

J. M. Krainyk, Cand. Sc. (Eng.), Doctoral fellow; V. O. Perov

CONSTRUCTION PECULIARITIES OF TPC-DECODER SEPARATE BLOCKS ON THE BASE OF FPGA

The paper considers the problem, dealing with the construction of the certain blocks for the reconfigurable decoder of Turbo-Product-codes. Principles of organization of the blocks of minimal values position search in the input vector and universal shift unit were suggested, they enable to provide the reuse of the logic resources of the programmable logic microcircuits for processing of the codes of various structure. Principles of blocks organization are suitable for usage both in specialized microcircuits and in the programmable logic integrated circuits (PLIC), but the emphasis is made on the latter due to their universality and possibility of the multiple change of configuration structure. The developed searching unit uses in its structure two types of simple elements and on their base allocates values which are minimal. The structure, enabling to perform the search of the preset amount of the minimal values is proposed. In its turn, universal shift unit provides necessary relocations in the input or output vector of values, so that they can be written in the memory module or sent to the processing module. As a result of using this unit the parallel access to the values in the memory is provided, this increases considerably the rate of the decoding process. Besides, the unit is oriented on the work with the codes of different length and performs the values shift during minimal number of operation frequency cycles. Accordingly, irrespective of the code, used in the process of decoding, values shift will be performed during one and the same unit of time. In accordance with the principles, suggested in the paper, corresponding units are developed by means of the language of circuit engineering description VHDL. Simulation of the developed units operation is performed, it proved the correctness of their operation. These units in the decoder Turbo-Product-codes enable to improve the universality of the decoder, providing high carrying capacity.

Key words: decoder, unit, Turbo-Product-codes, PLIC.

Introduction

Algorithm of the decoding of noise-immune codes and, in particular, Turbo-Product-codes (TP-codes) obtained wide practical application and were investigated in numerous studies [1 – 8]. They are used in data transmission systems of various designations and in different physical environments. Nevertheless, not all the algorithms can be reduced to mathematical model in physical realization on the base of modern microcircuits. This is explained by the resources limitations, available in such microcircuits.

Reconfigured decoder on the base of programmable logic integrated circuits (PLIC) can perform decoding for various codes. At the same time, PLIC-based systems are reconfigured systems reconfiguration of the decoder implies the possibility to perform changes of the setting, as a result the decoder will be able to process the information, formed on the basis of the new code. Thus, two levels of reconfiguration can be allocated:

the first – hardware – is provided by the setting possibilities of PLIC itself;

the second level – logic – is provided by the realization on the level of the language of circuit engineering description with further conversion in the bit file of the microcircuit firmware.

The given research is devoted to the second aspect, particularly, in the part, concerning the construction of the universal blocks, which can perform separate stages of the decoding.

Depending on the settings the reconfigured decoder can perform the processing of the code words of different length. The problem of the reuse of the realized blocks resources for the processing of various codes is of great importance. Such concept will provide the possibility not to use the separate block for each processed code and decrease the usage of the resources at the expense of their reuse within one universal block which performs the preset function for all the codes. The greater is the number of such universal blocks, the more resources can be saved. This is the possible economic effect of the research.

Analysis of the scientific sources

Decoding of TP-codes, applying Chaise algorithms [2] is rather popular due to its simplicity and increased ability of faults correction than in hard decoder. It is used, also b , as the basic for other decoding algorithms [3]. Main idea of the algorithm is based on the assumption that the errors (if they are available in the code word), occurred in the process of the transmission in bits, have the least absolute value. Generation of test vectors is carried out by means of exhaustive search of all the possible binary values in the preset positions. For each code word matric calculation is carried out, it enables to evaluate the proximity to the correct code word. On the base of the matric value the basic solution to be used for further processing is selected. Chaise algorithm is referred to the list of the list-based algorithm, which try to find the correct solution on the base of the solutions list generation, among the elements of which these which meet the preset criteria, are selected.

In [4] the decoding block, which allows to process the data for the component of the code of 32 symbols of the length is presented. Such block finds 3 minimal values. Besides, for its correct operation only 32 values must be sent to it. That is why, for the realization of the reconfigured decoder the development of minimal values searching block, which can process the values of the codes of various length, is obligatory.

