Designing of Hierarchical Structures for Binary Comparators on FPGA/SoC

Valery Salauyou, Marek Gruszewski

Designing of Hierarchical Structures for Binary Comparators on FPGA/SoC

Valery Salauyou, Marek Gruszewski

Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland

v.salauyou@pb.edu.pl, m.gruszewski@pb.edu.pl

Abstract. The article considers the general synthesis technique of hierarchical tree structures on FPGA/SoC for binary comparators. Designing of first level comparators is given. The best hierarchical comparator structure for the specific FPGA/SoC family is found empirically by experimental researches. The offered method allows reducing an area from 5.3% to 43.0%, and for high bitwidth comparators (with an input word length 1024) by 2.225 times. In the conclusion additional opportunities of the offered method are marked, and main directions of further researches are presented.

Keywords: binary comparator, synthesis, FPGA, SoC, hierarchical tree structure, Verilog language

1 Introduction

At present, there is a constant tendency to increase the length of words in computer systems. The length of words increases rapidly in telecommunication systems and data transmission and processing devices. On the other hand, integrated circuits of a programmable logic such as Complex Programmable Logic Devices (CPLDs), Field Programmable Gate Arrays (FPGAs), and System on Chips (SoCs) are widely used in building digital systems [1]. Digital systems as a rule include different standard functional blocks one of which is the comparator of binary numbers [2]. The binary comparator is one of the fundamental components in digital systems with many applications such as the decoding of the microprocessor instruction sets, the renaming of the register files in a superscalar system, and the number magnitude comparison in an arithmetic logic unit.

When high bitwidth comparators are constructed, it is sufficient to implement only function G "greater-than" and function E "equal", because function L "less-than" can always be determined on the basis of two first functions: $L = \overline{G} \,\&\, \overline{E}$.

The design problem of binary comparators is examined in many articles. In [3], the comparator is offered to build like an adder by using of a generate function and a propagate function which can be realized by the Manchester carry chain.

In some designs, a pipeline processing of signals and a mechanism of power off are used to increase performance and reduction of energy consumption. In [4], the comparator using two-phase clocking dynamic CMOS logic with modified non-inverting all-N-transistor block is presented. The compared output of two 64-bit binary numbers is performed in 3.5 clock cycles. In [5], the comparator is based on the priority-encoding (PE) algorithm. The circuit is realized with a latch-based two-stage pipelined structure. The comparator designed with the proposed techniques is 16% faster, 50% smaller, and 79% more power efficient as compared with the all-n-transistor comparator [4]. In [6], the prefix tree structure's area and power consumption can be improved by leveraging two-input multiplexers at each level and generate-propagate logic cells at the first level, which takes advantage of one's complement addition. In [7], a priority encoding (parallel-MSB-checking) algorithm along with a new priority encoder design and a MUX-based comparator structure is proposed; the method allows to increase performance by 22% in comparison with [5]. In [8], other architectures use a multiplexer-based structure to split a comparator into two comparator stages; the method allows to increase performance by 28% in comparison with [7]. In [9], to reduce the long delays suffered by bitwise ripple designs, an enhanced architecture incorporates an algorithm that uses no arithmetic operations. This scheme detects the larger operand by determining which operand possesses the leftmost 1 bit after pre-encoding, before supplying the operands to a bitwise competition logic (BCL) structure. The BCL structure partitions the operands into 8-b blocks and the result for each block is input into a multiplexer to determine the final comparison decision. In [10], the comparator project combines a tree structure with a two phase domino clocking structure for speed enhancement. In [11], binary comparator is based on a novel parallel-prefix algorithm. The proposed design shows an energy dissipation reduction of 23% and a speed improvement of 7%. In [12], the comparator exploits a novel scalable parallel prefix structure that leverages the comparison outcome of the most significant bit, proceeding bitwise toward the least significant bit only when the compared bits are equal. This method reduces dynamic power dissipation by eliminating unnecessary transitions in a parallel prefix structure.

