*International Journal For Technological Research In Engineering*
*Volume 3, Issue 10, June-2016* *ISSN (Online): 2347 - 4718*

# EFFICIENT VLSI ARCHITECTURE OF TURBO DECODERS FOR LONG TERM EVOLUTION (LTE) COMMUNICATION

Reddy Soujanya A[1], Dr. N G Kurahatti[2]
[1]M.Tech Student, [2]Professor, Department Of ECE, EPCET, Bangalore, India

***Abstract: Wireless communication is the fastest developing portion of the communications industry. Channel coding is the most important part of digital wireless communication systems. The fading of signal due to multipath propagation causes errors which need to be corrected. The main aim of any channel coding schemes is to provide error-free data transmission by adding redundancy to information to check code and correct errors. Forward error correction (FEC) is favoured strait coding procedure if the most extreme permitted transference delay is small or during the returning strait is not accessible. One imperative category of forward error correction codes are turbo codes. Turbo codes accomplish greater coding benefit and foremost utilized for error emendation in high rate wireless frameworks. This paper depicts a turbo decoder for Long Term Evolution (LTE) standard, utilizing a MAP algorithm. Long-term evolution (LTE) is expected to accomplish maximum information assess in surplus of 300 Mbps for fourth era wireless transferral networks. Turbo codes predetermined strait coding plan in LTE, experiences small decoding throughput because of frequentative decoding calculation. An effective way to accomplish favourable throughput is to utilize maximum a posteriori (MAP) basics in equidistant.***

***Keywords: Turbo codes, LTE Standard, MAP decoder***

## I. INTRODUCTION

Since the emergence of digital communication systems, there has been a need for error correction. This is due to non-ideal nature of practical communication channels, which are often corrupted by noise. Error correction attempts to compensate for the errors introduced by noise. Many advanced wireless communication standards adopted turbo codes as the channel coding scheme due to its near Shannon error-correcting performance [2] [3]. Turbo codes were first introduced in 1993 by Berrou, Glavieux, and Thitimajshima. At the beginning, after proving the obtained decoding performances, the turbo codes were introduced in different standards as recommendations, while convolutional codes were still mandatory. The reason behind this decision was especially the high complexity of turbo decoder implementation. But the turbo codes became more attractive once the supports for digital processing, like Digital Signal Processor (DSP) or Field Programmable Gate Array (FPGA), were extended more and more in terms of processing capacity. Turbo codes are one of the most powerful types of Forward- Error-Correcting (FEC) channel codes. LTE stands for Long Term Evolution which is introduced by 3GPP to define a new high speed access method for mobile communication systems.

Recently, long-term evolution has been dominated as the next-generation wireless communication standard, which has long been seen as the first advancement towards stronger, faster and more efficient 4G data networks. The technology under LTE can currently reach downlink peak rates of 100Mbps and uplink speeds of 70Mbit/s. The LTE technology is also a scalable bandwidth technology for carriers operating anywhere from 20 MHz town to 1.4 MHz. Long Term Evolution offers some excellent advantages over current 3G systems including higher throughput, plug and play compatibility, FDD (Frequency Division Duplexing) and TDD (Time Division Duplexing), low latency and lower operating expenditures. It also offers legacy modes to support devices operating on GPRS systems. The turbo decoder, which is specified in LTE, reveals to be a limiting block toward these goals due to its iterative decoding nature, high latency, and significant silicon area consumption. Hence to overcome these decoding procedures is performed using the MAP algorithm. The paper is organized as follows. Section 2 describes the LTE coding scheme with the turbo encoder and turbo decoder. Section 3 presents the methodology. In Section 4, presents the implementation results that are obtained on Xilinx ISE and Modelsim. Section 5 presents the final conclusions.

## II. LTE CODING SCHEME

The coding scheme presented in LTE specification is a classic turbo coding scheme including a turbo encoder and turbo decoder.

