International Journal of Electronics Signals and Systems

Volume 4 | Issue 1 Article 7

July 2014

DESIGN AND IMPLEMENTATION OF TURBO CODER FOR LTE ON FPGA

SANTOSH GOORU
Dept. of Electronics and Communication Engineering, Thiagarajar college of Engineering, Madurai, India, santosh.gooru@gmail.com

DR. S. RAJARAM
Dept. of Electronics and Communication Engineering, Thiagarajar college of Engineering, Madurai, India, rajaram_siva@tce.edu

Recommended Citation

GOORU, SANTOSH and RAJARAM, DR. S. (2014) "DESIGN AND IMPLEMENTATION OF TURBO CODER FOR LTE ON FPGA," *International Journal of Electronics Signals and Systems*: Vol. 4 : Iss. 1 , Article 7.
DOI: 10.47893/IJESS.2014.1196
Available at: https://www.interscience.in/ijess/vol4/iss1/7

DESIGN AND IMPLEMENTATION OF TURBO CODER FOR LTE ON FPGA

GOORU SANTOSH[1], DR. S. RAJARAM[2]

[1]M.E. student, [2]Associate professor
Dept. of Electronics and Communication Engineering, Thiagarajar college of Engineering, Madurai, India
E-mail: santosh.gooru@gmail.com, rajaram_siva@tce.edu

Abstract- Recent wireless communication standards such as 3GPP-LTE, WiMax, DVB-SH and HSPA incorporates turbo code for its excellent performance. This work provides an overview of the novel class of channel codes referred to as turbo codes, which have been shown to be capable of performing close to the Shannon Limit. It starts with a brief discussion on turbo encoding, and then move on to describing the form of the iterative decoder most commonly used to decode turbo codes. Here, Turbo decoder uses original MAP algorithm instead of using the approximated Max log-MAP algorithm thereby it reduces the number iterations to decode the transmitted information bits. This paper presents the FPGA (Field Programmable Gate Array) implementation simulation results for Turbo encoder and decoder structure for 3GPP-LTE standard.

Keywords- *Turbo codes, Turbo Encoder, Turbo Decoder, SISO(Soft Input Soft Output) Decoder, Iterative Decoder, MAP, Channel coding, Convolutional codes, FPGA (Field Programmable Gate Array), 3rd Generation Partnership Project (3GPP),Long Term Evolution (LTE)*

I. INTRODUCTION

Modern wireless communication standards rely on powerful channel coding to ensure reliable (error free) transmission. Channel coding introduces a controlled amount of redundancy into the transmitted data stream, which is then exploited in the receiver to correct transmission errors induced by noise or interference present in the wireless channel. Turbo codes, first proposed in 1993 [1], represent a breakthrough in channel coding techniques, since they have the potential to enable data transmission at rates close to the Shannon limit. They have been adopted for error control coding in the high speed downlink packet access (HSDPA) standard by the third-generation partnership project (3GPP), which considerably enhance the throughput for data-centric 3G modems.LTE specifies the use of turbo-codes to ensure reliable communication.

Turbo codes were introduced in 1993 by Berrou, Glavieux and Thitimajashima [1], [2], reported extremely impressive results for a code with a long frame length. Since its recent invention, turbo coding has evolved at an unprecedented rate and has reached a state of maturity within just a few years due to the intensive research efforts of the turbo coding community. Simply put,a turbo code is formed from the parallel concatenation of two codes separated byan interleaver. Although the generalconcept allows for free choice of the encoders and the interleaver, most designsfollow the ideas:

- The two encoders used are normally identical
- The code is in a systematic form, i.e. the input bits also occur in the output
- The interleaver reads the bits in a pseudo-random order.

The remainder of the paper is organized as follows. Section II reviews the principles of turbo-encoding and decoding and details the algorithm used for SISO decoding and interleaver architecture presented. The FPGA implementation Architecture presented in section III and Simulation results for turbo codes presented in section IV and we concluded in section V.

II. SYSTEM OVERVIEW

A. TURBO CODES

Turbo codes, capable of achieving close-to-Shannon capacity and amenable to hardware-efficient implementation, have been adopted by many wireless communication standards, including HSDPA and LTE. The turbo encoder specified in the LTE standard is illustrated in Figure 1 and consists of a feed-through, two 4-state recursive convolutional encoders (CEs), and an interleaver. LTE employs a rate 1/3 parallel concatenated turbo code. The corresponding encoder is comprised of two rate 1/2 recursive systematic convolutional encoders, as shown in Figure 1.

