Design and VLSI Implementation of a High Throughput Turbo Decoder

Aso.M.Raymond
ME Communication Systems
Rajalakshmi Engineering College
Chennai

Dr.C.Arun
Associate Professor & Head of IT Department
RMK College of Engineering and Technology
Chennai

ABSTRACT

Turbo codes are one of the most efficient error correcting code which approaches the Shannon limit. However the major drawback of turbo codes is its high latency due to its iterative decoding process. The high throughput in turbo decoder can be achieved by parallelizing several Soft Input Soft Output(SISO) units together. In this way, multiple SISO decoders work on the same data frame at the same time. When more number of SISO decoders is connected parallel, the turbo interleaver creates a bottleneck in the system due to the contentions it introduces in accesses to memory. This delays the decoding process. In this paper, an advanced parallel interleaver called Quadratic Permutation Polynomial (QPP)interleaver is used which resolves the memory collisions introduced by parallel SISO decoders. The required area for the chip can be reduced by the help of efficient utilization of the SISO decoders. A method called Next Iteration Initialization is also used in order to reduce latency produced by a turbo decoder. The proposed Turbo decoder is expected to provide a throughput above 100Mbps.

General Terms

Error correction, Channel coding, MAX-LOG-MAP algorithm,

Keywords

Parallel processing,SISO decoders, Next Iteration Initialization, QPP interleaver

1. 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 back-channel 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 has been first introduced in 1993 By Berrou, Gavieux and Thitimajshima,[1] 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 convolutional 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[2] is to reestablish the transmitted data from the received systematic bitstream and the two parity check bitstreams, 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 a-priori1 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. A-posteriori1 data from decoder1 becomes a-priori2 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.

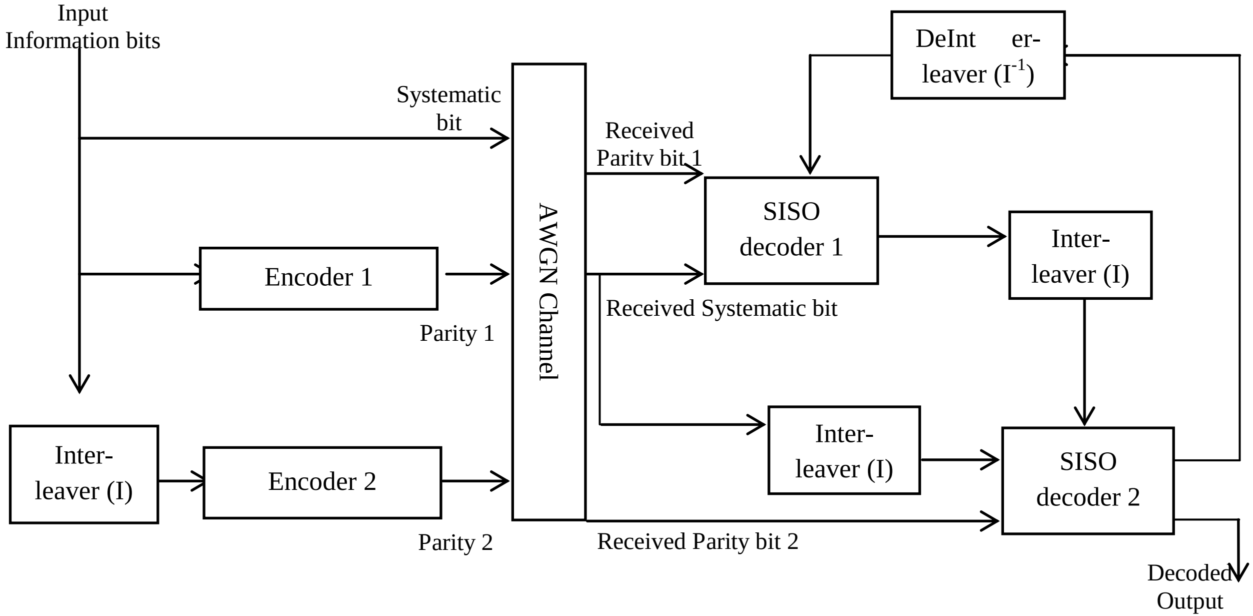

Fig 1: Conventional Turbo encoder and decoder

These memory collisions can be reduced by using a parallel interleaver. Such a type of interleaver is called Quadratic Permutation Polynomial(QPP)[3] 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.

