http://dx.doi.org/10.5755/j01.eee.21.1.6657 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 21, NO. 1, 2015

Increasing the Data Transmission Robustness in Wsn Using the Modified Error Correction Codes on Residue Number System

Hu Zhengbing[1], Vasyl Yatskiv[2], Anatoliy Sachenko[2,3]
[1]*School of Educational Information Technology, Central China Normal University, No. 152 Louyu Road, 430079, Wuhan, China*
[2]*Research Institute for Intelligent Computer Systems, Ternopil National Economic University, 11 Lvivska St., 46020, Ternopil, Ukraine*
[3]*Department of Computer Science and Econometrics, Silesian University of Technology, Roosevelta St. 26-28, Zabrze, 41-800, Poland*
a_sachenko@ieee.org

***Abstract*—The WSN standard IEEE802.15.4 basically uses the unlicensed frequency of 2,4 GHz for data transmission in a variety of devices, standards and applications: IEEE802.11, Bluetooth and etc. In this paper we proposed modified correction codes – based on Residue Number System – to improve the data transmission robustness in WSN. These codes are characterized by high correction characteristics as well as the simplified coding procedure. There are given examples, and circuits of the both coder and decoder are designed for such modified correction codes using the Residue Number System and CPLD.**

***Index Term*—Wireless sensor networks, Error correction codes, Cyclic redundancy check codes, Residue number system.**

I. Introduction

Wireless Sensor Networks (WSN) are used in ecological, technical and medical monitoring systems [1]. WSN can control critical parameters of the industrial equipment such as a vibration, temperature, pressure enabling to avoid emergency cases. WSN have some advances including fast deployment, flexibility, self-organization and a possibility of the intelligent data processing.

The main constraint of WSN implementation is the low robustness of data transmission that is primarily caused by the high amount of obstacles in industrial enterprises. Therefore, increasing the robustness and decreasing the delay time of data transmission are fundamental issues for WSN. Retransmission or error correction codes are proposed to increase the robustness of WSN [2]. However, using packet retransmission over the channel with the high error probability causes decreasing the general capacity and cannot guarantee stringent requirements for time delay [3].

The check of data packets integrity in the WSN respectively to the IEEE 802.15.4 standard is made on the basis of Cyclic Redundancy Check (CRC) code. The frame check sequence (FCS) has capacity 16 bits and calculated over the MAC header (MHR) and MAC payload parts of the frame (Fig. 1). The FCS calculation is used on the basis of the standard polynomial generator with a degree 16 [4]

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1. \tag{1}$$

The CRC code simply detects the presence of an error in the blocks of data. In such case a request for the retransmission of distorted packet is sent.

The following requirements are established for WSN correction codes: low complexity of coding/decoding; low hardware requirements for algorithm implementation (microcontroller speed, memory capacity); adaptive change of check symbols number according to channel parameters.

Octets 2	1	4 to 20	n	2
Frame control	Sequence number	Addressing fields	Data payload	Frame check sequence
MHR			MAC payload	MFR
0 - 127 Octets				

Fig. 1. IEEE 802.15.4 Data Frame structure.

In [2], [5] the Reed Solomon codes are proposed for the error correction using in WSN. But there is a problem of changing (increasing/ decreasing) the check symbols number in those codes. For instance, if we need to turn from the number of check symbols $r = n - k$ to the bigger value $r' = n' - k$ we must re-count all check symbols [6].

Taking into account advantages of the Residue Number System (RNS), such as the independence, small capacity, equality of residues and the possibility to perform parallel mathematical calculations [7], it is expedient to study its implementation in correction codes.

II. Residue Number System Correction Codes

We should note the RNS is used now for solving the specialized problems only. It’s conditioned by transformation

Manuscript received March 10, 2014; accepted November 2, 2014.
This research was supported by China International Science and Technology Cooperation Project (CU01-11) / State Agency on Science, Innovation and Informatization of Ukraine, Ukrainian-Chinese project No. M/23/2013.

necessity of the binary code – which is implemented by universal computers – into RNS code and reverse transformation to submit information for the user.