The first component encoder receives un-coded (systematic) data bits x_k in natural order and outputs a set of parity bits x_k^{p1}.The second CE receives an interleaved sequence $x_{\pi(k)}$ of the information bits, where stands for the interleaved address associated with address, and generates a second sequence of

parity bits . The systematic bits and the two sets of parity bits are then modulated onto an analog waveform (according to the employed communication standard) and sent over the radio channel. On the other side of the wireless link, a demodulator is responsible for the reconstruction of the transmitted bits from the received signal. However, since this signal is usually distorted by noise and interference, the demodulator can only obtain estimates of the systematic and two sets of parity bits. These estimates are provided to the subsequent turbo decoder in the form of log-likelihood ratios (LLRs), and which express the ratio between the probabilities of the transmitted bits being 0 and being 1, given the received analog signal.

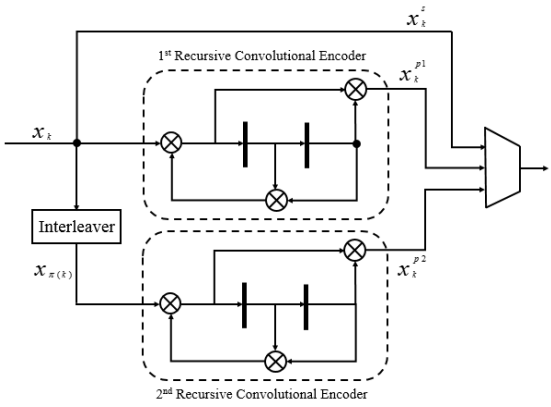

Figure 1.Architecture of Turbo Encoder

B. TURBO DECODING ALGORITHM

Decoding of turbo codes is usually performed with the BCJR algorithm.The basic idea behind the turbo decoding algorithm is to iterate between two soft-input soft-output (SISO) component decoders as illustrated in Figure 2. It consists of a pair of decoders which work cooperatively in order to refine and improve the estimate of the original information bits. The first and second SISO Decoder performs decoding of the convolutional code generated by the first or the second CE, respectively. A turbo-iteration corresponds to one pass of the first component decoder followed by a pass of the second component decoder. The operation performed by a single component decoder is referred to as a half-iteration.

The component decoders compute a-posteriori probabilities of the transmitted systematic bits from the LLRs of the (interleaved) systematic bits, the associated parity bits and the a-priori information). The latter is set to zero for the first half-iteration in the first turbo iteration. In subsequent iterations, each component decoder uses the so-called extrinsic information) of the other component decoder in the preceding half-iteration as a-priori information.

$$L_e(x_k) = L_{map}(x_k) - \{L_a(x_k) + L_c * (x_k^s)\} \qquad (1)$$

The decoder information is cycled around the loop until the soft decisions converge on a stable set of values. The latter soft decisions are then sliced to recover theoriginal binary sequence.

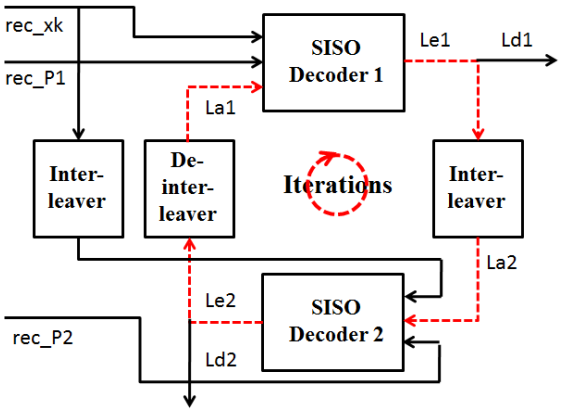

Figure 2.Architecture of Turbo decoder

C. SISO DECODER

A soft-in-soft-out (SISO) decoder receives as input a "soft" (i.e. real) value ofthe signal. The decoder then outputs for each data bit an estimate expressing theprobability that the transmitted data bit was equal to one. In the turbo-decoder under consideration in this paper, the maximum a-posteriori (MAP) algorithm of Bahl, Cocke, Jelinek and Raviv [7] (BCJR algorithm) is used for the SISO component decoders, because it shows the best error rate performance. The MAP algorithm minimizes the probability of bit error by using the entire received sequence to identify the most probable bit at each stage of the trellis. The MAP algorithm does not constrain the set of bit estimates to necessarilycorrespond to a valid path through the trellis. So the results can differ from those generated by a Viterbi decoder whichidentifies the most probable valid path through the trellis. The MAP soft decisions are defined as the log likelihood ratio:

$$L_{map}(x_k) = \text{Log}\left(\frac{\Pr[x_k = +1|y]}{\Pr[x_k = -1|y]}\right) (2)$$