In numerous papers [5 – 7] the emphasis is made on general architecture of the decoder, whereas the separate blocks, the decoder consists of, require more detailed description. In particular, if the suggested algorithm is oriented on the realization in the hardware PLIC platform.

In [8, 9], the method of the decoding and general architecture of the decoder for the realization of the base of PLIC are presented . This work is a logic continuation of the above-mentioned work and the emphasis is made on the realization of separate modules within the decoder.

The aim of the research is the increase of the decoder universality for the possibility of the messages processing on the base of various codes.

To reach this aim it is necessary to perform the study regarding the organization of the blocks of the minimal values search and universal shift block for the reconfigured TP-codes decoder. Thus, it becomes possible to perform decoding on the base of various codes with the possibility of changing the regulations for the horizontal and vertical code elements. Microcircuits PLIC are selected as the target platform for the realization of such decoder, that is why, the characteristic features of this type of microcircuits must be taken into account. The result of the research, carried out must be the methods of the specified block construction for reconfigured decoder of the base of PLIC, that enables to perform the codes processing of various length and, accordingly, make this element of the data transmission system more universal.

Main part

The decoding cycle of one code word starts from the supply of the values vector at the input of the decoding block. In this vector it is necessary to find the specified number of minimal values. To determine one position, the absolute value of which in the code word does not exceed any other value in the vector, it is sufficient to use the block, schematic structure of which is presented in Fig. 1. It is connected with other similar blocks in the serial channel. Such blocks are used in the considered research [4, 5].

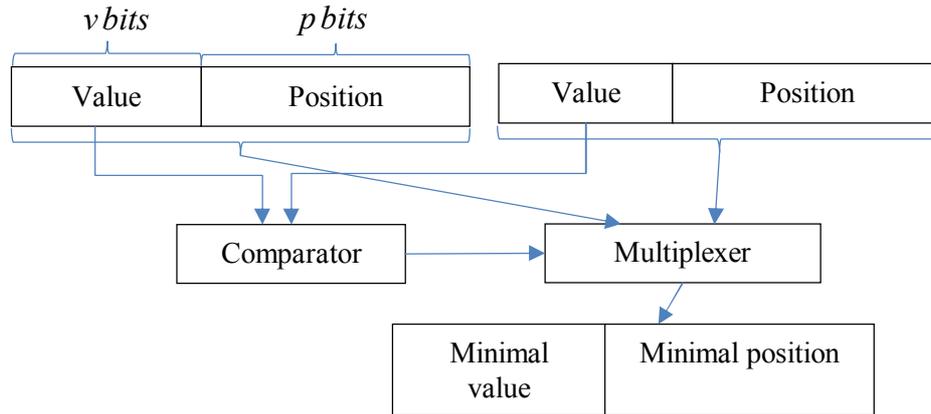


Fig. 1. Comparison block of two values

This block allows to find one value, however for generation of the test vectors list, it is necessary to define several positions with the minimum absolute values. The values, not selected as minimal, are rejected at a certain level by such blocks, that is why, it is impossible to determine correctly the positions of the minima, using such blocks. The characteristic feature of the chain of the comparison blocks is that the value which turned out to be less of the two, is not rejected at once but is supplied to the input of the parallel chain. Both input values, supplied to the input of such element, as a result, can be selected as a finite minima. If one of the values was rejected immediately, then, on the condition of the comparison of two minima, one of them would be lost and the position of the minimum would be calculated incorrectly. Organization of the parallel stage gives the possibility to the rejected value to pass on the parallel branch and be among the minima. Values, supplied to the input of the block is the combination of the value itself and its index. The fixed length is allotted for the value, the length equals v bits. The length p bit is allotted for the value of the index. For the operation of such stage another type of blocks, shown in Fig. 2, is necessary. The block of this type are used in [4, 5] for the decoding.