All of the aforementioned works achieve high-performance operations using dynamic logic. While dynamic logic has demonstrated superior performance, as compared with static logic, it is not suitable for low-power operation. On the other hand, using of static logic allows reducing power consumption significantly. In [13], a new tree structure comparator with a pre-encoding scheme is proposed; one is particularly suitable for implementation on static logic to ensure low-power consumption. In [14], some modifications have been done in binary comparator [13] design to improve the speed of the circuit. In [15], a single-cycle tree-based binary comparator with constant-delay (CD) logic is presented. The proposed comparator with CD logic is 20% faster or 17% more energy-efficient compared to a comparator implemented with just the static logic.

In [16], the design of digital comparator with two different parallel architectures is proposed. These comparators are realized in Verilog and simulated with Xilinx ISE 8.2i platform. Simulation results show that the first proposed architecture has 23.769

% less combinational delay and the second proposed architecture has a combinational delay of 35.218 % less compared to the traditional design.

Reviewing of known methods of comparator designing showed that the majority of the methods (except for [16]) are intended to implement of the comparators on application specific integrated circuits (ASICs) and are not suitable for designing the comparators on the programmable logic.

This paper presents a universal method for designing of hierarchical tree structures of comparators on the programmable logic. Universality of the offered method is that the design of the comparator is described entirely in Verilog language; it is applicable to all classes of the programmable logic: CPLD, FPGS and SoC; it is also applicable to all CMOS process. The offered method allows changing the number of logical levels in a tree structure of the comparator in a broad range, as a result the user can select a trade-off of cost against performance. The comparator diagram is completely combinatorial circuit which does not contain clock signals therefore it does not require additional circuits for generating clock signals (unlike the considered known methods). Besides, in the offered approach there are no bitwise carries, and parallelism of the diagram provides high speed.

Section 2 executes the review of related researches. Section 3 describes the general synthesis technique the hierarchical structures for the binary comparators. Section 4 considers synthesis the comparators of the first level. Section 5 discusses experimental results. Conclusions presents additional opportunities of the offered method and also represent the main directions of further researches.

2 Related Research

This work is a continuation of researches to find the effective design methods for binary comparators based on a programmable logic [17,18,19,20]. In [17], the following methods of the comparator synthesis are considered: parallel, sequential, parallel-sequential, and with adder using. For each method two ways of implementation are offered: graphic and in the AHDL (Altera Hardware Description Language) language. The experimental researches are performed by the Altera MAX+PLUS II platform for 64-bit comparators and the results are compared to the Altera parametrized function lpm_compare.

In [18], the method of the comparator design in the form of a hierarchical structure is offered. The experimental researches are executed by Altera MAX+PLUS II platform. The offered method is compared to the sequential and parallel methods [17], and also to the Altera parametrized function lpm_compare, and to the method that implemented in the AHDL language compiler. At the first level of the hierarchical structure the 4-bit comparators are used. These comparators are built by four methods: the lpm_compare function, the AHDL language, and the parallel and sequential method [17].

In [19], the hierarchical structures of 64-bit comparators were researched by Quartus II 13.1 platform for Altera CPLDs and FPGAs. The 4-bit comparators formed a first level of the hierarchical structure. Two methods for building the first level

comparators are used: Verilog language and pm_compare function. A connection of the comparators in the hierarchical structures for different configurations was executed by the graphics editor of Quartus II platform. By this method 15 hierarchical structures of the 64-bit comparator were constructed. The second group of the comparators is made by similar structures, but LCELL [21] buffers were set on outputs of the comparators of the first level. Such parameters as the implementation cost, the performance, and the power consumption were researched. Results were compared to the lpm_compare function. For CPLDs the offered method allows reducing the implementation cost by 32%, the delay by 44%, and the power consumption by 18%. For FPGAs the method allows reducing the implementation cost by 17% and the delay by 26%, thus power consumption does not change.