The numerator sum is over all possible state transitions associated with a '1' data bit, and the denominator over all possible state transitions associated with a '0' data bit.After simplification a posteriori likelihood ratio can be written as

$$L_{map}(x_k) = \log\left(\frac{\sum_{(s',s):\, x_k^s=1} \alpha_{k-1}(s').\gamma_k(s',s).\beta_k(s)}{\sum_{(s',s):\, x_k^s=0} \alpha_{k-1}(s').\gamma_k(s',s).\beta_k(s)}\right) (3)$$

Where $\gamma_k(s',s)$ represents the Branch metric computation for trellis at time instant 'k' moving from predecessor state s' to present state s can be calculated by

$$\gamma_k(s',s) = \exp[\tfrac{1}{2}(x_k * L_a(x_k) + x_k * L_c * x_k' + P_k * L_c * P_k')] (4)$$

and for AWGN channel $L_c = \frac{2}{\sigma^2}$

Here L_c-channel constant, σ – channel variance

$\alpha_k(s')$represents the forward state metric computed during forward recursions can be calculated by

$$\alpha_{k-1}(s') = \sum_{s} \alpha_k(s) * \gamma_k(s', s) \quad (5)$$

and $\beta_k(s)$ represents the backward state metric computed during backward recursions can be calculated by

$$\beta_{k-1}(s') = \sum_{s} \beta_k(s) * \gamma_k(s', s) \quad (6)$$

From the above computed LLR's we can find the extrinsic information using the equation (1), and interleave these results in a-priori information to the other decoder.

D. INTERLEAVER

The choice of the interleaver is a crucial part in the turbo code design.Interleavers scramble data in a pseudo-random order to minimize the correlation of neighboring bits at the input of the convolutional encoders.In this work, Quadratic Polynomial Permutation (QPP) interleaver is used. For an information block size K, address computation of QPP interleaver of size K is defined by the following polynomial

$$\pi(i) = (f_1 * i + f_2 * i^2) \bmod K \quad (7)$$

Where $0 \leq i \leq K-1$ the sequential index of the bit position after interleaving is, $\pi(i)$ is the bit index before interleaving corresponding to position 'i', and f_1 and f_2 are the coefficients that define the permutation. Possibilities of these coefficients are related to the factorization of K. For e.g., when K is even, the conditions are:

- f_1 is odd (relatively prime to K), and
- All prime factors of K are also factors of f_2.

III. FPGA IMPLEMENTATION

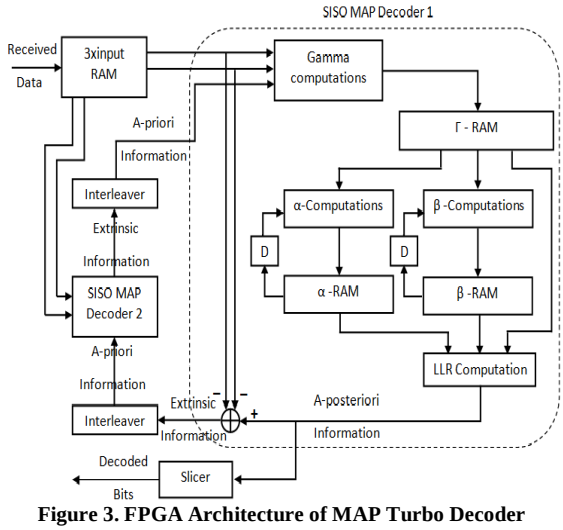

Figure 3. FPGA Architecture of MAP Turbo Decoder

The above Figure 3 represents the Field Programmable Gate Array (FPGA) architecture of Turbo decoder structure using the MAP algorithm. The '3xinput RAM' receives and stores the systematic and parity bits sent through the channel. After receiving data, 'Gamma computation' calculates all the branch metrics according to the trellis diagram for all states using equation (4) and stores them in the 'Γ-RAM'. The 'α-computation' calculates the forward state metrics using above mentioned equation (5) and computes during the forward recursions. The 'β-computation' calculates the backward state metrics using above mentioned equation (6) and computes during the backward recursions. The log likelihood ratio is computed by using the 'LLR computation' which uses the equation (3). The generated LLR is also known as the a-priori information generated from the corresponding SISO decoder. The extrinsic information can be generated by using the equation (1) and it can be done by adder shown in the above figure3. After interleaving, the generated extrinsic information acts as a-priori information to the other SISO MAP decoder. Similarly SISO MAP Decoder 2 also performs the same above operation but it takes the inputs as interleaved version of systematic and parity 2 bits. After some iterations, extrinsic information generated from both the decoders converges to some fixed values, then no need to go for the further iterations and slicer decodes the information bits from the soft a-posteriori information