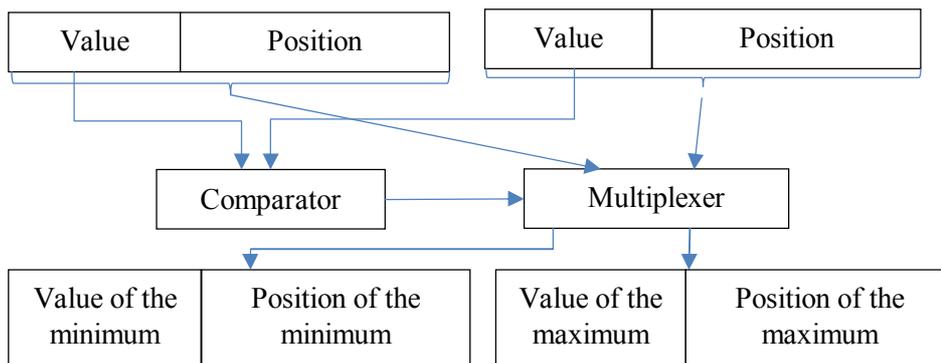


Fig. 2. Block of the comparison of two values for organization of the parallel chain

Unlike the previous block, in this case the multiplexer has two outputs. One of them is intended for choosing the minimal value. In case of the equality of the values the preference is given to the value with the smaller index. The value, that was not chosen as a minimal will be sent to another output.

In Figs. 1, 2 the blocks are presented in the forms of the registers. Also they could be presented as the blocks with the corresponding number of inputs and outputs, since the registers can be considered as the separate components, used for connecting blocks of the combinational logic.

Another important moment in the process of minima searching block operation is the possibility to process the input data with various length of the code word. For this purpose, the value that shows maximal absolute value in the system is sent to the input registers, indices of which exceed the length of the code word. Also it is necessary to provide that all the resulting values of minima positions enter the range, determined by the length of the code – $[0, n - 1]$.

In case, when the values beyond the range were not eliminated (it is possible, for instance, when all the input values are the same or for certain variants of the code words), they must be eliminated directly at the last stage to form correctly the test vectors.

We will pass directly to the organization of minima search stage.

Serial connection of the above -mentioned blocks in the stage with several levels enables to determine the minimal element with the smallest index (on condition of the equality of the values). However, for the operation of the algorithm on the base of generation of the test vectors list one value is not sufficient, that is why ,such level can be repeated only to a certain moment, when the determined amount of values will be allocated, the final set of values must be selected among them.

At the first level of such stage the blocks with two outputs are involved, this enables to sort the values. Their number at the first level is $n_{\max} / 2$. As a result $n_{\max} / 2$ values, are obtained, they enter the first stage and other part enters the parallel stage. Two halves are processed, applying various blocks. The first part enters the blocks with double output, whereas, the second part enters the blocks with a single output. Thus, at the second level, $n_{\max} / 4$ blocks with the double output and the same number of blocks with a single output must be involved. Characteristic feature of processing the values, which entered the second part, is that, practically, it is an independent stage, that as a result must supply a single value. Regarding the first part, that entered the second level at the input of the blocks with the double output, the same procedure is applied for the processing of the results of this part, as for the second level. As a result, half of the obtained values will be sent is to the input of the stage, that contains $n_{\max} / 8$ blocks with double output (it will be $n_{\max} / 4$ from the total amount of values, correspondingly). Such procedure is repeated till the moment when the first minimal value is found, this value is obtained at the output of the last block with the double output (output, that indicates the minimal value). Total number of levels, necessary for the organization of the first part of such stage is

$$lev_first_stage = \log_2 n_{\max} . \quad (1)$$

At the output of the first stage of the stage such amount of values will be obtained:

$$first_stage_outs = lev_first_stage + 1. \quad (2)$$

One of the obtained values will belong to the set of minima positions values.

Accordingly ,it is necessary to process $first_stage_outs - 1$ values. $first_stage_outs - 2$ values are the outputs of the substages, formed from the blocks with a single output and one more value is selected as the value that lost in comparison at the last level with the usage of the block with the double output.

The second level of the minima set search stage is formed on the base of the assumption:

$$first_stage_outs > k. \quad (3)$$

It is necessary to allocate from the found values the necessary amount. It is connected with the fact that the generation and further processing of a large number of testing vectors requires large resources and may not correspond to the constrains, regarding logic resources in the chosen microcircuit of PLIC.

