IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Design of Turbo Coder for LTE using Verilog HDL

1. Dr.MD.Hayath Rajvee, 2.D.VenkataMalleswari, 3.K.Nandini, 4.V.SriNeeha, 5.V.Sujatha, 6.D.Yogeswar Reddy

#1 Professor in Department of ECE, PBR Visvodaya Institute of Technology and Science,Kavali.

#2#3#4#5#6 B.Tech in ECE, PBR Visvodaya Institute of Technology and Science, Kavali.

***Abstract*—Turbo codes are error correction codes that are widely used in communication systems.Turbo codes exhibits high error correction capability as compared with other error correction codes. This paper proposes a Very Large Scale Integration (VLSI) architecture for the implementation of Turbo decoder. Soft-in-soft out decoders, interleavers and deinterleavers is used in the decoder side which employs Maximum-a-Posteriori (MAP) algorithm. The number of iterations required to decode the information bits being transmitted is reduced by the use of MAP algorithm. For the encoder part, this paper uses a system which contains two Recursive convolutional encoders along with pseudo-random interleaver in encoder side.The Turbo encoding and decoding is done using Octave, Xilinx Vivado, Cadence tools.The system is implemented and synthesized in Application Specific Integrated Circuit (ASIC).Timing analysis has been done and GDSII file has been generated.**

***Index Terms*—Turbo codes, Channel coding, Interleaver, SISO, Iterative decoder, MAP, Cadence, NClaunch, Xilinx Vivado**

I . INTRODUCTION

In a communication system, when data is transferred from the source system to a destination system, errors can be present in the received signal at the source end. So error correction is required to retrieve the original message.Turbo codes, which were first introduced in 1993, represent a quantum leap in channel coding techniques and a turning point for modern digital telecommunication.Turbo codes is one of existing powerful error correcting codes.Turbo codes has inspired the coding community with the possibility of using an iterative decoding technique that relies solely on simple constituent code to achieve close channel capacity.

Turbo coder architecture (Fig 1)comprises of turbo encoder and turbo decoder. Encoder consists of two Recursive Convolutional Encoders(RSC) and interleaver. In this paper, pseudo-random interleaver is used due to which the interleaved version of the code tends to be long and scrambled, that gives good performance of random codes. In turbo code implementation, RSC encoders are employed rather than conventional convolutional encoders since it generates low weight parity codes. MAP algorithm is used for the decoding of turbo encoded data in which errors are intentionally added and verified an error free decoded data after decoding.

II. LITERATURE REVIEW

Kavinilavu, Salivahanan, and Bhaaskaran in their work [1] develop and integrates the Convolutional encoder and Viterbi decoder. In ModelSim 10.0e they have planned and modeled and synthesized using XILINX-ISE 12.4i.Max Log MAP algorithm-based turbo decoder output variations on implemented with fixed point, Vedic and Booth multipliers have been presented by authors in paper [2]. A basic turbo coding strategy is proposed in [3] to maximize the error quality of a standard rate-1/3 turbo code. In the paper [4] the authors introduced the high-speed turbo SISO Decoder. Standardization operation was applied to state metric branch values rather than branch metric values. A minimum- power, and area-efficient turbo soft-output (SISO) decoder based on the Viterbi algorithm are proposed in [5]. The paper presents the implementation of SOVA decoder for different constraint lengths. Compared to a conventional SOVA decoder applica- tion, simulation results show power savings and area savings. The designers in [6] developed a turbo decoder architecture in which utilizes both parallel SISO decoder tier and parallel trellis stage level.In the LTE-Advanced standard, the authors in paper [7] present the design and implementation of a memory- reduced Turbo decoder on the field programmable gate array (FPGA). In this paper [8] the author presents a summary of the architecture issues relevant to turbo decoders. As a feature of the code's key parameter, an evaluation of different types of turbo decoders is carried out. The authors in paper

[9] proposed a new low-complexity Min-Log-MAP algorithm variant in their study. The encoding of tail-biting codes using hierarchical input encoders is done in paper [10], which is an important design criterion. Authors in their work [11] explore the different methods used in turbo codes to end trellis in the

recursive systematic convolutional encoders. With minimum generator polynomials, the author in [12] has achieved very low rate convolutional codes. For different code rates and for different constraint lengths, the authors in the paper [13],the Sum Product Algorithm (SPA) and One Step Majority Logic Decoding Algorithm (OSMLGD)is put together to decode the LDPC codes and their individual performances were compared.

