An Error-Correcting Line Code for a HEP Rad-Hard Multi-GigaBit Optical Link

G. Papotti [a, b], A. Marchioro [a], P. Moreira [a]

[a] CERN, 1211 Geneva 23, Switzerland
[b] Dept. of Information Engineering, Università di Parma, Italy

giulia.papotti@cern.ch

Abstract

This paper presents a line encoding scheme designed for the GBT ASIC, a transceiver under development for a multi-gigabit optical link upgrade of the TTC system. A general overview of issues related to optical links placed in radiation environments is given, and the required properties of the line code discussed. A scheme that preserves the DC-balance of the line and allows forward error correction is proposed. It is implemented through the concatenation of scrambling, a Reed-Solomon error-correction scheme and the addition of an error-tolerant DC-balanced header. The properties of the code are verified for two different interleaving options, which achieve different error correction capability at different implementation costs. One of the two options was implemented in a fully digital ASIC fabricated in a 0.13 μm CMOS technology, and ASIC implementation details and test results are reported.

I. INTRODUCTION

The Timing, Trigger and Control (TTC, [1]) system is responsible for timing distribution from the RF generators down to the experiments at the LHC. At the experiments, together with the LHC clock, level-1 trigger and control information are broadcast from a master transmitter to many receivers placed inside the detectors. In the framework of future luminosity improvements of the LHC, an upgrade of TTC system is under development. In the new proposed version the optical link becomes bidirectional and the transceiver ASIC is named GigaBit Transceiver (GBT). This new link profits from a new technology for the ASIC implementation which allows an important speed improvement. While in the present TTC 2 bits per 25 ns are delivered, corresponding to one bit reserved for level-1 trigger and one for control information, more than 60 bits are available in the new version. This allows the transmission of complex trigger information as opposed to the present 1-bit value, the protection of the whole packet through error-correcting (EC) schemes, the transmission of a whole control command every bunch crossing (whereas in the present system it takes many).

A line code is required to provide a relatively high number of transitions on the serial bit stream in order to facilitate clock and data recovery (CDR) and in particular for low jitter. Moreover, the data stream has to be constituted of roughly the same number of zeros and ones in order to allow for ac-coupling in the receiver. As in our application Single Event Upsets (SEUs) on the photodiode (PD) are likely to be the main source of errors, the line code proposed here includes an EC scheme particularly targeted to this issue.

We propose a line code that uses the concatenation of a scrambler, a Reed-Solomon (RS) EC encoder/decoder and the addition of an error-tolerant header used for frame synchronization. The scheme is proposed in two options which achieve slightly different EC capabilities. Details about each line code building block and the differences between the two options are explained in the third section of this paper, simulation results are given in the fourth section.

A demonstrator ASIC for the first option was implemented in a 0.13μm CMOS technology, its details are described in the fifth section of the paper along with test results.

II. OPTICAL LINKS FOR HEP EXPERIMENTS

A. Requirements for general purpose links

The scheme of a general purpose optical link is shown in Fig. 1 and summarizes what the fundamental components of the system are.
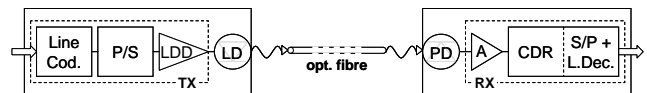

Figure 1: General scheme of an optical link. In the ASIC transmitter (TX), among other functions, line encoding and parallel-to-serial (P/S) conversion are performed. LDD is the laser driver and LD the laser diode. In the ASIC receiver (RX) amplification (A), CDR, serial-to-parallel conversion (S/P) and line decoding are performed.

The transmission of abundant timing information eases clock recovery circuitry design and operation, and allows the reconstruction of a low jitter clock. In a system based on non-return-to-zero data transmission, the timing information is contained in the level transitions between 1-bit values and 0-bit values. The abundance of these transitions is quantified by average and maximum run-length, that is the average and maximum length of strings constituted by identical values (zeros or ones). Additionally, the sequence of transmitted bits should not be periodic in order to keep pattern dependent jitter low. These requirements apply in particular to a system like the GBT on which the experiments rely for the precise delivery of the LHC clock and where the target jitter is as low as 20 ps rms.