Processing at the second stage of the chain is intended to reveal the rest of $k - 1$ values. Organization of this stage differs from the previous one . The general structure of the minima search blocks is shown in Fig. 3.

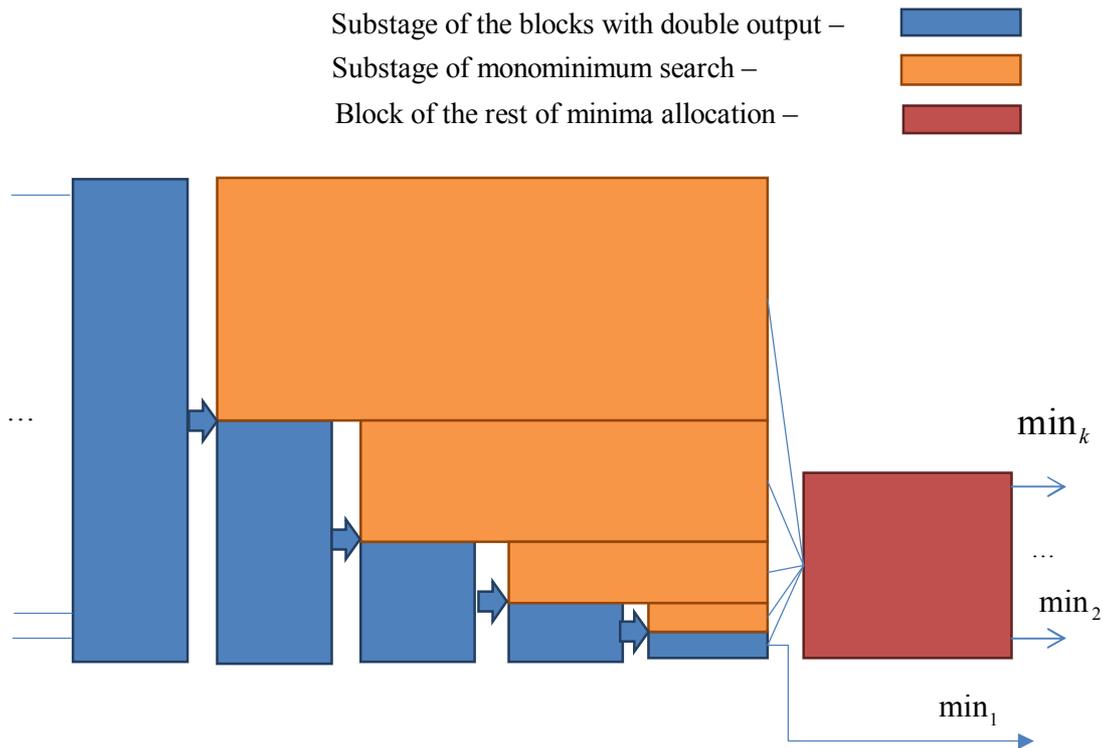


Fig. 3. Organization of the stage for minimum value position search

Special attention should be paid to the block of the rest of minima search, intended for the allocation, among the obtained values, of the necessary amount of minimal values at the output. Just so as for the previous blocks, the above-mentioned blocks are its structural elements. The number of the components level, which form the given block, will be determined as:

$$layer_count = first_stage_outs - 1. \quad (4)$$

It should be noted that the number of the components with a single output will be:

$$out1_count = layer_count - 1, \quad (5)$$

That is, with every level except the first one component will be rejected.

Presentation of the input data matrix in the hardware structure, as a rule, is organized using the register or block memory. Such memory provides that during one cycle only one value can be read from it. Accordingly, n_{max} modules of such memory are available in the decoder. They enable to obtain the necessary vector of the input data either for the row or for a column during one cycle. The problem is that in the process of the input data recording at the same addresses, the processed data for the following semi-iteration will be located in one memory module. That is why, for reading the input data it will be necessary to spend n_{max} cycles, instead of one cycle. It results in considerable loss of the decoder carrying capacity.

