FPGA Implementation of Turbo Codification Technique for Error Control in Communication Channels

Pablo Manrique Ramírez [1], Rafael Antonio Márquez Ramírez[2], Oleksiy Pogrebnyak[3],

Instituto Politécnico Nacional, CIC-IPN, Av.Juan de Dios Batiz s/n,
Colonia Nueva Industrial Vallejo, C.P. 07738, México D.F.
E-mail: [1]pmanriq@cic.ipn.mx, [2]ramr@super.unam.mx, [3]olek@pollux.cic.ipn.mx

Abstract. A hardware implementation of turbo coding technique for error detection and correction for data transmission is presented. The designed coder generates a more efficient transmission code with a high error correction capacity at decoder in real time. The implemented techniques provides the minimum possible energy consumption and is oriented to reach a transmission speed similar to the theoretical capacity of the communication channel "The Shannon's Limit". The turbo encoder and decoder were implemented in FPGA Spartan development system. With the implemented turbo coding, the number of erroneous bits for low signal to noise ratios decreases increasing the number of iterations starting with the second iteration.

Key words: error correction code, turbo code, FPGA implementation

1 Introduction

In the codification and information theory, an error correction code (ECC) is a code in which each one of the data signals observes specific rules of "construction", so that, depending on this construction, the received signal can be detected and be corrected automatically. It is used commonly in storage of computer data, for example in dynamic RAM, and data transmission. Some examples of ECC are: algebraic codes (block codes), Hamming code, Reed-Solomon code, Reed-Muller code, binary Golay code, convolutional code, turbo code, low density parity check code (LDPCC), etc. [1].

The simplest ECC can correct single bit errors (Single Error Correction) and detect double bit errors (double error detection). Other codes can detect or correct multiple bits errors. The two main classes of ECC are block codes and convolutional codes. In 1993 Claude Berrou, Alain Glavieux and Punya Thitimajshima from the Superior National School of Telecommunications of Bretagne, France developed the Turbo Codes, the most powerful ECC at the moment. They are a class of convolutional codes whose performance in terms of binary error rate (BER) approaches to the Shannon's limit [2].

The turbo codification techniques are based on convolutional algorithms that strengthen the data to be sent adding redundant information to data at the transmitter side,

Industrial Informatics
Research in Computing Science 31, 2007, pp. 203-212

and using iterative probability estimation algorithms at the receiver side [2],[3],[4]. They are those that more approach the theoretical limit of maximum rate of information transference on a channel with noise. We are able to aproach the AWGN capacity to a few fractions of a decibel [5]. Turbo codes does possible to increase the rate of data without the necessity to increase the transmission power, reason why also they can be used to diminish the amount of used energy to transmit to a certain rate of data.

Turbo codes can be used in various applications where one look for to obtain a maximum information transference in a limited bandwidth channel with the presence of noise. Such applications, for example, are: standards of cellular telephony of 3rd generation, satellite communications, future standards of satellite and mobile television, DVB-S (digital video broadcasting - satellite), DVB-H (digital broadcasting video - handheld) [6]. Now, turbo codes replace the Reed-Solomon codes used in telecommunications. In some future missions of the NASA space reconnaissance, the turbo codes will be used as a standard replacing RSV concatenated codes.

Actually, the refinement and implementation of the turbo codes are an active area of research in many universities and research centers.. In the paper, an attempt of efficient FPGA implementation of a turbo coder and decoder is presented.

2 Design of turbo coding

The Shannon's theorem is important in error correction. It describes the maximum attainable efficiency of an error correction scheme against the awaited noise interference levels as:

$$C = \lim_{T\to\infty} \frac{\log N(T)}{T}$$ were N(T) is the number of allowed signals of duration T . (1)

If the number of errors is smaller or equal to the correctable maximum threshold of the code, all the errors will be corrected. Thus, the error correction codes require more signal elements than necessary ones to transport the basic information [7], [8].

2.1 Encoder

The encoder is designed using a parallel concatenation of two recursive systematic encoders (RSC) and the decoder uses decoding rules on the base of iterative estimations of probabilities using identical blocks of decoding [9].

