Abstract
This work presents the coder and decoder of block 7B8B simplified with auxiliary channel. The coder 7B8B converts an input 7 bits word in an output 8 bits word. It transmits approximately an equal number of 1's and 0's to provide a DC constant component. It increases the transitions number, improves the system quality and security. The objective is also to improve the system potentialities with an auxiliary channel to monitor the communication (alarm.) The main channel is real, but the auxiliary channel is fictitious.

Keywords: Block Codes, Advanced digital systems, Transmission lines

1. Introduction
This work studies the coder and decoder of block 7B8B simplified with auxiliary channel [1–10].

The auxiliary channel (CA) does not consume physical resources. It is only achieved by programming the appropriate mBnB code in the PROMs (programmable ROM) of the encoder and decoder.

The code must satisfy the rule $n=m+1$ (m, n integers), to minimize the transmission elevation $txo = txi \times n/m$. Also, n must be even to guarantee a constant DC component with many transitions. Also, m must be low to minimize the system complexity.

The coder converts a seven bits word 7B in a word 8B. Posteriorly, the decoder converts newly the eight bits word 8B in the original word 7B, recovering the initial sequence.

Fig.1 shows the communication system with relevance for the blocks coder and decoder 7B8B.

The coder improves certain characteristics such as:

- Constant DC component what avoids signal fluctuation.
• Sufficient transitions for clock recovery and retiming.
• Independent of the bits sequence.
• Decodification independent of the state.
• Multiplication of errors is low.
• Possibility of errors detection.
• Information for alignment of blocks / words.

In emitter, the data source provides the data, the scrambler becomes the spectrum input independent, the line coder equalizes the number of 1's and 0's, the signal emission adapts the signal to the channel type.

In receiver, the signal receiver provides the electric signal, the synchronizer recoveries the clock, the decoder makes the inverse of the coder, the descrambler makes the inverse of the scrambler and the destination is the final.

The code 7B8B with auxiliary channel is the same normal code 7B8B, where only one word 7B is coded differently, when the CA is zero (CA=0) or one (CA=1).

We choose, for example, the word 7B number (127). Then, it is necessary that two forbidden words belong to the dictionary and codify the situation CA = 1. We choose, for example, the words 01111110, 10000001 (Tab.1).

