International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 13
October 2017

Memory-Reduced and Area Efficient Turbo Decoding Architecture

S. SURENDAR
M.Tech (VLSI)
Dept of E.C.E
Chaitanya Institute of Technology and Science,
Warangal, Telangana,India
Email: surya.somidi77@gmail.com

Dr. B. SHOBA RANI
Associate Professor
Dept of E.C.E
Chaitanya Institute of Technology and Science,
Warangal, Telangana,India
Email: suprashoba@gmail.com

ABSTRACT: A new compression technique known as Next Iteration Initialization (NII) metrics is proposed for modifying the storage demands of turbo decoders. The proposed method stores only the range of state metrics with two indexes of the maximum and minimum values, whereas the previous compression methods have to store all of the state metrics for initializing the following iteration. A hardware-friendly recovery strategy is proposed which can be implemented by simple multiplexing networks. Compared to the previous work, as a result, the proposed compression method reduces the required storage bits while providing the acceptable error-correcting performance in practice. The proposed architecture of this paper can analyze the logic size and area by using Xilinx 14.3.

I.INTRODUCTION

Turbo codes are one of the most powerful types of Forward Error-Correcting (FEC) channel codes. Since the emergence of digital communication systems, there has been a need for error correction. This is due to the non-ideal nature of practical communication channels, which are often corrupted by noise. Error correction attempts to compensate for the errors introduced by this noise. The advantages of forward error correction are that a backchannel is not required and retransmission of data can often be avoided (at the cost of higher bandwidth requirements, on average). FEC is therefore applied in situations where retransmissions are relatively costly or impossible.

Turbo codes have been first introduced in 1993 By Berrou, Gavieux and Thitimajshima, and provide near optimal performance approaching the Shannon limit. The channel coding scheme for Long Term Evolution (LTE) is Turbo coding. The Turbo decoder is typically one of the major blocks in a LTE wireless receiver. Turbo decoders suffer from high decoding latency due to the iterative decoding process, the forward backward recursion in the maximum a posteriori (MAP) decoding algorithm and the interleaving and deinterleaving between iterations.

Generally, the task of an interleaver is to permute the soft values generated by the MAP decoder and write them into random or pseudo-random positions. The Turbo encoding scheme in the LTE standard is a parallel concatenated convolution code with two 8-state constituent encoders and one interleaver. The function of the interleaver is to take a block of N-bit data and produce a permutation of the input data block. The job of the turbo decoder is to reestablish the transmitted data from the received systematic bit stream and the two

parity check bit streams, even though these are corrupted by noise.

The iterative turbo decoder consists of two constituent SISO decoders serially connected via an interleaver, identical to the one in the encoder, and a corresponding deinterleaver. When data arrives, it is first stored in memory. Turbo decoder uses various iterations for decoding. In the beginning of the first iteration, the apriori1 data is not available, and it is set to zero. Thus, only the systematic and the parity1 data are used by decoder1 to calculate the a-posteriori1 data. Aposteriori1 data from decoder1 becomes apriori2 data after interleaving, and it together with parity2 data and the interleaved systematic data, are used by decoder2 to calculate a-posterirori2 data.

Again the de-interleaved a-posteriori2 data becomes the a-prori data for decoder1 and the first iteration is finished. A turbo decoder goes through several iterations before the final data output can be retrieved as shown in fig.1.

To improve the correctness of its decisions, each decoder has to be fed with information that does not originate from itself. The concept of extrinsic information was introduced to identify the component of the general reliability value, which depends on redundant information introduced by the considered constituent code.

A natural reliability value, in the binary case, is the logarithm likelihood ratio (LLR).A high throughput Turbo decoder can be realized by parallelizing several MAP decoders, where each MAP decoder operates on a segment of the received code word. Due to the randomness of the Turbo interleaver, two or more MAP decoders may access the same memory at the same clock cycle which will lead to a memory collision. As a result, the decoder has to be stalled which consequently delays the decoding process.