III. IMPLEMENTATION

A. Architecture of Turbo Coder

Turbo encoder and decoder together comprises the Turbo coder architecture(shown in figure 1).Two identical Recursive convolutional encoders(RSC) and a pseudorandom interleaver constitutes the turbo encoder (figure 2).LTE employs a 1/3 rate parallel concatenated turbo code. Each RSC works on two different data. Original data is provided to the first encoder, while the second encoder receives the interleaved version of the input data. A specified algorithm is used to scramble the data bits and the method is called Interleaving. An appreciable impact on the performance of a decoder is seen with the interleaving algorithm when used. The RSC1 and RSC2 encoder outputs along with systematic input comprises the output of turbo encoder,that is, a 24 bit output is generated which is illustrated in figure 6. This will be transmitted through the channel to the Turbo decoder.A standard turbo decoder block diagram is shown in Figure 3 that contains two modules of SISO decoders together with two pseudorandom interleavers and a pseudorandom deinterleaver.

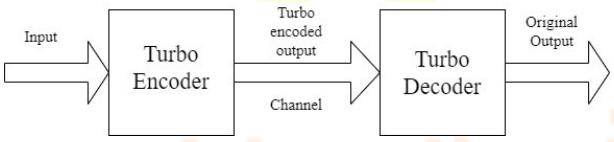


Fig. 1. Turbo Coder Block diagram

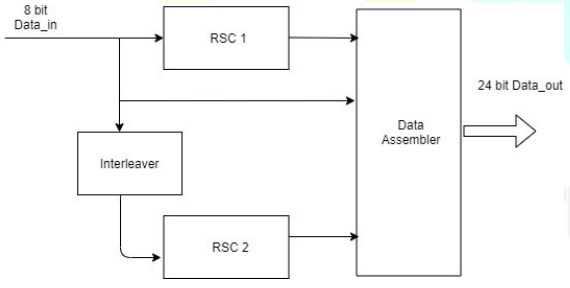


Fig. 2. Turbo Encoder Block Diagram

The usually used method of turbo code decoding is carried out using the BCJR algorithm.The fundamental and basic idea behind the turbo decoding algorithm is the iteration between the two SISO part decoders which is illustrated in figure 3. It comprises a pair of decoders,those which work simulateneously in order to refine and upgrade the estimate of the original information bits. The first and second SISO decoder, respectively, decodes the convolutional code generated by the first or second CE.A turbo-iteration corresponds to one pass of the first component decoder whichis followed by a pass of the second component decoder.

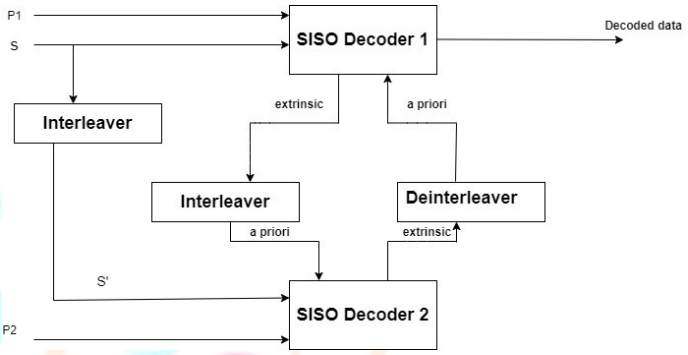


Fig. 3. Turbo Decoder Block diagram

B. SISO Decoder

The signal which is received at the input of a soft-in-soft- out (SISO) decoder is the real (soft)value of that signal.An estimate of each input bit The decoder then generates an approximation for each data bit expressing the probability that the transmitted data bit is equal to one.The maximum a-posteriori (MAP) algorithm is used in the turbo-decoder under consideration in this paper for the SISO component decoder.The MAP algorithm never restricts the set of bit estimates to correspond strictly to a valid path through the trellis. Therefore, the results produced by a Viterbi decoder that recognizes the most likely true path through the trellis should differ from those generated by that.