Fig. 1 shows an example of two RSC encoders that use a parallel concatenation scheme. Both elementary encoders (C_1 and C_2) use the same input signal d_k but in different sequence due to the presence of an interleaver. Thus, the turbo encoder can input the original data X_k followed by Y_{1k} sequence and later by Y_{2k} sequence. This way, the redundant parity information is added to the transmitted data making it more robust under the noise effects that will be added in the channel.

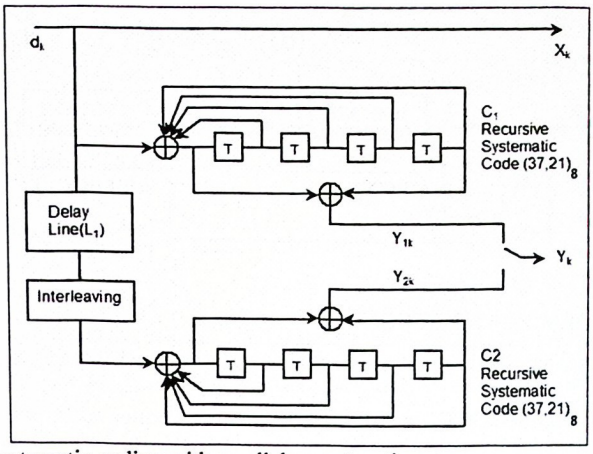

Fig. 1 Recursive systematic coding with parallel concatenation

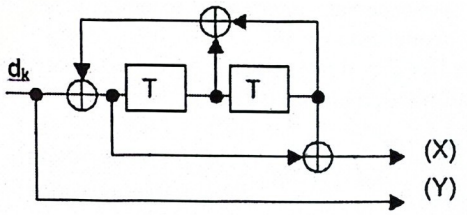

Fig. 2 RSC encoder with generators $[1, 05/07]_8$

The RSC encoder in Fig 2 was used as encoder component for the implementation. The interleaver process consists in generating a square matrix and arranging all the bits by row and reading by column in bit-reversing order. Although there are other ways and other pattern for the interleaver could be used, different patterns provide different results with significant BER differences, reason why the design of the interleaver significantly contributes in the general performance of a turbo code system.

2.2 Decoder

The original turbo decoder, is an array of decoders in a serial concatenation scheme united by interleaver [9].

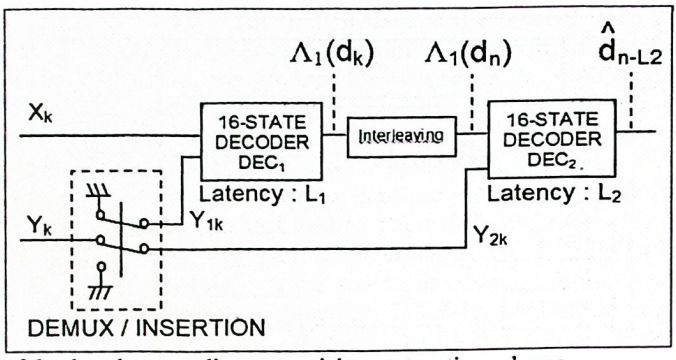

Fig. 3 Principle of the decoder according to a serial concatenation scheme

In Fig. 3, the decoder DEC_1 is associated with the sequence Y_1 and generates a soft decision. The decoding is made simultaneously for the full scope of the code word sequence. Such decoding provides good results, but it is not always feasible.

The output DEC_1 sequences are interleaved to enter at second decoder DEC_2 that is associated with the Y_2 sequences.

The logarithm of the likelihood ratio (LLR), $\Lambda_1(d_k)$ associated with each decoded bit by DEC_1 is an relevant information piece for DEC_2.

$$\Lambda_1(d_k) = \log \frac{P_r\{d_k = 1 \mid observation\}}{P_r\{d_k = 0 \mid observation\}} \tag{2}$$

where $P_r\{d_k=d \mid observation\}$, d=0,1 is the d_k data bit of the aposteriori probability (APP), and the observed data set is received from the transmission channel (y_1^N). Thus, the final decision of the decoded data becomes to be based on the sign of Λ_1.