The amplifying stages between the PD and the receiver have a high-pass response eliminating the low frequency content of the signal spectrum. For this not to distort the transmission, the data stream is required to have small low frequency content (i.e. DC-balanced data). This is equivalent to saying that the data should contain on average an equal number of zeros and ones on a time window with length proportional to the inverse of the low cutoff frequency.

B. Additional requirements for HEP applications

Optical links used in HEP applications have to be radiation resistant in order to be able to operate reliably in harsh radiation environments. The radiation hardness of the system is ensured through two approaches, one concerning the optical components and the other concerning the ASICs. When exposed to radiation, semiconductor laser diodes suffer a threshold shift, optical fibres "darken" (i.e. the attenuation increases) and PD responsivity decreases [2]. The optical components go through a thorough qualification process and an additional margin is built-in in the optical power budget. Concerning the electronic components, the effects can be classified into total dose effects and Single Event Effects (SEEs). The former effects can be minimized by adopting particular layout techniques (e.g. enclosed layout transistors, [3]). Concerning the latter effects, examples are single event latch-up and SEUs. Testing carried out on the technology to be used indicate that single event latch-up does not occur for the technology in use and the radiation levels expected at the LHC. SEUs are a concern for this technology, but ASICs can be made "SEU-tolerant" mainly by architectural and circuit techniques, e.g. by triple modular redundancy (majority voting) logic circuits [4].

Photodiodes are also sensitive to SEUs. Their function in the link is to detect light coming from the laser transmitter through the optical fibre. By means of the same physical principles, they also detect ionizing radiation. As no light detection translates into a 0-bit value, while light detection translates into a 1-bit value, ionizing radiation can in fact transform a logical 0 sent by the transmitter into a logical 1 detected by the receiver. Data transmission can be protected against such "fake signals" by implementing error correcting schemes in which the encoding is performed at the transmitter and the decoding at the receiver, i.e. after the PD. As SEE errors on the other components of the link can be easily dealt with, the proposed line encoding scheme addresses in particular the issue of SEUs on the PD.

Error cross-section studies for optical links components to be used in HEP or satellite environments were reported [5, 6, 7]. It is established that the major error cause is PD SEUs. The error probability decreases with an increase on the level of incident optical power, it increases with data rate and has a maximum when the angle of incidence between the particles and the PD is 0 degrees (i.e. grazing angle). Considering fluxes easily exceeding 10^6 particles s^{-1} cm^{-2} in the inner parts of the detectors at the LHC [8] and average reported error cross-sections of $10^{-6}cm^2$ [6, 7], a bit error rate of 10^{-10} can be estimated for the link under development. According to this bit error rate value, the probability of two error events falling in the same frame is 10^{-17} (Bernoulli trials, [9]).

Even though system performance depends highly on the launched optical power and the PD has not been chosen yet, the estimated bit error rate is not acceptable as it is (1 error/s per link), and thus EC capability has to be implemented in the system for reliable operation. At least single error event correction has to be included in the link in order to shift from the initial bit error rate value of 10^{-10} to 10^{-17} for double events (corresponding to 1 error per hour on 10000 links).

Low latency is fundamental in the GBT link as one of its functions is to deliver level-1 trigger information shortly after a collision event. This is a limiting factor in the choice of the EC scheme as many of the EC algorithms proposed in literature and utilized in commercial links are more computationally intensive.

III. LINE CODE ARCHITECTURE

The line code is designed to provide the data with the following properties: 1) a high number of transitions; 2) DC-balance; 3) non-periodicity; 4) frame synchronization capability; 5) high efficiency; 6) EC protection; 7) low latency (as only a time latency equivalent to few LHC clocks can be spent for line coding).

While the last two points are specified in particular for the GBT optical link, the previous ones are more general and of concern for many commercial links. For this reason, commercially adopted schemes were evaluated for our application, even though none of them is found fully compliant with our needs. The 8b/10b scheme [10] for example is difficult to combine with error-correcting codes while maintaining a good efficiency: single errors on the line are converted into bursts, which then require a strong error-correcting scheme, and consequently lead to a less efficient use of bandwidth. The 64b/66b scheme, used in 10G Ethernet physical layer (IEEE 802.3ae standard) could satisfy most of our needs apart from SEU tolerance.

