**Valery SALAUYOU**
BIALYSTOK UNIVERSITY OF TECHNOLOGY
45A Wiejska St., 15-351 Bialystok, Poland

# The use of hierarchical structures for design of high-speed digital comparators on FPGA/SoC

**Abstract**

This paper presents a design method of high-speed digital comparators on FPGA/SoC by means of hierarchical structures. A synthesis technique of hierarchical structures for comparators is offered. In this technique, the comparator best hierarchical structure is empirically found for a certain FPGA family. The proposed method allows reducing a delay for 256-bits comparators by 1.245 to 2.516 times as compared with a traditional approach, and for 512-bits comparators by 3.399 times. The method also allows reducing an area by 40.2% on occasion.

**Keywords**: comparator, high-speed, hierarchical structures, system on chip, field programmable logic array, FPGA, SoC.

## 1. Introduction

A digital comparator is one of the most fundamental components in digital systems for many applications such as digital signal processing, parallel computing, sorting and searching data, image processing, and 3D graphics. The evolution of microelectronics has resulted in appearance of system-on-a-chip (SoC) microcircuits with scale of integration that is enough to implement the complex digital systems. As a rule, SoC includes field-programmable logic arrays (FPGA). Consequently, the design on FPGA and SoC of high-speed area-efficient comparator architecture becomes very important.

The digital comparator is a combinational circuit that compares two binary numbers, A and B, and forms three logic functions: "greater than" $G$ (A > B), "equal to" $E$ (A = B), and "less than" $L$ (A < B). Note that when creating comparators for longer inputs (64 bit and more) it is enough to implement only two functions "greater than" $G$ and "equal to" $E$ as the function "less than" $L$ can be always defined by the logic equation $L = \overline{G} \,\&\, \overline{E}$, where the character «&» means logical AND.

In the last few years, the design of high-speed, low-power, and area-efficient binary comparators has received a great deal of attention. A high-performance tree-structure comparator using all-N-transistor (ANT) dynamic CMOS logic is proposed in [1]. In [2], a single-cycle two-phase comparator is proposed by using a priority-encoding algorithm. It shows 16% performance enhancement over [1]. A parallel-most-significant bit (MSB)-checking algorithm is proposed in [3] by introducing a new static priority encoder and a MUX-based comparator structure. In [4,] there is described a new comparator with bitwise competition logic (BCL) to detect the earliest first "1" away from the MSB after pre-encoding the inputs. In [5], there are presented tree-based comparators in which dynamic Manchester structures are used to facilitate the comparison process. In [6], a comparator design exploits a 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. A single-cycle binary comparator utilizing a radix-2 tree structure is proposed in [7]. In [8], a single-cycle tree-based binary comparator with constant-delay (CD) logic is presented.

The considered projects of comparators are intended for building the comparators on Application Specific Integrated Circuits (ASICs) and do not suit creation of comparators on FPGA/SoC. In [9], the designing method of digital comparators on FPGA/SoC as hierarchical structures is offered. 4-bit comparators are used at the first level of the hierarchical structures that are built using four methods by means of the function lpm_compare from Altera, a language AHDL, the parallel method, and the sequential method. The proposed method has been compared with sequential and parallel methods, the function lpm_compare, and the synthesis method of the design software MAX+PLUSII. In [10], a synthesis method on FPGA/SoC of hierarchical structures for binary comparators is presented. The method is aimed at creation of comparators at the minimal cost.

This paper presents a design method on FPGA/SoC of the hierarchical structures for high-speed digital comparators. The offered method allows changing the number of logical levels in the hierarchical structure of the certain comparator in a broad range. As a result the user can select a trade-off the cost against the performance. The comparator diagram is completely combinatorial circuit which does not contain clock signals therefore the comparator diagram does not require additional circuits for generating clock signals. Besides, in the offered approach there are no bitwise carries, and parallelism of the hierarchical structures provides high speed.