2. TURBO DECODING ALGORITHM

There are two main algorithms in the component of the SISO decoders. They are MAP decoding and SOVA decoding. The MAP decoding algorithm is based on a posteriori probabilities(APP). The SOVA decoding algorithm is based on ML probabilities. Both of the algorithms use iterative technique to achieve decoding performance. The MAP algorithm can outperform SOVA decoding by 0.5dB or more.

The MAP algorithm checks very possible path through the convolutional decoder trellis, so that it seems too complex for application in the most systems. The MAP algorithm is complex because it has large number of multiplications and exponentials. In the Log-MAP algorithm all the calculations are performed in log domain. In this paper, Max-log-map algorithm is used for decoding. The three algorithms called MAP, LOG-MAP and MAX-LOG-MAP algorithms have been compared both theoretically and practically and the best algorithm for high throughput is preferred as MAX-LOG-MAP algorithm. Four parameters are determined to find the accurate a-posteriori-probability for the received block. They are branch metric, forward state metric, reverse state metric and Log likelihood ratio (LLR). Radix-2 max-log-map decoding is used with the scaling of the extrinsic information which allows a close log-MAP decoding performance to be achieved. The scaling factor ranges from 0.6 to 0.8.

2.1 SISO Unit

Using two SISO blocks in a turbo decoder tends to occupy much area and hence the decoder is modified as shown in fig.2 in our proposed system.

2.1.1 Branch Metric computation unit

In the algorithm for turbo decoding[4] the first computational block is the branch metric computation. The branch metrics is computed based on the knowledge of input and output associated with the branch during the transition from one state to another. There are four states and each state has two branches, which gives a total of eight branch metrics. The branch metrics is given by [5] as

$$\gamma_k(s_k, s_{k+1}) = \frac{1}{2}[L(u_k)x_k^s + y_k^s x_k^s + y_k^p x_k^p] \qquad (1)$$

where s^k is the previous state and s^{k+1} is the next state, x^k= (x_k^s, x_k^p) is the input/output symbol of the encoder for each branch between state $s^k$to s^{k+1}yk= $(y_k^s, y_k^p)$is the received channel symboland L(u_k) is the *a-priori* information.