The proposed line code utilizes a concatenation of three operations in order to get to a line data stream that complies with all the requirements stated before. As can be seen in Fig. 2, the data is first scrambled, then encoded through a RS EC scheme and finally a header is added for frame recognition. Inverting the order of scrambling and EC would result in error multiplication due to the scrambler, and this would consequently impose the need for a stronger EC scheme.
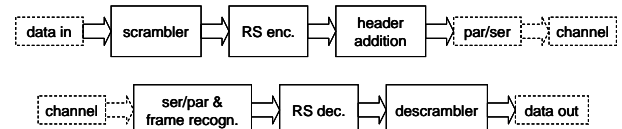

Figure 2: Block diagram of the proposed line encoding scheme: encoder diagram (top) and decoder diagram (bottom).

A. Scrambling

Scrambling [11] is a method of randomizing the statistics of a data stream. Some data patterns, as long strings of 1's or 0's, are commonly transmitted but problematic with regard to DC-balance and run-length issues. Through scrambling, these sequences are mapped to others which resemble closely real random data. This statistically guarantees DC-balance and a low average run-length on the scrambled bits.

A scrambler is typically implemented through an n-bit linear feedback shift register. Its output sequence is "pseudorandom", and for a constant input, the output has a period of 2^n-1. Thus higher n produces longer sequences.

In order to avoid the need for a synchronization code the scrambler is implemented as self-synchronizing. In this way, at the beginning of the transmission and in case of

synchronization loss there is no need to send a special synchronization code. As the TTC is currently implemented in a broadcast topology, self-synchronization is particularly effective since it allows a single receiver to reacquire synchronization without interfering with the operation of the other receivers. The implementation of a self-synchronizing scrambler and descrambler of length 63 is shown in Fig. 3.
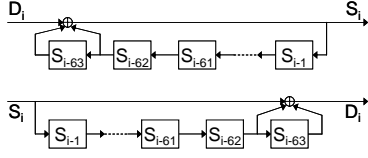

Figure 3: 63-bit scrambler (top) and descrambler (bottom). D_x is the user data and S_x is the scrambled data. The square blocks are memory elements; the sum is an exclusive-or operation.

B. Reed-Solomon error correction

RS codes [12] are linear cyclic block codes which treat groups of m bits, referred to as "symbols", as single entities. The correction of one symbol corresponds to the correction of all the m-bits that belong to it, making the codes very well suited for burst errors. Moreover, encoding and decoding are naturally implemented in parallel structures, allowing operation at a frequency that is m-times slower than line speed. The code construction is very flexible and they can be built to correct multiple errors. RS codes are also proven to be very efficient [12].

The chosen RS encoding scheme is systematic, i.e. the redundancy bits are appended without actually modifying the data word, so that the resulting RS encoded word after scrambling is still DC-balanced. Additionally, the pseudorandom characteristics of the data word are acquired also by the redundancy bits.

Once the width m of the symbol is chosen, the maximum number of symbols that belong to the same block is fixed ($N_{s,max} = 2^m-1$, or, in bits: $N_{b,max} = m \cdot (2^m-1)$), being the block the unit on which the error correction operation is performed. In the case of a 60-bit block, 15 4-bit symbols can be used. Also a shorter block length can be used if part of the available data space is considered filled by zeros. These in fact are not transmitted, but the feature is known at both ends of the link and used in the encoding and decoding algorithms, additionally allowing extra error detection. In the case above, if only 40 bits, that is 10 symbols, are needed out of 60, then 5 of 15 symbols can be zero-padded. If a number t_s of symbols needs to be corrected, then up to ($N_{s,max} - 2t_s$) symbols in the block can be used for data information.

The encoding procedure is performed via a division operation which is implemented through a feedback shift register. The decoding operation is more complicated and is divided in steps: 1) detection of errors; 2) identification of the corrupted symbol and 3) identification of the corrupted bits within a symbol. The case of single error correction (i.e. $t_s = 1$) uses a simple and fast algorithm. Due to the latency requirements of the link under study, $t_s = 1$ is chosen. Decoding is then implemented through a feedback shift-register followed by multiplication and look-up operation.

In order to accommodate for the cases in which the SEU error event extends over 2 bits that might belong to different symbols, and in order to extend the error correction capability, interleaved blocks are used.

C. Frame delimitation

The addition of a header block is imposed by the need of synchronizing the data stream at the frame level. Repeated recognition of a valid header in a fixed position in the frame allows for frame-locking. Repeated non-valid header recognition causes loss of frame lock.

