International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-10, Issue-2, March 2022
https://doi.org/10.55524/ijircst.2022.10.2.115
Article ID IRPV1054, Pages 611- 614
www.ijircst.org

Verilog HDL using LTE Implementation MAP Algorithm

T. Rama Krishna[1], T. Krishna Murthy[2], N. Vilasrao Sarode[3], P. Srilakshmi[4], V. Geetha Sri[5]

[1,2]Associate Professor, Department of Electronics and Communications Engineering, PACE Institute of Technology & Sciences, Ongole, Andhra Pradesh, India

[3]Assistant Professor, Department of Electronics and Communications Engineering, PACE Institute of Technology & Sciences, Ongole, Andhra Pradesh, India

[4,5]UG Student, Department of Electronics and Communications Engineering, PACE Institute of Technology & Sciences, Ongole, Andhra Pradesh, India

ABSTRACT- In many communication systems, turbo coding Techniques for Encoding and Decoding are employed to repair errors. As compared to other error correction codes, turbo codes provide great error correcting capabilities. For the implementation of the Turbo decoder, a Very Large Scale Integration (VLSI) architecture is suggested in this study. The Maximum-a-Posteriori (MAP) algorithm is employed at the decoder side, where soft-in-soft-out decoders, interleaves, and deinterleavers are all used. The usage of the MAP algorithm reduces the quantity of iterations necessary to decode the information bits being transferred. This research employs a system for the encoder component that consists of two recursive convolutional encoders and a pseudorandom interleaver on the encoder side. Tools from Octave and Xilinx Vivado are used for the Turbo encoding and decoding. The system is synthesised and implemented using a specialised integrated circuit.

KEYWORDS- Encoding, Decoding Techniques. MAP, Xilinx, vivado.

I. INTRODUCTION

A very efficient method of error correction is turbo coding, which has recently had a significant impact on channel coding. One of the most popular and effective coding methods for reducing bit errors in digital communication is turbo coding. The magnitude comparator will be used to incorporate the turbo encoder as a module into the embedded In-Vehicle Device module. It has been demonstrated that the Turbo encoder parallel computing variant system's processing time increased according to chip size. Using the suggested rationale produced efficient space and power utilisation. The turbo encoder utilising a magnitude comparator in the suggested design has a somewhat complicated circuit Many digital microprocessors, application-specific circuits, and other devices use arithmetic circuits that execute digital arithmetic operations. Using 32-bit operands, addition, subtraction, multiplication, and division operations are performed using a high-speed ASIC version of a floating point arithmetic unit.Together with exceptional handling, pre-normalization unit and post-normalization unit discussions are also covered [3]. Turbo code is used for good performance in wireless communication technologies as 3GPP-LTE, WiMAX, DVB-SH, and HSPA. Several wireless communication technologies, such as HSDPA and LTE, have used turbo codes because they can achieve near to Shannon capacity and can be implemented with hardware efficiency. The number of times the transmitted information bits must be decoded using the turbo decoder is decreased since it employs the original MAP algorithm rather than the estimated Max log-MAP method. It takes less iterations to decode the bits of transferred information as a result. turbo coding the information by concatenating convolutional encoders in parallel with pseudorandom interleave bits that must be sent through a network. It produces interleaved and non-interleaved parity bits as well as systematic bit sequences. A-priori probability values, or soft-demodulated values of transmitted bits, are given to individual MAP decoders. [6] Such MAP decoders use on an algorithm that analyses the a-priority probabilities of systematic and parity bits to create the a-posteriori probability values of the transmitted information bits. This algorithm operates on the idea of trellis graph.

II. LITERATURE REVIEW

The fundamental turbo coding approach for optimising the error quality of a typical rate-1/3 turbo code. The authors described the high-speed turbo SISO Decoder in their publication [1]. Rather than branch metric values, the standardisation process was performed to state metric branch values. [2] Proposes a minimum-power, area-efficient turbo soft-output (SISO) decoder based on the Viterbi algorithm. In their work [3], Kavinilavu, Salivahanan, and Bhaaskaran design and combine the Convolutional encoder and Viterbi decoder. They planned, modelled, and synthesised using XILINX-ISE 12.4i in ModelSim 10.0e. Authors showed Max Log MAP algorithm-based turbo decoder output variations on implemented with fixed point, Vedic, and Booth multipliers in article

The SOVA decoder implementation for various constraint lengths is included in the article. According to simulation results, space and power use can be reduced in comparison to a traditional SOVA decoder application. The creators of [4] created a turbo decoder architecture that makes use of

both parallel trellis stage level and parallel SISO decoder layer. The proposed paper [5] present the design and implementation of a memory-reduced Turbo decoder on the field programmable gate array in the LTE-Advanced standard (FPGA). The author summarises the architecture concerns affecting turbo decoders in this study [6]. The key parameter of the code allows for the evaluation of several turbo decoder varieties. In their investigation, the authors of article [8] suggested a brand-new Min-Log-MAP algorithm variation with a low level of complexity.