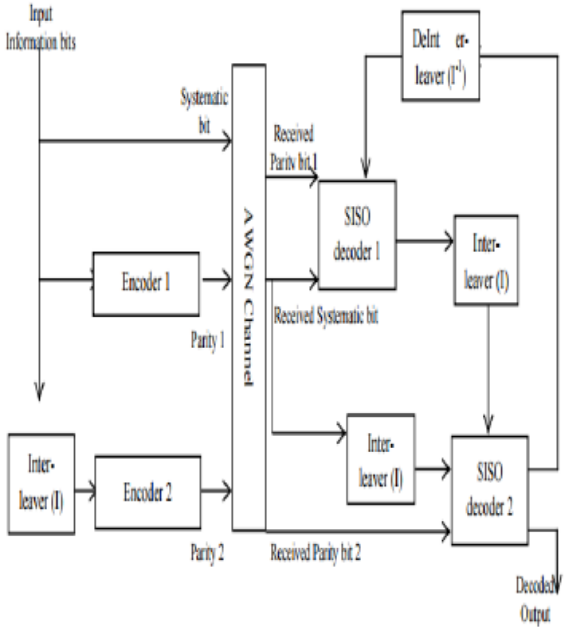

Fig 1. Conventional Turbo encoder and decoder

These memory collisions can be reduced by using a parallel interleaver. Such a type of interleaver is called Quadratic Permutation Polynomial (QPP) interleaver which can generate destination addresses on the fly. An enhanced QPP interleaver is proposed which is recursive in nature and it can permute the data without receiving the entire block of data. A QPP deinterleaver is just the inverse of QPP interleaver. The major advantages of turbo codes are its high BER because of the iterative algorithm used in turbo decoder. Soft In Soft Out (SISO) Decoder is used in turbo codes which enables us to get soft decisions rather than hard decisions.

II.PREVIOUS WORKS

A. Conventional Turbo Decoding Architecture

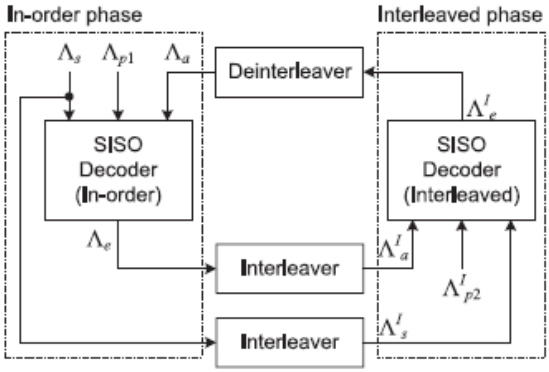

Fig2. Generalized turbo decoding architecture

Fig. 2 describes the generalized turbo decoding architecture based on the soft-input soft-output (SISO) decoders. The turbo decoder alternatively processes two decoding phases, i.e., in-order and interleaved phases. In the figure, the input log-likelihood ratio (LLR) sequences of the systematic bits and parity bits are denoted as Λs (or $\Lambda I\ s$) and $\Lambda p1$ (or $\Lambda I\ p2$), respectively, where superscript I denotes the sequences related to the interleaved phase.

Based on the input LLRs and *a priori* information Λa (or ΛIa), a SISO decoder generates *a posteriori* information, i.e., the extrinsic information Λe (or ΛIe), which will be *a priori* information of the opposite phase after passing through an interleaver (or deinterleaver). As the two different phases are exclusive in time, only one SISO decoder is normally adopted in practice for realizing the time-interleaved process.

B. Sliding-Window Technique With NII Metric Compressions

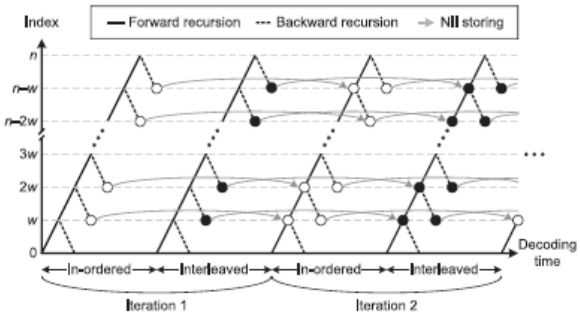

Fig 3 Sliding-window-based turbo decoding with NII technique

The sliding-window technique is widely accepted for the recent turbo decoders to reduce the size of internal buffers. Fig. 3 illustrates the decoding procedure for the *n*-bit codeword associated with sliding windows of *w* bits. Based on the maximum *a posteriori* (MAP) decoding algorithm, each sliding window first computes state metrics recursively in forward direction by using the corresponding branch metrics.

For the sake of simplicity, *k* forward state metrics of the *i*th trellis step are defined as $\alpha i(0), \alpha i(1), \ldots, \alpha i(k-1)$. Then, as shown in Fig. 3, the backward recursion computes the extrinsic information of each trellis step as well as the next state metrics in backward direction. Similar to the forward state metrics, *k* backward state metrics of the *i*th trellis step are represented as $\beta i(0), \beta i(1), \ldots, \beta i(k-1)$.