At the same time, different headers are required to distinguish different packet types, i.e. user data, trigger data or idle pattern. The patterns used for the headers are 6 or 8-bit long (depending on the interleaving option) and are chosen to be DC-balanced and redundant so that they are still recognizable even if corrupted by an SEU event.

D. Coding options

Our scheme proposes two interleaving options: one is based on 4-bit RS symbols and interleaving of two RS blocks, while the other one is based on 3-bit RS symbols and four-block interleaving. The first interleaving option is shown in Fig. 4 and the second in Fig. 5.
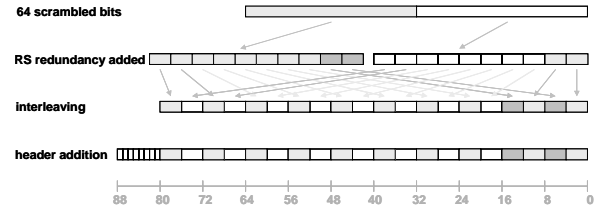

Figure 4: Modification of the data word through the line encoding building blocks, first option.

In the first option 64 user bits are first scrambled. Then two blocks of 32 bits each are RS encoded: 2 redundancy symbols (8 bits in total) are added to each of the two blocks, which are then interleaved. Last, an 8-bit header is appended, leading to a total frame length of 88 bits, a total link speed of $f_{bit} = 88 \cdot 40$ MHz $= 3.52$ GHz and an overall line code efficiency of 72.7%. The overall encoding process takes one $T_{FRAME} = 88 \cdot T_{bit}$, while the decoding process takes 2 T_{FRAME}.
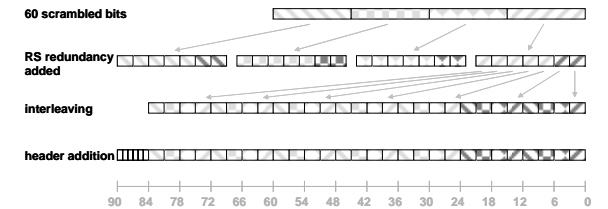

Figure 5: Modification of the data word for the second option.

The main differences between the two options are the number of interleaved blocks and the block widths. In the second option (see Fig. 5) first 60-bit data are scrambled, then four 15-bit blocks are RS encoded (24 redundancy bits are added) and interleaved. A 6-bit header is then added, for a total 90-bit frame, a link speed of $f_{bit} = 90 \cdot 40$MHz $= 3.60$ GHz and an efficiency of 66.7%. The overall encoding process

takes one $T_{FRAME} = 90 \cdot T_{bit}$, while the decoding process takes $2\,T_{FRAME}$.

The summary of the parameters for the two options is given in Table 1. In the table, m is the RS symbol width, L is the number of interleaved blocks, N_s is the number of symbols per RS block, out of which K_s are information symbols. N_b is the total number of bits after RS encoding, K_b of which are information bits. H is the header bit length and N_{tot} is the total frame bit length. The efficiency R is the ratio between the number of information bits and total frame length.

Table 1: Line code characteristics for the two presented options.

code characteristic	first option	second option
scrambler order n	63	60
m	4	3
L	2	4
N_s	10	7
$N_b = N_s L m$	80	84
K_s	8	5
$K_b = K_s L m$	64	60
H	8	6
$N_{tot} = N_b + H$	88	90
$R = K_b / N_{tot}$	72.7%	66.7%
$f_{bit} = 1 / T_{bit}$ (GHz)	3.52	3.6

IV. PROPERTIES OF THE PROPOSED CODING SCHEME

A. *Error-correction capability*

The proposed scheme allows correcting any single upsetting event per frame, even if the event extends over two consecutive bit periods.

Due to interleaving, additional error correction capability is obtained, so that independent double errors per frame can sometimes be corrected depending on the position in which they fall in the frame. The two errors can be tolerated if one falls on the header and one on the RS segment. In case the two errors fall on the RS segment, then two cases must be distinguished. If they fall in different RS blocks, they can be corrected. If they fall in the same RS block, but not in the same symbol, then they cannot be corrected and the packet is corrupted. Due to the choice of implementing the scrambler as self-synchronizing, this last case causes the frame that follows the corrupted one to be incorrect as well. Statistical calculations have been carried out to explore all different possibilities for two events falling in the same frame: the first interleaving option can correct 62% of them, while the second option can correct 81%. The first option though allows for additional error detection due to the use of zero-padding.