There are developed some correction codes to detect and correct errors in RNS [7]. But for implementation of these codes in data transmission systems, particularly in WSN, it is necessary to convert data into Residue Number System that requires additional time and extra computing facilities.

Let's consider a system with the bases of $p_1, p_2, \dots, p_i, \dots, p_n$, where a number A is presented by residues

$$A = (b_1, b_2, \dots, b_i, \dots, b_n), \tag{2}$$

where $b_i = A\left(\bmod\ p_i\right)$, and let's introduce the additional check module $p_n < p_{n+1}$.

In the existing correction codes of RNS (R – RNS codes) the value of the check number [7]

$$b_{n+1} = A\left(\bmod\ p_{n+1}\right), \tag{3}$$

that is the calculation of check number requires the initial value of the number A.

There are the two existing approaches of errors correction using RNS correction codes [7], [8]:

– Calculation of the number projection with a followed correction and two possible modes:
1. If received number is greater than the working range then an error exists;
2. There are calculated the number projection and basic numbers by turn throw of modules. If the number projection fits the working range then a given module has the error.

– Renewing the positional representation of the number from received residues on the Chinese Theorem of Residues with the three possible modes:
1. If received number is greater than the working range then an error exists;
2. Residues are thrown by turn. If received positional representation of the number is correct then the error exists at the thrown residue;
3. A correct number can be received using the residue from the number positional representation.

Described steps above require essential computing costs that limit their implementation in WSN.

R – codes RNS have limitations imposed by the choice of increasing sequence of relatively prime modules. It is necessary for test modules to have identical digits. Otherwise, there is a problem of efficient maintaining of check symbols. Therefore authors proposed modified correction codes based on RNS and described below.

III. MODIFIED CORRECTION CODES ON RNS

In the proposed modified correction codes on RNS (M – RNS codes) the data packet in binary code must be transmitted and divided into blocks of the equal length (quaternion or bytes) [9]

$$\begin{gathered} a_j^1 \dots a_2^1\ a_1^1\ a_0^1,\ a_j^2 \dots a_2^2\ a_1^2\ a_0^2, \dots \\ \dots, a_j^i \dots\ a_2^i\ a_1^i\ a_0^i, \dots, a_j^k \dots a_2^k\ a_1^k\ a_0^k, \end{gathered} \tag{4}$$

where a^i is the binary digit, $j = 4, 8$.

At that the value of check number

$$\begin{gathered} \bar{b}_{k+1} = (v_1 \times b_1 + v_2 \times b_2 + \dots \\ \dots + v_i \cdot b_i + \dots + v_k \cdot b_k)\left(\bmod\ p\right), \end{gathered} \tag{5}$$

where v_i are coprimes coefficients of a linear form with p; b_i is the byte of data in binary or decimal system

$$\begin{gathered} b_i = a_7^i \dots a_3^i\ a_2^i\ a_1^i\ a_0^i = \\ = a_0 \times 2^0 + a_1 \times 2^1 + a_2 \times 2^2 + \dots + a_7 \times 2^7. \end{gathered} \tag{6}$$

Let's assume, that in the process of data transmission the error occurs in one of the data blocks (the bit stream from 1 to 8 binary digits was distorted), and instead A we get

$$A' = (b_1, b_2, \dots, b_i', \dots, b_k, \bar{b}_{k+1}). \tag{7}$$

Then we calculate the value of the check number

$$\bar{b}_{k+1}' = (v_1 \times b_1 + \dots + v_i \times b_i' + \dots + v_k \times b_k \left(\bmod\ p\right). \tag{8}$$

The difference between check symbols can be expressed as $\text{¢} = \bar{b}_{k+1}' - b_{k+1}$. So, if $\text{¢} = 0$ then there is no error; if $\text{¢} \neq 0$ then the error occurs.