Before starting the backward recursion, it is important to properly initialize the starting confidence levels of each backward state. In the NII technique, as shown in Fig. 3, the final backward states of each window boundary are stored to be used for the starting points at the next backward recursion of the corresponding phase.

Assuming that all of the state metrics are normalized by the zeroth state, i.e., $\alpha i(0) = \beta i(0) = 0$, which is popularly applied to the turbo decoder for reducing the internal computing resolution, the conventional NII scheme requires the dedicated storage of $2 \times (k - 1) \times d \times n/w$ bits, where d stands for the bit-width of a state metric. For the practical turbo decoder of LTE advanced systems ($k = 8$), for example, a total of 32 256 bits have to be stored for realizing the conventional NII scheme, where d and w are assumed to be 12 and 32, respectively.

To reduce the storage demands caused by the NII scheme, the static encoding method is widely used in the recent turbo decoders. In the algorithm, the dedicated transfer function is introduced to restrict NII metrics to the power of twos.

III. PROPOSED NII METRIC COMPRESSION

In the contemporary turbo decoder, the max-log-MAP decoding algorithm is widely adopted due to the simple max operations instead of complicated max-star operations in the MAP algorithm. As the max-log-MAP decoding only focuses on the trellis path having the maximum reliability, it is necessary to determine the most reliable state at the initializing process of each sliding window. Unlike the previous works preserving each state metric value as much as possible, the proposed NII metric compression considers the range of state metrics denoted as Δx, i.e., the difference between the maximum and minimum state values among the wxth backward state metrics, $\beta wx(\cdot)$.

It is possible to make the bit-width of Δx smaller than d, the bit-width of each state metric, as the saturation of ranges exceeding a certain value is acceptable without degrading the BER performance. Based on the numerous simulations, only 8 bits are enough to represent Δx, while each original state metric requires at least more than 12 bits to prevent overflows in the LTE-advanced systems. To give a hint for the confidence levels of each state at the recovery process, in addition, we store the indexes of the maximum and minimum states, represented as $I\text{MAX}x$ and $I\text{MIN}x$, respectively.

Conceptually, the proposed compression method tries to offer the precise information of the peak differences by sacrificing the accuracy of each state metrics. Hence, the proposed algorithm reduces the number of storage bits in effect without degrading the error-correcting capability. When the size of a sliding window is set to 32, for the case of 6144-bit turbo codes, our compression scheme uses only 5376 bits for NII information, which is 6 times less than the conventional algorithm.

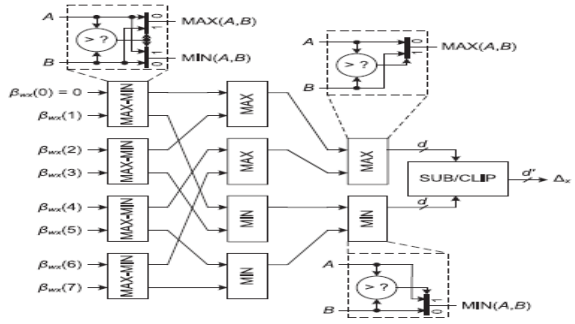

Fig 4. Cost-effective compressing process of eight state metrics based on the Proposed NII metric compression

Similar to the previous works, the proposed NII metric compression also requires additional computations for the encoding and decoding processes. To compensate the computational complexity

overheads, it is important to reduce the number of comparisons, which are much more time-consuming than the encoding networks.

In the proposed compression, as exemplified in Fig. 4 dealing with eight backward states denoted as $\beta wx(\cdot)$, the number of comparisons can be greatly reduced by sharing the intermediate results. To find Δx efficiently, three basic modules are utilized in the proposed architecture: the MAX–MIN, MIN, and MAX modules.

As detailed in Fig. 4, the MAX–MIN module finds both the minimum and maximum values between two inputs by utilizing one comparator and two multiplexors. Note that the simplified modules MIN and MAX, associated with one comparator and one multiplexor, are also introduced for performing MIN(A,B) and MAX(A,B), respectively.

The SUB/CLIP unit computes the final output Δx with the reduced bit-width. It is well known that IMAXx and IMINx can be easily generated by collecting the prior comparison results. Note that it is impossible to reduce the number of comparisons at the previous algorithm as all of the compressing processes are independent of each other.

Since the maximum and minimum values are generated at the same time by taking into account all of the states, on the other hand, we can reduce the number of comparisons by utilizing MIN–MAX modules to share the comparators of each processing as shown in Fig. 4. Note that the storage demands of the proposed algorithm are proportional to log2 k, whereas the previous solutions are proportional to k, requiring a much larger memory size. Moreover, the additional comparisons of the proposed work are also remarkably relaxed by more than 50%, compared to those of the static compressing method.

