Reprint

A CMI Decoder with Single Bit Error Correction Capabilities

A. Maniatopoulos, T. Antonakopoulos and V. Makios

The IEEE Singapore International Conference on Networks and Information Engineering '95

SINGAPORE, JULY 1995

A CMI Decoder with Single Bit Error Correction Capabilities

A. Maniatopoulos, T. Antonakopoulos and V. Makios
Laboratory of Electromagnetics, Telecommunications and Information Processing Division
Department of Electrical and Computer Engineering, University of Patras, 26500 Patras, Greece
Tel: +30 (61) 997 286, Fax: +30 (61) 997 342, e-mail: antonako@ee.upatras.gr

Abstract

This paper presents the architecture of a new Coded Mark Inversion (CMI) decoder with improved Bit Error Rate (BER) performance. This new decoder initially processes the received data as NRZ unencoded data at twice the original frequency and then uses a pattern matching circuit for achieving bit-level synchronization and detecting corrupted bits. The decoding and the error correction procedures are finally performed by using a simple sequential circuit. The error correction function is based on a state transition diagram by minimizing the probabilities for state transition between valid and invalid states.

1. Introduction

In baseband communication systems various coding schemes are used for alleviating distortion and noise introduced in the communication channel and for improving the quality of clock recovery mechanisms [1]. Line codes have been used in metallic cable systems and in optical fiber systems. Such codes include Alternate Mark Inversion (AMI), CMI, Differential Mark Inversion (DMI), DmB1M [2],[3],[4]. The unipolar codes used in high speed systems satisfy the bit-sequence independence condition and have DC balance.

The CMI code has been extensively used in coaxial cable and optical fiber communication systems [3], [4] and recently it has been approved for the coding scheme of the electrical interface of the User-Network Interface (Physical Medium Dependent (PMD) sublayer) of B-ISDN [5], [6]. The next section presents a concise description of the CMI code and its properties, and identifies the redundancy introduced by this coding scheme. Section 3 proposes an error correction function that can be applied in CMI coded bit streams by exploiting their code redundancy, while Section 4 presents the architecture of a decoder based on this error correction function for achieving higher BER performance.

2. The CMI Code and its Properties

The Coded Mark Inversion (CMI) coding scheme is a two-level code in which encoded signals for binary '1' remain at the same level during the bit interval but alter level between successive marks, while the encoded signal for binary '0' changes from '0' to '1' in the middle of the symbol interval [3]. Figure 1 shows an example of a CMI coded bit stream. In [1], two different CMI codes are described which differ on how the binary

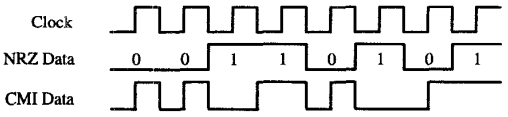


Fig. 1 The CMI code

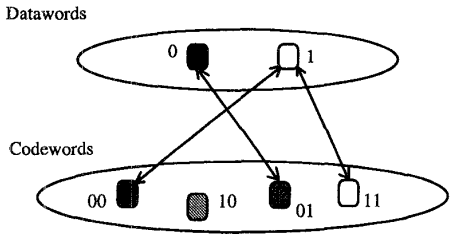


Fig. 2 The mapping of the CMI code spaces

'0' is encoded. In CMI 1, the binary '0' is coded as '10', while in CMI 2, the binary '0' is coded as '01'. Although the proposed decoder is based on CMI 1 code, it can be easily modified to support CMI 2, due to the similarities of the two codes.

The CMI code is a binary code operating at twice the channel rate and achieves DC balance and adequate transmission density for clock recovery at the receiving side. CMI decoders use transitional edges of the received signal for clock extraction, observe data transitions over one-and-a-half symbol duration and make a binary decision based on the most likely transition pattern [3], [4]. Although improved performance is achieved against symbol timing jitter, no error correction is performed for exploiting the redundancy introduced during the coding process. The redundancy, introduced by the CMI code, is due to the use of a part of the available codewords space.

Each data word (1 bit length) results to one or two valid codewords (2 bits length) and the datawords space (2 values) is mapped to the three of four available values of the codeword space, as it is shown in Figure 2. The forth codeword, although is never generated by a CMI coder, can be received at a CMI decoder due to errors introduced during codeword transmission. As it will be proved, the probabilities for receiving the invalid codeword is different for each dataword and this can be used for achieving single bit error correction in some cases.