To avoid such loss of the decoder performance, it is necessary to organize the process of recording and reading of the resources in PLIC memory that would allow to perform reading both of rows and columns during one cycle. The base for such organization is universal barrel shifter, which provides shift/rotation of the input vector at the preset number of positions. It is connected to the block, that performs reading/writing of the data from the memory. In its turn block of reading/writing is connected to n_{max} blocks of memory. It transfers data from the memory to the universal barrel shifter. An important moment to be taken into account for the reconfigured decoder

is the necessity of the correct execution of the shift for the words of the different length. The variant of the rotation realization for the basic lengths codes setting is possible. In this case, the processing of the vectors is carried out according to the adjustments of the basic code and no correspondence between real lengths must be taken into account at the following stages of the processing, to make the final result correct. As the basic code lengths it is recommended to select such values which are the results of the exponentiation of two (for instance, 16, 32, 64, 128). For the codes, the length of which is not equal the value n_{max} the shift of only the part of the complete set of data that corresponds to the real code length, is performed. This can be written as:

$$shift_vals_denom = 2^{\log_2 n_{max} - \log_2 n_{cur}}, \quad (6)$$

where $shift_vals_denom$ – is a divisor for actual number of values to be shifted; n_{cur} – real length of the processed code.

In its turn, the number of values, to be shifted, is calculated as:

$$shift_count = \frac{n_{max}}{shift_vals_denom}. \quad (7)$$

From the point of view of the realization the module of the universal barrel shifter is logic resources-intensive as it must provide a great number of the interconnections of various logic components. That is why, this factor must be taken into account when PLIC final microcircuit is to be chosen, this circuit must contain the sufficient amount of logic resources for performing words shift with the preset maximum length. In general, such module consists of n_{max} multiplexers, each of which has n_{max} inputs. As a result one value is obtained at the output of each multiplexer, according to the value of the counter. General diagram of this module is presented in Fig. 4.

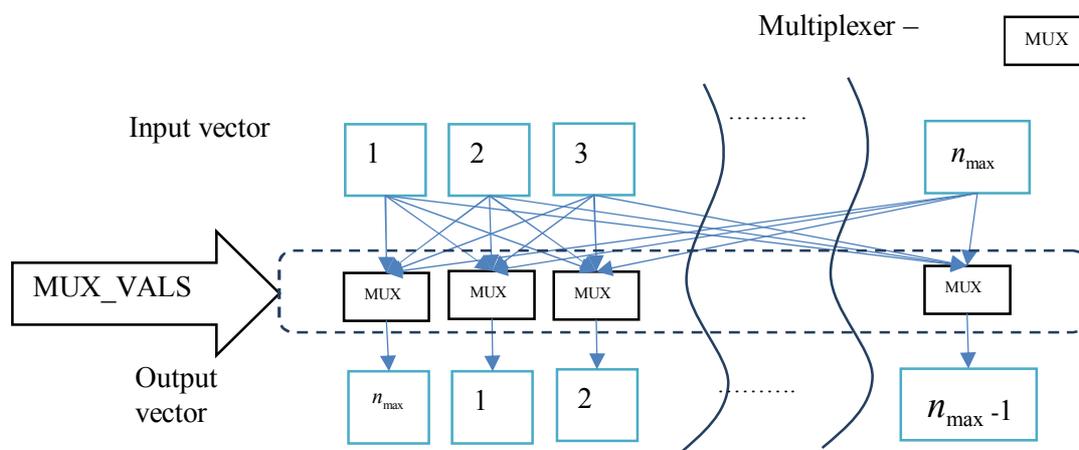


Fig. 4. Connection between input and output vectors in the barrel shifter

To select correctly the vector from the memory, the values of MUX_VALS addresses must be different. Calculation of the MUX_VALS value is performed due to the application of counters, which are specified the initial value, after that they start performing computations. After reaching the maximum value of the computation, which is determined by the length of the processed code, the transition to 0 value is performed and computation continues. In case, when the value of the counter is decremented after reaching 0 value, the transition to maximum value with further decrement occurs, if necessary.