| word | +4 Disp. | +2 Disp. | 0 Disp. (8B) | -2 Disp. | -4 Disp. |
|------|----------|----------|-------------|----------|----------|
| 0-   |          |          |             |          |          |
| 1-   |          |          |             |          |          |
| 2-   |          |          |             |          |          |
| 3-   |          |          |             |          |          |
| 4-   |          |          |             |          |          |
| 5-   |          |          |             |          |          |
| 6-   |          |          |             |          |          |
| 7-   | 00101011 |          |             |          |          |
| 8-   | 00101101 |          |             |          |          |
| 9-   | 00110011 |          |             |          |          |
| word | +4 Disp. | +2 Disp. | 0 Disp. (8B) | -2 Disp. | -4 Disp. |
|------|---------|---------|--------------|----------|---------|
| 10-  | 00110101|
| 11-  | 00110110|
| 12-  | 00111001|
| 13-  | 00111010|
| 14-  | 00111100|
| 15-  | 01000111|
| 16-  | 01001011|
| 17-  | 01001101|
| 18-  | 01001110|
| 19-  | 01010011|
| 20-  | 01010101|
| 21-  | 01010110|
| 22-  | 01011001|
| 23-  | 01011010|
| 24-  | 01011100|
| 25-  | 01100011|
| 26-  | 01100101|
| 27-  | 01100110|
| 28-  | 01101001|
| 29-  | 01101010|
| 30-  | 01101100|
| 31-  | 01110001|
| 32-  | 01110010|
| 33-  | 01110100|
| 34-  | 01111000|
| 35-  | 10000111|
| 36-  | 10001011|
| 37-  | 10001101|
| 38-  | 10001110|
| 39-  | 10010011|
| 40-  | 10010101|
| 41-  | 10010110|
| 42-  | 10011001|
| 43-  | 10011010|
| 44-  | 10011100|
| 45-  | 10100011|
| 46-  | 10100101|
| 47-  | 10100110|
| 48-  | 10101001|
| word | +4 Disp. | +2 Disp. | 0 Disp. (8B) | -2 Disp. | -4 Disp. |
|------|----------|----------|--------------|----------|----------|
| 49-  |          |          | 10101010     |          |          |
| 50-  |          |          | 10101100     |          |          |
| 51-  |          |          | 10110001     |          |          |
| 52-  |          |          | 10110010     |          |          |
| 53-  |          |          | 10110100     |          |          |
| 54-  |          |          | 10111000     |          |          |
| 55-  |          |          | 11000011     |          |          |
| 56-  |          |          | 11000101     |          |          |
| 57-  |          |          | 11000110     |          |          |
| 58-  |          |          | 11001001     |          |          |
| 59-  |          |          | 11001010     |          |          |
| 60-  |          |          | 11001100     |          |          |
| 61-  |          |          | 11010001     |          |          |
| 62-  |          |          | 11010100     |          |          |
| 63-  |          |          | 11011000     |          |          |
| 64-  |          |          | 11100001     |          |          |
| 65-  |          |          | 11100100     |          |          |
| 66-  |          |          | 11101000     |          |          |
| 67-  |          |          | 11110000     |          |          |
| 68-  | 0001111 | 00000111   | 00001011     |          |          |
| 69-  |          |          | 00001011     |          |          |
| 70-  | 0010111 | 00001011   | 00001011     |          |          |
| 71-  |          |          | 00001011     |          |          |
| 72-  | 0011011 | 00001101   | 00001101     |          |          |
| 73-  |          |          | 00001101     |          |          |
| 74-  | 0011101 | 00010011   | 00010011     |          |          |
| 75-  |          |          | 00010011     |          |          |
| 76-  | 0100111 | 00011010   | 00011010     |          |          |
| 77-  |          |          | 00011010     |          |          |
| 78-  | 0101111 | 00011010   | 00011010     |          |          |
| 79-  |          |          | 00011100     |          |          |
| 80-  | 0101110 | 00100011   | 00100011     |          |          |
| 81-  |          |          | 00100011     |          |          |
| 82-  | 0110111 | 00101010   | 00101010     |          |          |
| 83-  |          |          | 00101010     |          |          |
| 84-  | 0110110 | 00101010   | 00101010     |          |          |
| 85-  |          |          | 00101010     |          |          |
| 86-  | 0111011 | 00110001   | 00110001     |          |          |
| 87-  |          |          | 00110001     |          |          |
| word  | +4 Disp. | +2 Disp. | 0 Disp. (8B) | -2 Disp. | -4 Disp. |
|-------|----------|----------|--------------|----------|----------|
| 88-   | 01110001 |          |              | 00110100 |          |
| 89-   | 01110100 |          |              | 00111000 |          |
| 90-   | 01111000 |          |              | 01000011 |          |
| 91-   | 10001111 |          |              | 01000101 |          |
| 92-   | 10010111 |          |              | 01000110 |          |
| 93-   | 10011011 |          |              | 01001001 |          |
| 94-   | 10011101 |          |              | 01001010 | 01001100 |
| 95-   | 10011110 |          |              | 01001100 |          |
| 96-   | 10100111 |          |              | 01010001 |          |
| 97-   | 10101011 |          |              | 01010010 |          |
| 98-   | 10101101 |          |              | 01010100 |          |
| 99-   | 10110010 |          |              | 01011000 |          |
| 100-  | 10110111 |          |              | 01100001 |          |
| 101-  | 10111011 |          |              | 01100010 |          |
| 102-  | 10111101 |          |              | 01110000 |          |
| 103-  | 10111110 |          |              | 10000011 |          |
| 104-  | 10111100 |          |              | 10000111 |          |
| 105-  | 10110101 |          |              | 10001001 |          |
| 106-  | 10101101 |          |              | 10001010 | 10001100 |
| 107-  | 10101111 |          |              | 10001110 |          |
| 108-  | 10100111 |          |              | 10010010 |          |
| 109-  | 10100110 |          |              | 10010100 |          |
| 110-  | 11010001 |          |              | 10100110 |          |
| 111-  | 11010011 |          |              | 10100100 |          |
| 112-  | 11010101 |          |              | 10100001 |          |
| 113-  | 11010110 |          |              | 10110000 |          |
| 114-  | 11010100 |          |              | 10110100 |          |
| 115-  | 11011000 |          |              | 10110001 |          |
| 116-  | 11010011 |          |              | 10100010 |          |
| 117-  | 11001011 |          |              | 10100100 |          |
| 118-  | 11000011 |          |              | 10101000 |          |
| 119-  | 11000101 |          |              | 10110000 |          |
| 120-  | 11001010 |          |              | 11000001 |          |
| 121-  | 11001100 |          |              | 11000001 |          |
| 122-  | 11100001 |          |              | 11000010 |          |
| 123-  | 11100010 |          |              | 11000100 |          |
| 124-  | 11100100 |          |              | 11001000 |          |
| 125-  | 11100000 |          |              | 11000000 |          |
| 126-  | 00111111 |          |              | 00000011 |          |
From a total of 256 words 8B, we use 186 +2 words for the dictionary (+4D=3, +2D=56, 0D=70, -2D=56, -4D=3). This is, all the 0 disparity words (70), all the +2 and -2 disparity words (56 + 56) and only three +4 and -4 disparity words (3 + 3). We opted by the two (+one) words +4D (63- 00111111, 252-11111100, 126- 01111110 AC) and -4D (3- 00000011, 192- 11000000, 129- 10000001). So, 188=70+2*56+2*2+2*1.