The line code error correction capability improves the link performance from an uncoded transmission BER α to a coded Frame Error Rate $FER = 3 \cdot 10^3 \cdot \alpha^2$ for the first option and $1.5 \cdot 10^3 \cdot \alpha^2$ for the second one. This corresponds to an improvement from $\alpha = 10^{-10}$ to FER = 3.8 errors/hour on 10k links for the first option or 1.9 err./h. for the second option.

B. *Implementation complexity*

A Verilog model for the two interleaving options schemes was written, simulated and synthesized using a commercial standard cell library. A comparison of the two options in terms of implementation complexity is carried out based on synthesis. In Table 2 the number of cells and power consumption of encoder and decoder blocks for the two options is reported. The results are based only on the blocks that perform scrambling, RS encoding/decoding and control logic. It can be concluded that the second option halves the error probability at the price of increased power consumption and decreased efficiency.

Table 2: RS code complexity for the two line code options. The power consumption is normalized to the value obtained for the encoder for the first interleaving option.

Module	Cell count	Power cons. (a.u.)
enc.1st option	1066	1
enc. 2nd option	1098	1.23
dec. 1st option	2794	2.04
dec. 2nd option	2402	2.56

C. *DC-wander simulation results*

Excessive DC-wander impacts on system performance by effectively closing the eye diagram. In order to verify the amount of DC-wander generated by this scheme, the high level model and the synthesized netlist were extensively simulated. The baseline wander behaviour is studied through a finite time step simulation of high-pass filtering the line data stream (at f_{bit}) with a 100 KHz cutoff frequency. Histograms of the obtained wander probability on 5 Mb data streams are plotted in Fig. 6 (top) for the first option and (bottom) for the second one. Different user data (random data, constant 0's and 1's, idle frames) show qualitatively the same wander, with means of the order of 0.01% of signal amplitude, and maximum sigma of 0.47%. These values for the DC-wander impact on the system by generating a BER well below 10^{-22}, consequently negligible with regard to other noise sources.
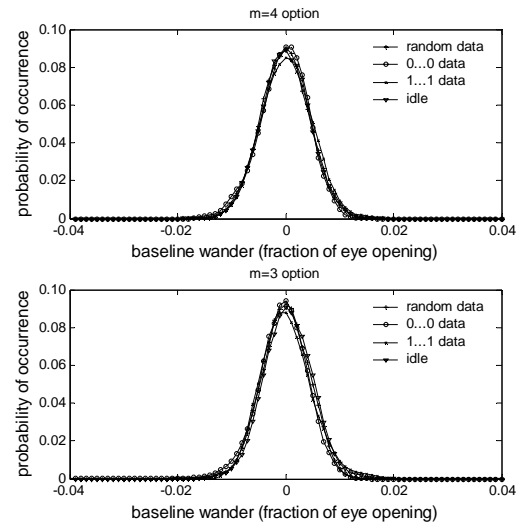

Figure 6: Baseline wander probability of occurrence.

Simulation results show that for both options of the proposed encoding scheme the average run-length is less than 2 bits as consequence of data scrambling, while the maximum run-length is strictly limited by the presence of the header to 80 bits for the first option and 84 bits for the second one.

V. Demonstrator asic implementation

A. *Functionality*

A test chip for the first option of the encoding scheme was fabricated in a commercial 0.13 µm CMOS technology. Encoder and decoder are implemented according to the block diagrams show in Fig. 7. The 64 user bits enter the encoder bytewise due to pad number limitations. The data packet is then scrambled, the RS redundancy bits and the header are added and finally the 88-bit word is serialized and driven out of the chip. In the decoder the inverse process is implemented. First a frame synchronization block identifies the frame boundaries, then the word is checked for errors through the RS decoder, finally descrambled and multiplexed out in 8-bit groups. Data or idles can be sent, and scrambling or RS encoding can be bypassed for testability.
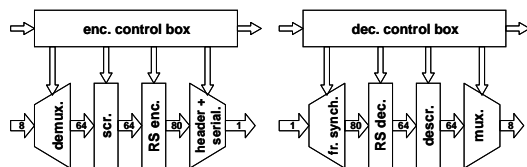

Figure 7: Block diagram of the encoder (left) and decoder (right) implemented in the ASIC. Details about control signals cannot be discussed due to space limitations.