To detect the error in any data block it is necessary to provide a condition that every value ¢ corresponds to one error value. To satisfy this condition the coefficients of linear form must be coprime with p and the value of check module must be equal to $p \geq 2^m$, where m – is the number of binary digits in the selected block (if we divide the packet into bytes then $m = 8$).

To estimate the effectiveness of the correction code it is necessary to know the correlation between redundancy and error detection or error correction possibilities.

The minimal code distance of the modified RNS code is expressed as $d_{\min} = n - k + 1$, where n is the code length, k is the number of data blocks. The modified RNS code can detect therefore the all t errors under condition $d_{\min} \geq t + 1$; if $d_{\min} \geq 2 \cdot t + 1$ then this code can correct the all t errors. To provide the assured error correction in one data block we need to introduce the two check numbers.

If we introduce the two or more check numbers, we can select:
1. Equal check modules and different coefficients of the linear form;
2. Different check modules and equal coefficients of the linear form.

Values of the linear form coefficients – which are required for calculating the check numbers – are calculated according to (5) and illustrated in the Table I. Since modules p_i are selected to be equal the coefficients must be different ones as well as be coprimes with p_i . If this condition is broken then values of check numbers will be the same, i.e. $b_{k+i} = b_{k+j}$, $i \neq j$ for different data.

TABLE I. THE VALUES OF THE COEFFICIENTS OF THE MODIFIED RNS CODES.

Check numbers b_{k+j}	**Coefficients** $v_1 - v_{17}$	**Module** p_i
b_{k+1}	7, 71, 151, 239, 13, 79, 163, 251, 17, 83, 167, 59, 67, 137, 227, 103, 41	256
b_{k+2}	11, 73, 157, 89, 241, 19, 173, 97, 41, 109, 197, 113, 103, 163, 47, 127, 79	
b_{k+3}	13, 79, 163, 251, 23, 97, 179, 103, 43, 113, 199, 41, 107, 167, 17, 173, 101	
b_{k+4}	17, 83, 167, 103, 29, 101, 181, 71, 47, 127, 211, 239, 41, 79, 151, 13, 67	
b_{k+5}	19, 89, 103, 173, 163, 37, 7, 71, 151, 239, 17, 79, 157, 149, 43, 181, 59, 113	
b_{k+6}	23, 97, 107, 179, 59, 29, 73, 157, 241, 19, 71, 239, 151, 61, 229, 11, 131	
b_{k+7}	29, 101, 241, 59, 181, 271, 73, 163, 89, 251, 23, 67, 43, 41, 19, 109, 137	
b_{k+8}	31, 79, 103, 113, 191, 57, 29, 101, 41, 193, 241, 97, 239, 19, 67, 13, 139	

Let's consider an example of error detection using the modified RNS code. We select as a check module $p = 16$; coefficients of the linear form $v_1 = 3$, $v_2 = 5$, $v_3 = 7$; data blocks in the decimal system: $b_1 = 2$, $b_2 = 10$, $b_3 = 9$.

The value of check number

$$\bar{b}_{k+1} = (3\times 2 + 5\times 10 + 7\times 9)\,(\mathrm{mod}\ 16) = 7. \tag{9}$$

Let's introduce the error in the first block $b_1' = 4$, while the value of check number

$$\bar{b}'_{k+1} = (3\times 4 + 5\times 10 + 7\times 9)\,(\mathrm{mod}\ 16) = 13. \tag{10}$$

Thus misalignment value $¢ = \bar{b}'_{k+1} - b_{k+1} = 13 - 7 = 6$, so the error is detected.

Experiments for the check module value $P \in 2^3 \div 2^8$ were executed. It was established the modified RNS code with one check module can provide the 100 % of error detection in one data block, and the 87 % in two data blocks. Moreover the correction capabilities depend on check module also.

IV. Error Correction

Let's use the concept of an alternative set for errors correction [10]. The alternative set for modified RNS codes is the set of data blocks

$$\Theta = \left(b_{¢}^1, b_{¢}^2, \ldots, b_{¢}^k\right). \tag{11}$$

The error in position of each of them can lead to some misalignment value $¢_i$ which is determined by the condition $0 < ¢_i < p_{k+1}$. The error value $x_{a_i}^i$ can be determined per each b_i block (Table II).

