FPGA Implementation of a Modified Turbo Encoder

Rajagopal.A
Research Scholar
Department of E&C,
DSCE,VTU
Bengaluru, India

Karibasappa.K
Professor
Department of E&C,
DSCE,VTU
Bengaluru, India

Vasundara Patel K.S
Associate Professor
Department of E&C,
BMSCE,VTU
Bengaluru,India

ABSTRACT
Turbo convolutional codes (TCC) are excellent error correcting codes, however TCC decoding based on *A-Posteriori Probability* (APP) algorithm is computationally complex and the complexity is not significantly reduced even if puncturing mechanism is used. To overcome the above disadvantage turbo codes need to be concatenated with other coding techniques such that the decoding complexity is significantly reduced and at the same time the Signal to Noise Ratio (SNR) can be as close to the Shannon limit as possible. In this paper one such modification is described, whereby convolutional coding as well as block coding technique of Zig-Zag codes will be used. First the simulation results of the encoder using MATLAB are presented and then the FPGA results using Artix-7 board will be shown.

General Terms
Error correcting codes and source encoding techniques

Keywords
Recursive Systematic Coder (RSC), Zig-Zag codes, Interleaver, parallel concatenation, Artix-7 FPGA.

1. INTRODUCTION
The purpose of a communication system is to transport an information bearing signal from a source to a user destination via a communication channel. One of the major reasons for the continuous growth in the use of digital communication is its ability to reduce cost of any communication links to almost any location with higher power efficiency. In a digital transmission system, error control is achieved by the use of channel coding schemes. Channel coding schemes protect the signal from the effects of channel noise and interference and ensure that the received information is as close as possible to the transmitted information. They help to reduce the Bit Error Rate (BER) and improve reliability of information transmission. Channel coding schemes involve the insertion of redundant bits into the data stream that help to detect and correct bit errors in the received data stream. Due to the addition of the redundant bits in channel coding, there is a decrease in data rate and bandwidth is expanded. There are two types of channel codes namely convolutional codes and block codes. Block codes accept a block of k information bits, perform complex algebra or finite field arithmetic, and produce a block of n code bits. These codes are represented as (n, k) codes. The encoder for a block code is memory less, which means that the n digits in each code-word depend only on each other and are independent of any information contained in previous code-word. Some of the common block codes are Hamming codes and Reed Solomon (RS) codes. On the other hand, convolutional codes are designed for real-time error correction. The code converts the input into one single code-word. The encoded bit depends on both previous bit and current bit information. Turbo codes [1][2][3**]** are examples of convolutional codes. In the past few years, several approaches to achieve low complexity Turbo-like code designs have appeared in the literature[4][5][6**]**. Modified Turbo Code (MTC) provides a good compromise between complexity and error performance. Modified Turbo Code is a concatenation of convolutional as well as block code where the convolutional code is a Recursive Systematic Convolutional (RSC) code and the block code is a Zig-Zag code.

The rest of the paper is organised as follows: Section 2 describes about RSC codes, Section 3 describes about Zig-Zag codes, Section 4 explains the Modified Turbo encoder structure as well as MATLAB simulation results and FPGA implementation results and finally Section 5 concludes the paper.

2. RECURSIVE SYSTEMATIC CONVOLUTIONAL (RSC) CODES
The RSC encoder **[2]** is obtained from a conventional non-recursive non-systematic convolutional encoder by feeding back one of its encoded outputs to its input. Figure 1 shows a conventional rate *r = 1/2* convolutional encoder with constraint length K=*3*. In the figure 'D' indicates delay and the '+' indicates modulo-2 addition.