The remaining (70- 2) prohibited words (out of dictionary) are [0, 1, 2, 4, 5, 6, 8, 9, 10, 12; 16, 17, 18, 20, 24, 32, 33, 34, 36, 40, 48; 64, 65, 66, 68, 72, 80, 95, 96, 111, 119, 123, 125, (withdrawal 126), 127; 128, (withdrawal 129), 130, 132, 136, 144, 159, 160, 175, 183, 187, 189, 190, 191; 207, 215, 219, 221, 222, 223, 231, 235, 237, 238, 239, 243, 245, 246, 247, 249, 250, 251, 253, 254, 255].

To guarantee a null DC component is necessary after a +2 disparity word to send a -2 disparity word, although between them can appear zero disparity words 0D (don’t affects DC) and vice versa. Also, after a +4 disparity word it must send a -4 disparity word, although between them can appear zero disparity words 0D and vice versa.

The previous table needs to discriminate 5 disparity types (+4, +2, 0, -2, -4), what complicates the implementation.

But, only two words 7B (126, 127-(127)) needs to be codified with words 8B of disparity (+4, - 4). So, 126 (00111111, 11000000) and 127 (11111100, 00000011). These only 4 words change the DC component very little.

Then, we can consider the words +4D as being +2D and the words -4D as -2D, so we have only 3 types of disparity (+2D, 0D, -2D) and the previous table is much simplified.

Alternatively, in the simplification table, we could have opted to code the 3 words 7B (126, 127, (127)) directly as three words 8B (00111111, 11000000, (01111110)) and include them in the words of null disparity 0D.

The above simplified table produces the followings tables of codification and decodification. The 7B8B code simplification becomes it similar with the 3B4B and 5B6B.

### 2. Tables of Codification and Decodification

With the code 7B8B auxiliary, we can obtain the table of codification 7B8B and the table of decodification 7B8B.
A DC constant component implies to send an equal number of 1's and 0's. Then, after to send a positive disparity word, ignoring the 0 disparity word, is necessary to send a negative word and vice-versa. The 7B word is converted in the corresponding 8B word (Tab. 3).

This table is programmed in the PROM (Programmable Read Only Memory) of the coder.

**2.2. Table of decodification (PROM - Decod)**

The table of decodification must be programmed in the PROM of the decoder. The decoder 7B8B inverted can recover newly the original words 7B (Tab 4).

The total words are 70+ 68+ 56+ 56+ 2+ 2+ 1+1 = 256.

The words with don’t care (11) are forbidden, can be associated with for example (11 - - - -) ≡ (11 0101010).

**3. Pair Coder and Decoder 7B8B**

To make the codification and decodification is necessary to project (create) the hardware of the coder and decoder.

With the previous table simplification, the state diagrams of the coder and decoder 7B8B are equal to the 3B4B and 5B6B, what facilitates extremely the project.
3.1. Coder 7B8B