Fig. 4 contains the simplified structure of the module, that does not take into account the fact, that the number of values to be shifted does not always correspond to the number of inputs.

For the suggested blocks of the decoder the realization on the language of circuit engineering

description VHDL is carried out. Simulation of their operation in the environment MentorGraphicsModelSim 10.1 c is carried out. The results of the simulation of the block of minima search are presented in Figs. 5 and 6.

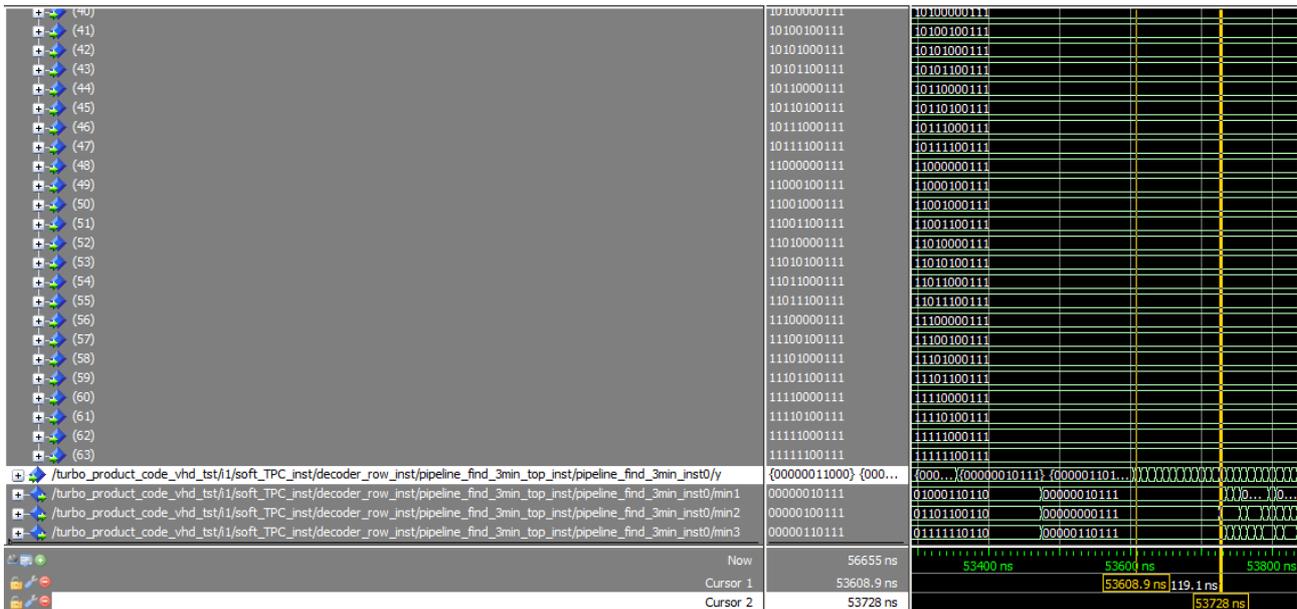


Fig. 5. Screenshot of the block of minima search operation simulation

Fig. 5 shows how the data are sent to the module and what data are obtained as a result. Vector, sent to the input, is divided into two parts. Each input value in the example contains 11 bits: seven are used for the position presentation, and the last four – for the value itself. Values, identical by the module, were sent to the input, that is why, as a result, 7 values were obtained. It is worth paying attention to the indices of the values, which were selected during the input data supply: they correspond to the positions 0, 1 and 3.