Turbo Encoder:

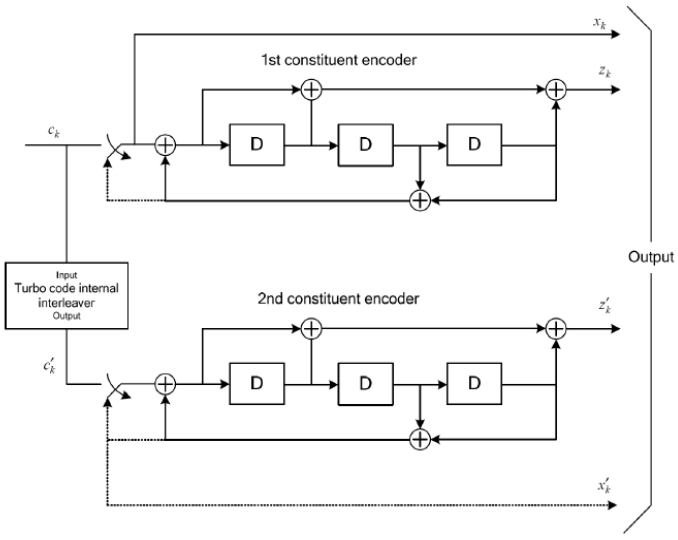


Fig 1: Turbo Encoder

  

The LTE Turbo encoder includes one interleaver block and two constituent encoders. The two constituent encoders utilized are similar with generator polynomials g0 (D) = 1 + D2 + D3 and g1(D) = 1 + D + D3.This is depicted in Figure 1. The information bits along with the parity bits that are been generated by two convolution encoders are transmitted. Initially each shift register is set to zero i.e the convolutional encoder is in zero state. The information block $C_k$ is the input to the LTE turbo encoder which can be seen in the figure 1. The K bits relating to the input are transmitted to the output as systematic bits in a steam of $X_k$. At the same time, the first constituent encoder processes the information block which results in the parity bits $Z_k$. The second constituent encoder handles the interleaved version of the information block $C'_k$ which results in the parity bits $Z'_k$. The interleaver is a device that permutes the data sequence in some predetermined manner. Interleaver is used to scramble bits before being input to the second encoder. This makes the output of one encoder different from the other encoder. Thus, even if one of the encoders occasionally produces a low-weight, the probability of both the encoders producing a low-weight output is extremely small. This improvement is known as interleaver gain. Another purpose of interleaving is to make the outputs of the two encoders uncorrelated from each other. Combining the systematic bits and the two streams of parity bits, the following sequence is obtained at the output of the encoder:

$X_1, Z_1, Z'_1, X_2, Z_2, Z'_2 \ldots. X_k, Z_k, Z'_k$.

At the end of the coding process, in order to drive back the constituent encoders to the initial state, the switches from Fig. 1 are moved from position A to B.