TABLE II. ERROR VALUE PER EACH DATA BLOCK.

Misalignment value: $¢$	**Data block:** b_i					
	b_1	b_2	$\ldots$	b_i	$\ldots$	b_k
1	$x_{b_1}^1$	$x_{b_2}^1$	$\ldots$	$x_{b_i}^1$	$\ldots$	$x_{b_n}^1$
2	$x_{b_1}^2$	$x_{b_2}^2$	$\ldots$	$x_{b_i}^2$	$\ldots$	$x_{b_n}^2$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$p_k - 1$	$x_{b_1}^{p_k-1}$	$x_{b_2}^{p_k-1}$	$\ldots$	$x_{b_i}^{p_k-1}$	$\ldots$	$x_{b_n}^{p_k-1}$

According to [2] the misalignment value

$$\begin{aligned} ¢ = \bar{b}'_{k+1} - b_{k+1} = (v_1\times b_1 + v_2\times b_2 + \ldots + v_i b_i' + \\ + \ldots + v_k \times b_k)\,(\mathrm{mod}\ p_{k+1}) - (v_1\times b_1 + v_2\times b_2 + \ldots + \\ + v_i b_i' + \ldots + v_k b_k)\,(\mathrm{mod}\ p_{k+1}) \ . \end{aligned} \tag{12}$$

When the error (data distortion) occurs in the data block the misalignment value $b_i' = b_i + x$, where x is the error value. The error value depends on data capacity m and can be expressed by the condition: $-(2^m - 1) \le x \le 2^m - 1$.

Let's find the misalignment value at error in the data block b_i using (12)

$$¢_i = v_i \times x\ (\mathrm{mod}\ p_{k+1}). \tag{13}$$

To provide the unambiguous correction of the single error (error in one data block) it is necessary to provide a condition: the alternative set Θ consists of one data block per each misalignment value. Then the value of the check module

$$p \ge 2\times n \times \left(2^m - 1\right), \tag{14}$$

where n is a number of data blocks.

Let's consider the example of alternative sets usage for errors localization. Let the data consist of three blocks with the capacity of three bits each. Coefficients of the linear form can be found experimentally (using designed software):$\neg_1 = 54$, $\neg_2 = 25$, $\neg_3 = 67$. According to (14) the minimal theoretical value of the check module $p_{k+1} \ge (2\cdot 3\cdot (2^3 - 1)) = 42$. It's established experimentally (using designed software) that minimal value of the check module $p_{k+1} = 73$ allowing to get different misalignments. Then according to (12) the misalignment

$$\begin{aligned} ¢_i = \bar{b}'_{k+1} - \bar{b}_{k+1} = \\ = \left(54\times b_1' + 25\times b_2 + 67\times b_3\right)(\mathrm{mod}\ 73) - \\ - \left(54\times b_1 + 25\times b_2 + 67\times b_3\right)(\mathrm{mod}\ 73), \end{aligned} \tag{15}$$

and misalignments per each data block are equal to:

$$\begin{cases} \text{¢}_{1i} = 54 \times x_{b_i} \pmod{73}, \\ \text{¢}_{2i} = 25 \times x_{b_i} \pmod{73}, \\ \text{¢}_{3i} = 67 \times x_{b_i} \pmod{73}. \end{cases} \tag{16}$$

Misalignments values are calculated (Table III) in a case when the error appears in the one date block only.

TABLE III. MISALIGNMENT VALUES FOR ALL POSSIBLE ERRORS.

Error values: x_i	Misalignment value: ¢_{1i}	Misalignment value: ¢_{2i}	Misalignment value: ¢_{3i}
1	54	25	67
2	35	50	61
3	16	2	55
4	70	27	49
5	51	52	43
6	32	4	37
7	13	29	31
-1	19	48	6
-2	38	23	12
-3	57	71	18
-4	3	46	24
-5	22	21	30
-6	41	69	36
-7	60	44	42

As it can be seen from the Table III misalignment values at the 3 columns are so different, that enables to detect and correct the error in any data block.