B. *Implementation details and test results*

The whole encoder block utilizes 1700 cells for a power consumption of 25 mW/GHz, while the decoder block counts 5000 cells for a power consumption of 50 mW/GHz. The total area of the die is (1.3×1) mm^2, while the active areas are (0.15×0.35) mm^2 for the encoder and (0.35×0.35) mm^2 for the decoder. An image of the die is shown in Fig. 8.
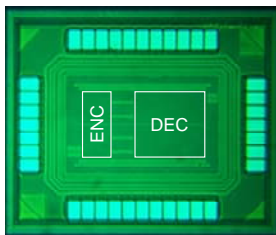

Figure 8: Picture of the die. The 36 bondpads and the power rings are visible around the encoder and decoder active areas.

Through a digital tester, test vectors can be applied to the inputs of the die via a board and socket interface. The die signal outputs are checked against vectors obtained from simulation and the correct operation of the ASIC can thus be verified. Test input vectors were applied to the encoder and decoder for verifying data or idle transmission while applying different control settings. In the decoder case, also data corrupted by errors were used to verify the operation of the error correcting circuitry. The encoder and decoder were additionally tested in back-to-back mode. The tests were carried out at frequencies below 50 MHz due to limitations imposed by the test board. All tests were successful. The power consumption measurement is in agreement with the simulated values. The ASIC implementation does not take into account radiation resistance of the ASIC itself, characteristic which will be dealt with in future work.

VI. Conclusions

We reported the study for a line code to be used in the GBT ASIC transceiver. The line code performs error correction while guaranteeing a DC-balance well below 1% of signal amplitude and average run-length of 2 bits. The code is based on the concatenation of scrambling, a Reed-Solomon error-correcting scheme and the addition of a DC-balanced error-tolerant header. Two different coding options are explored which correct all single upsetting events and different fractions of double events per frame, at different implementation costs. One of the two options was implemented in a fully digital 0.13 µm CMOS technology demonstrator ASIC for which implementation details and test results are given.

VII. References

[1] B. Taylor, "Timing Distribution at the LHC", Proc. 8th Workshop on Electronics for LHC Experiments," Colmar, France, 9-13 September 2002, CERN 2002-003, pp. 63-74.

[2] K. Gill et al., "Radiation damage studies of optoelectronic components for the CMS tracker optical links", Proc. of the 3rd Workshop on Electronics for LHC Experiments, London, 1997, pp. 276-281.

[3] G. Anelli, M. Campbell, M. Delmastro et al., "Radiation-tolerant VLSI circuits in standard deep submicon CMOS technologies for the LHC experiments: practical design aspects", IEEE Transactions on Nuclear Science, vol.46, no.6, December 1999, pp.1690-1696.

[4] S. Niranjan, J. F. Frenzel, "A comparison of fault-tolerant state machine architectures for space-borne electronics", IEEE Transactions on Reliability, vol.45, no.1, March 1996, pp.109-113.

[5] F. Faccio, G. Berger, K. Gill, M. Huhtinen, A. Marchioro, P. Moreira, F. Vasey, "Single event upset tests of an 80-Mb/s Optical Receiver", IEEE Transactions on Nuclear Science, vol.48, no.5, October 2001, pp.1700-1707.

[6] C. J. Marshall, P. W. Marshall, M. A. Carts, R. Reed, S. Baier, K. LaBel, "Characterization of transient error cross sections in high speed commercial fiber optic data links", IEEE Radiation Effects Data Workshop, 2001, pp142-145.

[7] P. W. Marshall, P. T. Wiley, R. N. Prusia, G. D. Rash, H. Kim, K. A. LaBel, "Proton induced bit error studies in a 10Gb/s Fiber Optic Link" IEEE Transactions on Nuclear Science, vol.51, no.5, October 2004, pp.2736-2739.

[8] "CMS Tracker project tech. design report", CERN LHCC 98-6, 1998.

[9] A. Papoulis, "Probability, random variables, and stochastic processes", 3rd ed., McGraw-Hill International Editions, pp.43-47.

[10] A. X. Widmer, P. A. Franaszek, "A DC-balanced, partitioned-block, 8B/10B transmission code," IBM Journal R&D. 27, 440-451 (1983).

[11] E. Lee, D. Messerschmitt, "Digital communication", 2nd ed., Kluwer Academic Publishers, pp. 591-603.

[12] S. Lin, D. J. Costello, "Error Control Coding", 2nd ed., Prentice Hall, 2004, ch.2-7.