III. PROPOSED METHOD

The Proposed paper shows that Encoding and decoding techniques using for MAP Algorithm. The MAP algorithm's intricate nature makes it challenging to implement. The computations of the values are in the log domain and the MAP method will mostly do the approximation, which is the key distinction between the MAP and Max-Log-MAP algorithms. The approximations are simple to use. By making tiny corrections at each stage to optimise the operation, the Max-Log-MAP method reduces or completely avoids approximations of the MAP algorithm. Effective calculations that are close to the MAP method are produced by this modest adjustment.

Figure 1: Block Diagram of map algorithm

A. Encoding Technique for Map Algorithm

The MAP method reduces the possibility of a bit mistake by identifying the most likely bit at each trellis point utilising the whole sequence that was collected. Assume a channel output of y that the decoder got for a frame of N symbols that were encoded with m bits.

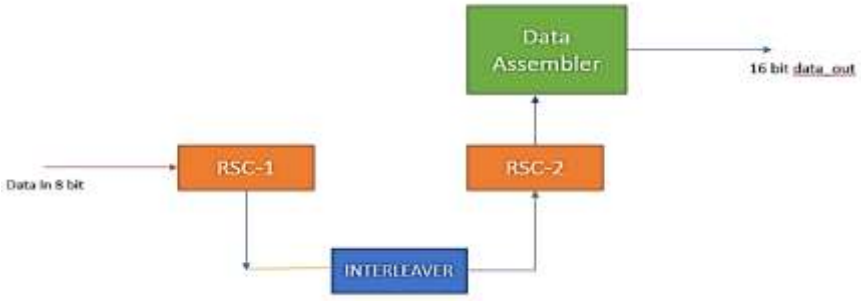

Figure 2: Block diagram of Encoding Method

The complexity of the MAP algorithm's architecture makes it challenging to implement. The fundamental distinction between the MAP and Max-Log-MAP algorithms is that the calculations of the values are done in the log domain, whereas the MAP method will mostly do approximation. It's simple to use the approximations as values. By adding a tiny adjustment at each stage to optimise the operation, Max-Log-MAP reduces or completely avoids approximations of the MAP algorithm. Effective calculations that are close to the MAP technique can be achieved with this minor tweak.

The four main components of the Log MAP decoder are: Branch Metric Unit (BMU) for calculation, Forward State Metric Unit (FSMU) for verification, Backward State Metric Unit (BSMU) for correlation, and Log Likelihood Ratio (LLR) for assessment. Figure 3.3 provides an illustration of the Log-MAP decoder's calculation stages. The analogue signal that was received was scaled up by the quantization procedure. The computation is done in branch metric units; in addition to the BMU, this step also contains forward state metric units

B. Design of the Turbo Coder

Together, Turbo encoder and decoder make up the Turbo coder architecture. The turbo encoder is made up of two identical Recursive convolutional encoders (RSC) plus an interleaver that generates pseudorandom numbers (figure 2). Turbo concatenated parallel coding is used by LTE. Every RSC uses two separate sets of data. The first party receives original data. While the interleaved version of the input data is sent to the second encoder. The data bits are scrambled using a specific algorithm, which is referred to as interleaving.

C. Turbo decoder

The BCJR algorithm is used to decode turbo codes in the most common way. The turbo decoding algorithm's essential and core concept is the iteration between the two SISO component decoders, as seen in figure 3. It consists of two decoders that operate in parallel to enhance and improve the estimation of the original information bits. The convolutional code produced by either the first or second CE is respectively decoded by the first and second SISO decoders. The first component decoder makes one

pass during a turbo-iteration, which is followed by the second component decoder doing another pass.

D. Deinterleavering and Interleaving

An essential building piece in the channel coding of turbo codes is the interleaver. In order to fulfil the decoding requirements of the parallel turbo decoder, the interleaver or deinterleaver is likewise developed in parallel. Memory contention is a relatively prevalent interleaving issue that is remedied by using the contention free interleaver. The block size in the suggested design is N,

$$(i) = \mathrm{a}(i) mod NA(i) = \mathrm{b}_1 i_1 + \mathrm{b}_2 i_2 \qquad (1)$$

Where, i=0, 1, 2,..,N-1

Where b1 is an odd number and b2 is an even number, 'i' is the index number of input data. yi and Π(i) is the index number after interleaving.