V. THE EXPERIMENTAL RESEARCH OF MODIFIED CORRECTION CODES

Modified RNS error correction codes are proposed to be used for increasing the data transmission robustness in WSN.

It's reasoned in [2] that most appropriate to store the check numbers from all blocks in the end of the packet. Such check numbers location enables to receive data packets by wireless sensors which don't use the additional RNS coding scheme.

Let's divide data (MHR field and MAC payload) into five blocks (D1 – D5) for the implementing the RNS error correction code at the MAC layer (Fig. 2). The code length $n_i = 25$ bytes, the data block number $k_i = 17$ bytes. The check blocks R1 – R5 are formed into a corresponding data block and stored after the last data block.

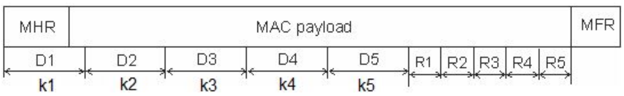

Fig. 2. The data frame structure using coding.

After receiving the packet we use the CRC code to check the packet integrity. If there is no error then we don't use the RNS code for the additional check. If the error is detected then we use the modified RNS code to correct the error.

Encoding time was studied in the Matlab environment for the following codes: RNS correction code (R – RNS codes), RNS modified correction code (M – RNS codes) and Reed Solomon code RS (127, 87) with parameters: $n = 25$, $k = 17$, $r = 8$, $t = 4$. The results are given in Table IV.

As it can be seen from Table IV, the correction M - RNS code allows to reduce the encoding time in five times in comparison with the correction R- RNS code and Reed Solomon RS (127, 87) code at the about same code rate.

TABLE IV. THE PARAMETERS OF THE CODES.

Code	Encoding time (ms)	Code rate, $R = k / n$
R – RNS code	117,5	0,64
M – RNS code	22,5	0,68
RS (127, 87)	109,7	0,68

WSN belong to systems of the digital communication. Those systems are used, as the effectiveness criterion, a formula E_b / N_0 , where E_b is the bit energy, N_0 is the spectral density of chaos power. Using the noiseless coding [11] we may determine the coding effectiveness

$$G = \left(\frac{E_b}{N_0}\right)_u - \left(\frac{E_b}{N_0}\right)_c \ (dB), \tag{17}$$

where $(E_b / N_0)_u$ and $(E_b / N_0)_c$ is a required value E_b / N_0 without coding and with coding correspondingly.

Estimation of the coding effectiveness is executed in Communications Toolbox 3.0, Matlab using the graphical environment BERTool.

Graphical curves (Fig. 3) are obtained for the modulation Offset Quadrature Phase-shift keying (OQPSK), which is used in the standard IEEE802.15.4, and the proposed correction code with a minimal coding distance $d_{\min} = 9$.

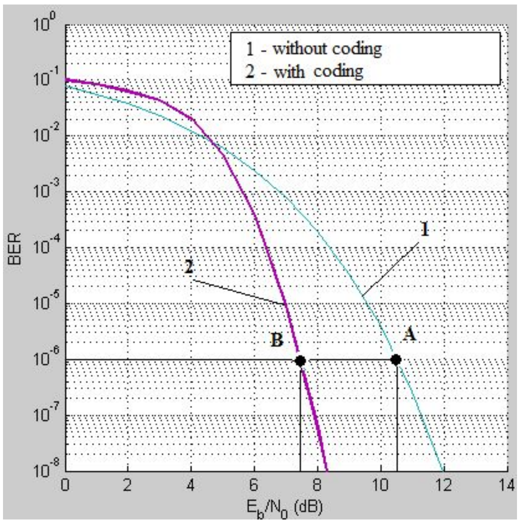

Fig. 3. Comparison of the data transmission veracity using a circuit without coding and with coding: BER – a probability of the bit error.