In [20], the combined technique was used for designing the 128-bit and 256-bit comparators. The 64-bit comparators made the first level. The 64-bit comparators were built by a parallel-sequential method [17] from sections on 2-, 4-, 8-, 16-, and 32-bit. The hierarchical structure of the 128-bit and 256-bit comparators was designed by the graphics editor of Quartus II platform. The offered method, in comparison with the lpm_compare function, allows reducing the implementation cost by 13% and the delay by 18% for 128-bit comparators, and for 256-bit comparators, respectively, by 19% and by 54%.

In [18,19,20], the methods of designing comparators generally on 64 bits were considered. In [18], for researches the outdated MAX+PLUS II platform and the AHDL language that supported only by Altera tools was used. In [18,19], the lpm_compare function and the AHDL language were used to implement comparators of the first level. Besides, the graphics editor of Quartus II platform was used to design hierarchical structures of comparators. The specified shortcomings do not allow the synthesis method [18,19,20] of the hierarchical comparator structures to be the universal method claiming for broad application.

This paper offers the synthesis method of the hierarchical structures of the high bitwidth comparators in Verilog language. This method has next distinctive features: all elements of hierarchical structure are described only in Verilog language, the graphics editor is not used, and the lpm_compare function is not used. Therefore the offered method is universal and can be used for the implementation of the binary comparators on any integrated circuits (CPLD, FPGA, SoC or ASIC) by design tools of any vendors. Single restriction is that these design tools must support the Verilog language.

3 A General Synthesis Technique of the Hierarchical Structures for Binary Comparators

The proposed hierarchical structure for binary comparators is shown in Fig. 1. It consists of modules of the first level comparators $CMP_1,\ldots,CMP_N$ and the combinatorial circuit CL. Each comparator of the first level CMP_n represents the binary T-bit comparator which realizes two functions: the function "greater-than" g_n and the function "equal" e_n, where $n=\overline{1,N}$.

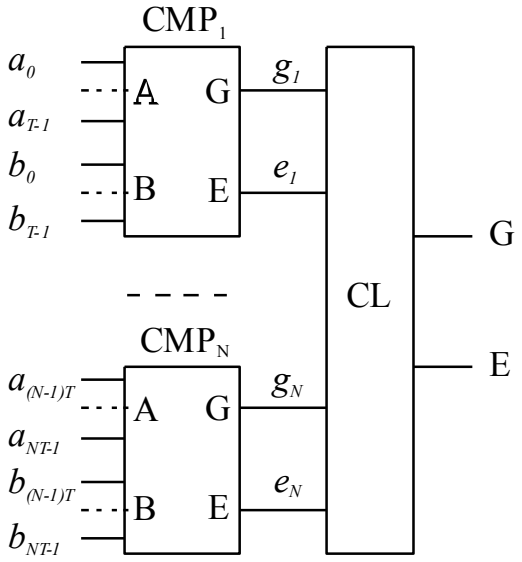

Fig. 1. The generalized hierarchical two-level structure of the comparator *C_M_NxT*

The combinatorial circuit CL calculates values of the output functions "greater-than" G and "equal" E for entire hierarchical structure. Inputs for the combinatorial circuit CL are values of output functions of first level comparators. For computing the values of the G and E functions the following equations are used:

$$G = g_N + e_N \,\&\, g_{N-1} + e_N \,\&\, e_{N-1} \,\&\, g_{N-2} + \ldots + e_N \,\&\, e_{N-1} \,\&\, \ldots \,\&\, e_2 \,\&\, g_1; \quad (1)$$

and

$$E = e_1 \,\&\, e_2 \,\&\, \ldots \,\&\, e_N. \quad (2)$$

The hierarchical structure in Fig. 1 is described by the formula:

$$C\_M\_NxT, \quad (3)$$

where C is the abbreviation of the word "Comparator", M is the width (the number of bits) of input words of the comparator, N is the number of the first level comparators, and T is the width of input words of first level comparators. In the formula (3) the following condition must be executed: M = NT.