## 2. Synthesis technique of hierarchical structures for comparators

The hierarchical two-level structure of comparators is shown in Fig. 1. It consists of modules of the first level comparators $CMP_1,\ldots,CMP_{N1}$ and the combinatorial circuit CL. Each comparator of the first level $CMP_n$ is an $M_1$-*bit* comparator that realizes the function "greater than" $g_n$ and the function "equal to" $e_n$, where $n = \overline{1, N_1}$.

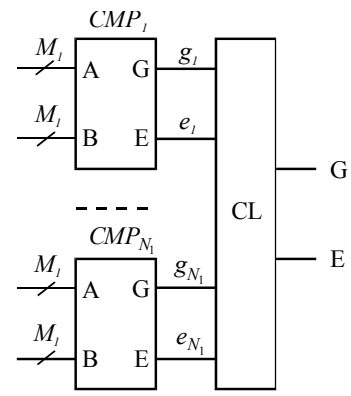


Fig. 1. The two-level hierarchical comparator with structure $C\_M_2 \times N_1 \_ M_1$

The combinational circuit CL based on the values of the functions generated by the comparators of the first level calculates values of output functions "greater than" $G$ and "equal to" $E$ for all hierarchical structure by means of the following logical equations:

$$G = g_{N_t} + e_{N_t} \,\&\, g_{N_{t-1}} + e_{N_t} \,\&\, e_{N_{t-1}} \,\&\, g_{N_{t-2}} + \ldots + \\ + e_{N_t} \,\&\, e_{N_{t-1}} \,\&\, \ldots \,\&\, e_2 \,\&\, g_1 \quad ; \qquad (1)$$

and

$$E = e_1 \,\&\, e_2 \,\&\, \ldots \,\&\, e_{N_1} \, . \qquad (2)$$

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

$$C\_M_2 \times N_1 \_ M_1 ,$$

where $C$ is the abbreviation of the word "Comparator", $M_1$ is the width (the number of bits) of input words of the comparator, $N_1$ is the number of comparators on the first level, $M_2$ is the width of input words of the two-level comparator, $M_2 = N_1 \cdot M_1$.

The hierarchical two-level structure of Fig. 1 can be used as comparators of the first level. In a similar way, the hierarchical structure of the comparator with the longer inputs can be constructed.

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

$$C\_M_T \times N_{T-1}\_M_{T-1} \times N_{T-2}\_...M_2 \times N_1\_M_1, \quad (3)$$

where $M_t$ is the width of input words of the comparators on level $t$, $t = \overline{1,T}$, $N_{t-1}$ is the number of the comparators on the previous level $t$-1 from which comparators of the level $t$ are built, $T$ is the number of the logic levels (a depth) of the comparator.

In the formula (3), the following conditions must always be satisfied:

$$M_t = N_{t-1} \cdot M_{t-1} \qquad \forall t = \overline{2,T}. \quad (4)$$

Generally, the multi-level hierarchical structure of the comparator can be shown as in Fig. 1, where $CMP_1,\ldots,CMP_{N1}$ are the comparators of the first level, and the combinational circuit CL realizes hierarchical connections of the functions "grater then" and "equal to" that are determined by the formula (3).

Synthesis of the comparator hierarchical structure is executed based on the formula (3) at satisfying the conditions (4). Note that for the same comparator, different hierarchical structures with different depth, a different number of modules, and different width of input words at each level can be constructed.

The basis of logical elements of the modern FPGA and SoC is made by the functional generators look up table (LUT). The LUT represents memory RAM which can be programmed on implementation of any logic function from a small number of arguments. As the number of inputs of the functional generators LUT for the majority of FPGA families is equal to 4, it is offered for all hierarchical structures of comparators at the first level to use 2-bit comparators. It allows realizing each function of comparators of the first level on one LUT.