As it can be seen from a Fig. 3 (points A and B) the use of noiseless coding allows to reduce a relation E_b / N_0 from 10.5 dB to 7.5 dB at retaining the given veracity of the data transmission, i.e., the coding effectiveness equals 3 dB.

VI. THE ENCODER/DECODER HARDWARE FOR THE MODIFIED CORRECTION CODE BASED ON RNS

To reduce a load on the wireless controller we propose to implement the encoder/decoder on CPLD. For this purpose (5) can be written in the form

$$\begin{aligned} \bar{b}_{k+1} = (v_1 \times b_1 \pmod{p} + \ldots + v_i \times b_i \pmod{p} + \\ + \ldots + v_k \times b_k \pmod{p}) \pmod{p}. \end{aligned} \tag{18}$$

On the base of the expression (18) a general scheme of the encoder is synthesized (Fig. 4).

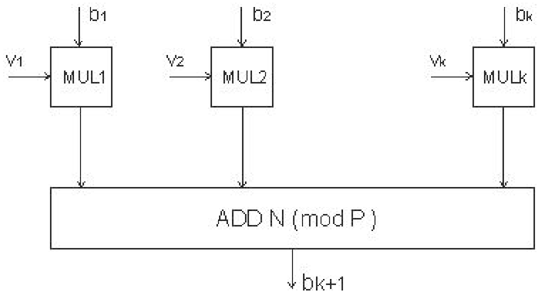

Fig. 4. Encoder scheme: MULi – the modular p multiplication blocks; ADD – the modular p addition block.

Encoder/decoder for the RNS Modified Correction codes with the three data blocks and the one check number is implemented on CPLD (Fig. 5).

The encoder scheme is described in the Verilog language within the environment of Quartus II and it's implemented on Altera CPLD, the integrated circuit family MAX II, EPM240T100C3. There are used the following hardware resources: the total logic elements 141/240 (59 %). The maximum delay of forming the check number is 25 ns. It has established experimentally the encoder implementation - based on (18) – decreases the delay of the check number calculation approximately in 30 %, due to the parallel operation of the modular p multiplication blocks.

The decoder consists of the two units: the unit of error detection and the unit of data correction (see Fig. 5). In the first unit the check numbers are formed and misalignment s is calculated on base of received data. Then, in the second unit, the data correction is running per each data block on the base of misalignment tables [10].

The decoder is implemented on Altera CPLD, Family MAX II, Device EPM570T100C3. There are used the following hardware resources: the total logic elements 334/570 (59 %). The maximum delay of the check number calculation is 49 ns.