IV. RESULTS

E. TURBO ENCODER OUTPUT

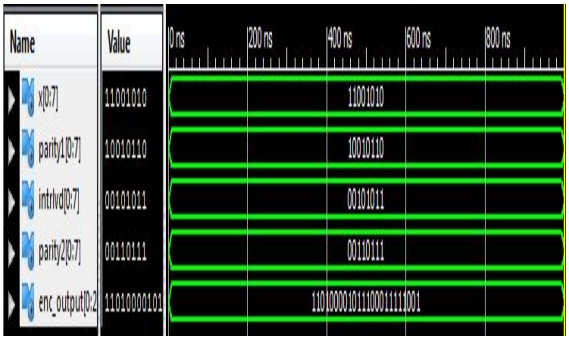

Figure 4. Output of Turbo encoder

x [in] - Input information bits (or)Systematic bits
intrlvd [out] - Interleaved information bits
parity1 [out] - Parity bits generated from the 1st Recursive Convolutional Encoder
parity2 [out] - Parity bits generated from the 2nd Recursive Convolutional Encoder
enc_output [out] - Rate 1/3 Turbo code generated from the Turbo encoder which is concatenation of systematic bits, parity1 bits and parity2 bits.

The information bits and the parity bits are mapped to symbols thentransmit over the wireless medium.

F. SISO DECODER OUTPUT

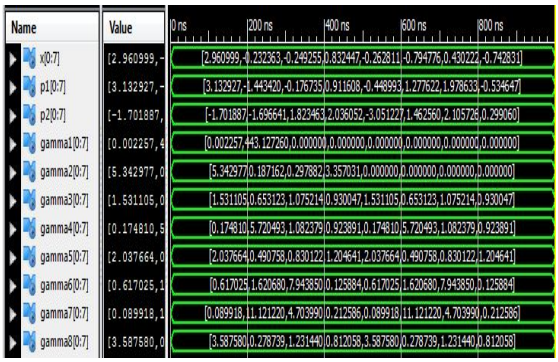

Figure 5.Simulation Result of Branch Metrics

x [in] - Received Systematic bits
p1 [in] - Received Parity1 bits
p2 [in] - Received Parity bits
gamma1 to gamma8 [out] - Computed Branch metrics of iteration 1 for SISO decoder 1 using the equation (4).

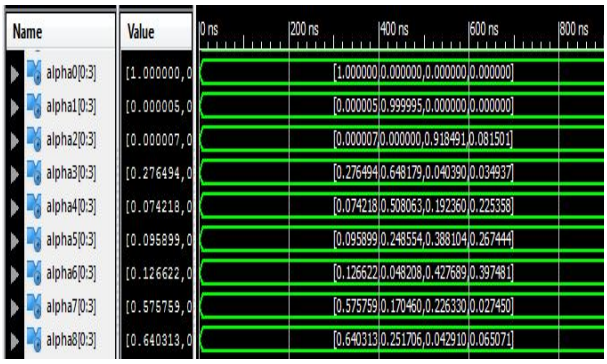

Figure 6.Simulation Result of Forward State Metrics

alpha0 to alpha8 [out] – Computed Forward state metrics of itrearatin1 for SISO decoder1 using the equation (5).

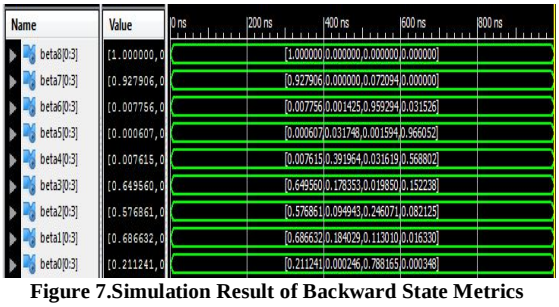

Figure 7.Simulation Result of Backward State Metrics

beta8 to beta0 [out] - Computed Backward state metrics of iteration1 for SISO decoder1 using the equation (6).

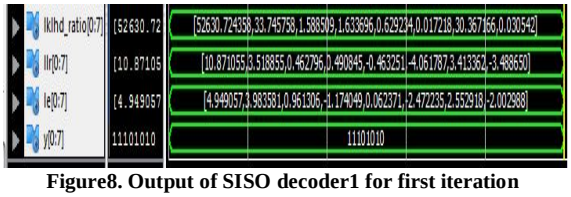

Figure8. Output of SISO decoder1 for first iteration

llr [out] - Log likelihood ratio ( )
le [out] - Extrinsic information generated from the decoder1 for 1st iteration
y [out] - Decoded information bits from decoder1 for first iteration.

