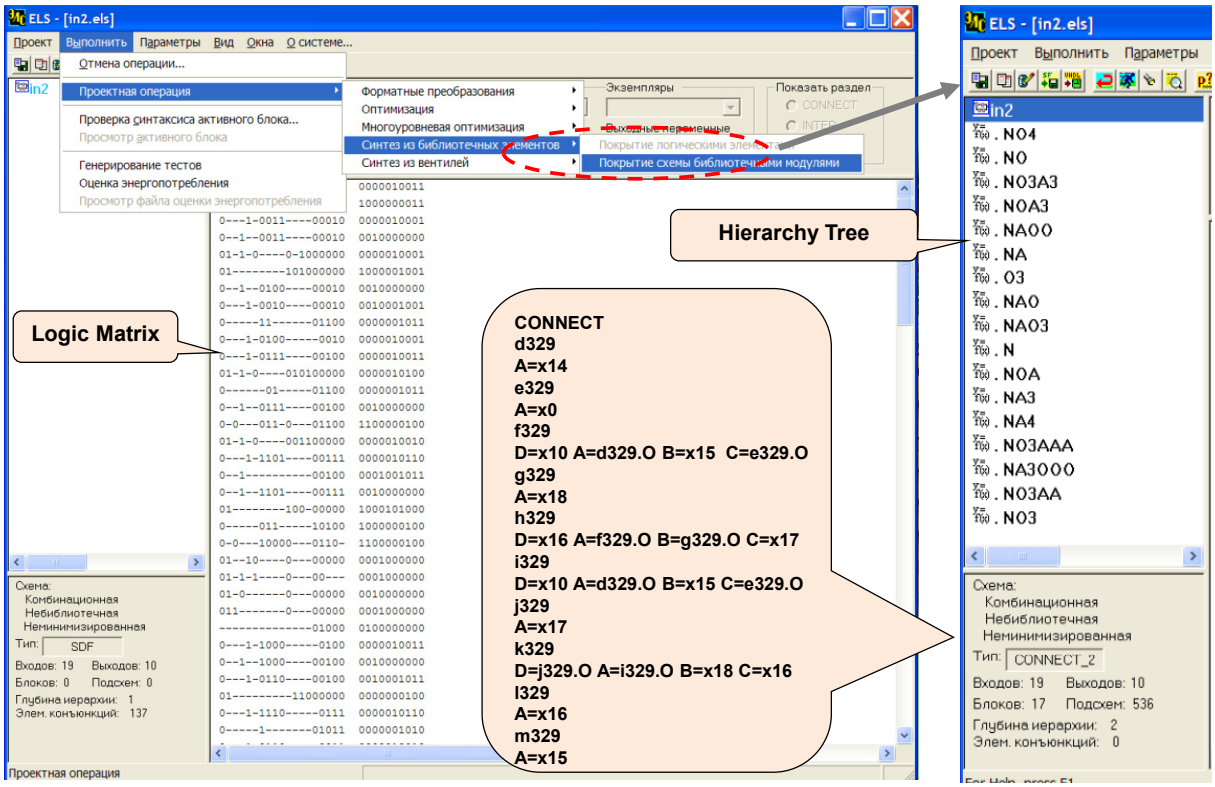

Figure 1. Technology mapping CAD system

Conclusion

The program implementations of the proposed methods are included as project operations in the software system for energy-saving logical synthesis [Bibilo, 2012] developed in the United Institute of Informatics Problems of NAS of Belarus. The system is intended for cell library design automation of custom very large-scale integration CMOS circuits. Figure 1 shows the typical window of the system in the case when multi-level *k*–bounded AND-OR network is covered with CMOS cell library instances. The estimations of complexity and power consumption are accepted as optimality criteria when designing CMOS circuits.

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

[Bibilo, 2012] P.N. Bibilo, L.D. Cheremisinova, S.N. Kardash, N.A. Kirienko, V.I. Romanov, D.I. Cheremisinov. Low-Power Logical Synthesis of CMOS Circuits Automation. In: Programnaia indzeneria, 2013, no 8, pp. 35–41 (in Russian).

[Balasubramanian, 2007] P. Balasubramanian, C. H. Narayanan, K. Anantha. Low Power Design of Digital Combinatorial Circuits with Complementary CMOS Logic. In: Intern. Journal of Electronics, Circuits and Systems, 2007, vol. 1, no 1, pp. 10–18.

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

[Cheremisinov, 2011] D.I. Cheremisinov, L.D. Cheremisinova. Low Power Driven Minimization of Two-Level CMOS Circuits. In: Informacionnie technologii, 2011, no 5, pp. 17–23 (in Russian).

[Cheremisinova, 2010] L.D. Cheremisinova. Low-power synthesis of combinational CMOC networks. In: Informatics (UIIP of NAS of Belarus), 2010, no 4, pp. 112–122 (in Russian).

[Cheremisinova, 2013] L.D. Cheremisinova, N.A.Kirienko. Low power driven synthesis of multi-level Logical circuits. In: Informacionnie technologii, 2013, no 3, pp. 8–14 (in Russian).

[Kirienko, 2015] N.A. Kirienko, D.I. Cheremisinov, L.D. Cheremisinova. Optimization of multi-level representations of logic circuits to reduce power consumption and VLSI chip area. In: Vesti of NAS of Belarus. Ser. physics-mathematics sciences, 2015, no 2, pp. 103–111 (in Russian).

[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.

[Toropov, 2001] N.R. Toropov. Minimization of a system of Boolean with output polarity assignment. In: Proc. of the 4th Intern. Conf. on Computer-Aided Design of Discrete Devices, Minsk, Rep. of Belarus, Nov.10–12, 2001, vol. 2, pp. 92–104 (in Russian).

[Zimmermann, 1997] R. Zimmermann, W. Fichtner. Low-power logic styles: CMOS versus pass-transistor logic. In: IEEE Journal of Solid-State Circuits, 1997, vol. 32, no. 7, pp. 1079–1090.

Authors' Information

Dmitry Cheremisinov *– The United Institute of Informatics Problems of National Academy of Sciences of Belarus, leading researcher, Surganov str., 6, Minsk, 220012, Belarus; e-mail: cher@newman.bas-net.by*

Major Fields of Scientific Research: Logic design automation, System programing

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