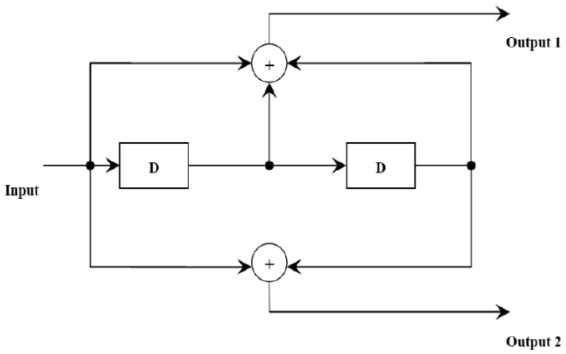

Fig 1: Conventional convolutional encoder

A generator polynomial is defined in the convolutional encoder for each adder**[6].** It shows the hardware connections of the shift register taps to the modulo-2 adders. A "1" represents a connection and a "0" represents no connection. The generator polynomials for the above conventional convolution encoder are given as *g1* = [111] and *g2* = [101] where the subscripts 1 and 2 denote the corresponding output terminals. The generator matrix of the convolutional encoder is a k-by-n matrix. The element in the *i*throw and *j*thcolumn indicates how the *i*thinput contributes to the *j*thoutput. The generator matrix of the above convolutional encoder is given by

$$G = [g1, g2] = [111,101]$$

The conventional encoder can be transformed into an RSC encoder by feeding back the first output to the input. The generator matrix of the encoder then becomes

$$G = [1, g1/g2]$$

Where '1' denotes the systematic output, feed forward output is denoted by g2 and g1 is the feedback to the input of the RSC encoder. Figure 2 shows the resulting RSC encoder of r=1/2 and K=3.

It is suggested that good codes can be obtained by setting the feedback connection of the RSC encoder to a primitive polynomial, because the primitive polynomial generates maximum-length sequences which adds randomness to the turbo code.

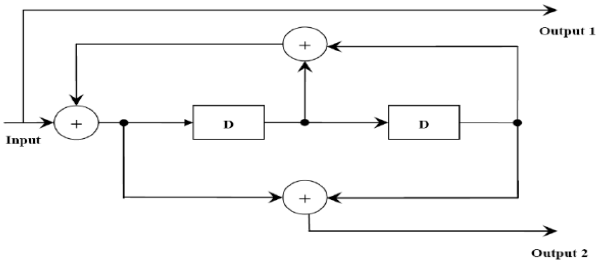

Fig 2: RSC encoder

3. ZIG-ZAG CODES

Zig-Zag codes **[7] [8]** can be described using the following graph shown in figure 3 where a sequence of N data bits are arranged in an (I*J) array where the white nodes are information bits and the black nodes are parity bits. The alphabet 'I' (rows) is used to denote the number of segments in the graph and 'J' (columns) to denote the number of information bits on each segment.

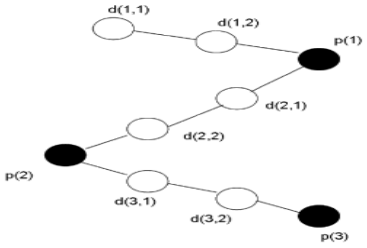

Fig 3: Zig-zag graph

In other words, 'I' denotes the total number of parity bits and 'J' denotes the number of information bits per parity bit. In this particular graph example, I = 3 and J = 2.

For example, Data Sequence = 011001; therefore number of data sequence N=6, Data is now arranged in an array of I×J, here N=I*J. Number of parity generated depends on value of 'I' (rows). So in-order to generate 3 parity bits for 6 bits of data sequence I=3 and J=2. With the aid of the below flow shown in figure 4, the parity bits are computed as follows:

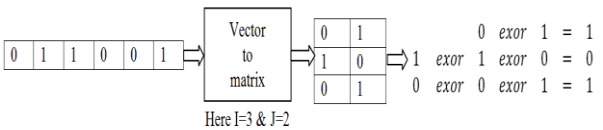

Fig 4: Calculation of Zig-Zag parity bits

To enable the encoding and decoding process, the information bits and parity bits are stored in the matrix D and vector P, respectively.

$$D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} P = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

4. TURBO ENCODER AND IMPLEMENTATION RESULTS

Figure 5 shows the overall block diagram of the modified turbo code structure. Since RSC codes have better error-correcting capability, the data bits are encoded by RSC encoder generating Systematic data *d* and RSC parity bits $r_1^{(1)}$, , $r_2^{(1)}$, , $r_L^{(1)}$.[4]

Systematic data is nothing but combination of information and trellis termination bits known as Tail bits. This systematic data is given to the zigzag encoder for generating Zig-Zag parity. Since the error-correcting capability of the zigzag code itself is weak since it has minimum distance d_{min}=2 for any pair so in order to build a powerful code, several constituent zigzag codes have to be concatenated.

