# Ber Analysis of Turbo Code Interleaver

Prabhavati D. Bahirgonde
Research Student of Solapur University, Solapur
Walchand Institute of Technology,
Ashok Chowk, Solapur

Shantanu K. Dixit, PhD
Professor in Electronics & Telecommuncation
Department, Walchand Institute of Technology,
Ashowk Chowk, Solapur

## ABSTRACT
This paper presents a low complexity interleaver design that facilitates high throughput Turbo decoding required for next generation wireless systems. When a parallel decoder structure is considered, interleaver design is the most important issue. In such parallel decoder, the contention problem occurs when more than one extrinsic value references to the same memory block for read or write purpose. This paper focuses on the alternate method for QPP interleaver which shows improved BER performance for large frame size. Bit reversed indexing is used to generate interleaved addresses. A counter is used to generate sequential address as well as interleaved address. The number of address lines of memory which stores data , depends upon frame size of data. In this paper, a comparison is made between best proved interleaver and proposed interleaver on the basis of BER performance for different number of iterations, different frame size and different decoding algorithms.

## General Terms
Interleaver, Turbo Decoder, Bit reverse indexing.

## Keywords
QPP; BER; APP; MAP

## 1. INTRODUCTION
The fourth generation (4G) wireless systems are supporting to very high data rate from 100Mbps to 1Gbps. This gives an increase in complexity of the wireless receiver. Among the different channel coding, Turbo code is one of most significant and interesting code. Turbo code as an error correction code was proposed by Berrou in 1993, which is close to the Shannon limit of error correction capability[1]. There are significant challenges for Turbo decoder in hardware implementations of reducing complexity and area. High throughput is also one of the key area in which research is going on. Increase in throughput also increases complexity. The Turbo decoder uses iterative algorithm, interleaver and de-interleaver. The algorithms used for decoding are MAP (Maximum A Posteriori), Log MAP, approximated Log MAP, Max Log MAP, constant Log MAP. The complexity can be reduced in these algorithms but the focus is on interleaver. To obtain high throughput, Turbo Decoders are used in parallel. Due to parallelism, the contention problem occurs. To avoid this contention, there is need of efficient interleaver. Normally used interleavers are random interleaver and block interleaver. The Quadratic Permutation Polynomial (QPP) interleaver is best which avoids contention. Here bit reverse indexed interleaver is discussed which shows improvement in BER performance. Actual implementation of the system is on FPGA. Here ,the focus is only on an interleaver , so its implementation on FPGA is discussed.

The organization of this paper is as follows: Section 2 contains description of Turbo encoder and decoder. Different interleavers with proposed interleaver is discussed in section 3. MATLAB based simulation results are presented in section 4. Finally, conclusion on performance of proposed interleaver in section 5.

## 2. TURBO CODES
### 2.1 Turbo Encoder
The 3GPP LTE standard Turbo code is parallel concatenated convolutional code with an interleaver. It consist of two RSC encoder. First encoder receives data sequentially while second encoder through interleaver. The code rate of a turbo code is 1/3. The structure of turbo encoder is shown in figure 1. The interleaver shuffles incoming data bits in a specific manner.

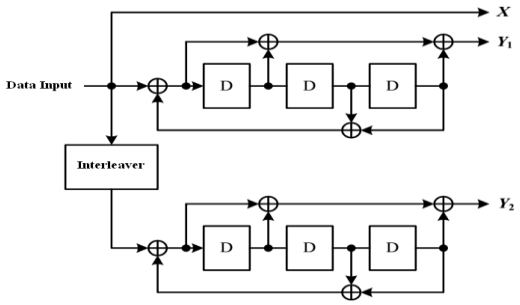


**Fig. 1. Structure of Turbo Encoder**

Interleaving the encoded message before transmission and deinterleaving after reception causes a burst of channel errors to be spread out in time and thus to be handled by the decoder as if they were random errors. So to randomize this error interleaver is suitable. The decoding algorithm employed in Turbo decoders is MAP algorithm proposed by Bahl *et al.* But this decoding algorithm is very complex as it requires more number of multiplications and additions. To reduce complexity, algorithm is converted into log domain. The Turbo decoding functionality is described in figure 2. While decoding, SISO decoder get input as intrinsic LLRs from the channel and extrinsic LLRs from another SISO decoder through interleaver ($\Pi$) or de- interleaver ($\Pi^{-1}$). As Turbo decoder uses iterative decoding, increase in iterations improves the performance at the cost of increase in the complexity. Decoding performed by a first decoder is considered as half iteration and output of this decoder is given

as input to second decoder. When a second decoder performs decoding , full iteration will be completed. The random interleaver degrades the performance and also adds decoding latency.