1) The MAP Algorithm : The MAP algorithm minimizes the likelihood of bit error by using the entire sequence that was obtained to figure out the most likely bit at each trellis point. Consider a frame of N coded symbols consisting of m bits and the channel output received by the decoder as y. For every d^{sym}, a MAP decoder provides a 2^m *a posteriori* probabilities. The hard decision on the value j that is equal to d^{sym}, helps to maximize the a_j*posteriori* probabilities. It is expressed injoint probabilities as:

sy

$$Pr(d_i^s ym = j/y) = \frac{p(d_i^s ym = j, y)}{\sum_{k}^{2} m - 1(p(d_i^s ym = k, y)}$$

The trellis form of the code allows the decomposition of computing the joint probabilities among the former and latter observations.The *Forward recursion metric* $\alpha_i(S)$ used in decomposing is shown in Equation 2. It provide the probabilties of state *S* instantly at *i* acquired from previous values from the channel.Backward recursion metric $\beta_i(S)$ is also used to find the probabilities of the state calculated using the forthcoming values from channel and *Branch metric* $\gamma_i(S', S)$.

$$Pr(d_i^{sym} = j|y) = \sum_{j}^{(S^i,S)/d_i^{sym}=} \alpha_i(S')\gamma_i(S', S)\beta_{i+1}(S) \quad (2)$$

And the branch metric is given by

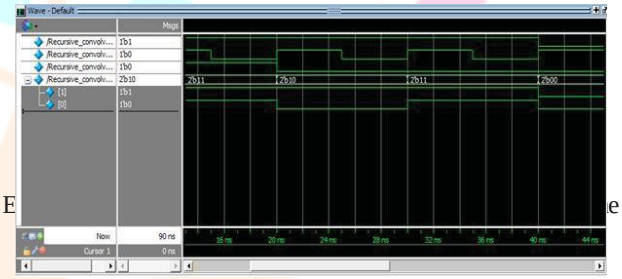
$$\gamma_i(S', S) = p(y_i|x_i).Pr^a(d_i^{sym} = d_i^{sym}(S', S)) \quad (3)$$

where,
$p(y_i|x_i)$ =channel transition probability,
x_i = *i*th transmitted modulated symbol and
y_i = *i*th received symbol.

For an equiprobable source, the *a priori* probability is $1/2^m$. In E e input of symbol channel.

a Xilinx based software suite. It may be used for HDL design synthesis and analysis. Recursive convolutional encoder, Turbo encoder decoder simulation outputs are performed using Xil- inx Vivado and Octave. The netlist is generated from RTL using Cadence NCLaunch during synthesis. Nclaunch is a GUI that helps in managing large-scale design projects and allows to customize and launch Cadence simulation software. RTL compiler Ultra is used for logic synthesis and analysis for digital designs. Physical design (Floor-planning, placement, routing) is done using Encounter tool. The Cadence imple-

mentation tool uses netlist as input and does optimization, placement, and routing.



C. Interleaver

Choosing the interleaver is a significant part of the turbo code design. Interleavers scramble data in a pseudorandom order to lessen the resemblance between adjacent bits at the input of the convolutional encoder.The interleaver is used on both the encoder part and the decoder part. It produces a long block of data on the encoder side, while it compares two SISO decoders' output in the decoder portion and helps to fix the error. Pseudo-random deinterleaver functions in a complimentary manner of pseudo-random interleaver.