IEEE Catalogue No. 95TH8061
0-7803-2579-6/95/$4.00©1995IEEE

3. The Decoding Procedure

Before describing how the redundancy by the proposed decoder for performing single bit error correction is exploited, the CMI code state diagram, depicted in Figure 3, has to be described. The diagram is composed of four stages, three of which are valid due to the coding scheme while the fourth state, in the centre of the diagram, is a code violation as a result of transmission error(s). The squared brackets near the symbol states indicate the probability that a symbol was encoded with the respective pattern. The arrows show transition from the initial pattern to the finally received patterns due to transmission errors. It is obvious that the occurrence of the '10' pattern is either the result of a single bit error to a '1' binary symbol (coded either as '00' or as '11') with probability 2p(1-p) or the result of two consecutive bit errors to a '0' binary symbol with probability p^2, where p is the probability that a single bit has been corrupted and using the assumption that the error probability of a single bit is independent of the error probabilities of its previous and subsequent bits. Using the value of 10^{-9} as an indicative value of BER in coaxial cable transmissions, the probability that the '10' code violation is the result of a corrupted '1' symbol is approximately 2.10^9 times greater than the probability of a corrupted '0' symbol. Thus whenever an '10' pattern has been detected it can be assumed to be an '1' symbol corrupted by noise.

Figure 4 shows all possible CMI coded patterns with a single error bit. For each incoming pattern (except '000'), there are two possible CMI coded error-free patterns, depending on how the previous '1' dataword was encoded. For each CMI coded error-free pattern, there are two single bit error patterns in the codeword in the middle of the pattern. The 'corrected CMI pattern' column is generated by applying the previously mentioned correction procedure, by minimizing the error probability. For example, in the second and third cases of the '010' input pattern, the error has not been detected, since it generates a valid code, while in the first and forth cases the error has been detected due to code violation and finally corrected successfully. As it can be seen from this Figure, the 8 initial patterns result to 15 CMI coded patterns with 30 different cases of single error bit occurrence. By using the proposed error correction procedure, in 8 cases of the 30 in total, error correction can be achieved.

The CMI code redundancy can be exploited by individually processing the received signal as an unencoded NRZ signal at 2F Mbps instead of as an F Mbps CMI coded signal and use a pattern matching circuit for detecting the occurrence of a '10' pattern. By observing the relation of the NRZ signal and the respective symbols, the '10' pattern during a CMI symbol interval can be used either to detect symbol boundaries, when no phase synchronisation has been achieved, or to detect the occurrence of an error at least at one half of this symbol. This procedure is implemented in the proposed CMI decoder.

Figure 5 shows the block diagram of a CMI coder and decoder. The CMI coder is composed of four blocks. The Clock Delay and the Data Delay blocks are used for allowing the Memory Cell to acquire the new bit, and if it

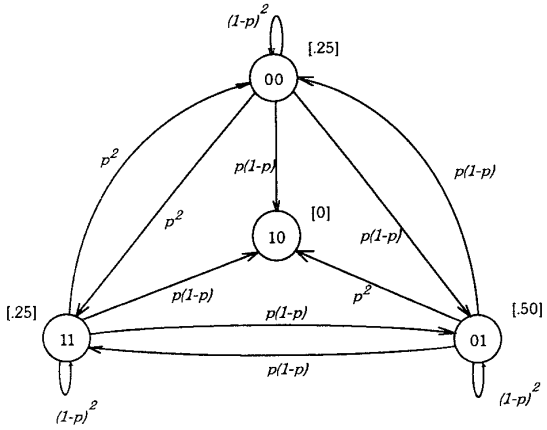
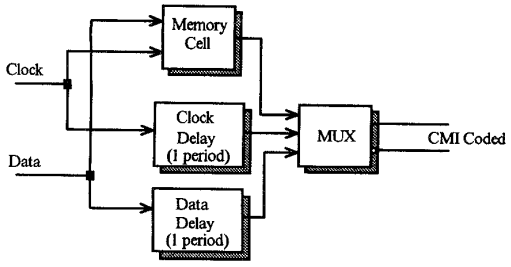