Turbo Decoder

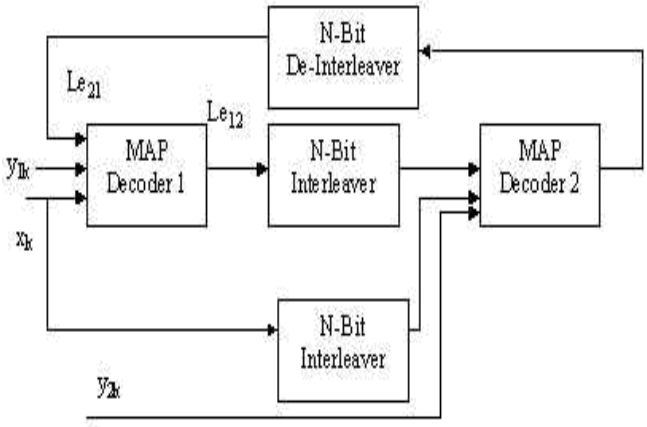


Fig 2: Turbo Decoder

The decoding plan of LTE turbo decoder is depicted in Fig. 2. In the decoder architecture, there are 2 decoders corresponding to 2 RSCs. Due to the existence of a feedback path as demonstrated, the operation of the turbo decoder is done in an iterative way. For every full iteration decoder comprises of two half iterations such that one for every constituent code it is used. The planning of the decoder is such that the MAP decoder 1 starts working amid the primary half iteration and MAP decoder 2 works amid the second half cycle. The 1st MAP decoder inputs are the tainted systematic array of bits $X_k$, the parity stream of bits $Y_{1k}$ from the first RSC encoder, and from 2nd MAP decoder the deinterleaved extrinsic information. To the 2nd MAP decoder the inputs are the tainted interleaved systematic bit stream, parity stream of bits $Y_{2k}$ from second RSC encoder, and from 1st decoder the interleaved extrinsic information. The Maximum A Posteriori (MAP) algorithm is utilized by two Recursive Systematic Convolutional (RSC) decoders. The exemplary algorithm gives the very good decoding executions, yet it experiences a very high complexity during implementation and low decoding throughput. For these reasons the MAP algorithm is used as a reference for targeted decoding performances. The MAP algorithm, which provides the a posteriori probability for each bit, is used in iterative decoding of turbo codes. The MAP algorithm provides the probability of the decoded bit $u_k$ being either +1 or −1 for the received symbol sequence *y* by calculation of the values as

$$L(u_k|y) = \log [(p(uk = +1|y) / p(uk = -1|y]$$

Where $p(u_k = +1| y)$ and $p(uk = -1|y)$ denote the probabilities of bit $u_k$ being +1 and −1, respectively.

## III. METHODOLOGY

A code rate of (1/3) is considered. Architecture of MAP decoder is divided into different sub-blocks. The flow of MAP decoder is depicted in Fig 3.

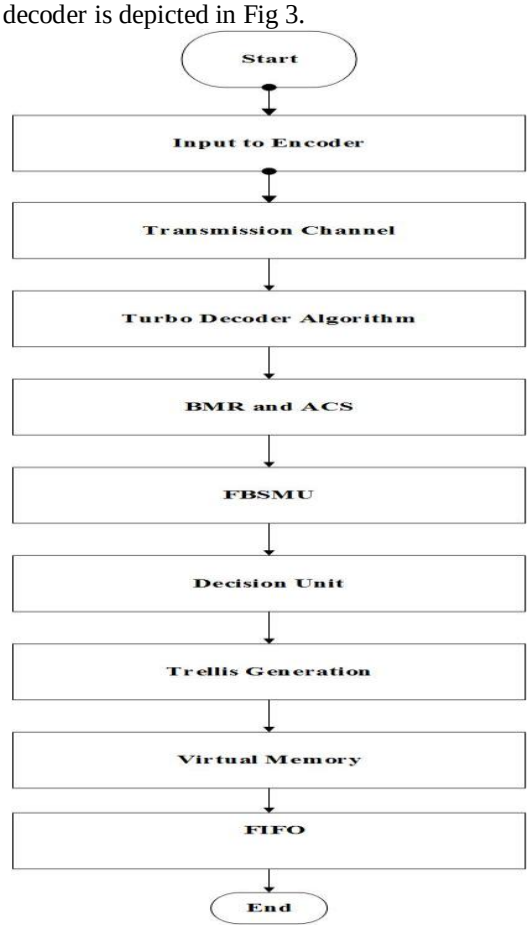


Fig 3: Flow Diagram of MAP decoder

  