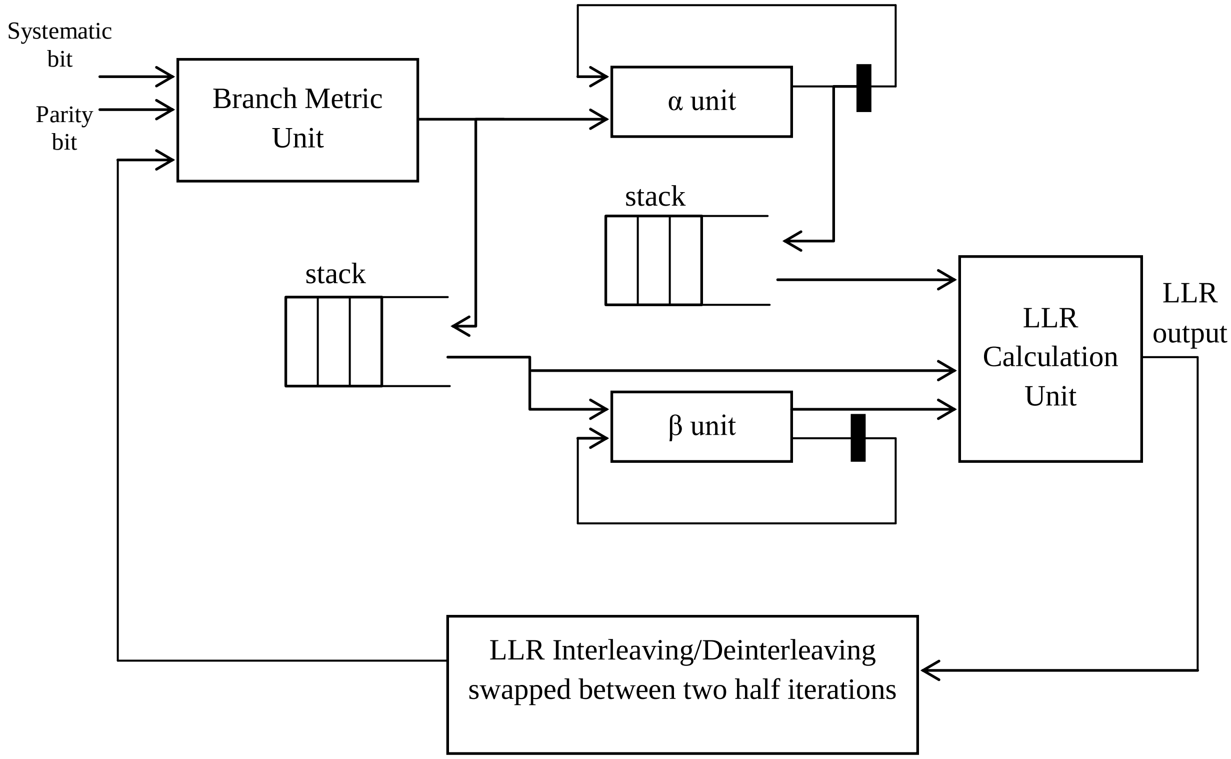

Fig 2: Modified turbo decoder

2.1.2 Forward State Metrics

$$\alpha(s_k) = max^*[\alpha_{k-1}(s_{k-1}) + \gamma_k(s_{k-1,}s_k)] \qquad (2)$$

2.1.3 Backward State Metrics

$$\beta(s_k) = max^*[\beta_{k+1}(s_{k+1}) + \gamma_k(s_k, s_{k+1})] \qquad (3)$$

2.1.4 LLR Unit

$$L_k = \max_{(s_k,s_{k+1}),1}{}^*[\alpha(s_k) + \gamma_{k+1}(s_k, s_{k+1}) + \beta_{k+1}(s_{k+1})] - \max_{(s_k,s_{k+1}),0}{}^*[\alpha(s_k) + \gamma_{k+1}(s_k, s_{k+1}) + \beta_{k+1}(s_{k+1})] (4)$$

The max star operator employed in the above descriptions is defined as follows

$$max^*(a,b) = \log(e^a + e^b) = \max(a,b) + \log\,(1 + e^{-|a-b|}) \qquad (5)$$

For MAX-LOG-MAP algorithm, the function is approximated as

$$max^*(a,b) = \max\,(a,b) \qquad (6)$$

2.2 Parallel Processing and Next Iteration Initialization

To reduce the decoding latency, the sliding window algorithm is often used [6]. In order to achieve high throughput of 4G mobile communication systems, eight SISO decoders as shown in fig.2.are connected in parallel. First, the coded block will be partitioned into eight equally sized sub-blocks. Each SISO unit is responsible for the processing of one sub-block. The processing of one sub-block is considered as one parallel window with size S_{pw}. Hence in total eight parallel windows can be processed in parallel, which reduces the processing time by eight times. To significantly reduce the latency produced by the turbo decoder, a method called Next Iteration Initialization is used in this paper along with windowing. The computed metricvalues at the border of the sliding windows will be stored in a buffer to be used as initial value in the next iteration. Here the state metrics in the previous iterations are close to state metrics in the next iteration at the same trellis. So the previous iteration metric values can be considered as the initial value for the next iteration. But an additional memory has to be allotted to store these metric values. Hence the turbo decoder latency is reduced at an expense of additional memory.