The hierarchical two-level structure in Fig. 1 can be used as comparators of the first level. By similar way the hierarchical structure for the comparator with the high value of the M can be constructed. The example of the multi-level hierarchical design for the comparator C_16_2x8_2x4_2x2 is shown in Fig. 2. Here the 2-bit comparators C_2 are located at the first level, and the combinatorial circuits CL make all subsequent comparator levels.

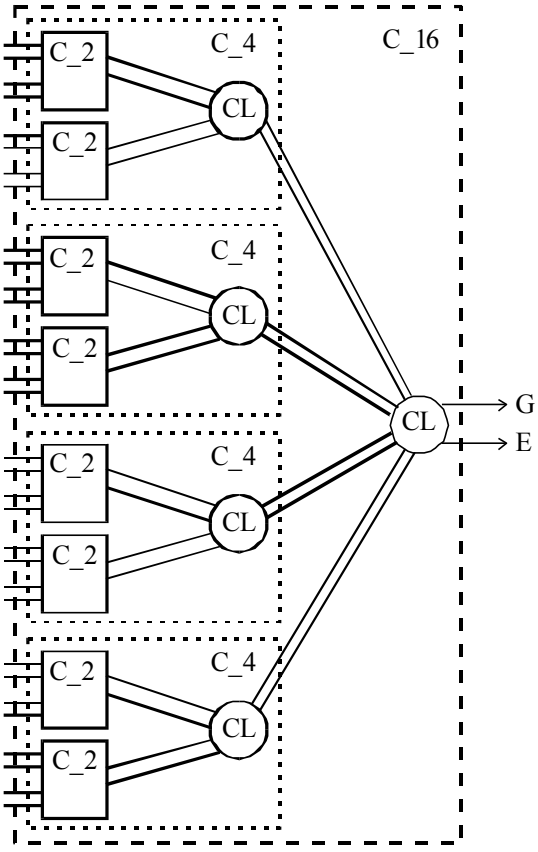

Fig. 2. Implementation a multi-level structure of the comparator *C_16_4x4_2x2*

The generalized hierarchical structure of the binary comparators is described by the formula:

$$C\_M\_N_Q x T_Q \_ N_{Q-1} x T_{Q-1} \_ \dots \_ N_1 x T_1, \tag{4}$$

where Q is the number of levels in the comparator structure, N_q is the number of the comparator modules on the level q, and T_q is the bitwidth for comparators on the level q, $q=\overline{1,Q}$.

In a formula (4) the following conditions must always be satisfied:

$$M = N_Q T_Q;$$

$$T_q = N_{q-1} T_{q-1}, \forall\ q=\overline{1,Q}\,. \tag{5}$$

The number Q of the levels in the formula (4) determines a depth of the combinatorial circuit CL in Fig. 1. Taking into account the first level comparators, the total number of the logic levels in the hierarchical structure (4) is equaled of the Q + 1.

The synthesis method of the hierarchical structures of comparators is based on the formula (4) and the conditions (5). Note that several hierarchical structures can correspond to one comparator. These structures can have the different numbers of levels and vertices at each level. Besides, the hierarchical structures can have different comparators at the first level which are distinguished by the lengths of input words.