By identifying the most likely bit at each trellis point using the whole sequence that was collected, the MAP method reduces the possibility of bit mistake. Assume a channel output of y received by the decoder for a frame of N coded symbols consisting of m bits.

$$\mathrm{G}_r(d^{sym} \overline{=} j_i|y) = \sum_{(S^t,S)/d^{sym}=j} \alpha i \ (S')\gamma i(S', S)\beta i{+}1(S) \quad \ldots\ldots\ldots\ldots\ldots(2)$$

The breakdown of computing the joint probabilities among the earlier and later observations is made possible by the trellis form of the code. Equation 2 illustrates the Forward recursion metric applied when deconstructing. It immediately provides, at time I the probabilities of state S based on channel data from earlier iterations. The probabilities of the state determined using the upcoming values from the channel and Branch metrics are also found using the backward recursion metric.

IV. RESULTS AND DISCUSSION

Verilog HDL is used in this study to simulate turbo encoders and decoders. A software package built on Xilinx is called Viv ado Design Suite. HDL designs may be created and analysed using it. The Turbo encoder decoder and recursive convolutional encoder are simulated using Xilinx Viv ado and Octave. The net list is built using RTL. The RTL is used for logic synthesis and analysis in digital designs. Regarding physical design (floor layout, placement, and routing). The implementation tool optimises, places, and routes a net list as input.

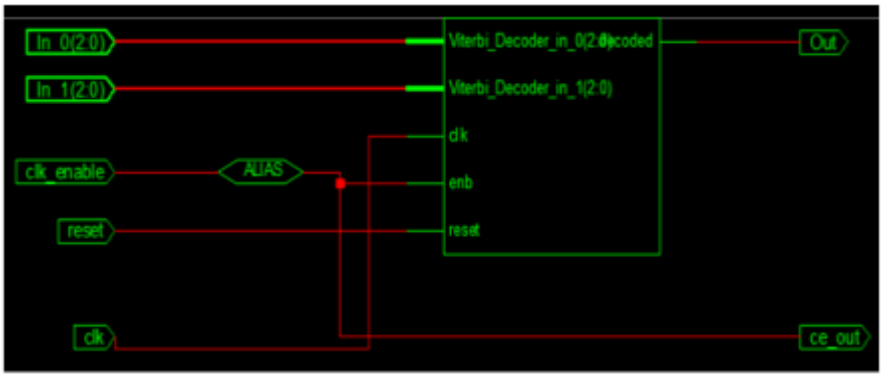

Figure 3: Synthase Result for decoder

The foundation for modalism simulates design. The top module, which displays the implementation of the Viterbi decoder, calls the numerous sub units. The turbo decoder of two inputs with variable data size is formed in this case by the cascaded Viterbi Decoder.

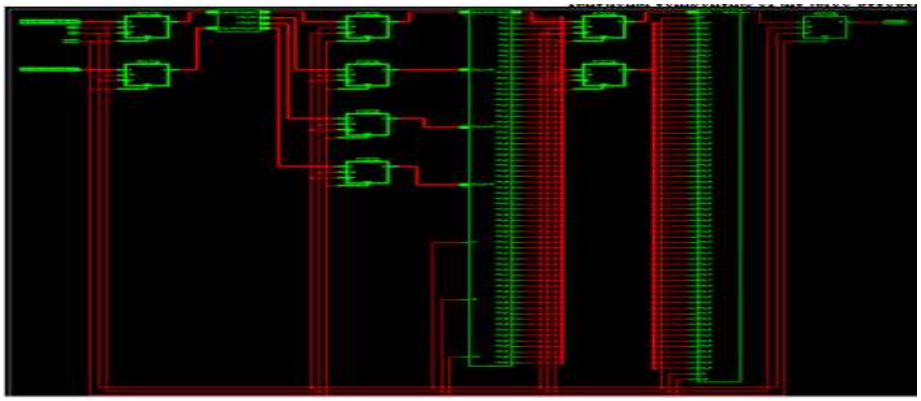

Figure 4: RTL Schematic

Enable signals in the map are used to activate the MAP decoder, which analyses the soft values to produce decoded a-posteriori LLR values. It is crucial to keep an eye on these LLR values while they are processed by the FPGA-based MAP decoder. As a result, these values can be observed utilising multichannel logic analyzers.