2.3 Proposed Parallel Interleaver

When the degree of parallelism increases in a turbo decoder, memory contention issue arises and it may lead to extra delay in the circuit. Hence there is a need for parallel interleaver[7] which should be capable of generating addresses on the fly for all the parallel SISO decoders.

The quadratic permutation polynomial (QPP) interleaver guarantees the desirable contention-free property for parallel memory access and has been adopted in the 3GPP LTE for turbo coding. The QPP interleaver[8] can be expressed via a simple mathematical formula. Given an information block length N, the x-th interleaving output position is specified by the quadratic expression:

$$f(x) = (f_2x^2 + f_1x) \bmod N \qquad (7)$$

where parameters f1 and f2 are integers and depend on the block size N (0 ≤ x, f1, f2 < N). For each block size, a different set of parameters f1 and f2 are defined. In LTE, all the block sizes are even numbers and are divisible by 4 and 8. Moreover, the block size N is always divisible by 16, 32, and 64 when N >= 512, N >= 1024, and N >= 2048, respectively. By definition, parameter f1 is always an odd number whereas f2 is always an even number[9][10].

This QPP interleaver is used to produce addresses recursively throughout the decoding process. The f(x) function is the basic function and the corresponding addresses for all the SISO decoders are generated recursively on the fly. The hardware implementation is made easier by storing the basic function in an LUT and computing other functions with the help of the basic LUT. The interleaving addresses of the entire decoder are obtained with the replication of the hardware and from the LUT providing the basic functionality.

To perform deinterleaving, when the interleaved data is read, the original location of this data can also be retrieved in the same time and become the deinterleaving destination address. QPP interleaver can support a block size from 40bits to 6144bits.

3. 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. The Bit error rate performance of a high throughput decoder with parallel processing is shown in fig.3. The Frame Error Rate performance is shown in fig 4. It is noted that an optimum BER performance has been deduced wirh a minimum SNR. The design parameters are shown in table 1.

The matlab simulation takes hours together to run and hence the hardware implementation is carried out in VLSI. The circuit of SISO decoder has been described in VHDL and synthesized using Xilinx ISE and simulated using ModelSim.

Table 1: Turbo Decoder Parameters

Generator matrix	[13 11]
Constraint length	4
Code Rate	1/3
Decoder Algorithm	MAX LOG MAP
Frame size, N	1024
QPP function	f(x)= (31*x+64*x^2)%N
Number of parallel SISO	8
Number of iterations	5

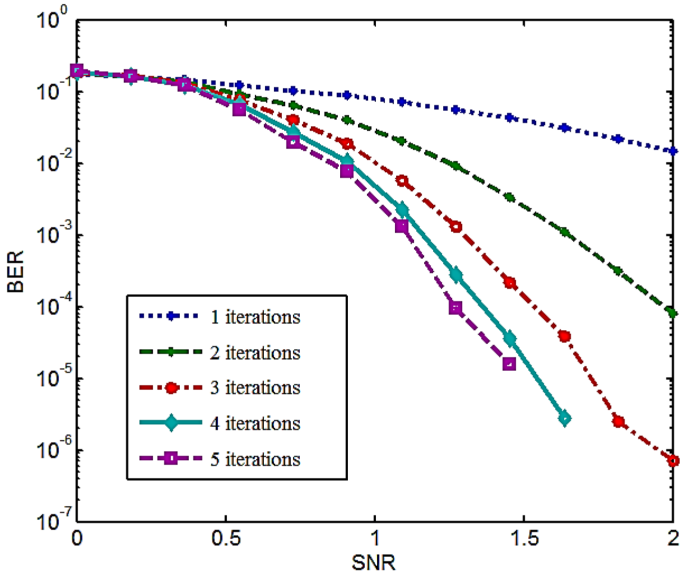

Fig 3: BER performance of turbo decoder(1024 bit frame)