As decoders are convolutional, these will have a certain amount of memory reason why they can be considered like "state machines", then, at time of deciding if d_k data is a 1 or a 0 the decoder will be in some of those states. $S_k = s$. Considering this already can be defined the conditional probabilities of the possible d_k value since the data set (y_1^N) has been received; and if in addition we considered that the received data set can be separated in the observed data before moment k, the present observation at moment k and the future observation after moment k

$$y_1^N = \left\{ y_1^{k-1}, y_k, y_{k+1}^N \right\} \tag{3}$$

it is possible to pass (2) to

$$\Lambda(d_k) = \log\left(\frac{P(d_k = 1, y_1^N)/P(y_1^N)}{P(d_k = 0, y_1^N)/P(y_1^N)}\right) = \log\left(\frac{\sum_{s'} P(s_{k-1} = s', d_k = 1, y_1^N)/P(y_1^N)}{\sum_{s'} P(s_{k-1} = s', d_k = 0, y_1^N)/P(y_1^N)}\right).$$

With a little manipulation of course, and using some of the BAHL's concepts in the algorithm [4], LLR becames to

$$\Lambda(d_k) = \log\left(\frac{\sum_{d=1} P(s_{k-1} = s', d_k = d, y_1^{k-1}, y_k, y_{k+1}^N)/P(y_1^N)}{\sum_{d=0} P(s_{k-1} = s', d_k = d, y_1^{k-1}, y_k, y_{k+1}^N)/P(y_1^N)}\right),$$

$$\Lambda(d_k) = \log\left(\frac{\sum_{d=1} P(s_{k-1} = s', y_1^{k-1})P(y_k^N / s_{k-1} = s')P(d_k = d, y_k / s_{k-1} = s')/P(y_1^N)}{\sum_{d=0} P(s_{k-1} = s', y_1^{k-1})P(y_k^N / s_{k-1} = s')P(d_k = d, y_k / s_{k-1} = s')/P(y_1^N)}\right).$$

With a little more manipulations and doing:

$$\alpha_k(s) = P(s_k = s, y_1^k)$$

$$\beta_{k-1}(s') = P(y_k^N / s_{k-1} = s')$$

$$\gamma_k(s', s) = P(d_k = d, y_k / s_{k-1} = s')$$

one can finally obtain LLR as

$$\Lambda(d_k) = \log\left(\frac{\sum_{d=1} \alpha_{k-1}(s')\beta_k(s)\gamma_k(s', s)/P(y_1^N)}{\sum_{d=0} \alpha_{k-1}(s')\beta_k(s)\gamma_k(s', s)/P(y_1^N)}\right) \quad (5)$$

where: $\alpha_k(s)$ is the probabilities join in s state in a k time given the last observations (fordward metric [2]), $\beta_{k-1}(s')$ is the conditional probability of the future observations given s' state in a k-1 time (reverse metric [2]), $\gamma_k(s', s)$ is the transition probability that the s_{k-1} state transits to s_k in a time k caused by d_k input.

Making simplifications and a little more of manipulation, a recursive version of $\alpha_k(s)$ and $\beta_{k-1}(s')$ can be obtained

$$\alpha_k(s) = \sum_{s'/s_k = s} P(s_{k-1} = s', y_1^{k-1})P(d_k = d, y_k / s_{k-1} = s')$$

$$\alpha_k(s) = \sum_{s'/s_k = s} \alpha_{k-1}(s')\gamma_k(s', s) = \sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s),$$

$$\beta_k s_{-1}(\ ') = \sum_{s/s_{k-1}=s'} P\,)(\ {}^{N}_{k+1}s'\ _k \mp\ )P(d_k = d, y_k / s_{k-1} = s')$$

$$\beta_{k-1}(s') = \sum_{s/s_{k-1}=s'} \beta_k(s)\gamma_k(s', s) \tag{6}$$

with the initial conditions $\alpha_0(s) = \begin{Bmatrix} 1 & s=1 \\ 0 & s \neq 1 \end{Bmatrix}, \beta_0 = \begin{Bmatrix} 1 & s=1 \\ 0 & s \neq 1 \end{Bmatrix}$.