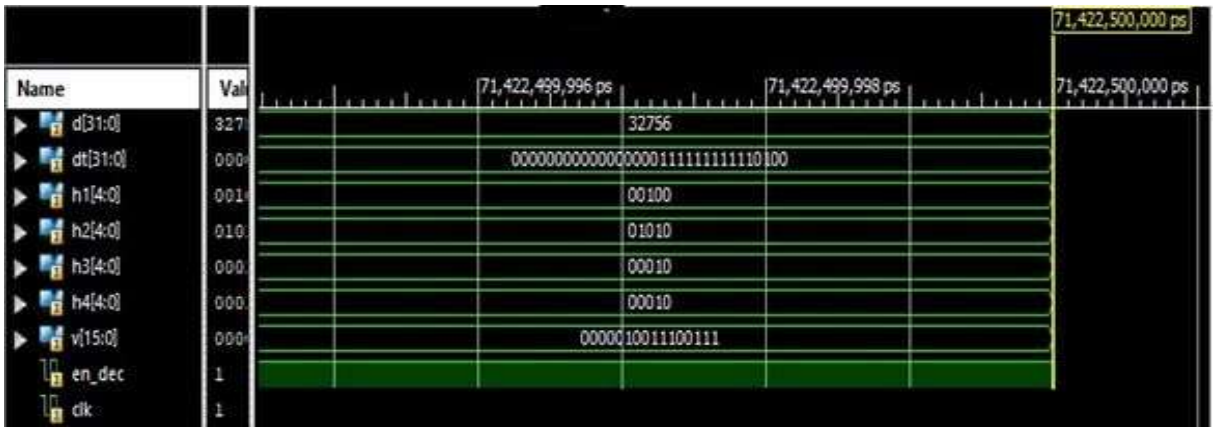

Figure 5: Simulation Results

V. CONCLUSION

An illustration of 4G LTE wireless networks' power usage is provided. Also covered are the main power dissipation sources in turbo decoders. There are many potential methods for reducing power consumption, including contention-free interleavers, memory cuts for parallel access, parallel architecture, and SDR for iteration termination. The examination of the effects of architecture and algorithm level optimisations on power usage and throughput is suggested. In this study, the primary functional components of the turbo decoder are examined. Examples of interleaving blocks are provided, along with their importance. Several design and implementation difficulties with regard to the integration of VLSI chips are addressed at the heart of the turbo decoder.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] Santosh Gooru, S. Rajaram, “Design and Implementation of Turbo Coder for LTE on FPGA”, International Journal of Electronics Signals and Systems (IJESS), Vol: 4, No: 1, pp: 48 – 52, 2014.

[2] M Siva Kumar, S Syed Shameem, M.N.V. Raghu Sai, Dheeraj Nikhil,P. Kartheek, K Hari Kishore, “Efficient and low latency turbo encoder design using Verilog - HDL”, International Journal of Engineering & Technology, Vol: 7, No: 1.5, pp: 37 – 41, 2018.

[3] Akshaya V, Sreehari K N, Anu Chalil, “VLSI Implementation of Turbo Coder for LTE using Verilog HDL”, Proceedings of Computing Methodologies and Communication, ISBN:978- 1-7281-4889-2, 2020.

[4] Manjunatha K N, and Vaibhav a Meshram, “Design and FPGA Implementation of Power Efficient Turbo Decoder for 4G LTE Standards”, Vol: 12, No: 21, pp; 10921 – 10925, 2017.

[5] Rami Klaimi, Charbel Abdel Nour and Catherine Douillard, “Low-complexity decoders for non- binary turbo codes”, IEEE 10th International Symposium on Turbo Codes & Iterative Information Processing, pp: 1 – 5, 2018.

[6] Tepoju Vivek Vardhan, Bandi Neeraja, Boya Pradeep Kumar, Chandra Sekhar Paidimarry, “Implementation of Turbo Codes Using Verilog- HDL and Estimation of its Error Correction Capability”, IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics, pp: 75 – 79.

[7] Moon Gi Seok, Dae Jin Park, Geun Rae Cho, Tag Gon Kim, “Framework for Simulation of the Verilog/SPICE mixed model: Interoperation of Verilog and SPICE simulators using HLA/RTI for model reusability”, doi: 978-1-4799-6016-3/14.

[8] [8] C. Lin, C. Chen and A. Wu, Area-Efficient Scalable MAP Processor Design for High- Throughput Multi standard Convolutional Turbo Decoding," IEEE Transactions on Very LargeScale Integrated (VLSI) Systems, vol. 19, no. 2, pp. 305-318, 2011.