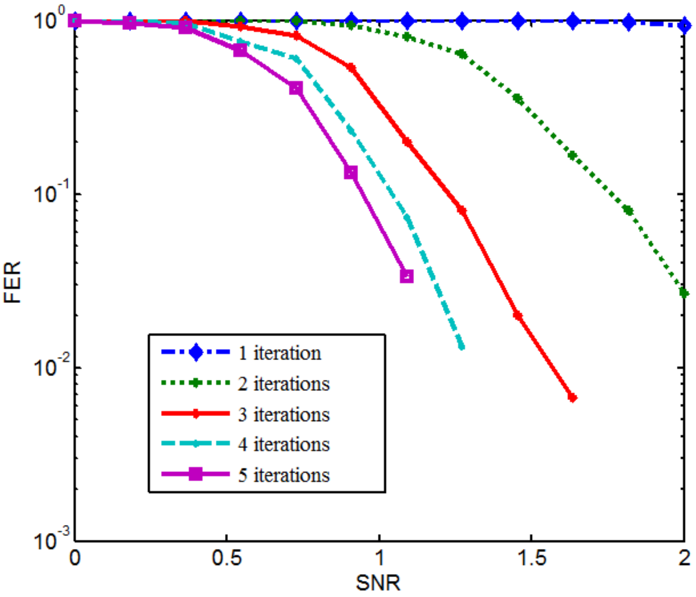

Fig 4: FER performance of turbo decoder

Table 2: Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	485	69120	0%
Number of Slice LUTs	3051	69120	4%
Number of fully used LUT-FF pairs	431	3105	13%
Number of bonded IOBs	18	440	4%
Number of BUFG/BUFGCTRLs	1	32	3%

The SISO decoder module has been synthesized using Xilinx and the design utilization summary is shown in table II. 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.

4. CONCLUSION

In this paper, a parallel turbo decoder has been adopted and QPP interleaver is used to avoid memory contention issues. To reduce latency a method called Next Iteration Initialization (NII) has been adopted. NII increases the throughput with the expense of additional memory. The parallel turbo decoder has been implemented in Matlab and the BER performance was studied. SISO decoder has been synthesized and the timing details have been analyzed. The designed radix-2 turbo decoder is expected to provide a throughput of more than 100Mbps when implemented in hardware. At high SNR, throughput is further improved by using a stopping criterion.

5. REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes,” in Proc. 1993 *InternationalConference on Communications* (ICC ’93), Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] Claude Berrou, Alain Glavieux and Punya Thitimajshima, “Near shannon limit error – correcting Coding and decoding : turbo-codes” IEEE Transactions on Communications, 44:1261 – 1271,Oct.1996.

[3] Third Generation Partnership Project, “3GPP home page” www.3gpp.org.

[4] Vogt.J and Finger.A, “Improving the max-log-MAP turboDecoder,” Electron. Lett., vol. 36, no.23, pp.1937-1939, Nov 2000.

[5] K. Loo, T. Alukaidey, and S. Jimaa, “High performance parallelised 3GPP turbo decoder," in *IEEE Personal Mobile Communications Conference*, April 2003, pp. 337-342.

[6] A.J. Viterbi. An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes. *IEEE J.Sel. Areas Commun.*, vol.16:pp.260–264, Feb.1998.

[7] R. Dobkin, M.Peleg, and R.Ginosar. Parallel interleaver design and vlsi architecture for low-latency map turbo decoders. *IEEE Trans.VLSI Syst.*, 13(4):427–438, 2005.

[8] Y.Sun, Y.Zhu, M.Goel, and J.R.Cavallaro. Configurable and Scalable High Throughput Turbo Decoder Architecture for Multiple 4G Wireless Standards. In *IEEE International Conference on Application-Specific Systems, Architectures and Processors(ASAP)*, pages 209–214, July2008.

[9] J. Sun and O.Y.Takeshita. Interleavers for turbo codes using permutation polynomials over integer rings. *IEEE Trans.Inform.Theory*, vol.51:101– 119, Jan.2005.

[10] O.Y.Takeshita.Onmaximumcontention-freeinterleaversand permutation polynomialsoverintegerrings. *IEEE Tran .Inform.Theory*, vol.52:1249–1253,Mar.2006.