VERILOG IMPLEMENTATION OF TURBO ENCODER AND DECODER USING LOG MAP BASED ITERATIVE DECODING

A. Hinduja¹, Sudheer Kumar Terlapu²

¹PG Scholar (M. Tech Student, 2nd Year), Department of Electronics and Communication Engineering-VLSID.
²Associate Professor, Department of Electronics & Communication Engineering, Sri Vishnu Engineering College for Women, Bhimavaram, India.

ABSTRACT: Field Programmable Gate Arrays (FPGAs) provide the flexibility in operation and function by a simple change in the configuration bit stream. With increasing demand for different data rates and services for communication systems, reconfigurability is of great importance. Another cause for the limitation in BER performance is a poor interleaver design. Due to highly correlated sequences, the BER decreases to a certain level from where there is no further improvement in the decoding process. Low complexity turbo-like codes based on simple two-state trellis or simple graph structure results in decoder with low complexity. Out of this Convolution encoder and turbo codes are widely used due to the excellent error control performance. The most popular communications decoding algorithm, the iterative decoding requires an exponential increase in hardware complexity to achieve greater decode accuracy. This work focuses on the realization of turbo encoder and decoder using Log-Map based Iterative decoding technique. In this work an 8 bit turbo system was performance of turbo decoder found in terms of decoder iterations.

KEYWORDS: Convolution Codes, Turbo Codes, VLSI Code Design.


INTRODUCTION: Digital communications has better error correction capability when compared to analog communications.[1] Data transmission in discrete messages provides for greater signal processing capability.[2] The ability to process a communications signal means that the errors caused due to noise or other impairments in the course of transmission can be detected and corrected.[4] In addition, digital systems offer faster data processing.[3] They are not only more cost-effective but are less subject to distortion and interference which makes them more reliable than the analog systems. Turbo codes are finding use in 3G/4G mobile applications where designers seek to achieve reliable information transfer over bandwidth or latency-constrained communication links in the presence of data-corrupting noise.[5]

Turbo encoders and decoders are key elements in today’s communication systems to achieve data reception with the fewest possible errors the best.[7] The above Figure shows the Turbo System where Encoder generates a multiplexed code of two coders fed with direct and interleaved data.[8] Forward error correction (FEC) is a technique for error control during data transmission, whereby redundant information is added to the original data, which allows the receiver to detect and correct errors without the need to resend the data.[6]

The main advantage of FEC is to avoid the retransmission at the cost of higher bandwidth requirements on average and therefore is employed in the situations where retransmission is relatively costly or impossible.[10] The original information may or may not appear in the encoded output. Turbo system is an example of Forward error correction.[9] The original information may or may not appear in the encoded output. Turbo system is an example of Forward error correction.[9]

[11] Shows design and implementation aspects of parallel turbo-decoders that reach the 326.4 Mb/s LTE peak data-rate using multiple soft-input soft-output decoders that operate in parallel. [12] shows the design of new turbo codes that can achieve near-Shannon-limit performance. The design criterion for random interleavers is based on maximizing the free distance of the turbo code, i.e., the minimum output weight of codewords due to weight-2 input sequences. An upper bound on the free distance of a turbo code has been derived. [13] shows development of an application specific design methodology for low power solutions.[14]

The methodology starts from high level models which can be used for software solution and proceeds towards high performance hardware solutions.[15] The effect on performance due to variation in parameters like frame length, number of iterations, type of encoding scheme and type of the interleaver in the presence of additive white Gaussian noise has been studied with the floating point C model.[16] In order to obtain the effect of quantization and word length variation, a fixed point model of the application has also been developed.[17] The MAP turbo decoder is too complex to be implemented due to the large number of multiplications and the need of non-linear functions.

For that reason, two simplified versions of it were proposed in the past, namely Log-MAP and Max-Log-MAP.[18] The latter algorithm is sub-optimum in terms of bit error rate (BER) performance but easier to be implemented, as it requires only additions and the max operator. Another sub-
optimum algorithm that is suitable for turbo decoding is the soft output Veterbi algorithm (SOVA). It is a modified Veterbi algorithm (VA) that produces, in addition to the most likely path sequence, a reliability value of each estimated bit.[19]

So we introduce turbo encoder and decoder with SISO coding technique. It proposed reconfigurable turbo decoder. In turbo decoder, there is less delay and it occupies less area when compared to conventional Veterbi Decoder. In most of real time applications like bit error rate and recently in digital wireless communications, the turbo codes are used for error correction. We also implemented turbo system in MATLAB to verify the bit error rate performance and found that bit error rate decreases with increase in number of iterations. It is very efficient to communication systems. The main advantage of Turbo code Decoder is it has low bit error rate.

ENCODING OF TURBO CODE: As conventional code, the encoder for a Turbo code accepts k-bit blocks of the information sequence u and produces an encoded sequence (code word) p of n-symbol blocks. Moreover, each encoded block depends not only on the corresponding k-bit message block at the same time unit, but also on m previous message blocks.

Turbo encoders and decoders are key elements in today’s communication systems to achieve possible data reception with the fewest possible errors the best Turbo System where Encoder generates a multiplexed code of two coders fed with direct and interleaved data.

**Fig. 3: Encoder State Diagram Implementation**

Forward Error Correction (FEC) is a technique for error control during data transmission, whereby redundant information is added to the original data, which allows the receiver to detect and correct errors without the need to resend the data. Figure 3 depicts the implementation of RSC state table shown in figure 2. The Turbo encoder comprises two RSC encoders and an interleaver as shown in figure 5. Here the purpose of interleaver is to scramble input information so that there is no correlation between the data applied to the encoders.