The codification table is programmed in a PROM, having a flip flop that guards the information of the parity P (positive or negative) of the last transmitted word, according the status diagram of Fig.2.
**TABLE 4**: Table of decodification 7B8B aux (PROM-Dec)

| 8B word $B_i$... $B_0$ | +2 Disp. $C_iC_{i+1}P_iP_{i+1}A_iA_0$ | 0 Disp., $\delta D$ $C_iC_{i+1}P_iP_{i+1}A_iA_0$ | -2 Disp. $C_iC_{i+1}P_iP_{i+1}A_iA_0$ |
|------------------------|----------------------------------|----------------------------------|----------------------------------|
| 0- 0 0...00            | 56w 7                            | 0011- 70w 7                      | 56w7                             |
| 1- 0 0...01            | (+2D)                            | 0011- (0D)                       | (-2D)                            |
| 2- 0 0...10            |                                  | 0011-                            |                                  |
| ...                   | +2w7(+4D)                        | +6w7 Pro)                        | +2w7(-4D)                        |
| 126-0 1...10           | 1010- ‘1’ S                      |                                  |                                  |
| 127-0 1...11           |                                  | 0011-                            |                                  |
| 128-1 0...00           |                                  | 0011-                            |                                  |
| 129-1 0...01           |                                  |                                  | 1001- ‘1’ S                      |
| 130-1 0...10           |                                  | 0011-                            |                                  |
| ...                   | +1w7(+4D)                        | 0000                             | +1w7(-4D)                        |
| 192-11...00            |                                  | 0110- ‘0’                        |                                  |
| ...                   |                                  |                                  |                                  |
| 255-11...11            |                                  |                                  | 0011-                            |

**Figure 2**: States diagram of the codification (memory element)

This states diagram implemented in the flip flop controls the codification mode of the PROM (mode 0 or mode 1). The PROM is the core of the coder. The table, in M=0 is the mode 0 and in M=1 is the mode 1. The table of codification is in the PROM and the states diagram (controller) is in the flip flop memory (Fig.3).

**Figure 3**: PROM with the table of codification 7B8B and flip flop

The flip flop receives the disparity $P$ of the anterior word and leads to the correct codification mode (0 or 1). If the disparity $P$ is null, it maintains the codification mode, otherwise it switches to the other codification mode.

DOI 10.18502/keg.v5i6.7087
In the coder, the data enters in serial 7B, is converted to parallel, coded 7B8B, newly converted to serial 8B and after sent. For this is necessary an input shift register serial-parallel, a normal register, a memory PROM and an output parallel-serial shift register. So that this components (architecture) work correctly is necessary a controller based in a clock synthesizer involving two counters of module 7 and 8 and a PLL (Phase Lock Loop) (Fig.4).

![Figure 4: Coder 7B8B](image)

The clock generator is the reference mark (beat) that marks the operating rhythm of the global system. In the auxiliary channel CA (127, 255, 383, 511), when CA=0 are send the words of NORMAL (11111100, 11000000), when CA=1 are sent the words of ALARM (01111110, 10000001).

### 3.2. Decoder 7B8B

The decodification table 7B8B is programmed in a PROM (Dec), but is necessary that the input words are aligned.

To align correctly the input words 8B is necessary an error detector that detects the forbidden words that don’t belong to the codification dictionary and still disrespect the accumulate disparity rule. When occur more than x errors in M, there is suspicion of misalignment and then is activated the alignment mechanism.
The error detector receives information $P_1P_0$ of the words forbidden (not belongs to the dictionary) and still disrespects to the accumulate disparity rule and leads to the error state $S_e$, with output $Z = 1$ (Fig. 5).

![States diagram of the errors detector](image1.png)

**Figure 5:** States diagram of the errors detector

The states diagram conduct to the states table, which leads to the circuit of errors detector (Fig. 6).

![States table and respective errors detector](image2.png)

**Figure 6:** States table and respective errors detector

The errors detector was implemented with conventional logic (Karnaugh maps), but it could be implemented in a PROM with 2 flip flops (states table) or in a PLD (states diagram). The PROM (Decod) with the decodification table 7B8B and the errors detector has the aspect of Fig. 7.

![PROM with the decodification table 7B8B and Detector](image3.png)

**Figure 7:** PROM with the decodification table 7B8B and Detector
So that the PROM (Decod) can work correctly are needed various auxiliary circuits.

In the decoder, the data enters in serial 8B, it is converted to parallel, decoded 7B8B and after converted newly to serial 7B recovering the original sequence.

For this, is needed an input shift register serial - parallel, a normal register, a memory (PROM) and an output parallel-serial shift register. So that this components (architecture) work in perfect harmony is need the controller based in the clocks synthesizer of two counters of module 7 and 8 and a PLL (Fig.8).

The controller of module 8 can change provisionally the counting module for 8+1=9, in order to make the word alignment, if there is more than x=8 word errors in M=64 words in the beginning and after 4 word errors in M=64.