Since the forward and backward probabilities are expressed in a recursive form, it is possible to think in a recursive scheme that could be of two forms: with feedback loop and concatenated. Fig. 4 [9] shows the feedback decoder, where W_{2k} represents the forward and backward probabilities that now are feedback to DEC_1 like a third input parameter (z_k).

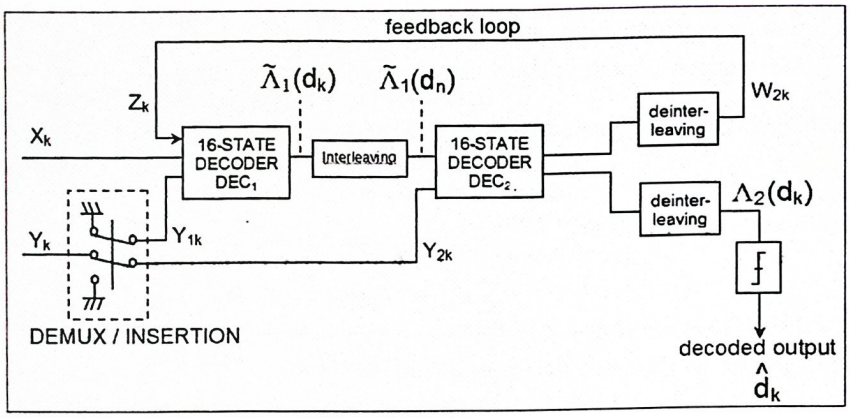

Fig. 4 Feedback decoder (assuming a 0 internal delay)

Because the first decoder DEC receives additional redundant information (z_k), this can improve the performance significantly. The turbo code term arises from this iterative scheme of decoder, remembering the turbo ignition principle in motors. Note this additional information (*extrinsic* [2], [9]) comes from an iterative previous step.

For a "modular concept" the decoder generates a delay caused by DEC_1 and DEC_2. The interleaver and deinterleaver imply that z_k information must be used through an iterative process as shown in Fig. 5, where global decoder circuit is compound of P serially concatenated identical elementary decoders. The p^{th} DEC decoder input is formed by the output sequence of demodulator $(y)_p$ through a delay line and an extrinsic information $(z)_p$ generated by the (p-1)th DEC decoder.

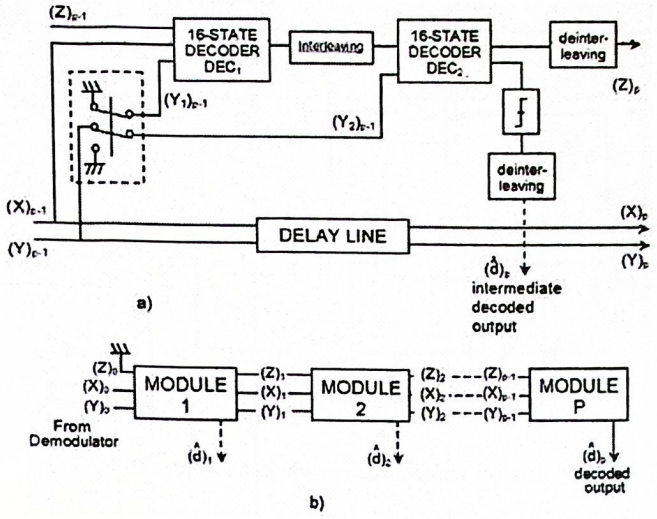

Fig. 5 a) Module decoder (level p); b) modular decoder corresponding to an iterative process of feedback decoding

3 Implementation and results

The turbo encoder and decoder was implemented in Xilinx FPGA Spartan3 development system. The turbo encoder was integrated with both encoders RSC in Fig 2 concatenated in parallel and separated by the interleaver as in Fig 1. At the output of the encoder, the original data X_k are followed by the first parity sequence Y_{1k} next by the second parity sequence Y_{2k}. The original information and the parity information redundant data are passed through the line encoder to the power line transmission channel as in Fig 7, and are transmitted in a manner more robust to the noise effects that will be added by the channel. At the channel output, the signal is mixed with additive channel noise, this signal passes through the line decoder to determine the metric that will be submited to the statistical evaluation of the turbo decoders ("maximum a priori" MAP decoder, see Fig 6).