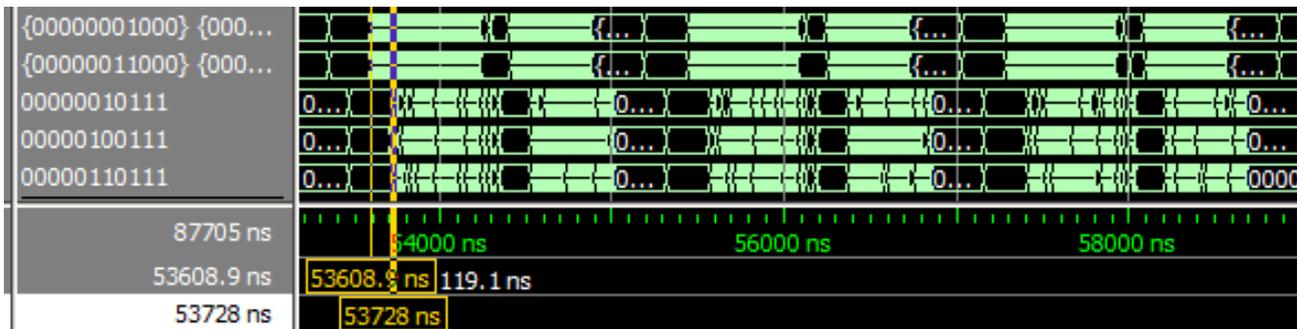


Fig. 6. Screenshot of the block of minimum search iterations simulation

Fig. 6 presents the performing of the simulation according to the iteration (scale is smaller as compared with Fig. 5). It shows that the block does not operate continuously but with the intervals. Accordingly, two such blocks for the processing of the rows and columns must be available.

By means of Dataflow tools in the environment ModelSim the following diagram of the minima search block on the level of the register logic is obtained (Fig. 7).

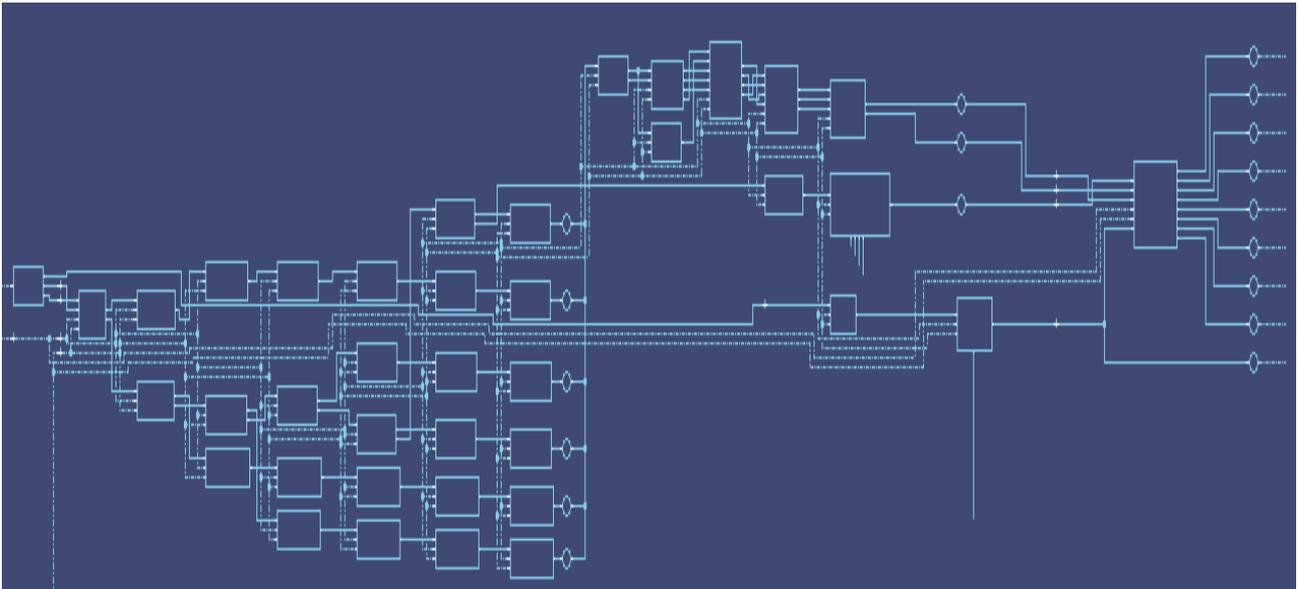


Fig. 7. Structural organization of minima search block

Automatic tools of similar circuits construction do not always generate the best variant of the structural diagram, however, it is sufficient to learn the internal structure of the block.

The realization of the universal barrel shifter for the codes with the basic length of 32, 64 and 128 bits is performed similarly.