IV.IMPLEMENTATION RESULTS

The proposed turbo decoder utilizes MAX-LOG-MAP algorithm for decoding the received code word. A maximum of eight iterations are performed in order to find the final LLR value. An AWGN channel is preferred for communication. Under high bit error rate performance, a stopping criterion is used which stops the iterations in between and provides high throughput. Frame error limit is found and it is taken as the termination factor. Parallel processing is employed and a total of eight parallel SISO decoders are used. The initial value of the forward and backward state metrics of particular iteration is taken from the previous iteration.

Additional memories are allotted to store the next iteration metric values. It is noted that an optimum BER performance has been deduced with a minimum SNR. The SISO decoder module has been synthesized using Xilinx. It operates at a maximum frequency of 103.6MHz and the minimum period is 9.652ns. The proposed decoder has presented a better frequency when compared to the existing decoders. Based on the fixed-point simulation result, the finite word-length implementation leads to negligible BER performance degradation from using the floating-point representation.

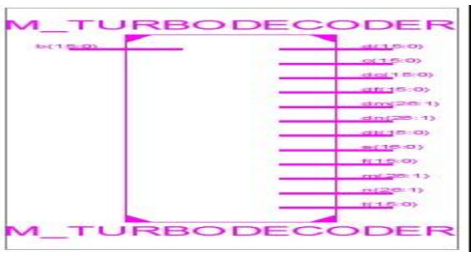

Fig 5. RTL SCHEMATIC

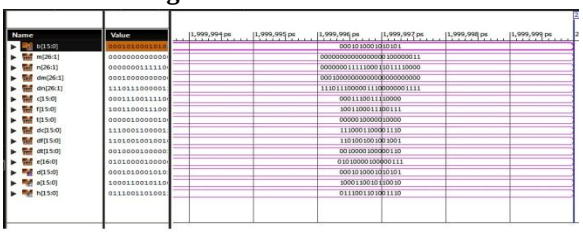

Fig 6. OUTPUT WAVEFORM

V.CONCLUSION

A new NII metric storing method has been proposed to reduce the memory demands of turbo decoders. By storing the precise ranges rather than the individually compressed metrics, the proposed algorithm remarkably reduces the size of NII metric memory while achieving an attractive error-correcting capability. As compared to the previous algorithms, the proposed compressing technique allows a low computational complexity in encoding and decoding of NII metrics which leads to the cost-effective turbo decoder architecture.

VI. REFERENCES

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

[2] *Multiplexing and Channel Coding (Release 11)*, 3GPP TS 36.212 v11.3.0, Jun. 2013.

[3] G. Wang *et al.*, "Parallel interleaver design for a high throughput HSPA+/LTE multi-standard turbo decoder," *IEEE Trans. Circuits Syst.I, Reg. Papers*, vol. 61, no. 5, pp. 1376–1389, May 2014.

[4] R. Shrestha and R. P. Paily, "High-throughput turbo decoder with parallel architecture for LTE wireless communication standards," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 9, pp. 2699–2710, Sep. 2014.

[5] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A lowcomplexity turbo decoder architecture for energy-efficient wireless sensor networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vo1. 21, no. 1, pp. 14–22, Jan. 2013.

[6] C. Benkeser, C. Roth, and Q. Huang, "Turbo decoder design for high code rates," in *Proc. IEEE Int. Conf. VLSI-SoC*, 2012, pp. 71–75.

[7] C. Roth, S. Belfanti, C. Benkeser, and Q. Huang, "Efficient parallel turbodecoding for high-throughput wireless systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 6, pp. 1824–1835, Jun. 2014.

[8] J.-H. Kim and I.-C. Park, "Double-binary circular turbo decoding based on border metric encoding," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 1, pp. 79–83, Jan. 2008.

[9] I. Yoo, B. Kim, and I.-C. Park, "Memory-optimized hybrid decoding method for multi-rate turbo codes," in *Proc. IEEE Veh. Technol. Conf. Spring*, 2013, pp. 1–5.

Authors:

S. SURENDAR studying M.Tech (VLSI) from Chaitanya Institute of Technology and Science, Warangal, Telangana,India.

Dr. B. SHOBA RANI working **as** Associate Professor in Dept of E.C.E from Chaitanya Institute of Technology and Science, Warangal, Telangana, India.