Fig. 3 The CMI code state diagram

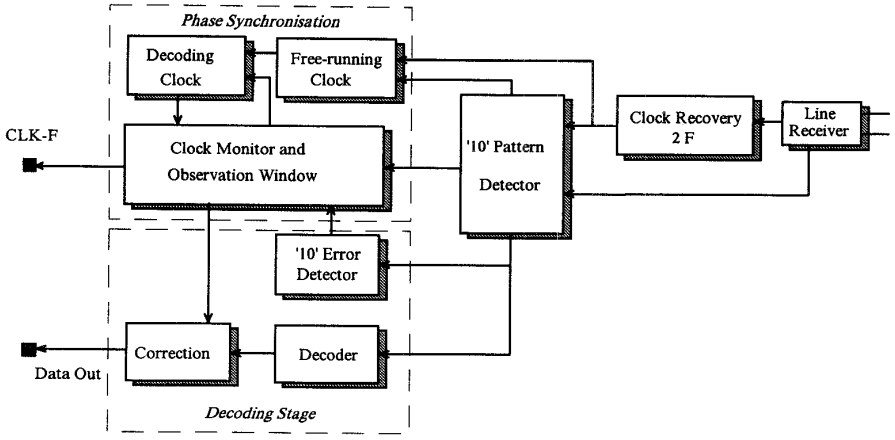
Incoming Pattern	CMI Coded Error-Free Pattern	Single Bit Error Pattern	Corrected CMI Pattern	Outgoing Pattern
000				010
				010
001				011
				011
				011
				011
100				110
				110
				110
				110
101				111
				111
				111
				111
010				010
				000
				000
				010
011				011
				001
				001
				011
110				100
				110
				110
				100
111				101
				111
				111
				101

Fig. 4 CMI decoding and single bit error correction.

is equal to '1', to compare it with the previous coded value of '1' for generating the new value. The output of the Data Delay block is used as the control signal of the multiplexer at the output stage. If this control signal is '1', the Memory Cell value is used as the CMI output, otherwise a clock period is used.



(a) The coder



(b) The decoder

Fig. 5 The CMI coder-decoder block diagram

If F is the frequency of the unencoded data, the generated CMI bit stream has frequency equal to 2F and that is the decoder operational frequency. The choice of a decoder operating at twice the frequency of the unencoded data, was preferred to an asynchronous decoding logic at the system frequency for reasons of higher tolerance in noise conditions. According to [4], a deviation of the rising edge of a CMI signal should not be higher than T/4 from its central position, for a decoder operating at the some frequency with the coder. For an STM-1 interface this timing tolerance is 1.6ns

The decoder is based on the detection of '10' patterns for performing either synchronization or single bit error detection. The decoder has two modes of operation, the sync acquisition mode and the decoding mode. After power-up, the circuit enters the sync acquisition mode, waiting for the stabilization of the clock recovery module and then waits the detection of a '10' pattern for detecting the boundaries of different codewords. Upon achieving synchronization, the detection of a '10' pattern is due to a transmission error and the output is forced to '1' for implementing the error correction function.

As it is indicated in Figure 5b, the clock recovery is performed at 2F MHz, the '10' pattern detection is applied and then the CMI decoding is performed. The NRZ signal contains the necessary timing information for clock extraction since it contains at least one edge over a one and a half codeword duration. The decoder is divided in two parts: the first part is used for phase synchronisation while the second part performs CMI decoding and error correction following the method previously described. The phase synchronisation part consists of the decoding clock, the free-running clock and the clock monitor modules. The decoding clock is stable and keeps its phase even under the occurrences of errors in the received signal. The free-running clock responds immediately to phase variations when '10' patterns are detected. The clock monitor module determines the phase of the clock in the decoding process by changing the phase of the decoding clock using a type of hysteresis. Whenever an '10' pattern is detected, the clock monitor isolates the decoding clock from the free-running clock for n-clock cycles (the number of clock-monitor stages that determine the decoder response time and its inertia to transmission errors). After n-clock cycles, the decoding clock is resynchronized to the incoming data stream phase. This clock isolation procedure increases the decoder endurance to transmission errors and determines the minimum distance of single bit errors that can be corrected by the proposed decoder.