Conclusions

The paper proposes and describes the principles of construction of separate blocks for the organization of TP-codes decoding by the reconfigured decoder. Main attention is paid to the block of minimal by the amplitude values in the code word positions search and universal barrel shifter. Methods of these blocks organization, which unlike the already known studies, allow to perform the processing of the code words of different length, are proposed. The suggested block of minima search is universal for codes processing with the preset maximal length. Repeated usage of the resources in the block of minima values search enables to increase the universality of the device. Due to this ability it becomes possible to perform the decoding of not only one set of codes, but several different codes simultaneously with the possibility of changing the settings for both horizontal and vertical code components. The characteristic features of their realization for the reconfigured decoder are studied. It was determined that such blocks can perform the processing for codes with different code word length and what rules should be kept during the development of these blocks to provide such a possibility. Main principles of the construction of the blocks of minima values search for the formation of the test vectors are determined. The structure of the universal barrel shifter and how it can perform its functions for the processing of the code words of different length is suggested. Due to the application of the suggested blocks the decoder obtains the possibility to perform correctly the processing of the code words for various codes. Simulation of the decoder operation, using ModelSim environment for the codes with basic length 32, 64, 128 bits and derivative codes was carried out. Testing of the decoder operation was performed on the board AlteraDE0-Nano, proposed for the studies by the company «Viacom» (Kyiv).

REFERENCES

1. Berrou C. Near Shannon limit error-correcting coding and decoding : Turbo-codes / C. Berrou, A. Glavieux, P. Thitimajshima // IEEE ICC'93. – 1993. – P. 1064 – 1071.
2. Chase D. A class of algorithms for decoding block codes with channel measurement information / D. Chase // IEEE Trans. Inform. Theory. – Jan. 1972. – Vol. IT-18. – P. 170 – 182.

3. Pyndiah R. Near-Optimum Decoding of Product Codes: Block Turbo Codes / R. Pyndiah // IEEE Transactions on Communications. – August 1998. – Vol. 46, No. 8. – P. 1003 – 1010.
4. Turbo product code decoder without interleaving resource: From parallelism exploration to high efficiency architecture / C. Leroux, C. Jego, P. Adde [et al.] // Journal of Signal Processing Systems, Springer. – № 64 (1). – 2011. – P. 17 – 29.
5. Megha S. Iterative Decoding Algorithm for Turbo Product Codes / S. Megha // International Journal of Innovative Research and Advanced Engineering (IJRAE). – April 2014. – Vol. 1, Issue 2. – P. 151 – 154.
6. Han J. High Speed Max-Log-MAP Turbo SISO Decoder Implementation Using Branch Metric Normalization / J. Han, E. Erdogan, T. Arslan // Proceedings of the IEEE Computer Society Annual Symposium on VLSI. – 2005. P. 173 – 178. DOI: 10.1109/ISVLSI.2005.37.
7. Mathana J. FPGA Implementation of High Speed Architecture for Max Log Map Turbo SISO Decoder / J. Mathana, Dr. Rangarajan // International Journal of Recent Trends in Engineering. – 2009. – Vol. 2, No. 6. – P. 142 – 146.
8. Krainyk Y. Low-complexity high-speed soft-hard decoding for turbo-product codes / Y. Krainyk, V. Perov, M. Musiyenko // 2017 IEEE 37th International Conference on Electronics and Nanotechnology (ELNANO). – 2017. – P. 471 – 474.
9. Hardware-Oriented Turbo-Product Codes Decoder Architecture / Y. Krainyk, V. Perov, M. Musiyenko [et al.] // Conference Proceedings of IEEE IDAACS-2017. – Bucharest, 2017. – P. 151 – 154.

Editorial office received the paper 02.03.2018.

The paper was reviewed 16.03.2018.

Krainyk Yaroslav – Cand. Sc. (Eng.), Doctoral fellow, Senior Lecturer with the Department of Computer Engineering, e-mail: codebreaker7@ukr.net, yaroslav.krainyk@chmnu.edu.ua.

Perov Vladislav – Post Graduate with the Department of Compute Engineering, e-mail: vlad.perov92@gmail.com.

Chernomorsk Petro Mokhyla National University.