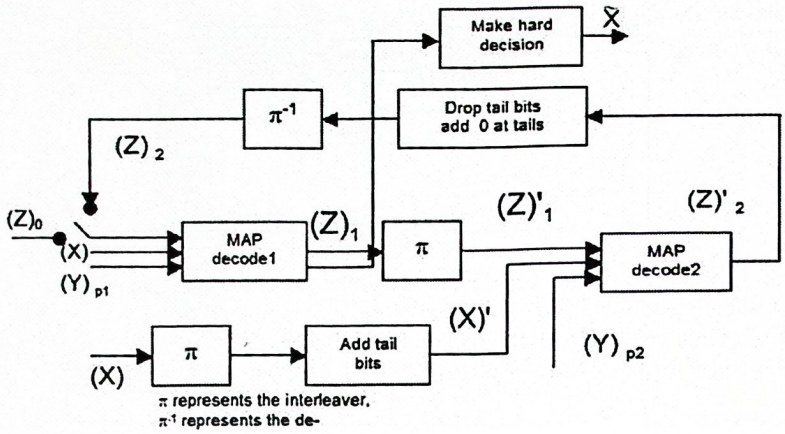

Fig. 6 Turbo decoder with iterative serial concatenation implemented in FPGA

Figure 6 shows the block diagram of the turbo decoder implemented, the most complex component is the MAP decoders, because there occurs the most intensive computing process. This process implies sums, multiplications, divisions, logarithms and exponents. Due to the nature of the MAP process the forward, backward and transition metrics can be estimated in parallel with the respective limitations of hardware resources. The logarithms and exponents are computed with iterative algorithms that restrict the performance.

The results of Berrou et al.[9] reproduced in Fig 8 show that for any signal to noise ratio greater than 0 dB, BER decreases as function of iterations p. The codification gain is high for the first values of p (p = 1, 2, 3). Thus, for p=18, for example, BER is smaller than 10^{-5} with a signal to noise ratio E_b/N_0= 0,7 dB. Shannon establish that, for a binary modulation with .5 transmission rate, P_e=0 for E_b/N_0=0 dB (several authors take $P_e = 10^{-5}$ as a reference). With a parallel concatenation of encoders RSC and feedback decoding, the performance is at 0,7 dB from Shannon's limit. Modifying the amount of memory of encoders and decoders one can obtain some degradation and inefficiencies in BER and the correction capacity added by the encoders C_1 and C_2 [9].