The input data stream from the encoder is transmitted over the communication channel. At the decoder end firstly Branch Metric Calculation (BMC) and Branch Metric Routing (BMR) is done.BMR calculation is done to compute the distance between the receiving symbols. The branch matrix values are the input to the ACS. Add Compare Select (ACS) unit involves adders, multiplexers which act as a compactor and subtractors. The sequence of input data stream is given to the four adder's i.e. two inputs to each adder. The resultants of addition bits are subtracted. The mux acts as a comparator with a select line that compares the output from two adders. The subtracted values are compared in comparator unit and resultant are the decoded bits and also extrinsic information to second MAP decoder. The real time incoming LLRs and L(U*k*) i.e systematic and two parity bits are fed to the BMR unit which computes the parent branch metrics. These branch metrics are passed through the registers. Parent branch metrics from BMR are fed to the sub-block ACS and FBSMU. FBSMU is a simple block comprising of BMR, ACS and SMU unit. The FBSMU consists of SMU (State Metric Unit) units where each SMU unit comprises of a shift unit and a latch. The shift and latch units are used to get the different combination of the output. Inputs for FBSMU are the parent branch metrics and backward state metrics of the preceding states, and the outputs are computed values of backward state metrics for the current states. The outputs of FBSMU units are also buffered. Processing unit is particular combination of interconnected FBSMU's. The encoded data bits are received at this unit such as pattern and parity bits i.e. symbol0 and symbol1.The bits are processed in the FBSMU during each clock cycle and results the final decoded output.

## IV. IMPLEMENTATION RESULTS

Fig 4 and 5 shows the RTL schematic diagram of turbo encoder and decoder implemented using Xilinx 14.7.

Fig 6 shows the simulation results for secure transmission of the data streams simulated using Modelsim 6.3. Initially reset is '1'and no operation is performed.When reset is '0'enc_bit_in process starts by transmitting the data bits.During reset is '0'and when dec_valid_in goes high and the output starts.Thus the output can be seen on dec_bit_out line after some latency.