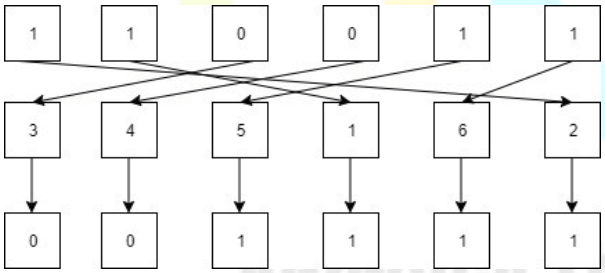


Fig. 4. Psuedo-random interleaver:working principle

I. RESULTS AND DISCUSSION

The turbo encoder and decoder simulations are done using Verilog HDL in Xilinx Vivado 2018.3.Vivado Design Suite is

Fig. 5. Output of RSC encoder using Xilinx Vivado. 1/2 rate RSC encoderis shown.That is, for 1 input bit, 2 output bits have been generated

```
p =  2
States of shift register:
x =

   0   0
   1   0
   0   1
   0   0

Code vector:
c =

   1   1   0   0   1   1   1   0
```

Fig. 6. Octave output :RSC encoder

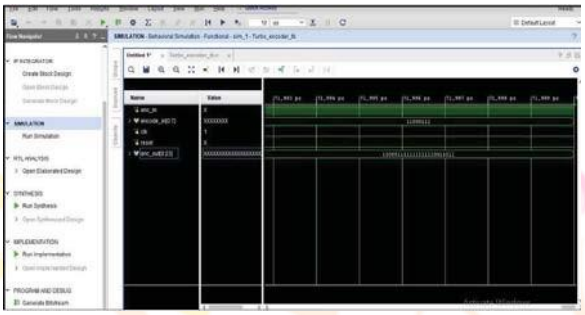
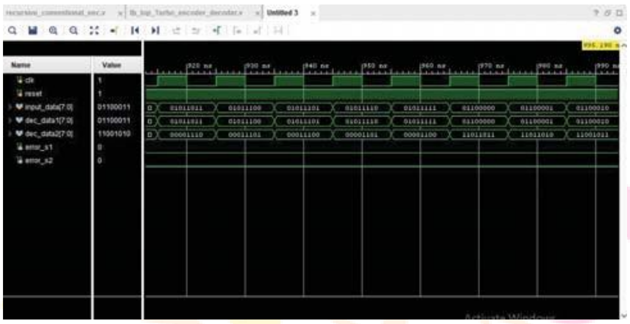


Fig. 7. Simulation output of Turbo Encoder using Vivado.The encoder outputis in concatenated form which is of 24 bits which is transmitted through the channel.

```
X =

   1   1   1   1   1   0   0   0

Interleaver =

   4   1   2   7   8   3   5   6

code =

 Columns 1 through 20:

   1   1   1   1   1   0   0   0   1   1   0   0   1   0   1   0   1   1   0   1

 Columns 21 through 24:

   0   0   1   0
```

Fig. 8. Octave Output:Turbo Encoder.
X is the random input generated which is interleaved(randomised) and finallyappended to get 24 bits after encoding.

Fig. 12. Layout of Turbo decoder



The two SISO component decoders generate dec_data1 and dec_data2 which is of 8 bits each and error s1 and error s2 are the corresponding errors calculated by the respective component decoders. Both of the SISO decoders generate zero error.

```
    Pin               Type                   Fanout Load Slew Delay Arrival
                                                    (fF) (ps)  (ps)   (ps)
-----------------------------------------------------------------------------
input_data[0]     in port                         2  1.6    8    +8       0 F
c1/input_data[0]
  g35/A                                                          +0       0
  g35/Y           INVXL                           1  0.0    6   +12      12 R
  c1/data_in[0]   pseudo random interleaver                      +0      12
-----------------------------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : input_data[0]
End-point    : c1/c1/data_in[0]
```