The clock recover (synchronizer) is the beat that marks the rhythm of operation of the entire receiver. In auxiliary channel CA, when appear the words NORMAL (11111100, 11000000), C1C0(SR)=01, is made the latch Reset. When appear the words ALARM (01111110, 10000001), C1C0(SR)=10 is made the latch Set (LED on). For other words C1C0=00, the latch maintains the state.

4. Project, Tests and Results

We present bellow the project, tests and results [5].
4.1. Project

In the pair Coder - Decoder, such as in a digital communication system, is necessary to use good project techniques to create the hardware (architecture) with the able potentiality to execute the desired task. However, is still necessary to make them communicate using a clock synthesizer (controller) that gives unity to the system parts and whose clock is the beater that marks the rhythm. After designed, the project was tested theoretically in the paper, with a sequence 1’ and 0 alternated, however actually the simulation is the tool normally used. Following, the pair Cod - Decod was mounted and tested in an ebonite breadboard. Finally, with previous performance guaranties, was implemented in printed circuit boards.

4.2. Tests

The pair Cod - Decod was initially connected between itself by a cooper line. In the transitory state, during the power on, occur the inherent errors until to establish the word alignment, which after in the permanent state soon finished and no more appeared.

4.3. Results

The pair Cod-Decod was integrated in the global system with transmission by fibre optic with a gap (air space) that simulated 50 km.

With a degraded signal in its relation signal - noise SNR in the minimum threshold of the CCITT (Comité Consultatif International Téléphonique Télégraphique) / ITU (International Telecommunication Union) the ratio error bits/ total bits BER (Bit Error Rate) of the system (BER - measured) was $10^{-15}$ and was lesser than the values allowed by the CCITT (BER - CCITT) of $10^{-12}$.

This errors 3 - 5 in a period of 24h, have nothing with the pair Cod - Decod, but yes with the fortuitous errors of the synchronizer decision when deciphers ‘1’ or ‘0’ in a signal strongly degraded and indecipherable at human eye.

5. Conclusions

We studied the block coder 7B8B with auxiliary channel.

The coder prepares the data to be transmitted with greater quality and security.
The pair coder decoder 7B8B is based in the code 7B8B programmed in a memory PROM and input serial-parallel and output parallel - serial.

The pair coder decoder needs in the emitter of a clock generator and in the decoder of a synchronizer (clock recovery) that are the reference clock of the synthesizer that controls the work rhythm of all the system.

In the transitory state of start, it arise the inherent alignments errors, that quickly disappear when the permanent state is reached.

With the pair Cod - Decod integrated in the global system, it arise some errors, which are related with the synchronizer decider that is disturbed by the noise.

Anyway, the obtained BER - measured was 10-15 what is lesser than the allowed BER -CCITT that is 10-12.

Acknowledgments

The authors are grateful to the program FCT (Foundation for the Science and Technology).

References

[1] D. A. Huffman, “A method for the construction of minimum redundancy codes”, in Proceedings IRE Vol.40 pp.1098-1101, 1962.

[2] H. Lohscheller, “A subjectively adapted image communication system”, IEEE Transactions on Comm. Com.32, pp.1316-1322, Dec. 1984.

[3] S. A. Vanstone, P. Oorschot, “An introduction to error correcting codes with applications”. Klumer Academic Publishers, 1989.

[4] ITU-T Recommendation H.261, “Video codec for audiovisual services at px64 kbps”, 1993.

[5] A. Hefez, “Introdução à teoria dos códigos”. UNICAMP, 1994.

[6] Joan L. Mitchell, William B. Pennebaker, Chad E. Fogg, Didier J. LeGall, “MPEG video compression standard”, 1996.

[7] Roopali Garg, “Line Code for Optical Fiber Communication Systems”, IETE Journal of Education, Vol. 40, No. 3-4, PP. 93-100, Ponjab - India, Jul - Dec 1999.

[8] Demir Oner, “Criteria for Choosing Line Codes in Data Communication”, Journal of Electrical and Electronics Engineering, Vol. 3, No. 2, PP. 843-857, Istambul 2003.
[9] V. Sneha Latha et al, “Performance Evaluation of Different Line Codes”, Indian Journal of Computer Science and Engineering, Vol.2, No.4, PP. 575-588, India, Aug-Sep 2011.

[10] Ankit Gupta, Gurashish Singh, “Implementation and Analysis of Different Line Coding Schemes using Verilog”, Internat. Journal of Science, Engineering and Technology Research, Vol. 5, I. 2, PP. 395-401, India, Feb. 2016.