Let $A = (a_2, a_1)$ and $B = (b_2, b_1)$ be input words of the 2-bit comparator. The Boolean function “equal to” for the 2-bit comparator is not minimized and it has the following form:

$$E = \bar{a}_2 \,\&\, \bar{a}_1 \,\&\, \bar{b}_2 \,\&\, \bar{b}_1 + \bar{a}_2 \,\&\, a_1 \,\&\, \bar{b}_2 \,\&\, b_1 + a_2 \,\&\, a_1 \,\&\, b_2 \,\&\, b_1 + a_2 \,\&\, \bar{a}_1 \,\&\, b_2 \,\&\, \bar{b}_1. \quad (5)$$

he Boolean function “greater than” for the 2-bit comparator after minimization has the following form:

$$G = a_2 \,\&\, \bar{b}_2 + a_1 \,\&\, \bar{b}_2 \,\&\, \bar{b}_1 + a_2 \,\&\, a_1 \,\&\, \bar{b}_1. \quad (6)$$

Hence the specific hierarchical structure of the comparator is defined by the formula (3) in the case of satisfying the conditions (4). The comparators of the first level are realized based on the logical equations (5) and (6), and the combinational circuit CL is defined by the logical equations (1) and (2) on each level of the hierarchical structure.

Open is a question: what formula (3) of the comparator hierarchical structure is the best at implementation of the certain size comparator for the specific family FPGA or SoC. The response to the matter is empirically fined in the offered approach by execution of a large number of the experimental research.

## 3. Experimental results

The experimental research were performed on FPGA and SoC families of the firm Altera by means of design software Quartus II. The comparator designs were coded in the Verilog language. The synthesis results of the comparators were compared to traditional implementation of the comparator in the Verilog language when functions are "greater then" and "equal to" described by means of the corresponding operations.

The experiments were made as follows. For comparators on 4, 8, 16, 32, 64, 128 and 256 bits all possible hierarchical structures were constructed (i.e. 128 designs together with the comparator on 2 bits). For creation of the 512-bit comparators, the structures with the best results were selected from the comparator designs on smaller a number of bits. As a criterion of optimization, the maximum high-speed performance was used. In total, 58 design of the 512-bit comparators were constructed.

At synthesis all settings of parameters were accepted by default of the software Quartus II. As the parameter of optimization, the maximum time delay of signal passing of from inputs to outputs in nanoseconds was considered.

Table 1 contains the values of the signal delays $D_T$ in the case of building the comparators by the traditional method, where C_*n* is *n*-bit comparator; *min* and *max* are the minimum and maximum value of the delay for the comparator C_*n*.

Tab. 1. The maximum time delay $D_T$ of signal passage (in nanoseconds) for the comparators implemented by the traditional method

| Family | C_4 | C_8 | C_16 | C_32 | C_64 | C_128 | C_256 | C_512 |
|---|---|---|---|---|---|---|---|---|
| MAX II | 6.3 | 7.1 | 7.4 | 9.5 | 9.0 | 17.6 | (1) | (1) |
| MAX V | 11.5 | 14.3 | 15.4 | 16.9 | 19.7 | 22.1 | (1) | (1) |
| MAX 10 | 8.8 | 9.5 | 10.4 | 11.3 | 14.4 | 18.1 | (1) | (1) |
| Cyclone | 10.5 | 10.4 | 13.2 | 13.4 | 14.6 | 15.2 | (1) | (1) |
| Cyclone II | 11.2 | 12.4 | 13.4 | 13.3 | 16.8 | 22.4 | 34.5 | (1) |
| Cyclone III | 7.7 | 9.3 | 8.4 | 9.7 | 12.2 | 16.4 | 35.3 | (1) |
| Cyclone IV | 8.5 | 8.9 | 9.0 | 10.6 | 12.3 | 15.8 | 45.2 | (1) |
| Cyclone V | 9.3 | 11.6 | 16.3 | 15.8 | 26.3 | 27.0 | 63.4 | (1) |
| Arria | 11.0 | 11.9 | 19.3 | 16.5 | 18.9 | 38.6 | 76.7 | (1) |
| Arria II | 8.7 | 10.6 | 11.2 | 22.9 | 14.9 | 29.2 | (1) | (1) |
| Stratix | 11.2 | 10.6 | 11.7 | 12.7 | 13.7 | 16.6 | 20.1 | 27.2 |
| Stratix II | 10.0 | 11.2 | 13.9 | 12.6 | 14.2 | 24.7 | 47.4 | 105.3 |
| Stratix III | 9.4 | 9.8 | 11.5 | 11.0 | 14.1 | 19.0 | 34.7 | (1) |
| Stratix IV | 10.3 | 11.3 | 12.6 | 13.3 | 22.9 | 32.6 | 42.7 | (1) |
| *min* | 6.3 | 7.1 | 7.4 | 9.5 | 9.0 | 15.2 | 20.1 | 27.2 |
| *max* | 11.5 | 14.3 | 19.3 | 22.9 | 26.3 | 38.6 | 76.7 | 105.3 |