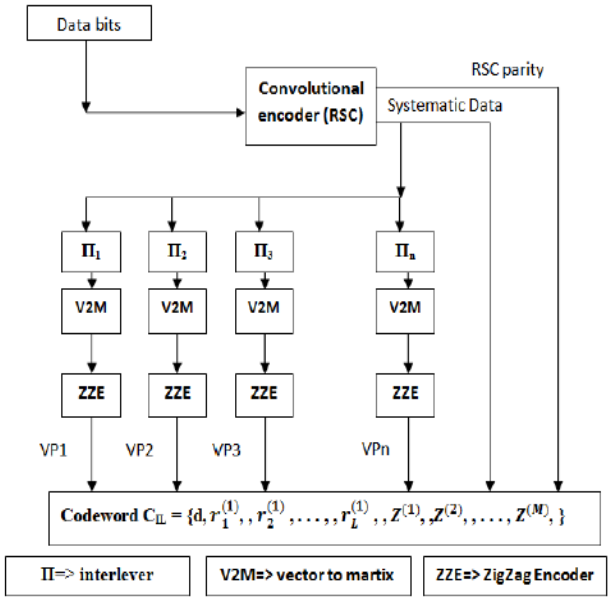

Fig 5: Modified Turbo Encoder

Parallel concatenation of M constituent encoders forms the overall encoder. Concatenated zigzag codes are attractive coding schemes since they offer excellent performance while having relatively low encoding and decoding complexity. As shown in figure 5 the systematic data from RSC are interleaved using interleaversπ_1, π_2, π_3,..., π_n and the interleaved systematic data are arranged in an array of J×K matrix and then the zigzag parity computed are computed as discussed in section 3 i.e. each generates a 1 bit parity, likewise all parity generated by different Zig-Zag Encoder (ZZE) i.e. $Z^{(1)}$, $Z^{(2)}$ . . . $Z^{(M)}$ are concatenated to form a zigzag parity. Code-word C_{IL} is formed by concatenating systematic data, RSC parity and Zig-Zag parity.

$$C_{IL} = \{d, r_1^{(1)}, r_2^{(1)} \ldots r_L^{(1)}, Z^{(1)}, Z^{(2)} \ldots Z^{(M)}\}$$

For MTC, code rate R_{IL} is given by R_{IL} =d/(2d+ (K*M)). Code rate can by adjusted by d, K and M.

4.1 Matlab Simulation Results

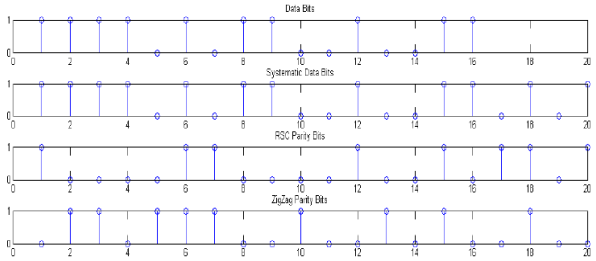

Fig 6: Output of MTC encoder

As shown in figure 6 input data bits [0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 1 1] are of 16 bits in length and the MTC encoder generates systematic data, RSC parity and Zig-Zag parity each of 20 bits in length.

4.2 FPGA Implementation Results

A Verilog code [9] is written for the encoder based on the latest ArtixNexys 4 FPGA family. From the device utilization summary the encoder used about 174 slice LUT's and 188 LUT Flip-Flop pairs.