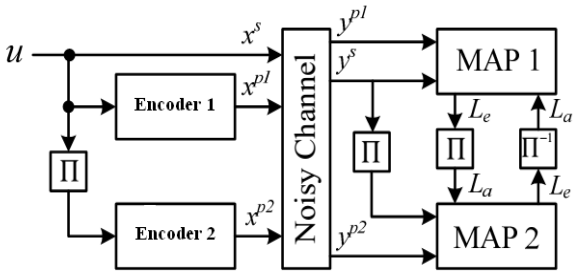


**Fig. 2. Turbo Encoder and Decoder**

The Turbo decoding process of 3GPP LTE Turbo code is effectively represented by using 8 state trellis diagram which describes all the possible state transitions through a graph representation. The 3GPP-LTE Turbo code uses 8-state trellis structure. To implement any Turbo decoding algorithm forward state metric, reverse state metric and branch metric should be computed. Using these metrics Log Likelihood Ratio (LLR) is calculated to obtain decoded data.

## 3. INTERLEAVERS

The interleaver size and structure considerably affect Turbo code error performance. The purpose of the interleaver in Turbo codes is to ensure that information patterns, which cause low-weight words for first encoder, are not interleaved to similar patterns for second encoder, thus improving weight spectrum of the code. Interleavers has been frequently used in a variety of communication systems. Generally, an interleaver was used to randomize the error locations. Since this randomness directly affects the decoding performance. However, random interleaver is not only difficult to implement, but also creates problems in parallel decoding due to memory conflict. Therefore, research for interleaver to improve performance during parallel decoding is becoming important task. Following are few interleavers which are discussed below:

*A. Random Interleaver*

It rearranges the elements of its input vector using a random permutations at encoder and decoder. It requires a lookup table which stores random addresses. It is not suitable for large frame size. Random interleaver degrades the performance of Turbo codes.

*B. S-random interleaver*

It is a pseudorandom interleaver which is an improvement to random interleaver. In S-random interleaver, any two input positions are separated with distance S.

*C. Block interleaver*

It accepts a set of symbols in blocks from the encoder and rearranges them, without repeating any of the symbols in the set. The number of symbols in each set is fixed for a given interleaver. Block interleaver may have good minimum distance, but the high multiplicity of low weight codewords makes this interleaver as unsuitable [2].

*D. QPP interleaver*

These are a class of deterministic interleavers based on quadratic permutation polynomial over ring of integers modulo N. The parameter selection of polynomial is based on minimum distance. It completes interleaving by making QP (Quadratic Polynomial) satisfy some conditions. In order to make F(x) be a QPP, f1≠ 0 and f2 =0 mod p. Where F(x)=f1.x + f2.$x^2$ is a permutation polynomial over $Z_P^n$. The QPP error function provides excellent error performance [3] and high throughput, and has advantages over earlier interleaver structures such as random and block interleaver. The QPP interleaver function can be represented in recursive manner [4]. To reduce hardware complexity interleaver addresses are stored in the lookup table form [5]. This look up table based approach is practically difficult to support multiple block size [6].

*E. Bit reverse indexed interleaver*

The frame size for 3GPP-LTE Turbo encoder is in the range of 40 to 6144. The interleaver is used in encoder to interleave the data which is stored in memory. Data from this memory goes to encoder 1 in a sequential manner, but it goes to encoder 2 through interleaver. For generating interleaving pattern a counter is used. The output bits of this counter are combined in the reverse way to generate address for interleaving. If suppose incoming data are 128 bits, we require memory of 128 locations and to address these locations 7 bit counter is required. For storing data ,dual port RAM is used which produces data for encoder1 as well as encoder2 simultaneously. To obtain interleaved data, counter outputs are connected in reverse way. This hardware implementation is very easy and less complex. Only the drawback of this design is if the frame is not a power of 2, there is wastage of memory.

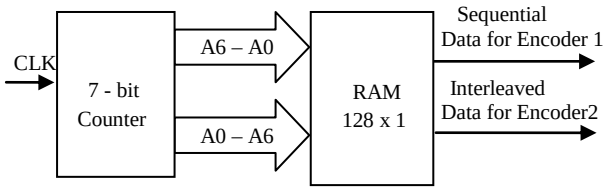


**Fig. 3. Address generator for Turbo Encoder**

The implementation of the counter, which generates sequential and interleaved addresses is done in Xilinx. The RTL schematic of this address generation unit is shown in figure 4. The value of n-bit counter depends on the frame size. Here, for simplicity, it is shown as a 3-bit counter. The value of n can be extended. The performance of this interleaved pattern is tested in MATLAB. If frame size is not a power of 2 such as 40, the value of n is 6. Here memory wastage of 24 locations occurs.