(1) - Comparator implementation is impossible because of a lack of pins or LUTs

Table 2 represents values of signal delays $D_H$ in the case of creating the comparators by means of the hierarchical structures.

Tab. 2. The maximum time delay $D_H$ of signal passage (in nanoseconds) for the comparators implemented by the hierarchical structures

| Family | C_4 | C_8 | C_16 | C_32 | C_64 | C_128 | C_256 | C_512 |
|---|---|---|---|---|---|---|---|---|
| MAX II | 6.4 | 6.6 | 7.3 | 8.9 | 9.3 | 18.9 | (1) | (1) |
| MAX V | 11.1 | 13.8 | 16.7 | 20.6 | 11.1 | 27.7 | (1) | (1) |
| MAX 10 | 8.8 | 10.7 | 13.0 | 12.7 | 14.4 | 19.9 | (1) | (1) |
| Cyclone | 10.8 | 11.5 | 12.9 | 15.5 | 16.9 | 17.6 | (1) | (1) |
| Cyclone II | 10.9 | 12.4 | 13.1 | 14.1 | 15.9 | 19.0 | 26.2 | (1) |
| Cyclone III | 7.6 | 9.3 | 9.2 | 9.9 | 12.0 | 13.2 | 28.3 | (1) |
| Cyclone IV | 8.6 | 8.9 | 10.5 | 12.1 | 11.6 | 13.2 | 40.8 | (1) |
| Cyclone V | 9.6 | 11.7 | 15.6 | 16.3 | 21.4 | 18.2 | 31.1 | (1) |
| Arria | 11.0 | 15.8 | 15.0 | 15.2 | 16.9 | 21.2 | 30.5 | (1) |
| Arria II | 8.8 | 9.1 | 10.7 | 16.9 | 14.8 | 18.3 | (1) | (1) |
| Stratix | 11.7 | 12.6 | 12.4 | 13.9 | 15.6 | 18.3 | 21.4 | 29.0 |
| Stratix II | 10.0 | 10.8 | 10.3 | 11.4 | 12.3 | 15.0 | 20.1 | 30.9 |
| Stratix III | 10.3 | 8.4 | 10.1 | 11.4 | 12.7 | 14.6 | 18.4 | (1) |
| Stratix IV | 10.4 | 11.5 | 10.9 | 12.4 | 16.5 | 19.2 | 26.0 | (1) |
| *min* | 6.4 | 6.6 | 7.3 | 8.9 | 9.3 | 13.2 | 18.4 | 29.0 |
| *max* | 11.7 | 15.8 | 16.7 | 20.6 | 21.4 | 27.7 | 40.8 | 30.9 |

(1) - Comparator implementation is impossible because of a lack of pins or LUTs

The results of comparing the time delay are presented in Table 3 for the comparators constructed by the traditional method and by means of the offered method.

The analysis of Table 3 shows that for families Cyclone, Stratix, MAX II, and MAX 10 the ratio $D_T/D_H$ basically is less than 1.0. Hence, it is recommended for these families to use the traditional method for synthesis of digital comparators.