**Fig. 4: Encoder of Turbo Code**

Figure 3 depicts the implementation of RSC state table shown in figure 2. The Turbo encoder comprises two RSC encoders and an interleaver as shown in figure 5. Here the purpose of interleaver is to scramble input information so that there is no correlation between the data applied to the encoders.

DECODING OF TURBO CODE: A turbo decoder is associate repetitive decoder, within which multiple decoders share chance info with each other in a repetitious fashion. The turbo decoder receives as its input a soft call worth from the rectifier.
This soft call worth can represent the chance that the transmitted bit was a one or zero.

A Parallel concatenated code decoder consists of two identical decoders, two identical interleavers, one deinterleaver, and a hard-decision output logic block, as shown in Fig. 6. Most A Posteriori (MAP) Soft-in, Soft-out (SISO) decoder is employed for the 2 decoders. The interleaver used is same because the interleaver employed in the encoder, and therefore the deinterleaver performs the alternative operation of the interleaver. The 3 outputs from the encoder: \( u, p, q \), are received into the decoder as \( u', p', q' \). The received bits: \( u', p', q' \) correspond to \( u, p, \) and alphabetic character once having been littered with the transmission.

![Fig. 5: PCCC Iterative Decoder Block Diagram](image)

![Fig. 6: SISO Decoder Block Diagram](image)

The decoder works on a 3K-bit square of got groupings \( u', p', q' \). Each 3K-bit piece of got information goes through various emphasis in the two decoder framework. The quantity of emphasis relies on the execution prerequisites and framework level contemplations. Once an adequate number of decoder cycles have been performed, the outward data from both decoders and the log-probability proportion of the got methodical images are utilized to process the decoded message.

A portion of the force of turbo codes originates from the way that the two encoded information groupings are decoded independently, yet the likelihood data from every decoder (Outward data) is shared. Since both decoders work on the same arrangement of data bits, every decoder has the capacity of helping the other. Subsequently, every cycle through the decoder gives an increment (to a point) in the likelihood of creating the right codeword at the decoder yield.

**INTERLEAVERS:** Interleavers are used in noncorrelation of two data outputs. Input bits are scrambles. Interleavers are classified into different types. They are Matrix interleavers, Random interleavers, Circular shift interleavers, odd and even interleavers. The purpose of the interleaver is to reorder a group of K input bits before they are encoded by the second RSC encoder. Typically, a turbo encoder interleaver is implemented by some type of pseudo-random algorithm. The interleaving algorithm that is used in the encoder can have significant impact on the performance of the decoder.[6] For the purposes of this study, only one interleaver type was used, a random K-bit interleaver.

The interleaver utilized as a part of the decoder must match the interleaver utilized as a part of the encoder. The interleaver will be distinctive regarding equipment usage, taking into account the way that delicate qualities (or multi-bit qualities) are going between the decoders. Consequently, the size and many-sided quality of the interleaver execution is identified with the quantity of bits (or quantization) chosen as the yield of every decoder. The deinterleaver performs the
inverse capacity of the interleaver; where the interleaver permutes the information, the de-interleaver de-permutes the information back to the first request.

**IMPLEMENTATION:** The schematic of the 8 bit turbo transmitter is shown below figure 10. It adds two coded bits for each transmitting bit, so the number of output bits from the transmitter is twenty four. In these eight bits are original bits, eight from first RSC and eight from second RSC whose input is interleaved version of input. The transmitter follows the state diagram of RSC to get work done.

![Fig. 9: Trellis for Decoder Implementation](image)

![Fig. 10: Implemented Transmitter](image)

The figure 12 below shows the wave forms of implemented encoder. The figure 13 below shows the wave forms of implemented decoder, the decoder having two RSC decoders with interleaver and De-interleaver blocks. Initially one decoder decodes the information and then the second decoder decodes information depending on the interleaved output of the first decoder and input, finally the output of the second decoder applied to De-interleaver to produce hard decision output.

![Fig. 11: Implemented Receiver](image)

![Fig. 12: Implemented encoder output wave forms](image)

These bits are transmitted over space to reach the receiver. The received signal is added up with noise so the receiver has to receive signals intelligently to avoid errors. The receiver implementation is shown in figure 11. It follows the Trellis (figure 9) to receive the information depending on present information being received and past information received. The receiver runs four state machines at a time with all possibilities to receive signals with less number of errors.
If the two decoders iterate for one more cycle, the information received can be a bit accurate. The accuracy further improved by increasing number of iterations. The figure 13 above shows the bit error rate considering white Gaussian noisy channel with binary phase shifting modulation technique.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Viterbi Decoder</th>
<th>Turbo Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (LUT's)</td>
<td>120 [14]</td>
<td>77</td>
</tr>
<tr>
<td>Delay (ns)</td>
<td>23.829 [14]</td>
<td>10.482</td>
</tr>
</tbody>
</table>

The implementation was compared with conventional Viterbi Decoder and observed that the proposed one occupies less area and produces less amount of delay.

**CONCLUSION:** The turbo encoder and Decoder were implemented in Verilog to deploy on to the FPGA. The synthesis done for the 8 bit transmitter and receiver for Spartan 3E Field programmable Gate Array. The decoder is able to receive information same as the information transmitted from the Transmitter. These parameters can have negative effects when very low BERs are simulated. Another cause for the limitation in BER performance is a poor interleaver design. Due to highly correlated sequences, the BER decreases to a certain level from the decoding process. This is the effect of the turbo code BER curve. The simulation result curve also presents a change in slope but at very low BER’s.

**REFERENCES:**