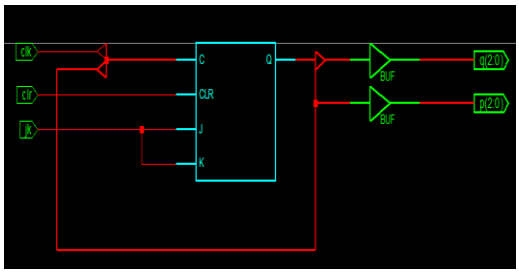


**Figure 4. RTL schematic of address generator**

## 4. SIMULATION RESULTS

In this section, performance of Turbo decoder is analysed based on interleaver which is used to randomize the error. For analysis, bit error probability is calculated for QPP interleaver and bit reverse indexed interleaver. The bit error rate of received signal depends on noise and interference of communication channel.This analysis is done in MATLAB. For analyzing Turbo decoder based on interleaver, following parameters are considered which are

i. Number of iterations during decoding

ii. Frame size

iii. Parallelism

The Turbo encoder and decoder used are as shown in figure 1 and 2. The decoding algorithms used are true a posteriori probability (APP), MAP and approximated MAP. For performing BER testing, random set of data is created and applied to encoder 1 directly and encoder 2 through interleaving. Random error is added and then the data is decoded. The error bits are calculated by comparing original data with the decoded data. The noise channel considered is AWGN. BPSK modulation is considered for testing the performance.

Figure 5 shows the BER performance with number of iterations as four and frame length 128. The parallelism is also considered. The performance of both interleaver is nearly same. If the number of parallel decoders are increased then BER performance slightly degrades.

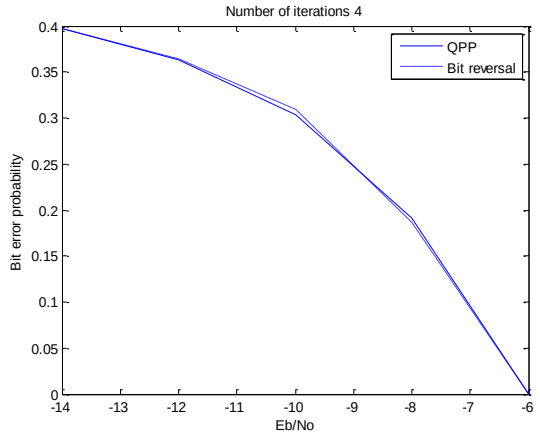


**Figure 5. BER Performance for Four Decoding Iterations**

Figure 6 shows the BER performance with number of iterations as six and frame length 128. The performance of both interleaver is nearly same. If the number of decoding iterations are less, performance of both interleaver is approximately same.

Figure 7 shows the BER performance with a number of iterations as six and frame length 512. In this case,the performance of bit reverse indexed interleaver is better than QPP interleaver.

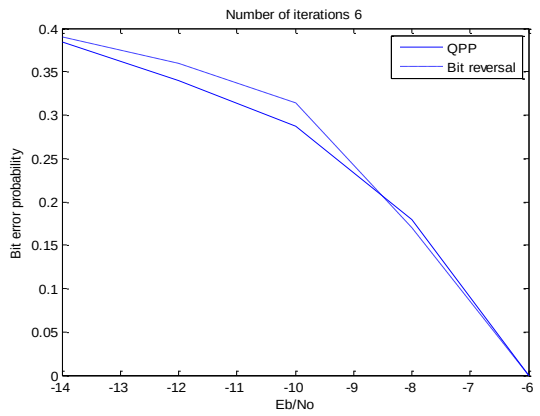


**Figure 6. BER performance for six decoding iterations**

So the performance of proposed interleaver is good for large frame size. While testing performance using the bit reversal technique, if frame size is not a power of 2 then extra zero bits are added to make frame size as power of 2.