For families Cyclone IV and MAX V the results of Table 3 are not unambiguous, for some comparators the best method is the traditional approach, and for other comparators the best method is the offered approach. Therefore it is necessary to apply both methods to these families and select the best result from the received results .

Table 3 analysis shows that for families Cyclone, Stratix, MAX II, and MAX 10 the ratio $D_T/D_H$ basically is less than 1.0. Hence it is recommended for these families for synthesis of digital comparators to use the traditional method.

Tab. 3. Comparison of a performance of comparators: ratio $D_T/D_H$

| Family | C_4 | C_8 | C_ 16 | C_ 32 | C_ 64 | C_ 128 | C_ 256 | C_ 512 |
|---|---|---|---|---|---|---|---|---|
| MAX II | 0.981 | 1.075 | 1.002 | 1.066 | 0.973 | 0.936 | (1) | (1) |
| MAX V | 1.036 | 1.041 | 0.925 | 0.819 | 1.783 | 0.798 | (1) | (1) |
| MAX 10 | 0.994 | 0.891 | 0.804 | 0.887 | 1.001 | 0.910 | (1) | (1) |
| Cyclone | 0.975 | 0.902 | 1.028 | 0.864 | 0.865 | 0.865 | (1) | (1) |
| Cyclone II | 1.029 | 0.997 | 1.021 | 0.939 | 1.054 | 1.176 | 1.320 | (1) |
| Cyclone III | 1.009 | 0.999 | 0.907 | 0.977 | 1.021 | 1.248 | 1.245 | (1) |
| Cyclone IV | 0.987 | 1.001 | 0.856 | 0.871 | 1.058 | 1.192 | 1.106 | (1) |
| Cyclone V | 0.974 | 0.986 | 1.043 | 0.966 | 1.232 | 1.482 | 2.036 | (1) |
| Arria | 1.000 | 0.757 | 1.285 | 1.087 | 1.116 | 1.815 | 2.516 | (1) |
| Arria II | 0.999 | 1.171 | 1.047 | 1.352 | 1.006 | 1.588 | (1) | (1) |
| Stratix | 0.957 | 0.837 | 0.947 | 0.915 | 0.882 | 0.908 | 0.942 | 0.938 |
| Stratix II | 1.000 | 1.031 | 1.345 | 1.103 | 1.159 | 1.642 | 2.360 | 3.399 |
| Stratix III | 0.909 | 1.173 | 1.136 | 0.966 | 1.115 | 1.295 | 1.880 | (1) |
| Stratix IV | 0.996 | 0.986 | 1.157 | 1.074 | 1.383 | 1.698 | 1.641 | (1) |
| min | 0.909 | 0.757 | 0.804 | 0.819 | 0.865 | 0.798 | 0.942 | 0.938 |
| max | 1.036 | 1.173 | 1.345 | 1.352 | 1.783 | 1.815 | 2.516 | 3.399 |

(1) - Comparator implementation is impossible because of a lack of pins or LUTs

For remaining families (Arria, Arria II, Cyclone II, Cyclone III, Cyclone IV, Cyclone V, Stratix II, Stratix III, and Stratix IV) using of the hierarchical structures exceeds the traditional method. For example, for implementation of 256-bit comparators this advantage makes for family Arria 2.516 times, for family Cyclone II 1.32 times, for family Cyclone III 1.245 times, for family Cyclone IV 1.106 times, for family Cyclone V 2.036 times, for family Stratix II 2.36 times, for family Stratix III 1.88 times, and for family Stratix IV 1.641 times. The greatest advantage of the hierarchical structures over the traditional approach is observed for family Stratix II at implementation of 512-bit comparators, it makes 3.399 times.

Fig. 2 shows an increase in the comparator delays for devices of the family Stratix II for using the traditional method and the offered method.