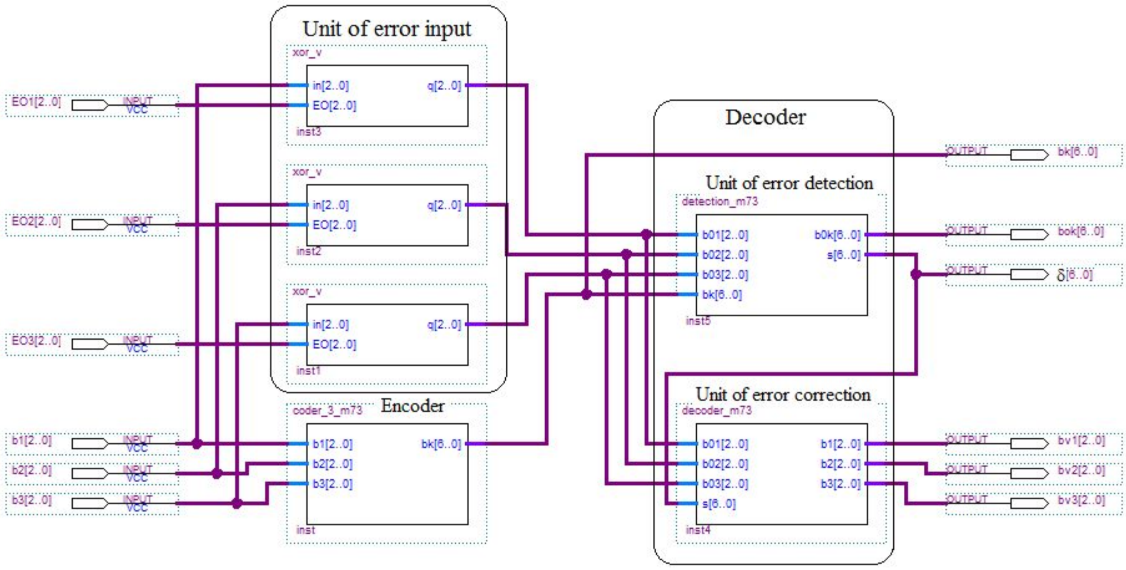

Fig. 5. CPLD implementation of the encoder/decoder for RNS modified correction codes: EO – input for error application; b1, b2, b3 – data inputs; bv1, bv2, bv3 – data outputs; s – misalignment; bk – received check number; b0k – calculated check number.

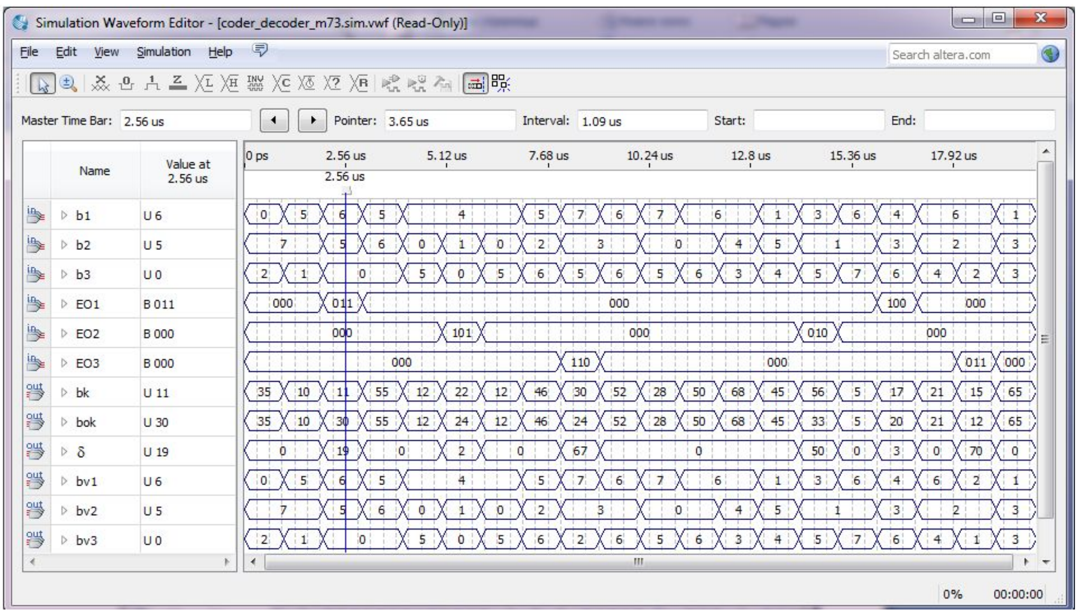

Fig. 6. Timing charts of encoder/decoder running.

The unit of error input is designed for the encoder/decoder verification enabling to insert errors in any of data block. For simulating the data corruption it's necessary to introduce the error vector in the EO input of the appropriate channel (see Fig. 5). A singular bit of the error vector converts the appropriate digit in the data block.

Executed simulation and verification of designed devices confirmed their proper running which is illustrated by timing charts (Fig. 6). To make more compact the representation values the error vector is given in binary notation, and the rest of values are given in decimal notation. For example let data distortion occurred in the block b2: $b2' = b2 \oplus e = 001 \oplus 101 = 100$, misalignment $¢_{2i} = 2$ at the error $e = 101$ of EO2 output (see Fig. 6). So according to Table III the misalignment $¢_{2i} = 2$ corresponds to the error $x_i = 3$, that is, for correcting the error it's necessary to provide: $b2 = b2' - x = 4 - 3 = 1$.