4 The Synthesis of the First Level Comparators

In the considered approach for all hierarchical structures of the comparators at the first level it is offered to use the 2-bit comparators. Let A = (a_2, a_1) and B = (b_2, b_1) be input words of the 2-bit comparator. The Boolean function "equal" for the 2-bit comparator is not minimized and it has the form:

$$E = \bar{a}_2 \& \bar{a}_1 \& \bar{b}_2 \& \bar{b}_1 + \bar{a}_2 \& a_1 \& \bar{b}_2 \& b_1 + a_2 \& a_1 \& b_2 \& b_1 + a_2 \& \bar{a}_1 \& b_2 \& \bar{b}_1. \quad (6)$$

The Boolean function "greater-than" for the 2-bit comparator after minimization has the following form:

$$G = a_2 \& \bar{b}_2 + a_1 \& \bar{b}_2 \& \bar{b}_1 + a_2 \& a_1 \& \bar{b}_1. \quad (7)$$

The logical equations (6) and (7) for the 2-bit comparators of the first level, the logical equations (1) and (2) for the combinatorial circuit CL in Fig. 1, and the formula (4) with constraints (5) are used to build the hierarchical tree structures of the comparators.

The hierarchical structure by the nature has the smallest time delay of the signal passing in comparison with the linear decomposition or the sequential implementation. Therefore it is possible to expect that hierarchical structures of the designed comparator will have low implementation cost and high performance.

It is an open question: what formula of the hierarchical comparator structure is the best at implementation for a particular comparator bitwidth on specific FPGA family. The response to this question is defined empirically by experimental researches.

5 Experimental Results

The following designs of hierarchical comparator structures were developed in Verilog language to check the efficiency of the offered method:

one structure of the 4-bit comparator
C_4_2x2;
two structures of the 8-bit comparators
C_8_2x4_2x2;
C_8_4x2;
four structures of the 16-bit comparators
C_16_2x8_2x4_2x2;
C_16_2x8_4x2;

C_16_4x4_2x2;
C_16_8x2;
eight structures of the 32-bit comparators
C_32_2x16_2x8_2x4_2x2;
C_32_2x16_2x8_4x2;
C_32_2x16_4x4_2x2;
C_32_2x16_8x2;
C_32_4x8_2x4_2x2;
C_32_4x8_4x2;
C_32_8x4_2x2;
C_32_16x2;
and similar 16 structures of the 64-bit comparators.

From the designed hierarchical comparator structures on 8-, 16-, 32-, and 64-bits the structures with the best results in implementation cost were selected. The following comparator structures were constructed from the selected comparator structures:

21 structures of the 128-bit comparators;
25 structures of the 256-bit comparators;
17 structures of the 512-bit comparators;
17 structures of the 1024-bit comparators.

For designing the hierarchical structures of comparators on 128 bits all the structures of comparators on 8, 16 and 32 bit, and the structure C_4_2x2 (in all 15 structures) were considered. From the structures of comparators on 64 bits the 6 structures, which are the best at an implementation cost at least for one family FPGA/SoC, were selected. Thus it was constructed 21 hierarchical structures of comparators on 128 bits. The hierarchical structures of comparators on 256-, 512-, and 1024 bits were similarly built.

All designed hierarchical comparator structures are described in Verilog and realized with Altera Quartus II 13.1 platform. The implementation cost was measured by the number of the functional generators LUT (Look-Up Table). The results received by the hierarchical comparator structures were compared to the standard (traditional) description of the comparator in Verilog language [22]. Note that the results obtained by the standard description exactly match the results received by the parametrized function lpm_compare.

Some families FPGA/SoC have the identical implementation cost for the offered hierarchical structures of comparators and for the standard description. Therefore all FPGA/SoC families can be divided into three groups:

group 1 – Arria II GX, Stratix III, and Stratix IV;
group 2 – MAX II, MAX V, Cyclone III, and Cyclone IV;
group 3 – Arria V GZ and Cyclone V.

The results of experimental researches are given in Table 1, where C_m is the comparator with a width of words on m bits, C_S is the implementation cost in case of standard approach, C_H is the smallest implementation cost received by using hierarchical structures, and C_S/C_H is a relation of the corresponding parameters.