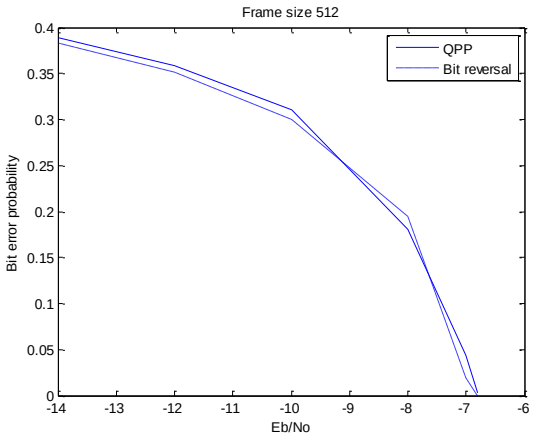


**Figure 7. BER performance for frame size 512**

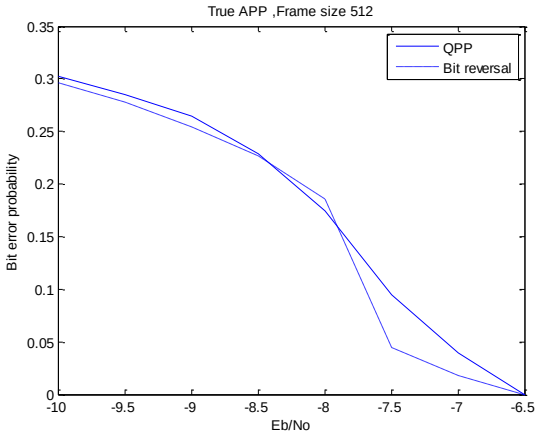


**Figure 8. BER performance for true APP decoding algorithm**

These above BER performances are tested for approximated MAP decoding algorithm which increases speed of operation [7]. The BER performance is also tested for true a posteriori probability (APP) and MAP (Maximum A Posteriori) decoding algorithm.These performances are shown in figure 8 and figure 9.

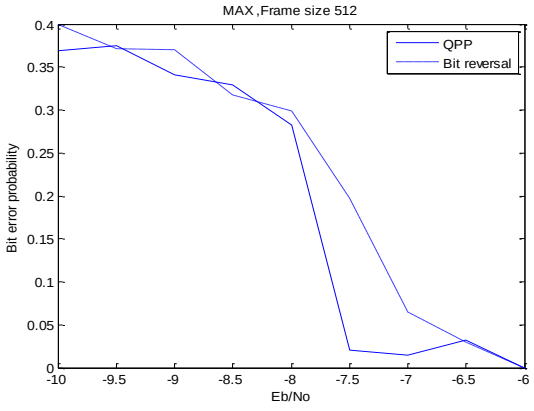


**Figure 9. BER performance for Max Log MAP algorithm**

## 5. CONCLUSION

The interleaver in Turbo codes play very important role in increasing performance of it .Interleaver reduces memory contention. The QPP interleaver is observed to be best interleaver which avoids contention problem, but its hardware implementation is quite complex. The proposed interleaver is simple to implement and it requires simple and less hardware. The de-interleaving required during decoding is also easy. The analysis done based on BER performance for QPP interleaver and bit reverse indexed interleaver .From the analysis BER performance of proposed interleaver is very close to QPP interleaver for less number of decoding iterations. As the number of iterations of decoding and frame size increases, the BER performance of proposed interleaver is better than QPP interleaver. This analysis is done with different decoding algorithms such as True APP and Max Log MAP. In all algorithms, BER performance of proposed interleaver is better than QPP interleaver.

## 6. REFERENCES

[1] Qu Wei, Information theory and coding theory ,Sciences Publishing House,2005

[2] A. Nimbalkar, Blankenship T. K., Classen B., Fuja T. E., Costello D. J., " Contention –Free Interleavers for High-Throughput Turbo Decoding", IEEE transactions on communication, vol. 56, No.8, 2008.

[3] O.Y.Takeshita,"On maximum contention free interleavers and permutation polynomial over integer rings," IEEE transaction on information theory, vol.52, no.3, 2006.

[4] Yang Sun, Joseph R. Cavallaro, "Efficient hardware implementation of highly-parallel 3GPP LTE/LTE-advance turbo decoder", INTEGRATION, the VLSI journal 44 (2011) 305-315.

[5] Chixiang Ma, Ping Lin, "Efficient implementation of Quadratic Permutation Polynomial Interleaver in Turbo Codes", International conference on WCSP 2009.

[6] Yang Sun, Joseph R. Cavallaro, Yuming Zhu, and Manish Goel "Configurable and Scalable Turbo Decoder for 4G Wireless receivers", 2010

[7] Michal Sybis and Piotr Tyczka, "Reduced complexity Log- MAP algorithm with Jensen inequality based non-recursive $max^*$ operator for turbo TCM decoding" EURASIP Journal on Wireless Communications and Networking 2013, 2013:238.