VII. Conclusions

Authors designed the modified correction codes on Residue Number System which allow to reduce the encoding time in five times in comparison with Correction R-code based on Residue Number System and Reed Solomon codes RS (127, 87) at about same code rate. Moreover proposed codes enable to decrease a relation E_b / N_0 in 3 dB at retaining the given veracity of the data transmission.

Authors developed the CPLD encoder/decoder for the modified correction codes based on Residue Number System which provide the correction of single errors (errors in the one block). Encoder and decoder schemes are described in the Verilog language within environment Quartus II and implemented on Altera CPLD, MAX II integrated circuit. There are used the following hardware encoder resources: the total logic elements equal 141 and the maximum delay of forming the check number is 25 ns. Hardware decoder resources are: the total logic elements - 334 and the maximum delay of check number calculation – 49 ns.

It should be noted the modified Residue Number System codes simplify the calculation of the check numbers. As a result we have the reduction of computational consumptions, which is very important for its implementation in self-contained power supply devices.

Proposed approach can be used for implementation of encoder/decoder with other code characteristics, in this case it's necessary to change check numbers and the number of data units in the correspondent block.

Taking into account low hardware resources for the implementing the encoder/decoder – based on modified correction Residue Number System codes – the developed devices can be used in wireless sensor networks for increasing the data transmission robustness.

VIII. Future Work

In the future we plan to implement the encoding/decoding algorithms in the wireless microcontrollers JN5148 and investigate their hardware and software complexity.

References

[1] I. F. Akyildiz, W. Su, Y. Sapnkarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", *Comput. Netw.*, vol. 38, pp. 393–422, 2002. [Online]. Available: http://dx.doi.org/10.1016/S1389-1286(01)00302-4

[2] Kan Yu, M. Gidlund, J. Akerberg, M. Bjorkman, "Reliable and low latency transmission in industrial wireless sensor networks", *Procedia Computer Science*, vol. 5, pp. 866–873, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2011.07.120

[3] A. Sikora, V. Groza, "Coexistence of IEEE 802.15.4 with other systems in the 2.4 ghz-ism-band", in *Proc. IEEE 3 Instrumentation and Measurement Technology Conf., (IMTC 2005)*, 2005, pp. 1786–1791.

[4] IEEE Standard for Part 15.4: Wireless Medium Access Control Layer (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std 802.15.4™-2006.

[5] M. Vuran, I. Akyildiz, "Cross-layer analysis of error control in wireless sensor networks", in *3rd Annual IEEE Communications Society Sensor and Ad Hoc Communications and Networks, (SECON 2006)*, 2006, pp. 585–594.

[6] V. Varhauzin, "Antinoise coding in the networks with packet switching", *TeleMultiMedia*, no. 9, pp. 10–16, 2005. (in Russian).

[7] I. Y. Akushskiy, D. I. Yuditskiy, *Machine Arithmetics in Residue Number System*, Moscow, Soviet Radio, p. 460, 1968. (in Russian).

[8] V. T. Goh, M. U. Siddiqi, "Multiple error detection and correction based on redundant residue number systems", *IEEE Trans. on Communications*, vol. 56, no. 3, pp. 325–330, 2008. [Online]. Available: http://dx.doi.org/10.1109/TCOMM.2008.050401

[9] V. Yatskiv, N.Yatskiv, Su Jun, A. Sachenko, Hu Zhengbing, "The use of modified correction code based on residue number system in WSN", in *Proc. 7-th IEEE Int. Conf. Intelligent Data Acquisition and Advanced Computing Systems, (IDAACS 2013)*, Berlin, Germany, 2013, vol. 1, pp. 513–516.

[10] I. Y. Akushskiy, I. T. Pak, "Questions of error-correcting coding position-independent code", *Problems of Cybernetics*, vol. 28, pp. 36–56, 1977. (in Russian)

[11] William Stallings, *Wireless Communications and Networking*. Upper Saddle River, NJ: Prentice Hall, 2002.