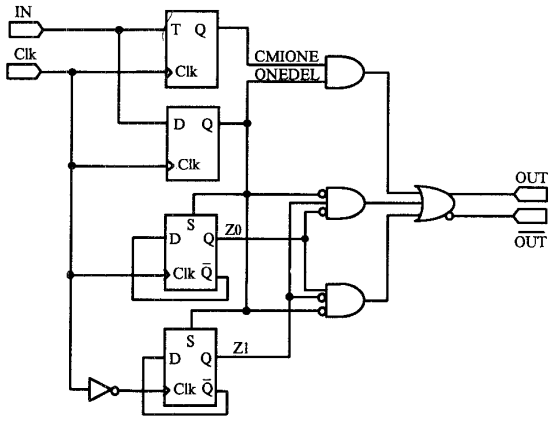


Fig. 6 The CMI coder

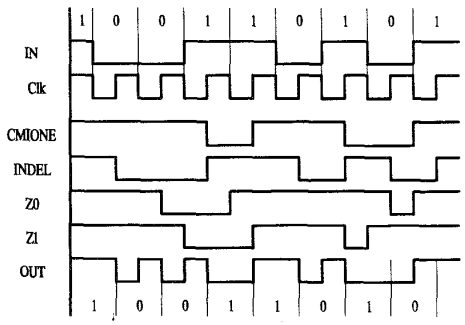


Fig. 7 The CMI coder timing diagram.

STM-1 SDH electrical interface following the G.703 CCITT Recommendation [6].

The CMI coder is depicted in Figure 6. The circuit receives the NRZ data with the clock at 155 MHz and the encoded data are generated with 3.5 nsec delay at 311 MHz. The coder and the decoder have been implemented using *ECLinPS* technology. The CMI coder timing diagram is shown in Figure 7. The diagram shows the timing performance at various stages of the CMI coder. In the test-bed implementation, the data are received from an 155-Mbit/s Synchronizer, the SYN155 device of Transwitch Inc. Although this device was developed for NRZ transmissions over fiber, it has been interfaced to a CMI coder for supporting also coaxial cable transmissions.

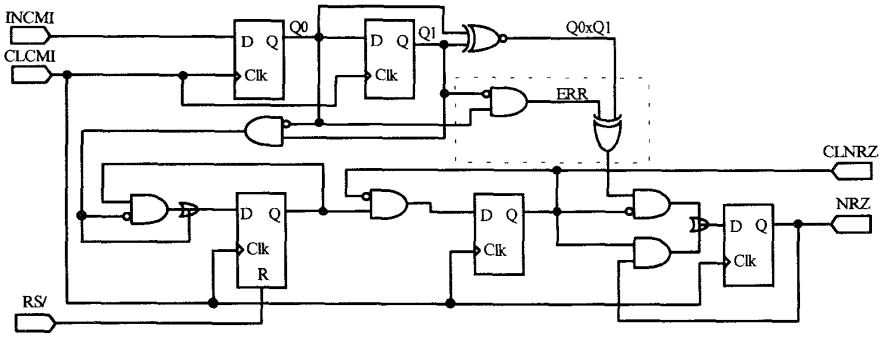


Fig. 8 The CMI decoder

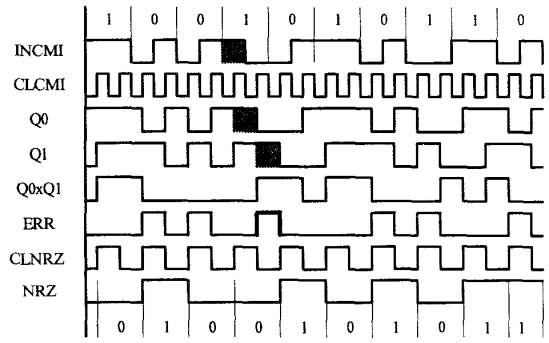


Fig. 9 The CMI decoder timing diagram

4. The CMI Coder-Decoder Implementation

This work originated from the need to implement a real-time test bed for emulating the traffic conditions of virtual channels in Asynchronous Transfer Mode (ATM) networks at 155 Mbps in the framework on the DISTIMA project of RACE [7]. During this work a reprogrammable hardware platform was developed using off-the-self components but mainly using Field Programmable GateArrays [8]. The Physical Medium Dependent (PMD) of this hardware platform was specified to implement the