Table 1. Research results of the hierarchical comparator structures for Altera FPGA/SoC

Comparator	**Arria II GX Stratix III Stratix IV**			**MAX II MAX V Cyclone III Cyclone IV**			**Arria V GZ Cyclone V**		
	C_S	C_H	C_S/C_H	C_S	C_H	C_S/C_H	C_S	C_H	C_S/C_H
C_2	2	2	1.000	2	2	1.000	2	2	1.000
C_4	4	4	1.000	6	6	1.000	3	3	1.000
C_8	8	9	0.889	13	11	1.182	6	6	1.000
C_16	20	19	1.053	27	24	1.125	12	12	1.000
C_32	41	37	1.108	53	48	1.104	28	23	1.217
C_64	82	74	1.108	107	96	1.115	59	46	1.283
C_128	141	149	0.946	213	194	1.098	119	96	1.240
C_256	254	302	0.841	427	392	1.089	236	165	1.430
C_512	521	587	0.888	853	794	1.074	(1)	(1)	-
C_1024	2645	1189	2.225	1707	1590	1.074	(1)	(1)	-

(1) – the compiler issues the message: not enough external outputs

The analysis of Table 1 shows that in all groups for comparators C_2 and C_4 the standard approach and the proposed hierarchical method show identical results. In group 1 the hierarchical method is better for comparators C_16, C_32 and C_64, and the implementation cost decreases from 5.3% to 10.8%. For the comparator C_1024 the hierarchical method exceeds the standard method by 2.225 times. In group 2 since the comparator C_8 and to the comparator C_1024 the hierarchical method is better than the standard method, and the implementation cost decreases from 7.4% to 18.2%.In group 3 the hierarchical method is better than the standard method for comparators from C_32 to C_256, and the implementation cost decreases from 21.7% to 43.0%.

In this work the efficiency of the considered synthesis method of the comparators was researched only concerning the implementation cost. Similar approach can be also used to minimize the delay or the power consumption.

Conclusions

In the provided approach 2-bit comparators were used at the first level of hierarchical structures. Comparators with different number of bits (e.g. 3, 4, 5) can also be used at the first level.

The offered synthesis method of hierarchical comparator structures can be also used to implement of comparators in other languages of hardware description, for example VHDL and SystemVerilog. The provided method can be used to find the best hierarchical comparator structure of a certain size for a specific FPGA or SoC family. The offered method can be enhanced by using of architectural features of the FPGA/SoC, for example fast carry chains, fast cascade chains, buffers LCELL.

Acknowledgements. This research was partially supported by Bialystok University of Technology, Poland, grant no. S/WI/1/2013.

References

1. Salauyou, V.V.: Designing of digital systems based on programmable logic integrated circuits. Hot line-Telecom, Moscow (2007) (Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. - Москва: Горячая линия-Телеком, 2007. - 636 с. Второе издание.)
2. Salauyou, V.V.: Designing of functional blocks of digital systems on programmable logic devices. Bestprint, Minsk (1996) (Соловьев В.В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах. - Минск: Бестпринт, 1996. - 252 с.)
3. Guangjie, W., Shimin, S., Lijiu, J.: New efficient design of digital comparator. In: 2nd International Conference on ASIC, pp. 263–266. IEEE Press, Shanghai (1996)
4. Wang, C. C., Wu, C. F., Tsai, K. C.: 1 GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking. IEE Proceedings-Computers and Digital Techniques. IET. 145(6), 433–436 (1998)
5. Huang, C. H., Wang, J. S.: High-performance and power-efficient CMOS comparators. IEEE Journal of Solid-State Circuits. 38(2), 254–262 (2003)
6. Cheng, S. W.: A high-speed magnitude comparator with small transistor count. In: 10th IEEE International Conference on Electronics, Circuits and Systems, 3, pp. 1168–1171. IEEE Press, Sharjah (2003)
7. Lam, H., Tsui, C.: High performance single clock cycle CMOS comparator. In: IEEE International Symposium on Circuits and Systems, pp.779–782. IEEE Press, Island of Kos (2006)
8. Lam, H. M., Tsui, C. Y.: A MUX-based high-performance single-cycle CMOS comparator. IEEE Transactions on Circuits and Systems II: Express Briefs. 54(7), 591–595 (2007)
9. Kim, J. Y., Yoo, H. J.: Bitwise competition logic for compact digital comparator. In: IEEE Asian Solid-State Circuits Conference, pp. 59–62. IEEE Press, Jeju (2007)
10. Perri S., Corsonello P. Fast low-cost implementation of single-clock-cycle binary comparator. IEEE Transactions on Circuits and Systems II: Express Briefs. 55(12), 1239–1243 (2008)
11. Frustaci, F., Perri, S., Lanuzza, M., Corsonello, P.: A new low-power high-speed single-clock-cycle binary comparator. In: 2010 IEEE International Symposium on Circuits and Systems, pp. 317–320. IEEE Press, Paris (2010)
12. Abdel-Hafeez, S., Gordon-Ross, A., Parhami B.: Scalable digital CMOS comparator using a parallel prefix tree. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 21(11), 1989-1998 (2013)