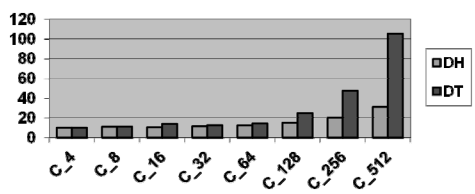


Fig. 2. The increase in comparator delays for devises of the family Stratix II at using the traditional method ($D_T$) and the offered method ($D_H$)

From Fig. 2, it is can be seen that when designing comparators by the traditional method the delay $D_T$ starts to increase quickly beginning with 128-bit comparators. In the case of usage of hierarchical structures, the delay $D_H$ increases noticeably slower. Therefore it is possible to expect that for creation of comparators with longer inputs (more than 512 bits) the advantage of hierarchical structures, in comparison with the traditional approach, will increase.

## 4. Conclusions

This paper presents the synthesis method of the hierarchical structures for binary comparators, which allows building high-speed comparators on FPGA and SoC. It opens new possibilities to designers of digital systems for a development of various blocks by using the high-speed comparators of longer inputs.

In the presented approach, at the first level 2-bit comparators were used. It is possible to use also the comparators on other number of bits, for example, 3, 4, and so on, as comparators of the first level.

The further improvement of the proposed method can be made by the way using of architectural FPGA and SoC singularities, for example, by using fast carry chains, cascade chains, buffers LCELL, by grouping comparator logic to clusters for implementation in one functional block, by implementation of comparators of the first level in memory blocks, and in blocks of distributed memory.

*The present study was supported by the grant S/WI/1/2013 from Bialystok University of Technology and funded from the resources for research by the Ministry of Science and Higher Education.*

## 5. References

[1] 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, 1998, vol. 145, no 6, pp. 433-436.

[2] Huang C.H., Wang J.S.: High-performance and power-efficient CMOS comparators. IEEE Journal of Solid-State Circuits, 2003, vol. 38, no 2, pp. 254-262.

[3] Lam H.M., Tsui C.Y.: A MUX-based high-performance single-cycle CMOS comparator. IEEE Transactions on Circuits and Systems II: Express Briefs, 2007, vol. 54. no. 7, pp. 591-595.

[4] Kim J.Y., Yoo H.J.: Bitwise competition logic for compact digital comparator. Proc of the IEEE Asian Solid-State Circuits Conference, 2007, IEEE, pp. 59-62.

[5] Perri S., Corsonello P.: Fast low-cost implementation of single-clock-cycle binary comparator. IEEE Transactions on Circuits and Systems II: Express Briefs, 2008, vol. 55, no. 12, pp. 1239-1243.

[6] 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, 2013, vol. 21, no. 11, pp. 1989-1998.

[7] Hauser A., Chichester I.: High-speed 64-bit binary comparator using two stages. European Journal of Engineering and Innovation, 2013, vol. 11, no. 2. pp. 29-38.

[8] 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, 2014, vol. 61, no. 1, pp. 160-171.

[9] 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, 2009, Vol. 54, no. 3, pp. 338–346.

[10] Salauyou V., Gruszevski M.: Designing of hierarchical structures for binary comparators on FPGA/SoC. Proc. of the 14th International Conf. Computer Information Systems and Industrial Management (CISIM 2015), Warsaw, Poland, September 24-26, 2015, Springer, pp. 386-396.

*Received: 08.03.2016 Paper reviewed Accepted: 02.05.2016*

**Valery SALAUYOU, DSc, PhD, eng.**

Valery Salauyou received the PhD and DSc degrees in computer science from the Belarusian State University Informatics and Radioelectronics, in 1986 and 2003, respectively. From 1992 he is Associate Professor at the Bialystok University of Technology. His research interests are in the area of VLSI design, with emphasis on logic synthesis, Embedded Systems, architectures of programmable logic (CPLD/FPGA/SoC), and optimization of FSMs.

*e-mail: v.salauyou@pb.edu.pl*