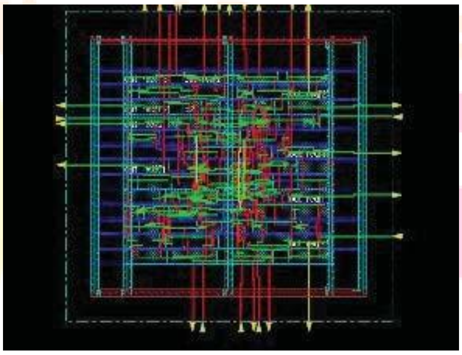


_Fig. 10. Timing report of Turbo encoder decoder

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
top_Turbo_encoder_decoder	8	70.055	666.255	736.311
c1	8	70.055	666.255	736.311
c1	0	0.000	0.000	0.000
c2	0	0.000	202.500	202.500
c2	0	0.000	0.000	0.000

Fig. 11. Power Utilisation Of Turbo encoder decoder

IV.CONCLUSION

Turbo encoding and decoding is done using Verilog HDL.Thedecoder is developed based on MAP algorithm. The MAP algorithm identifies the most probable bit of information that was sent.Original MAP algorithm is used because the number of iterations for process decoding is reduced. Many approximations on the MAP algorithm are usable, such as the Max-Log MAP algorithm, where computations are mostly in the logarithmic realm, thus facilitating the implementation of values and operations.Here, the decoder successfully corrects the error and retrieves the original message.Synthesis is done in Xilinx Vivado 2018.3 and results are tabulated.Synthesis is done using Cadence NClaunch to convert the RTL into netlist.Encounter tool is used to convert the netlist to phys- ical design.Floor planning, placement and routing are per- formed.GDS II file has been generated successfully.

REFERENCES

[1] V. Kavinilavu1, S. Salivahanan, V. S. Kanchana Bhaaskaran2 , Samiappa Sakthikumaran, B. Brindha and C. Vinoth "Implementation of Convo- lutional Encoder and Viterbi Decoder using Verilog HDL ", IEEE 3rd International Conference on Electronics Computer Technology,2011

[2] Aswathy Narayanan ; Senthil Murugan ; Ramesh Bhakthavatchalu, "Low Latency Max Log MAP based Turbo Decoder", International Conference on Communication and Signal Processing (ICCSP),2019

[3] C. Tanriover ; B. Honary ; Jun Xu ; Shu Lin", "Improving turbo code error performance by multifold coding",IEEE Communications Letters,2002

[4] Shweta Ramteke ; Sandeep Kakde ; Yogesh Suryawanshi , "HDL Imple-mentation of Turbo Decoder Using Soft Output Viterbi Algorithm",2015.

[5] J.H. Han ; A.T. Erdogan ; T. Arslan ,"Power and area efficient turbo decoder implementation for mobile wireless systems", IEEE Workshop on Signal Processing Systems Design and Implementation,2005

[6] Cheng-Chi Wong ; Ming-Wei Lai ; Chien-Ching Lin ; Hsie-Chia Chang ; Chen-Yi Lee,"Turbo Decoder Using Contention-Free Interleaver and Parallel Architecture",IEEE Journal of Solid-State Circuits",2010

[7] Yaqin Shi ; Ming Zhan ; Jie Zeng , "FPGA Implementation and Power Estimation of a Memory-Reduced LTE-Advanced Turbo Decoder", IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018.

[8] E. Boutillon, C. Douillard, and G. Montorsi , "Iterative decoding of concatenated convolutional codes: Implementation issues",Proc. IEEE, vol. 95, no. 6, pp. 1201–1227,June, 2007

[9] Rami Klaimi ; Charbel Abdel Nour ; Catherine Douillard ; Joumana Farah ,"Low-complexity decoders for non-binary turbo codes" ,IEEE 10th International Symposium on Turbo Codes Iterative Information Processing (ISTC), 2018

[10] Maria Kovaci ; Horia Balta,"Performance of trellis termination meth- ods for RSC component encoders of turbo codes", 10th International Symposium on Electronics and Telecommunications,IEEE,2012