13. Chuang, P., Li, D., Sachdev, M.: A low-power high-performance single-cycle tree-based 64-bit binary comparator. IEEE Transactions on Circuits and Systems II: Express Briefs. 59(2), 108–112 (2012)
14. Hauser, A., Chichester, I.: High-Speed 64-Bit Binary Comparator using Two Stages. European Journal of Engineering and Innovation. 11(2), 29–38 (2013)
15. Chuang, P. I. J., Sachdev, M., Gaudet, V. C.: A 167-ps 2.34-mW Single-Cycle 64-Bit Binary Tree Comparator With Constant-Delay Logic in 65-nm CMOS. IEEE Transactions on Circuits and Systems I: Regular Papers. 61(1), 160–171 (2014)
16. Deb, S., Chaudhury, S.: High-speed comparator architectures for fast binary comparison. In: Third International Conference on Emerging Applications of Information Technology, pp. 454–457. IEEE Press, Kolkata (2012)
17. Salauyou, V.V., Posrednikova, A.A.: An implementation on programmable logic comparators with high bitwidth. Chip-News, Engineering microelectronics. 9, 20–25 (2005) (Соловьев В.В., Посредникова А.А. Реализация на ПЛИС компараторов большой размерности. – Chip-News, Инженерная микроэлектроника, 2005, №9, с.20-25.)
18. Solov'ev, V.V., Posrednikova, A.A.: The hierarchical method of synthesis of large-capacity comparators with the use of programmable logic integrated circuits. Journal of Communications Technology and Electronics. 54(3), 338–346 (2009)
19. Salauyou, V., Gruszewski, M.: An implementation on CPLD/FPGA hierarchical comparators by parallel-sequential synthesis. Measurements, Automation, Control. 60(7) 474–476 (2014) (Salauyou V., Gruszewski M. Implementacja w strukturach CPLD/FPGA komparatorów hierarchicznych z wykorzystaniem równoległo-szeregowej syntezy // Pomiary, Automatyka, Kontrola, V.60, nr 7, 2014, s. 474-476.)
20. Salauyou, V., Gruszewski, M.: Hierarchical comparators – describe styles, synthesis results. Measurements, Automation, Control. 60(7), 498–500 (2014) (Salauyou V., Gruszewski M. Komparatory hierarchiczne – metody opisu, wyniki syntezy // Pomiary, Automatyka, Kontrola, V.60, nr 7, 2014, s. 498-500.)
21. Designing with Low-Level Primitives. User Guide. Altera Corporation, San Jose (2007)
22. Salauyou, V.: Bases of the hardware description language Verilog. Hot line-Telecom, Moscow (2014) (Соловьев В.В. Основы языка проектирования цифровой аппаратуры Verilog. - Москва: Горячая линия-Телеком, 2014. - 208 с.)