The circuit shown in Figure 8 implements the '10' pattern detection and the decoding stage of the CMI decoder presented in the previous section. The INCMI and CLCMI signals are the data and clock signals generated by the Clock Recovery module. The first two flip-flops implement the CMI autocorellation function, while the last two flip-flops implement the clock phase alignment. The third flip-flop is used by the Clock Monitor module for regaining synchronization after the detection of error conditions in the incoming data stream. Figure 9 gives an example of the decoding procedure implemented in the decoder of Figure 8. The shaded area indicates a corrupted bit which is passed through the error correction part and finally corrected. The ERR indication is valid only when the CLNRZ is low and in this case, the error correction is performed.

Figure 10 shows the PMD board developed for the DISTIMA project. It contains the SYN155 Synchronizer, a clock recovery module at 155 Mbps for interfacing the SYN155 with the CMI encoder and a clock recovery module at 311 Mbps for the CMI decoder.

5. Conclusions

The paper presented a new method for achieving single bit error correction in a CMI encoded bit stream by exploiting redundancy imposed by the CMI coding scheme. The decoder implements the mapping of a state diagram consisting of four states to a three states diagram using minimum error probabilities. The error correction capabilities of the decoder depend on the number of '1' in

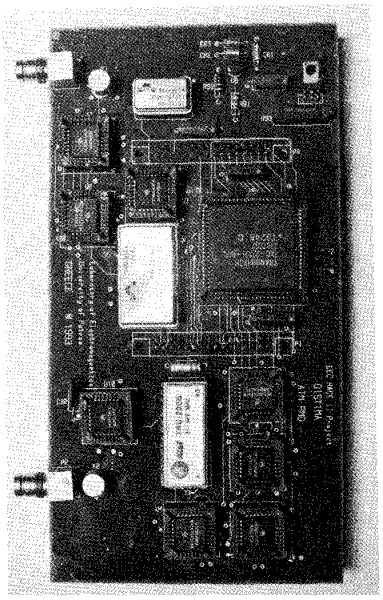


Fig. 10 The PMD board of the STM-1 interface

the initial data and on the distance of the adjacent corrupted bits. The presented decoder has been implemented using ECLinPS technology and has been used in the STM-1 electrical interface (155 Mbps) of the Synchronous Digital Hierarchy.

Acknowledgement

This work was partially supported by the "Research for Advanced Communications in Europe" (RACE) program, R2045 DISTIMA project.

References

[1] Yoshitaka Takasaki, "Digital Transmission Design and Jitter Analysis", Artech House, 1991.

[2] S. Kawanishi, N, Yoshikai, J. I. Yamada and K. Nakagawa, "DmB1M Code and its Performance in a very High-Speed Optical Transmission System". *IEEE Transactions on Communications*, Vol. 36, No 8, August 1988, pp. 951-956.

[3] K. Hagishima, Y. Kobayashi and K. Aida, "High-Speed CMI Optical IntraOffice Transmission System: Design and Performance", *IEEE Journal on Selected Areas in Communications*, Vol. SAC-4, No 9, December 1986, pp. 1451-1457.

[4] Y.H. Chew and Tjeng. T. Tjhung: "A New Edge Detector for CMI Optical Fiber Transmission with Improved Jitter Tolerance", *IEEE Journal of Lighwave Technology,* Vol. 11, No 4, April 1993, pp. 654-660.

[5] Sexton and A. Reid, "Transmission Networking: SONET and the Synchronous Digital Hierarchy", Artech House, 1992.

[6] CCITT Recommendation G.703, "Physical-Electrical Characteristics of Hierarchical Digital Interfaces", June 1990.

[7] DISTIMA project: "Annual Project Review Report", September 1994, 45/SIE/SN6/DS/B/A1-94/b1.

[8] C. Papadopoulos, A. Maniatopoulos, T. Antonakopoulos and V. Makios: "A Real-time Test-bed for Prototyping Cell-based Communication Networks", *The 5th IEEE International Workshop on Rapid System Prototyping,* Grenoble, France, June 1994.