In figure 8 likelihood ratios are computed according to equation (3) using the above shown simulation results of Branch metrics, Forward state metrics and Backward state metrics.

G. TURBO DECODER OUTPUT

In the figure 9 , p1, p2 represents the received noisy systematic, parity1, parity2 bits respectively, which are inputs to the turbo decoder. 'le_dcdr1_itrn1' represents the extrinsic information generated from the SISO decoder1 for the first iteration. The decoded information bits are same as transmitted bits except 1-bit error in the 3rd position. 'y_dcdr1_itrn1' represents the decoded information bits from the decoder1. Hence, the further iterations will correct the existing errors and generates the stable convergence results.'le_dcdr2_itrn1' represents the extrinsic information generated from the SISO decoder2 for the first iteration; 'y_dcdr2_itrn1' represents the decoded information bits from the decoder1.These are same as transmitted information bits but it is in the interleaved form. Even if proceed for further iterations will also produce the same results.

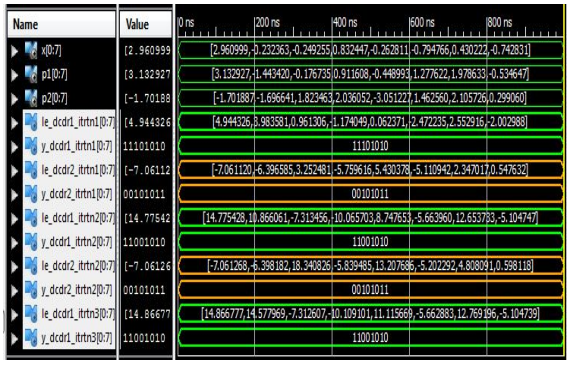

Figure 9. Output of the Turbo Decoder

V. SYNTHESIS REPORT

Table I Device Utilization Summary

Logic utilization	Used	Available	Utilization
Number of slice LUTs	139258	303600	45%
Number of bonded IOBs	1200	700	184%
Number of DSP48E1s	456	2800	16%

VI. CONCLUSION

This work provides brief discussion on one of the optimal channel coding technique such as Turbo

codes and gave brief analysis about the turbo encoder and decoder structure.n the turbo encoder side, FPGA simulation results have been generated for the parity bits of each individual convolutional encoder and overall rate 1/3 concatenated turbo code also generated and the simulation results are verified with the manual calculations.With respect to the turbo decoder, SISO decoder uses the MAP algorithm for decoding process. The SISO decoder FPGA results presented and also shown the turbo decoder output results for multiple iterations and these results are verified with the manual calculations.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding. Turbo codes," in Proc. Int. Conf.Communications, May 1993, pp. 1064–1070.

[2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," IEEE Trans. Commun., vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[3] ChristophStuder, Christian Benkeser, SandroBelfanti and QuitingHuang "Design and Implementation of Parallel Turbo Decoder for 3GPP-LTE," IEEE Journal of solid-state circuits, VOL. 46, NO. 1, JANUARY 2011.

[4] C. Benkeser, A. Burg, T. Cupaiuolo, and Q. Huang, "Design and optimization of an HSDPA turbo decoder ASIC," IEEE JSSC, vol. 44, no.1, pp. 98–106, Jan. 2008.

[5] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, "A 390 Mb/s 3.57mm 3GPP-LTE turbo decoder ASIC in 0.13 µm CMOS," in Proc.IEEE ISSCC Dig. Tech. Papers, San Francisco, CA, USA, Feb. 2010,vol. 1, pp. 274–275.

[6] J.H. Kim and I.-C. Park, "A unified parallel radix-4 turbo decoder for mobileWiMAX and 3GPP-LTE," in Proc. CICC, San Jose, CA, USA, Sep. 2009, pp. 487–490.

[7] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inf. Theory, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[8] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Inf. Theory, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[9] J. P. Woodard and L. Hanzo, "Comparative study of turbo decoding techniques: An overview," IEEE Trans. Veh. Tech., vol. 49, no. 6, pp. 2208–2233, Nov. 2000.

[10] R. Dobkin, M. Peleg, and R. Ginosar, "Parallel VLSI architecture for MAP turbo decoder," in Proc. IEEE Int. Symp. PIMRC, Sep. 2002, vol. 1, pp. 384–388.