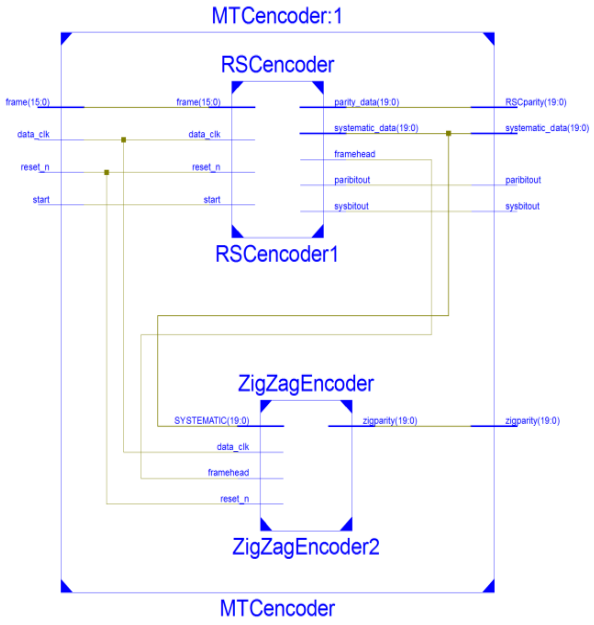

Fig 7: RTL schematic of MTC encoder

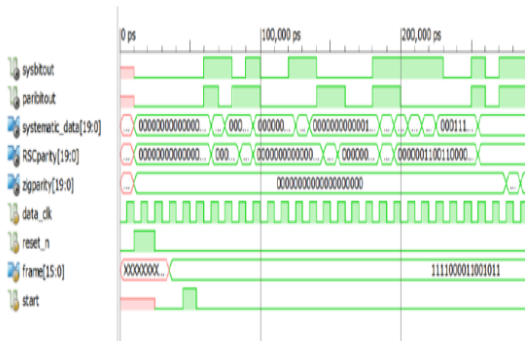

Fig 8: MTC encoder I-Sim simulation results

Simulation results of MTC encoder shown above in the graph is tabulated in table 1.

Table 1: Input and output values of encoder

ports	Bits	Input/output Data
Frame	Input(16 bits)	1111000011001011
start	Input(1 bit)	Start encoding process
Reset_n	input(1 bit)	Resets complete encoder
Data_clk	input(1 bit)	Clock source to encoder
Systematic Data	Parallel Output(20 bits)	10011111000011001011
RSC parity	Parallel Output(20 bits)	10000011001100001101
Zigparity	Parallel Output(20 bits)	10101100100110111011
sysbitout	Serial output(1 bit)	10011111000011001011
paribitout	Serial output(1 bit)	10000011001100001101

5. CONCLUSION

In this paper a new class of low complexity channel coding technique termed as Modified Turbo Code (MTC) encoding is implemented. MTC is a concatenation of block codes and convolutional codes and it was seen that concatenated zigzag codes are attractive coding schemes since they have relatively low encoding complexity as seen from the device utilization summary thus MTC was implemented using multiple concatenations of Zig-Zag codes (Block Code) and a Recursive Systematic convolutional (RSC) code. A design modification to existing MTC encoder is done where the Zig-Zag parity depends on the RSC output. This encoding structure will make the decoding process simpler at the receiver end where the Zig-Zag decoder could support the Log-Map Decoder by correcting the errors present in the tail bits which in-turn improves the performance of convolutional decoder with better improvement in trellis termination and iterative decoding process.

6. REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima., "Near Shannon limit error correcting coding and decoding: Turbo codes", IEEE International Conference on Communications, Geneva, Switzerland, May 2003.

[2] S. Lin and D. J. Costello, "Error Control Coding: Fundamentals and Applications," second edition, Prentice Hall: Englewood Cliffs, NJ, 2005.

[3] Moon T. K.: "Error correction coding- mathematical methods and algorithms", Wiley (2005).

[4] ArchanaBhise and Prakash D. Vyavahare, "Improved Low Complexity Hybrid Turbo Codes and their Performance Analysis," IEEE Transaction on Communications, Vol. 58, No. 6, June, 2010, pp.1620-1622

[5] A. Bhise and P. D. Vyavahare, "Low complexity hybrid turbo codes," in Proc. IEEE Wireless Commun.Netw. Conf., Las Vegas, pp 1525-1535, Mar. 2008.

[6] Anum Imran, "Software implementation and performance of UMTS Turbo code", Masters of Science thesis, Tampere University of Technology, March 2013.

[7] Li Ping, Xiaoling Huang, and Nam Phamdo, "Zigzag Codes and Concatenated Zigzag Codes," IEEE Trans.on Information Theory, Vol. 47, No. 2, Feb. 2001, pp. 800-807.

[8] Li Ping, "Turbo-SPC Codes," IEEE Trans. On Communications, Vol. 49, No. 5, May 2001, pp. 754-759.

[9] Peter J. Ashenden: "Digital design: An embedded systems approach using VERILOG", Morgan Kaufmann (2008).