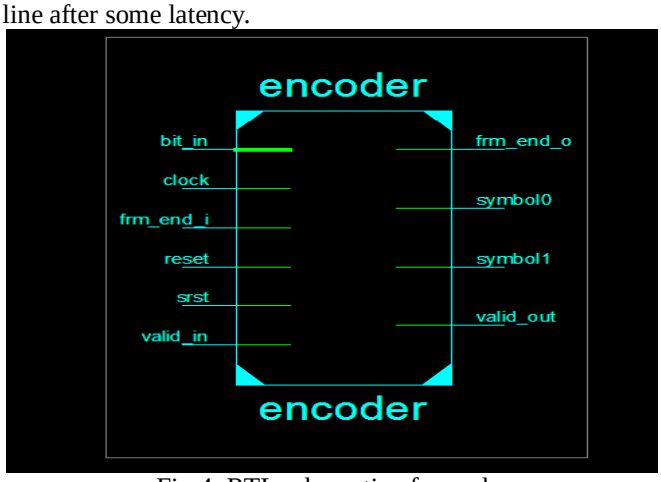


Fig 4: RTL schematic of encoder

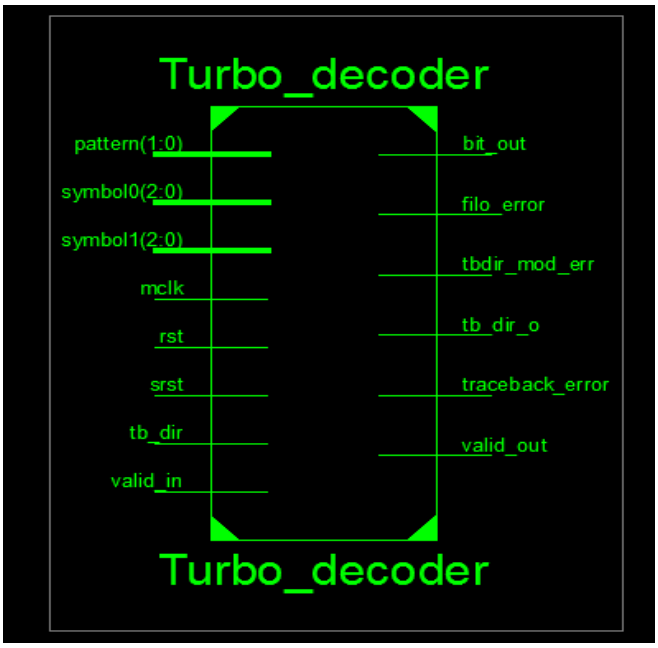


Fig 5: RTL schematic of Turbo dncoder

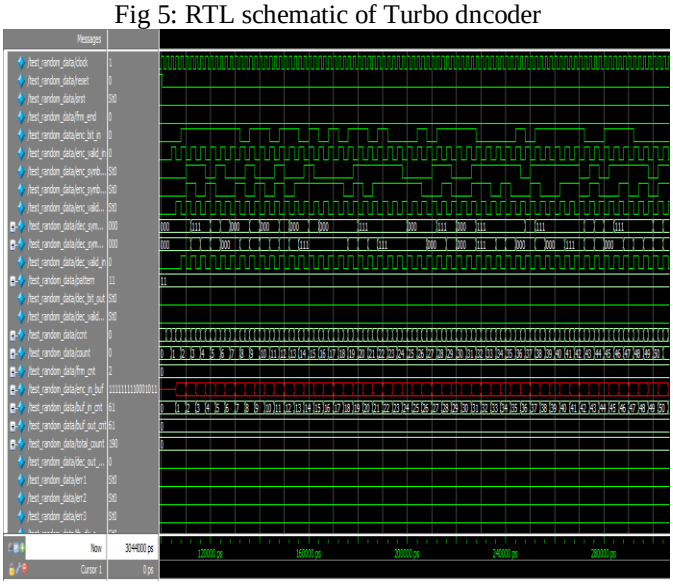

Fig 6: Simulation waveform

## V. CONCLUSION

The most important aspects regarding the FPGA implementation of a turbo decoder for LTE systems were presented in this paper. Turbo decoders are used for error detection and error correction for distance communication. Turbo codes, the described channel coding strategy in LTE, suffers from a low-decoding throughput because of its iterative decoding algorithm. A MAP based turbo decoder has been implemented which increases the throughput. This paper also proposes the adoption of an advanced interleaver, which reduces delay in decoding process and hence has less latency. Hence Turbo code is better in terms of performance, low latency, and computational complexity than other codes. The modelled hardware design is implemented on to the Artix7 FPGA and the design can work at a frequency of 230.289MHz and the device utilization is 2%.

  

REFERENCES

[1] Arash Ardakani and Mahdi Shabany "A Novel Area- Efficient VLSI Architecture for Recursion Computation in LTE Turbo Decoders", TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 62, NO. 6, JUNE 2015.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," IEEE Proceedings of the Int. Conf. on Communications,Geneva, Switzerland, pp. 1064-1070, May 1993.

[3] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," IEEE Trans. Communications, vol. 44, no.10, pp. 1261-1271, Oct. 1996.

[4] Rahul Shrestha and Roy Paily, Member, IEEE" Design and Implementation of a High Speed MAP Decoder Architecture for Turbo Decoding", 2013 26th International Conference on VLSI Design and the 12th International Conference on Embedded Systems.

[5] F. Maessen, A. Giulietti, B. Bougard, V. Derudder, L. Van der Perre, F. Catthoor, M. Engels IMEC Leuven, Belgium," Memory power reduction for the high-speed implementation of Turbo codes "in IEEE International Conference on Communications".

[6] Cristian Anghel, Valentin Stanciu, Cristian Stanciu, and Constantin Paleologu, Telecommunications Department, University Politehnica of Bucharest, Romania "CTC Turbo Decoding Architecture for LTE Systems Implemented On FPGA", ICN 2012: The Eleventh International Conference on Communication Networks.

[7] Patel Sneha Bhanubhai, Mary Grace Shajan, Upena D. Dalal "Performance of Turbo Encoder and Turbo Decoder for LTE"ISSN: 2277-3754 ISO 9001:2008 Certified International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 6 October 2012.

[8] Anum Imran," SOFTWARE IMPLEMENTATION AND PERFORMANCE OF UMTS TURBO CODE, Master of Science Thesis: 74 pages, 22 appendix pages. March 2013.

  