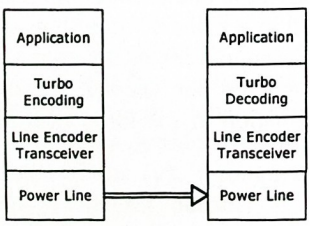

Fig. 7 Block diagram of turbo encoder-decoder test system

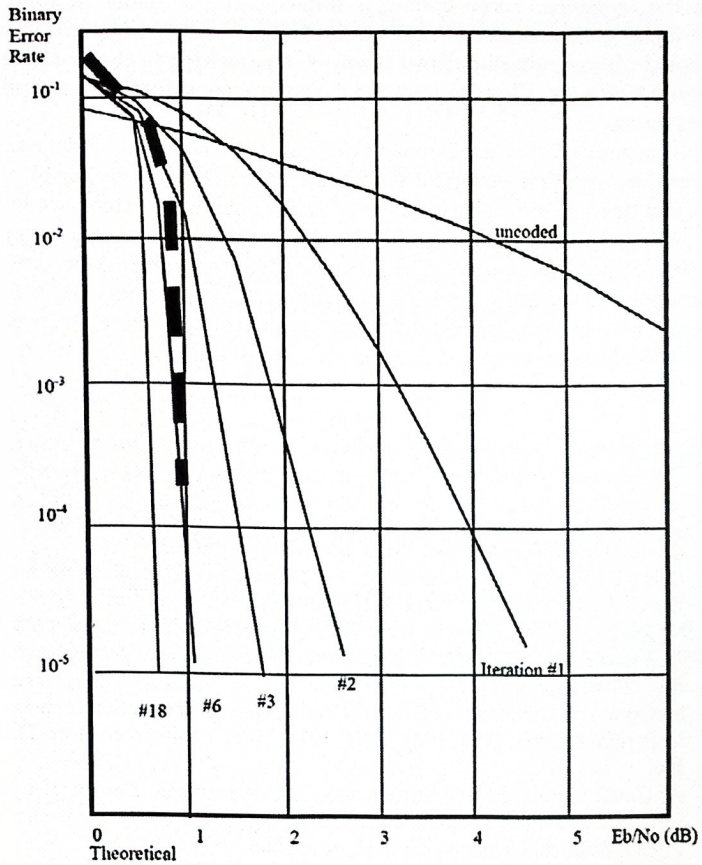

Fig. 8 Binary error rate given by iterative decoding (p=1,...,18) Of code of Fig. 1 (rate 1/2) interleaving (256x256).

Fig. 8 shows the obtained at moment by author proper results presented by the segmented bold line, for block sizes of 64 bit in approximately 6 iterations.

5 Conclusions

The development of turbo coding techniques is an intense activity in different academic and research centers. Recently, much of improvements, such as inventions, algorithms for encoders and decoders, save energy, diminution of components etc. are

presented in the literature. Turbo coding is implemented in many of high technology products that imply great amounts of data manipulation, not only in telecommunications.

The presented turbo coding implementation is approaching to the possible theoretical limit, although it needs the efforts to improve the energy consumption and optimization of computing resources.

The initial purpose of this development was an improvement of data transmissions between controllers using the electrical power line as a channel. Obviously, there are too many factors that degrade the signal quality. With this encoding system, we hope decrease the power consumed for line coding with the decrement of necessary components and energy for line coding of cousrse. The line encoder transceiver was developed with discrete components: opams, network transformers, capacitors, fast commutation transistors, etc. (very cheap). The final application depends on the existed demands that are out of our objectives now.

References

1. David J.C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, ISBN 0-521-64298-1 (2003)
2. Sklar, B., Digital Communications: Fundamentals and Applications,Second Edition, Upper Saddle River, NJ: Prentice-Hall, 2001.
3. Nguyen, Quang. High speed turbo codes decoder for 3G using pipelined SISO Log-map decoders architecture. *United States ICOM Technol ogies, Inc. (Wilmington, D)*, 2004, Patent 6813742.
4. McGraw-Hill Dictionary of Scientific and Technical Terms. Sci-Tech-Dictionary Copyright © 2003, 1994, 1989, 1984, 1978, 1976, 1974 by Mc-Graw Hill Companies Inc.
5. A. Graell i Amat, F. Brännström, and L. K. Rasmussen, Design of rate-compatible serially concatenated convolutional codes, *Int. Symp. on Turbo Codes and Related Topics*, Munich, Germany, paper 13, Apr. 2006.
6. Xilinx Inc. 3GPP2 Turbo Decoder v2.1. Product Specification LogiCORE Facts DS275, February 15, 2007.
7. Claude E. Shannon: A Mathematical Theory of Communication, *Bl System Technical Journal, Vol. 2* , pp. 379–423, 623–656, 1948.
8. C.E. Shannon, Communication in the presence of noise. *Proc. of the IRE, Vol. 37* pp.1021, Jan. 1949
9. C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. *In Proc. 9 International Conference on Communications (ICC9)* , Geneva, Switzerland, 1993.
10. L.R.Bahl, J.Cocke, F.Jelinek, and J.Racic. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans, Inform. Theory, Vol. IT-0* pp.284-287, 1974.